

Eigenspace Approach to Specific Emitter Identification of Orthogonal Frequency Division Multiplexing Signals

Peter H. Sahmel

Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science
in
Electrical Engineering

Jeffery H. Reed, Chair
Chad M. Spooner
Carl B. Dietrich

November 16, 2011
Blacksburg, Virginia

Keywords: Cyclostationarity, Eigendecomposition, Specific Emitter Identification, Hidden
Markov Models

Copyright 2011, Peter H. Sahmel

Eigenspace Approach to Specific Emitter Identification of Orthogonal Frequency Division Multiplexing Signals

Peter H. Sahmel

(ABSTRACT)

Specific emitter identification is a technology used to uniquely identify a class of wireless devices, and in some cases a single device. Minute differences in the implementation of a wireless communication standard from one device manufacturer to another make it possible to extract a wireless "fingerprint" from the transmitted signal. These differences may stem from imperfect radio frequency (RF) components such as filters and power amplifiers. However, the problem of identifying a wireless device through analysis of these key signal characteristics presents several difficulties from an algorithmic perspective. Given that the differences in these features can be extremely subtle, in general a high signal to noise ratio (SNR) is necessary for a sufficient probability of correct detection. If a sufficiently high SNR is not guaranteed, then some form of identification algorithm which operates well in low SNR conditions must be used. Cyclostationary analysis offers a method of specific emitter identification through analysis of second order spectral correlation features which can perform well at relatively low SNRs. The eigenvector/eigenvalue decomposition (EVD) is capable of separating principal components from uncorrelated gaussian noise. This work proposes a technique of specific emitter identification which utilizes the principal components of the EVD of the spectral correlation function which has been arranged into a square matrix. An analysis of this EVD-based SEI technique is presented herein, and some limitations are identified. Analysis is constrained to orthogonal frequency division multiplexing (OFDM) using the IEEE 802.16 specification (used for WiMAX) as a guideline for a variety of pilot arrangements.

This work made possible by Zeta Associates Inc., NSF grant CNS-0627436 (CT-ISG: Assuring Security in Spectrum Agile Radio Networks), and Wireless@VT affiliates.

Acknowledgments

I would like to thank my advisor, Dr. Jeff Reed, for providing the direction and focus for the bulk of this thesis. I would also like to thank Dr. Chad Spooner for his willingness to share his wealth of knowledge on Cyclostationarity, and for putting up with my barrage of questions. In addition, I would like to thank Dr. Carl Dietrich for his sound advice and words of wisdom.

Many thanks go out to the students in the MPRG whose friendship I've gained over the course of my studies. Greatly missed will be the lunch outings, social gatherings and humorous discussions.

I would also like to thank my employer, Zeta Associates, for allowing me to take a sabbatical from work in order to complete my master's level graduate studies. Without their continued support I would not have been able to spend the valuable time required to complete the program in a timely manner.

List of Abbreviations

| | |
|-------------|--|
| AWGN | Additive White Gaussian Noise |
| BPSK | Binary Phase Shift Keying |
| CAF | Cyclic Autocorrelation Function |
| CDP | Cyclic (or Cycle-Frequency) Domain Profile |
| CP | Cyclic Prefix |
| CR | Cognitive Radio |
| CSCF | Conjugate Spectral Correlation Function |
| CSCM | Conjugate Spectral Correlation Matrix |
| DFT | Discrete Fourier Transform |
| DL | Downlink |
| DSA | Dynamic Spectrum Access |
| EVD | Eigenvalue (or Eigenvector) Decomposition |
| FFT | Fast Fourier Transform |
| FOT | Fraction of Time |
| FUSC | Full Usage of Subcarriers |
| FV | Feature Vector |
| HMM | Hidden Markov Model |
| IFFT | Inverse Fast Fourier Transform |
| LOS | Line-of-sight |

| | |
|--------------|---|
| ML | Maximum Likelihood |
| NLOS | Non-line-of-sight |
| OFDMA | Orthogonal Division Frequency Multiple Access |
| OFDM | Orthogonal Division Frequency Multiplexing |
| OFUSC | Optional Full Usage of Subcarriers |
| PDF | Probability Distribution Function |
| PdF | Probability Density Function |
| PRBS | Pseudorandom Binary Sequence |
| PR | Pattern Recognition |
| PSD | Power Spectral Density |
| PSK | Phase Shift Keying |
| PUSC | Partial Usage of Subcarriers |
| QPSK | Quadrature Phase Shift Keying |
| RF | Radio Frequency |
| SCF | Spectral Correlation Function |
| SCM | Spectral Correlation Matrix |
| SEI | Specific Emitter Identification |
| SUI | Stanford University Interim |
| UL | Uplink |
| WGN | White Gaussian Noise |

Contents

| | | |
|----------|--|-----------|
| 1 | Introduction | 1 |
| 1.1 | Motivation | 1 |
| 1.2 | Literature Review | 3 |
| 1.3 | Organization | 4 |
| 1.4 | Contributions | 5 |
| 1.5 | Publications | 5 |
| 2 | Cyclostationarity | 6 |
| 2.1 | Introduction | 6 |
| 2.2 | Definitions | 7 |
| 2.2.1 | Stochastic Process Approach | 7 |
| 2.2.2 | Fraction-of-Time Approach | 9 |
| 2.2.3 | Cyclic Autocorrelation Function | 10 |
| 2.2.4 | Spectral Correlation Function | 10 |
| 2.3 | Computing Spectral Correlation | 12 |
| 2.3.1 | Frequency Smoothing Method | 12 |
| 2.3.2 | Strip Spectral Correlation Algorithm | 14 |
| 2.3.3 | Matrix Formulation | 17 |
| 3 | Hidden Markov Models for Pattern Recognition | 24 |
| 3.1 | Introduction | 24 |
| 3.2 | Eigenspace Approach to Feature Vector Extraction | 25 |

| | | |
|----------|--|-----------|
| 3.2.1 | Mathematical Basis | 25 |
| 3.2.2 | Example Using OFDM | 27 |
| 3.3 | Hidden Markov Models | 30 |
| 3.3.1 | Three Urns Example | 30 |
| 3.3.2 | General Discrete Case | 32 |
| 3.3.3 | Gaussian Mixtures | 33 |
| 3.3.4 | Configuring HMMs for Pattern Recognition | 34 |
| 3.3.5 | HMM Toolbox | 34 |
| 4 | Wireless Transmitter Simulation | 36 |
| 4.1 | Introduction | 36 |
| 4.2 | Orthogonal Frequency Division Multiplexing | 37 |
| 4.2.1 | WiMAX | 38 |
| 4.3 | Transmitter Component Modeling | 42 |
| 4.3.1 | Filter Modeling | 42 |
| 5 | Specific Emitter Identification | 47 |
| 5.1 | Introduction | 47 |
| 5.2 | Feature Vector Extraction | 48 |
| 5.2.1 | No Models Applied | 48 |
| 5.2.2 | Models Applied | 50 |
| 5.3 | Classification Test | 63 |
| 5.3.1 | Algorithm | 63 |
| 5.3.2 | Simulation and Results | 64 |
| 5.3.3 | Effect of Multipath on Discernability | 65 |
| 6 | Conclusions | 72 |
| 6.1 | Results | 73 |
| 6.2 | Advantages and Limitations | 73 |
| 6.3 | Future Work | 74 |

List of Figures

| | | |
|-----|---|----|
| 1.1 | Depiction of an SEI-capable wireless access point. | 2 |
| 2.1 | Diagram of a spectral correlation analyzer for a given f and α . The center frequency of each BPF determines f | 12 |
| 2.2 | Visualization of the complex demodulates. | 15 |
| 2.3 | Block diagram of the SSCA. | 17 |
| 2.4 | Mapping of the SSCA from k and q to f and α for the case of $N' = 8$ and $N = 32$ | 18 |
| 2.5 | Diagram of the SCM and its attributes. | 21 |
| 2.6 | Plots of the SCM and CSCM for BPSK using square-root raised cosine pulse shaping with values for roll-off parameter r of 0, 0.5 and 1, averaged over 500,000 symbols. | 22 |
| 2.7 | Plots of the SCM and CSCM for QPSK using square-root raised cosine pulse shaping with values for roll-off parameter r of 0, 0.5 and 1, averaged over 500,000 symbols. | 23 |
| 3.1 | Spectral correlation matrix eigenvalues and eigenvectors for OFDM with 8 pilots. | 28 |
| 3.2 | Conjugate spectral correlation matrix eigenvalues and eigenvectors for OFDM with 8 pilots. | 29 |
| 3.3 | Three urns example. | 31 |
| 3.4 | Trellis diagram for the three urns example. | 31 |
| 4.1 | Block diagram of simulation. | 37 |
| 4.2 | Block diagram of an OFDM transmitter | 38 |
| 4.3 | Pilot mapping for a PUSC cluster. | 40 |

| | | |
|-----|---|----|
| 4.4 | Pilot-only spectral plots of the selected 802.16 downlink types. The constant pilot in the FUSC spectrum appears higher powered than the variable pilots due to averaging. | 41 |
| 4.5 | Pseudorandom binary sequence generator used for 802.16 pilots. | 42 |
| 4.6 | Example of a spectral mask. | 43 |
| 4.7 | Spectral masks used for modeling spectrally non-flat RF components. | 45 |
| 5.1 | Legend for eigenvector and feature vector plots. | 50 |
| 5.2 | (a) and (b): Average of the dominant 6 (pilot) eigenvectors of the SCM and CSCM for OFDM downlink using 2.5 MHz bandwidth (256 subcarriers). (c) and (d): Mean and standard deviation of the feature vectors of the SCM and CSCM for OFDM downlink using 2.5 MHz bandwidth. Note that the FV of the CSCM initially appears more unique, and stronger, than that of the SCM. | 51 |
| 5.3 | (a) and (b): Average of the dominant 6 (pilot) eigenvectors of the SCM and CSCM for OFDMA downlink PUSC using 1.25 MHz bandwidth (128 subcarriers). (c) and (d): Mean and standard deviation of the feature vectors of the SCM and CSCM for OFDMA downlink PUSC using 1.25 MHz bandwidth. Here the multi-dimensional effect of the pilot eigenvectors is clearly seen as the pilot spectral energy is distributed across all 6 eigenvectors. | 52 |
| 5.4 | (a) and (b): Average of the dominant 6 (pilot) eigenvectors of the SCM and CSCM for OFDMA downlink FUSC using 1.25 MHz bandwidth (128 subcarriers). (c) and (d): Mean and standard deviation of the feature vectors of the SCM and CSCM for OFDMA downlink FUSC using 1.25 MHz bandwidth. Note, in the case of the SCM, the single constant pilot in the 5th dominant eigenvector and the even distribution of the remaining pilots in the 1st and 2nd dominant eigenvectors. | 53 |
| 5.5 | (a) and (b): Average of the dominant 6 (pilot) eigenvectors of the SCM and CSCM for OFDMA downlink optional FUSC using 1.25 MHz bandwidth (128 subcarriers). (c) and (d): Mean and standard deviation of the feature vectors of the SCM and CSCM for OFDMA downlink optional FUSC using 1.25 MHz bandwidth. Here the pilot spectral energy is distributed evenly across all 6 eigenvectors, in contrast to the previous 3 downlink types. | 54 |
| 5.6 | Mean and standard deviation of the SCM feature vectors of the 6 simulated emitters for OFDM downlink (2.5 MHz bandwidth). Notice that the statistics are unique for each emitter, and they closely follow the spectral masks in Figure 4.7. | 55 |

| | | |
|------|---|----|
| 5.7 | Mean and standard deviation of the CSCM feature vectors of the 6 simulated emitters for OFDM downlink (2.5 MHz bandwidth). Here it can be see that the statistics have greater deviation and lower uniqueness than those of the SCM for OFMD downlink. | 56 |
| 5.8 | Mean and standard deviation of the SCM feature vectors of the 6 simulated emitters for OFDMA downlink PUSC (1.25 MHz bandwidth). Notice the features appear in different eigenvectors depending on the shape of the spectral mask of each emitter. | 57 |
| 5.9 | Mean and standard deviation of the CSCM feature vectors of the 6 simulated emitters for OFDMA downlink PUSC (1.25 MHz bandwidth). Again, uniqueness is seen between the statistics of the emitters, but it is far less than for the case of the SCM for OFDMA downlink PUSC. | 58 |
| 5.10 | Mean and standard deviation of the SCM feature vectors of the 6 simulated emitters for OFDMA downlink FUSC (1.25 MHz bandwidth). Since the majority of the features are constricted to the 2 dominant eigenvectors, the uniqueness of the statistics is diminished compared to the case when the features are distributed throughout all of the eigenvectors. | 59 |
| 5.11 | Mean and standard deviation of the CSCM feature vectors of the 6 simulated emitters for OFDMA downlink FUSC (1.25 MHz bandwidth). The deviation is higher than the SCM case, however the pilot features are distributed through the eigenvectors. | 60 |
| 5.12 | Mean and standard deviation of the SCM feature vectors of the 6 simulated emitters for OFDMA downlink optional FUSC (1.25 MHz bandwidth). A much high deviation is seen in the statistics of the feature vectors, however they are unique for each emitter and distributed evenly throughout the eigenvectors. | 61 |
| 5.13 | Mean and standard deviation of the CSCM feature vectors of the 6 simulated emitters for OFDMA downlink optional FUSC (1.25 MHz bandwidth). These statistics have high deviation and low uniqueness. | 62 |
| 5.14 | Block diagram of HMM training for a single simulated emitter (RF model). | 63 |
| 5.15 | Block diagram of HMM training (emitter classification) for a single simulated emitter (RF model). | 64 |
| 5.16 | Confusion matrices for SEI classification tests of OFDM downlink with 10 dB and 0 dB SNR using SCM only. Results are excellent except for emitter 1 in the 0 dB SNR case. | 66 |

| | | |
|------|---|----|
| 5.17 | Confusion matrices for SEI classification tests of OFDM downlink with 10 dB and 0 dB SNR using SCM and CSCM. For the 0 dB SNR case there is a slight ambiguity between emitter 3 and emitter 6. | 66 |
| 5.18 | Confusion matrices for SEI classification tests of OFDMA downlink PUSC with 10 dB and 0 dB SNR using SCM only. Ambiguity between emitter 1 and emitter 2 is apparent. | 67 |
| 5.19 | Confusion matrices for SEI classification tests of OFDMA downlink PUSC with 10 dB and 0 dB SNR using SCM and CSCM. Note that results due to the lack of uniqueness between SCM feature vectors extracted from emitters 1 and 2 were worsened by the non-uniqueness of the CSCM feature vectors. | 67 |
| 5.20 | Confusion matrices for SEI classification tests of OFDMA downlink FUSC with 10 dB and 0 dB SNR using SCM only. Performance is acceptable for all but emitter 4 at 10 dB SNR. For the 0 dB SNR case performance is generally poor. | 68 |
| 5.21 | Confusion matrices for SEI classification tests of OFDMA downlink FUSC with 10 dB and 0 dB SNR using SCM and CSCM. Noting the overall poor performance, the non-uniqueness in the feature vectors of both the SCM and CSCM is clearly apparent. | 68 |
| 5.22 | Confusion matrix for SEI classification test of OFDMA downlink optional FUSC with 10 dB and 0 dB SNR using SCM only. Nearly perfect classification was obtained in both cases. | 69 |
| 5.23 | Confusion matrices for SEI classification tests of OFDMA downlink optional FUSC with 10 dB and 0 dB SNR using SCM and CSCM. Perfect classification was obtained in both cases. | 69 |
| 5.24 | Confusion matrix for SEI classification test of OFDMA downlink optional FUSC with 10 dB and 0 dB SNR using SCM only. The SUI-3 multipath propagation channel model was applied to each snapshot prior to classification. | 71 |

List of Tables

| | | |
|-----|---|----|
| 1.1 | Literature Survey on SEI | 4 |
| 3.1 | Pertinent HMM Toolbox Functions | 35 |
| 4.1 | Global WiMAX Parameters | 39 |
| 4.2 | Selection of 802.16 Downlink Types | 40 |
| 4.3 | Spectral Mask Parameters (Normalized Frequency) | 44 |
| 5.1 | Eigenvector to Feature Vector Mapping | 49 |
| 5.2 | SUI-3 Channel Model Parameters | 70 |

Chapter 1

Introduction

1.1 Motivation

Specific emitter identification (SEI) is the process of identifying a wireless transmitter based on the statistics of its physical RF waveform. Much like the fact that no two human fingerprints are alike, although many may appear similar or nearly identical, the process relies on the assumption that physical attributes of the transmitter itself affect the transmitted communication signal in such a way that no two RF waveforms are completely identical. However, while it may be true that every wireless device affects its own transmitted waveform in such a way as to be unique, it has not yet been shown, to the knowledge of the author, that these differences are large enough to be measurable using known techniques. So, while the term SEI may imply the ability to find a unique "fingerprint" for any given wireless device, this is not usually the case. However, SEI has been shown [1–3] to be useful for identifying broader classes of devices, such as the vendor of a device, or even the model of a device by a specific vendor. The work contained in this thesis does not examine the ability of SEI to uniquely identify a single device, but rather, it is rather purely evaluative of the degree to which several devices could potentially be distinguished given certain requirements on their construction.

Although performing SEI accurately has many applications, one important practical application for SEI is wireless security. While a wireless base station or access point, such as a Wi-Fi hotspot, may have built-in security measures like passwords, encryption keys etc., it may not have any method to protect against a user who is not authorized for access, but has obtained (maliciously or otherwise) the passwords and encryption keys which are normally used to prevent access. In such a scenario, SEI may allow the access point to differentiate an unauthorized user from an authorized user and allow or deny access accordingly. This is especially true if the unauthorized user is a spoofer, posing as a legitimate wireless device with non-standard or custom hardware and software.

A hypothetical environment for an SEI wireless access point is shown in Figure 1.1. Possible devices in the environment include known authorized devices (devices for which information is known a priori, and which are allowed access to the wireless access point), known unauthorized devices (that is a device for which information is known a priori, but which is denied access to the wireless access point), unknown devices (devices for which there is no a priori information, but it may be desirable to collect this information), and adjacent wireless access points (another access point for which a priori information is known) which may be more common in ad hoc wireless networks, but also may be easier to spoof.

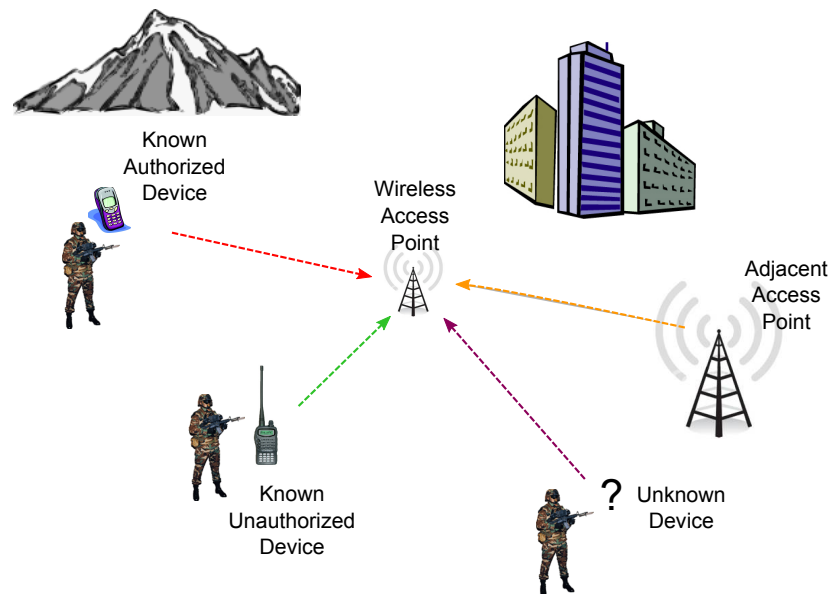


Figure 1.1: Depiction of an SEI-capable wireless access point.

One real world application for SEI is the environment of femtocells. Femtocells are home base stations for cellular communication devices, and as such they allow the user of a cellular device to enjoy a reliable connection whenever in the vicinity of the femtocell [4]. The behavior of a femtocell needs to be such that it mimics the behavior of larger macrocells while providing access to a much smaller footprint. Some femtocells, open femtocells, are configured such that they provide access for any network user within range. Other types, closed femtocells, provide access only for a small group of authorized users while denying access to all other users in the area. Since most cellular communication standards have no inherent access control scheme (other than on a network level), SEI may provide closed femtocells an additional means of separating users for which they were intended to provide enhanced service, from all other users. For open femtocells, SEI may help differentiate between outside users and the primary set of users, who are likely paying for its use, thus allowing the femtocell to give quality of service (QoS) priority those primary users when traffic due to outside users is particularly high.

Another potential real world application is the field of cognitive radio (CR), which has recently become a hot topic in the academic and wireless research communities. The concept of a cognitive radio entails a radio which is capable of making decisions based on parameters which it has collected from the surrounding environment. One of the more commonly proposed applications for CRs is dynamic spectrum access (DSA) [5], where so-called white space devices are permitted to transmit in unused portions of licensed spectrum. Although these devices are not licensed users of the spectrum, they are theoretically able to operate without impacting the primary licensed users by detecting when they are transmitting and ceasing transmission. The white space devices may even be capable of learning about which primary users tend to occupy certain bandwidths in a particular location. However, these devices, and CRs in general, are prone to security issues. Specifically, a white space device may be forced to cease transmitting if another wireless device posing as a licensed user begins transmitting immediately after the white space device begins transmitting causing it to falsely detect a primary user and shut off. This cycle can be repeated indefinitely causing a denial of service (DoS) attack. The addition of spectrum sensing and SEI technologies may help to mitigate this and other CR-related security holes [6].

What allows CRs to learn about their operating environment is the fact that they use a variety of sensing techniques. This includes characterizing and classifying any nearby wireless transmitters using techniques such as energy detection, modulation classification, position location, propagation channel measurement, and others. These techniques can be useful if the surrounding environment behaves in a way that the transmitter is unaware the CR is gathering this information. However, it has been shown [7] that legitimate transmitters can be misclassified if a CR attempts to blindly classify signals in its environment while a malicious transmitter is present. By adding SEI to list of techniques used to classify signals, a CR may add an extra dimension of robustness to environment awareness security problems by helping to differentiate between benign and malicious transmitters.

One reason that WiMAX was chosen for scrutiny in this work is due to its near-ubiquitous use of OFDM (or OFDMA) for data transmission. It has been shown [8] that channel-like fingerprints can be embedded in OFDM without destructively interfering with data transmission rates. Another reason that OFDM was chosen is that past research [1–3] has shown that the modulation scheme lends itself to performing SEI using cyclostationary analysis. It was out of this previous work that the inspiration for the eigendecomposition-based cyclostationary spectral analysis technique presented in this work arose.

1.2 Literature Review

Traditionally, SEI has implied the ability to identify radar transmitters [9], but more recently has been extended to wireless digital communication devices. Previously established SEI algorithms involve measuring time, frequency and phase-space features, such as the transient response of push-to-talk radios [10] or the RF filter shape of other devices. The wavelet

transform has also been proposed for the analysis of short-time-duration features [11]. More recent findings suggest that second-order spectral correlation features may be used to successfully, and reliably, classify a set of Wi-Fi network cards [2, 3]. This is backed up by further evidence that suggests Wi-Fi network cards may exhibit unique electromagnetic signatures [12]. It is with this perspective in mind that the use of second order cyclostationary analysis is proposed for extracting features for SEI, especially when used in conjunction with the EVD, of OFDM-based transmitters. A brief literature survey of publications on the topic of SEI is contained in Table 1.1.

Table 1.1: Literature Survey on SEI

| Author(s) | Year | Title |
|--------------------|------|---|
| L. E. Langley | 1993 | Specific emitter identification (SEI) and classical parameter fusion technology |
| A. M. Abdulla | 1996 | Identification of push-to-talk transmitters using wavelets and spectral correlation (Thesis) |
| K. I. Talbot et al | 2003 | Specific emitter identification and verification |
| K. A. Remley et al | 2005 | Electromagnetic signatures of WLAN cards and network security |
| K. W. Kim | 2008 | Exploiting Cyclostationarity for Radio Environmental Awareness in Cognitive Radios (Dissertation) |
| C. M. Spooner | 2008 | Specific emitter identification for modern communication signals |
| K. W. Kim et al | 2008 | Specific emitter identification for cognitive radio with application to IEEE 802.11 |

1.3 Organization

The remainder of this thesis is organized as follows. Chapter 2 covers a brief background necessary for understanding the application of second-order spectral correlation features of cyclostationary signals for the identification of such signals. Additionally, this chapter discusses algorithms for efficient digital computation of the 2nd order cyclic spectrum as well as a linear algebra approach which arranges the output of a cyclic spectrum analyzer in a matrix. Chapter 3 briefly introduces the eigenvector/eigenvalue decomposition, and a proposed method for how feature vector might be extracted from the eigenvectors of a spectral correlation matrix for pattern recognition. This chapter also introduces hidden Markov models (HMMs) and discusses their use in pattern recognition. Chapter 4 details several techniques for simulating the different components of a wireless transmitter relevant to performing SEI,

from generating OFDM modulation with pilots to simulating RF components. Chapter 5 details a full SEI simulation including the training of several HMMs and the results, in confusion matrices, of the testing of various WiMAX downlink types. Finally, Chapter 6 summarizes the results of the simulations, discusses the advantages and limitations of the approach to SEI as well as a discussion of possible future work.

1.4 Contributions

The overall goal of this thesis is to evaluate the performance of an eigendecomposition-based specific emitter identification technique for a variety of OFDM modulation schemes with variable conditions. From the simulations contained herein, in order to meet this goal, two main contributions have stemmed. The first main contribution of this thesis is to show an alternative linear algebra method of computing a 2nd order cyclic spectrum which organizes the output in a so-called spectral correlation matrix. The second main contribution is to show the utility of a new method of specific emitter identification which employs an eigendecomposition on this spectral correlation matrix for OFDM signals with strong correlated pilots.

1.5 Publications

- P. H. Sahmel, J. H. Reed, "Eigenspace Approach to Specific Emitter Identification", Proceedings of the 6th Karlsruhe Workshop on Software Radios, Karlsruhe, Germany, March 2010
- P. H. Sahmel, J. H. Reed, C. M. Spooner, "Eigenspace Approach to Specific Emitter Identification", *Frequenz, Journal of RF-Engineering and Telecommunications*, Vol. 64, November/December 2010
- (Expected 2012) P. H. Sahmel et al., "Specific Emitter Identification of Multi-pilot OFDM Signals"

Chapter 2

Cyclostationarity

2.1 Introduction

Many modern communications signals exhibit the property of cyclostationarity, especially those which are digital in nature. The philosophy behind the need for cyclostationary processing in digital communications is that, though all digitally-keyed signals are repetitive, they may not be periodic. Simply examining the autocorrelation function or the power spectrum of a binary phase shift keyed (BPSK) signal, for example, will only reveal some information about its symbol rate. However, by invoking the cyclic autocorrelation function, we find that periodicities do arise for cycle frequencies related to the symbol rate. Since the power spectral density is defined as the Fourier transform of the autocorrelation function, the cyclic spectrum is the Fourier transform of the cyclic autocorrelation function. Any additive sine-wave components which occur due to periodicity in the autocorrelation function at a given cycle frequency will be revealed in the cyclic spectrum.

The mathematical framework for Cyclostationarity was developed over the last 50 years by Gardner and others [13]. The seminal inspiration for the concept of cyclostationarity as a model for communications signals came from work by Bennett in 1958 [14] and Franks in 1969 [15]. By 1994 a book (edited by Gardner) [16] covering the use of cyclostationarity in multiple aspects of communications and signal processing had been published. Thus, a great deal of research on the topic of cyclostationarity has been done in the past, with most of the great milestones having been already set. However, this does not mean that new technical aspects are no longer uncovered. Many new applications of cyclostationary signal processing have to do with using previously established functions to solve newer problems, such as signal detection and characterization, signal classification, and emitter identification. Previous attempts to demonstrate the utility of the spectral correlation function for specific emitter identification have proven successful in the past [2,17], and it is out of these successes that the motivation to continue to find methods of using cyclostationary-based emitter identification

techniques arises.

While this chapter does not contain a comprehensive history of the theory of cyclostationarity, it does lay the groundwork necessary for computing the second-order spectral correlation function, an important aspect of cyclostationary-based of emitter identification. The remainder of this chapter is organized as follows. First, the elementary definitions for functions of a cyclostationary signal are given in Section 2.2. Section 2.3 covers methods of computing second order spectral correlation, with a focus on digital computation as well as efficient algorithms for these methods. This section also introduces the concept of a linear algebra-based method of computing spectral correlation, one which outputs the spectral correlation function as a matrix.

2.2 Definitions

A cyclostationary process is one whose statistics, such as mean and autocorrelation, vary cyclicly with time [18]. While such a process may not be periodic, in some cases its autocorrelation function may have one or multiple periods. Since the concept of cyclostationarity is rooted in the analysis of stochastic processes, it is helpful to begin with classical statistical definitions and then move into those associated with cyclostationary signals. The classic approach in stochastic signal processing is to define a random process as a random variable as a function of time, and then invoke ergodicity. However, cyclostationary definitions come directly from the fraction-of-time probabilistic approach, where ergodicity is avoided. From these definitions come the equations for the cyclic autocorrelation function and the spectral correlation function, which are the basic tools for analysis of cyclostationary signals.

2.2.1 Stochastic Process Approach

The definition of a stochastic process uses definitions from the classical theory of statistics and probability. This means that any stochastic process must be defined by an underlying probability distribution and density. This allows for the various moments and higher order functions to be defined.

A continuous-time real-valued stochastic process $X(t)$ with time-translation parameter t has probability distribution function (PDF) [13, 18–20]

$$F_{X(t)}(x) = \Pr(X(t) < x), \quad (2.1)$$

where t is real, x is real and $0 \leq F_{X(t)}(x) \leq 1$. Now if we let $\mathbf{x} = \{x_{t_1}, x_{t_2}, \dots, x_{t_n}\}$ correspond to a set of random variables

$$\mathbf{X}(t) = \{X(t + t_1), X(t + t_2), \dots, X(t + t_n)\} \quad (2.2)$$

with time offsets t_1, t_2, \dots, t_n we can define a joint probability distribution function for $\mathbf{X}(t)$ as

$$F_{X(t+t_1)\dots X(t+t_n)}(x_{t_1}, x_{t_2}, \dots, x_{t_n}) = \Pr(X(t + t_1) < x_{t_1}, X(t + t_2) < x_{t_2}, \dots, X(t + t_n) < x_{t_n}), \quad (2.3)$$

or more succinctly

$$F_{\mathbf{X}(t)}(\mathbf{x}) = \Pr(X(t + t_1) < x_{t_1}, X(t + t_2) < x_{t_2}, \dots, X(t + t_n) < x_{t_n}). \quad (2.4)$$

The probability density function (PDF) is then given by the partial derivative with respect to each of the parameters $\{x_{t_1}, x_{t_2}, \dots, x_{t_n}\}$ as

$$f_{\mathbf{X}(t)}(\mathbf{x}) = \frac{\partial^n}{\partial x_{t_1} \dots \partial x_{t_n}} F_{\mathbf{X}(t)}(\mathbf{x}). \quad (2.5)$$

We can now make the following definitions (for any n) with regards to stationary and cyclostationary processes [18]:

Definition 1. *The stochastic process $X(t)$ is a stationary (in the strict sense) process if and only if $F_{\mathbf{X}(t)}(\mathbf{x})$ is independent of the time-translation parameter t .*

Definition 2. *The stochastic process $X(t)$ is a cyclostationary (in the strict sense) process if and only if $F_{\mathbf{X}(t)}(\mathbf{x})$ is periodic in t with period T .*

The stochastic process $X(t)$ has also has mean and autocorrelation functions, which are defined as [20]

$$m_X(t) = \mathbb{E}\{X(t)\} = \int_{-\infty}^{\infty} x_t f_{X(t)}(x_t) dx_t \quad (2.6)$$

and

$$R_X(t, \tau) = \mathbb{E}\{X(t)X(t + \tau)\} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x_{t_1} x_{t_2} f_{\mathbf{X}(t)}(x_{t_1}, x_{t_2}) dx_{t_1} dx_{t_2}. \quad (2.7)$$

where $t = t_1$ and $\tau = t_2 - t_1$. This stochastic process is said to be cyclostationary in the wide sense if its mean and autocorrelation function are periodic with some period T_0

$$m_X(t + T_0) = m_X(t), \quad (2.8)$$

$$R_X(t + T_0, \tau) = R_X(t, \tau). \quad (2.9)$$

This is still true if the stochastic process is periodic with multiple periods.

2.2.2 Fraction-of-Time Approach

The fraction-of-time (FOT) probability framework stems from the lack of a need for a stochastic process framework in most engineering systems which involve the processing of a time series. The FOT approach works particularly well for communications systems because in almost all cases the output of a system will be a cyclostationary finite-duration time series. By taking an FOT approach to defining cyclostationary functions there is an assumption that a system output can be adequately modeled using infinite time averages. As long as this assumption remains in effect, Gardner argues [18] that an FOT probability framework can be developed alongside the stochastic process framework to the point that the latter is no longer necessary.

In the case of the FOT probability approach, the time-averaging operator $\langle \cdot \rangle$ is frequently used, which can be defined simply by letting the averaging period T tend to infinity in the finite time-averaging function $\langle \cdot \rangle_T$

$$\langle \cdot \rangle = \lim_{T \rightarrow \infty} \langle \cdot \rangle_T. \quad (2.10)$$

For the continuous-time case, the finite time-averaging function is defined as

$$\langle \cdot \rangle_T = \frac{1}{2T} \int_{-T}^T (\cdot), \quad (2.11)$$

and in the discrete-time case is defined as

$$\langle \cdot \rangle_T = \frac{1}{2T} \sum_{t=-T}^T (\cdot). \quad (2.12)$$

Using the time-averaging operator makes sense from an engineering perspective, and it also overcomes pitfalls which would otherwise arise in developing the remaining stochastic process framework for analysis of cyclostationary signals. Thus, for the remainder of this work we use this time-averaging operator when referring to a non-deterministic time series $x(t)$. Since an infinite time average cannot be computed in practice, the finite time-averaging function is often used in its place to compute an estimate.

2.2.3 Cyclic Autocorrelation Function

A cyclostationary signal will generally contain second-order periodicities. One second-order transformation commonly found in statistics is the autocorrelation function, which produces the time-delayed inner product of a signal, and is related to the variance. However, since cyclostationary signals have statistics which are periodic with some period T_0 , it is necessary to add a frequency shift α to the autocorrelation function. This leads to the cyclic autocorrelation function (CAF). The CAF of a continuous or discrete, real- or complex-valued signal $x(t)$ is defined as

$$R_{xx}^{\alpha}(\tau) = \left\langle (x(t)e^{-j\pi\alpha t}) (x(t-\tau)e^{j\pi\alpha(t-\tau)})^* \right\rangle, \quad (2.13)$$

which can be rewritten as

$$R_{xx}^{\alpha}(\tau) = \left\langle x(t)x^*(t-\tau)e^{-j2\pi\alpha t} \right\rangle e^{j\pi\alpha\tau}. \quad (2.14)$$

We see that the CAF simply reduces to the autocorrelation function for the case when $\alpha = 0$

$$R_{xx}^0(\tau) = \left\langle x(t)x^*(t-\tau) \right\rangle, \quad (2.15)$$

thus the CAF can be interpreted simply as an autocorrelation with an additional frequency shift α .

The conjugate cyclic autocorrelation function (CCAF), useful for analyzing complex-valued cyclostationary signals, can be easily obtained by replacing $x(t-\tau)$ in (2.14) with $x^*(t-\tau)$. Thus the equation for the CCAF is

$$R_{xx^*}^{\alpha}(\tau) = \left\langle x(t)x(t-\tau)e^{-j2\pi\alpha t} \right\rangle e^{j\pi\alpha\tau}. \quad (2.16)$$

The CCAF is especially useful for baseband signals where the non-conjugate CAF does not reveal all of the second order periodicities contained in a signal, such as BPSK and MSK.

2.2.4 Spectral Correlation Function

Cyclic features present in a cyclostationary signal can be observed directly by examining its cyclic spectrum. The aptly named spectral correlation function is the Fourier transform of the cyclic autocorrelation function (SCF), otherwise called the cyclic Wiener-Khinchin relation (or the cyclic Wiener relation). For a real or complex-valued continuous time series $x(t)$, the SCF is defined as

$$S_{xx}^\alpha(f) = \int_{-\infty}^{\infty} R_{xx}^\alpha(\tau) e^{-j2\pi f\tau} d\tau. \quad (2.17)$$

Because the SCF is a function of both the parameters f and α , it is defined on a two dimensional plane, sometimes called the bifrequency plane. The conjugate SCF (CSCF) can be obtained by applying the cyclic Wiener-Khinchin relation to the conjugate CAF, $R_{xx^*}^\alpha(\tau)$, that is

$$S_{xx^*}^\alpha(f) = \int_{-\infty}^{\infty} R_{xx^*}^\alpha(\tau) e^{-j2\pi f\tau} d\tau. \quad (2.18)$$

When $x(t)$ is a complex-valued baseband time series, both the non-conjugate and conjugate versions of the SCF are necessary, in general, to reveal all of the second-order cycle frequencies.

An alternative definition for the SCF, one which does not depend on the CAF, is [13]

$$S_{xx}^\alpha(f) = \lim_{\Delta f \rightarrow 0} \lim_{T \rightarrow \infty} \frac{1}{T} \int_{-T/2}^{T/2} \Delta f X_{1/\Delta f}(t, f + \alpha/2) X_{1/\Delta f}^*(t, f - \alpha/2) dt, \quad (2.19)$$

where $X_Z(t, f)$ is defined as

$$X_Z(t, f) = \int_{t-Z/2}^{t+Z/2} x(s) e^{-j2\pi fs} ds. \quad (2.20)$$

One method [18] for interpreting this definition of the SCF is to use an infinitesimally narrowband filter convolved with two frequency shifted versions of the signal $x(t)$, one shifted up by $\alpha/2$ and the other shifted down by $\alpha/2$. These can be denoted as $u(t)$ and $v(t)$, such that

$$u(t) = x(t) e^{-j\pi\alpha t}, \quad (2.21)$$

$$v(t) = x(t) e^{+j\pi\alpha t}. \quad (2.22)$$

Then the equation for the SCF can be written as

$$S_{xx}^\alpha(f) = \lim_{B \rightarrow 0} \frac{1}{B} \left\langle \left[h_B^f(t) \otimes u(t) \right] \left[h_B^f(t) \otimes v(t) \right]^* \right\rangle, \quad (2.23)$$

where $h_B^f(t)$ are the filter coefficients of a bandpass filter with center frequency f and bandwidth B .

2.3 Computing Spectral Correlation

The second order spectral correlation function, sometimes called the cyclic spectrum, is an easy to compute function which has the ability to reveal far more about a cyclostationary signal than its PSD. The cyclic spectrum can be computationally expensive if computed for the full range of α , therefore some algorithms exist which make its computation efficient. Such an algorithm may be referred to as a spectral correlation analyzer (or algorithm). A spectral correlation analyzer is an algorithm for estimating spectral correlation of some input signal over a finite time duration [18]. A block diagram for an example spectral correlation analyzer is shown in Figure 2.1.

Several methods for computing the cyclic spectrum are detailed below. The frequency smoothing method is ideal for computing the cyclic spectrum over a short time duration for a small number of cycle frequencies. The strip spectral correlation algorithm is a time smoothing method of computing the cyclic spectrum for a full range of cycle frequencies, but done efficiently using the FFT. Finally, the matrix formulation is a time smoothing method which computes the cyclic spectrum for a full range of temporal and cycle frequencies (f and α respectively), but in such a way that it is arranged in a square matrix, with the PSD along the diagonal.

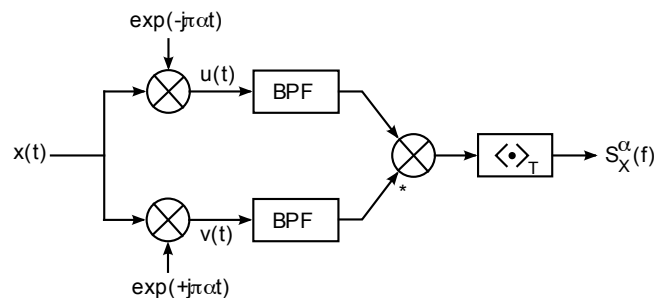


Figure 2.1: Diagram of a spectral correlation analyzer for a given f and α . The center frequency of each BPF determines f .

2.3.1 Frequency Smoothing Method

The first approach for computing spectral correlation is the frequency smoothing method (FSM) [21]. In general, the FSM is desirable when it is preferred to compute spectral correlation for a small subset of α instead of over the range of all α . To compute the spectral correlation function would require averaging over an infinite time period with infinitely narrow frequency resolution, thus in practice it can only be estimated, similarly to the power spectrum. The FSM uses the cyclic periodogram to estimate the spectral correlation function, analogously to how the periodogram is used to generate an estimate of the power

spectral density function.

We first examine the periodogram. A complex-valued, discrete-time (that is, sampled) signal $x(n)$ has discrete Fourier transform (DFT)

$$X(f) = \sum_{n=0}^{N-1} x(n)e^{-j2\pi fn}, \quad (2.24)$$

and its periodogram is defined as

$$I(f) = \frac{1}{N} |X(f)|^2. \quad (2.25)$$

For deterministic signals, the periodogram is a good measure of energy at a discrete frequency. However, for stochastic signals, the periodogram is not an accurate estimate of the true PSD. Fortunately, the PSD can be accurately estimated over a relatively short period by performing smoothing of the periodogram in the frequency domain. Thus we have the PSD estimate

$$\hat{S}(f) = g(f) \otimes I(f), \quad (2.26)$$

where $g(f)$ is a unit-area window function and \otimes denotes convolution.

In the same way, the cyclic periodogram can be calculated by adding a cyclic shift α to the periodogram

$$I^\alpha(f) = \frac{1}{N} X(f + \alpha/2)X^*(f - \alpha/2). \quad (2.27)$$

Since the DFT takes on discrete values of f , the cyclic periodogram is limited to the discrete values of the cyclic shift parameter $\alpha = k/N$, where $k = 0, \dots, N - 1$. Thus, the resolution in the cyclic domain is limited to $1/N$. An estimate of the spectral correlation function can then be obtained by smoothing the cyclic periodogram using [21]

$$\hat{S}_{xx}^\alpha(f) = g(f) \otimes I^\alpha(f) \quad (2.28)$$

$$= \frac{1}{N} \sum_{n=0}^{N-1} g(f - n/N)X(n/N + \alpha/2)X^*(n/N - \alpha/2). \quad (2.29)$$

In the same way, an estimate of the conjugate spectral correlation function, $\hat{S}_{xx^*}^\alpha(f)$, can be computed by replacing $I^\alpha(f)$ in (2.28) with the conjugate cyclic periodogram [21]

$$I_*^\alpha(f) = \frac{1}{N} X(f + \alpha/2) X(\alpha/2 - f). \quad (2.30)$$

The advantage of computing spectral correlation using the frequency smoothing method is that it can be implemented with an FFT, complex multiply, and convolution. Additionally, in the special case that the windowing function is rectangular and has unit area, the convolution of $g(f)$ with the cyclic periodogram can be computed at low cost. It can be shown [21] that this convolution then becomes a running sum, reducing the complexity. So for the spectral correlation function we have

$$\hat{S}_{xx}^\alpha(f) = \sum_{r=f-M/2}^{f+M/2} g(f-r) I^\alpha(r) \quad (2.31)$$

$$= \frac{1}{M+1} \sum_{r=f-M/2}^{f+M/2} I^\alpha(r), \quad (2.32)$$

where M is the length of the smoothing window. The sum in (2.32) can be computed efficiently by maintaining a running sum of the cyclic periodogram. That is, the sum does not need to be recomputed separately for each value of f , but rather (so long as it is done consecutively) can be computed by subtracting an old value and adding a new one.

2.3.2 Strip Spectral Correlation Algorithm

The strip spectral correlation algorithm (SSCA), sometimes also referred to as an analyzer, is an efficient algorithm for computing spectral correlation for the full range of cycle frequencies [22–24]. Unlike the frequency smoothing method, the SSCA is a time smoothing algorithm, and as such, it is useful for computing a potentially more crude estimate of the entire cyclic spectrum over a similar time duration. It is derived by reorganizing the definition for the time-smoothed cyclic periodogram which allows it to be computed using short and long discrete Fourier transforms. The short DFT channelizes the data while the long DFT performs smoothing over time. Additionally, the DFT can be computed efficiently using the FFT, resulting in a particularly fast method of computing spectral correlation (with some caveats).

The following definitions assume $x(n)$ to be a complex-valued discrete-time signal. We first define the time-smoothed algorithm for computing spectral correlation, which is derived from the time-smoothed cyclic periodogram [25]

$$S_{xx_T}^\alpha(n, f)_{\Delta t} = \langle X_T(n, f + \alpha/2) X_T^*(n, f - \alpha/2) \rangle_{\Delta t}, \quad (2.33)$$

where Δt is the averaging period and $X_T(n, f)$ are the complex demodulates of $x(n)$. These are defined as

$$X_T(n, f) = \sum_{r=-N'/2}^{N'/2-1} a(r)x(n-r)e^{-j2\pi f(n-r)T_s}. \quad (2.34)$$

Here, the complex demodulates are defined using a data tapering window $a(n)$, which, for the purposes of the simulations contained in this work, is assumed to be a Hamming window. However, any windowing function may be used. The time duration of the data tapering window is $T = N'T_s$, where N' is the number of demodulates and $T_s = 1/f_s$ is the sampling period of $x(n)$. A pictorial example of the demodulates are shown in Figure 2.2 [26].

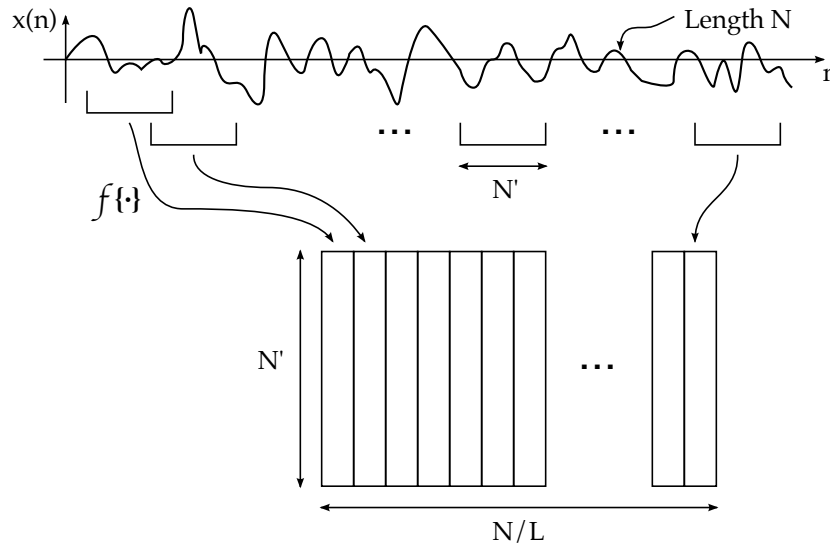


Figure 2.2: Visualization of the complex demodulates.

Since the cyclic periodogram is a finite time average, it is only an estimate of the spectral correlation function. However, this estimate improves as the averaging time Δt increases, and as the demodulate bandwidth $\Delta f = 1/T$ decreases. Thus, as $\Delta t \rightarrow \infty$ and $T \rightarrow \infty$ the time-smoothed cyclic periodogram should approach spectral correlation function $S_{xx}^\alpha(f)$ [25], that is

$$\lim_{T \rightarrow \infty} \lim_{\Delta t \rightarrow \infty} S_{xx_T}^\alpha(n, f)_{\Delta t} = S_{xx}^\alpha(f). \quad (2.35)$$

The time-smoothed periodogram can be interpreted as the time-average of the correlation of spectral components of the demodulates of $x(n)$ separated by a particular cyclic shift. Thus, for a time-smoothed estimate of the SCF averaged over N samples, we have [27]

$$S_{xx_T}^{\alpha}(n, f)_{\Delta t} = \sum_{m=-N'/2}^{N'/2-1} X_T\left(n+m, f+\frac{\alpha}{2}\right) X_T^*\left(n+m, f-\frac{\alpha}{2}\right) g(m) \quad (2.36)$$

where $g(t)$ is a unit area weighting function, such that $\sum_n g(n) = 1$. In order to compute the time-smoothed cyclic periodogram efficiently, it is desired to extend (2.36) to the case where it can be computed using the FFT. First note that the complex demodulates can be manipulated such that

$$X_T(n, f) = \left[\sum_{r=-\frac{N'}{2}}^{\frac{N'}{2}-1} a(r)x(n+r)e^{-j2\pi frT_s} \right] e^{-j2\pi fnT_s} \quad (2.37)$$

where the demodulate bandwidth $\Delta f = f_s/N'$. By inspection, it is apparent that the portion of (2.37) inside the brackets is a discrete Fourier transform, and thus can be computed efficiently using an N' -point windowed FFT.

It is also possible to extend (2.36) such that it can be computed using an N -point FFT. This requires a modification such that

$$S_{xx_T}^{\alpha_i+q\Delta\alpha}(n, f_j)_{\Delta t} = \sum_{m=-\frac{N}{2}}^{\frac{N}{2}-1} X_T(n+m, f_k) X_T^*(n+m, f_l) g(m) e^{-\frac{j2\pi mq}{N}} \quad (2.38)$$

where the following relationships are used

$$f_j = \frac{f_k + f_l}{2} = \frac{k+l}{2} \left(\frac{f_s}{N'} \right), \quad (2.39)$$

$$\alpha_i = f_k - f_l = (k-l) \left(\frac{f_s}{N'} \right). \quad (2.40)$$

The SSCA further reduces the complexity of (2.38) by setting $\alpha_i = f_k$, $f_j = (f_k - q\Delta\alpha)/2$, and by replacing the second demodulate, $X_T^*(n+m, f_l)$, with the original time-series $x^*(n+m)$ [22, 27]. This leads to

$$S_{xx_T}^{f_k+q\Delta\alpha}\left(n, \frac{f_k + q\Delta\alpha}{2}\right)_{\Delta t} = \sum_{m=-\frac{N}{2}}^{\frac{N}{2}-1} X_T(n+m, f_k) x^*(n+m) g(m) e^{-\frac{j2\pi qm}{N}}. \quad (2.41)$$

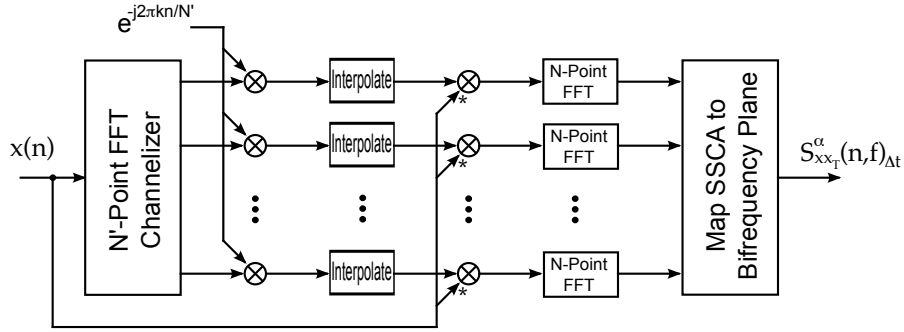


Figure 2.3: Block diagram of the SSCA.

where k is related to f_k by $f_k = k(f_s/N')$ for $k = -N'/2, \dots, (N'/2) - 1$. A block diagram representing the flow of data in the SSCA is depicted in Figure 2.3.

Since the SSCA as arranged in (2.41) is a function of q and k , it must be mapped back to the bifrequency plane as a function of f and α in order to be interpreted as the SCF. This mapping from k and q to f and α is given by

$$f = \frac{f_k - q\Delta\alpha}{2}, \quad (2.42)$$

$$\alpha = f_k + q\Delta\alpha, \quad (2.43)$$

where

$$k = -\frac{N'}{2}, \dots, \frac{N'}{2}, \quad (2.44)$$

$$q = -\frac{N}{2}, \dots, \frac{N}{2} \quad (2.45)$$

This mapping to the bifrequency plane is shown in Figure 2.4 [27]

It is also possible to generate an estimate of the conjugate SCF using the SSCA by replacing $x^*(n+m)$ with $x(n+m)$ in (2.41).

2.3.3 Matrix Formulation

The spectral correlation function can be formulated using matrix notation. First, it is necessary to define the windowed complex demodulates using a matrix (or snapshot) notation.

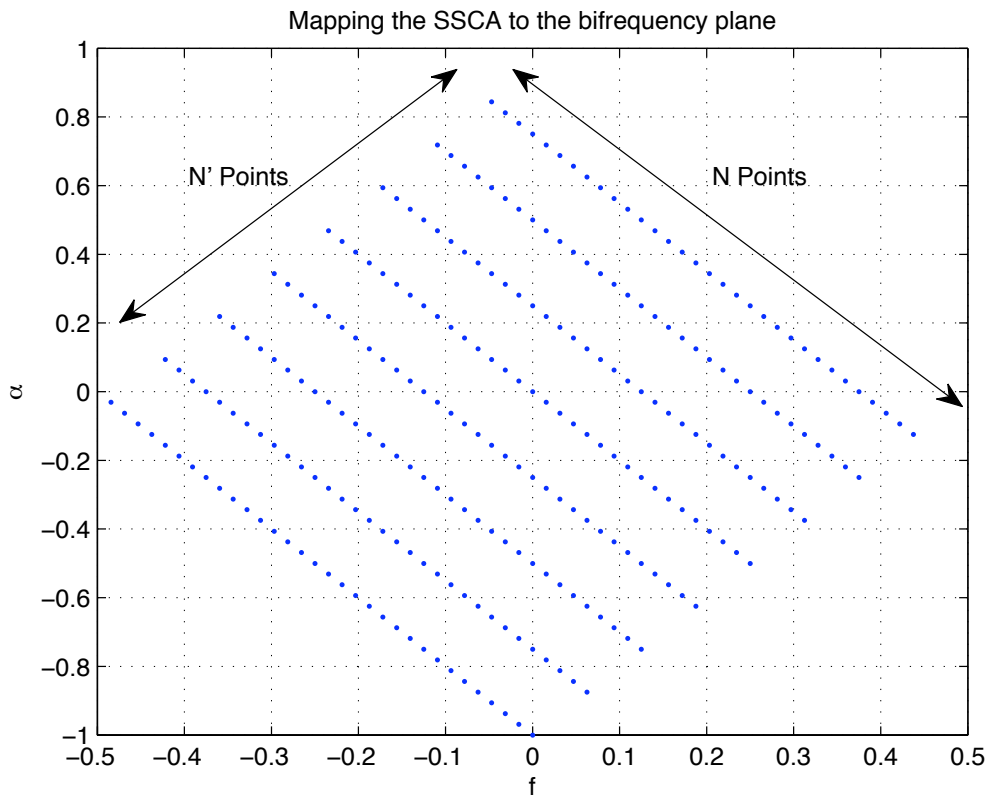


Figure 2.4: Mapping of the SSCA from k and q to f and α for the case of $N' = 8$ and $N = 32$.

Recall that the complex demodulates of a discrete-time signal $x(n)$ of length $N + N' - 1$ are defined as

$$X_T(n, f) = \sum_{r=-N'/2}^{N'/2-1} a(r)x(n-r)e^{-j2\pi f(n-r)T_s}, \quad (2.46)$$

where $a(n)$ is some appropriately chosen data tapering window, and $n = N'/2, \dots, N + (N'/2) - 1$. Now let i and j be indices into an $N' \times N$ matrix \mathbf{X} where i is the row and j is the column, such that the combination $\{i, j\}$ is an index to the element, denoted X_{ij} , of the matrix at the i^{th} row and j^{th} column of the matrix. The matrix \mathbf{X} of complex demodulates can then be formed by the combination of elements X_{ij} for $i = 1, 2, \dots, N$ and $j = 1, 2, \dots, N'$, where a single element of the matrix is defined as

$$X_{ij} \triangleq X_T(n_j, f_i) = \sum_{r=-N'/2}^{N'/2-1} a(r)x(n_j - r)e^{-j2\pi f_i(n_j - r)T_s}. \quad (2.47)$$

This allows the formation of a matrix whose rows are the complex demodulates of the signal $x(n)$, that is, each row is equivalent to the output of a filter having center frequency f_i and bandwidth $1/N'$ (for $f_s = 1$), and whose input is the signal $x(n)$. In fact, this is completely analogous to a filterbank, implemented using the DFT, whose N' filters, each with bandwidth $1/N'$, channelize $x(n)$ at the frequencies f_k for $k = 0, 1, \dots, N' - 1$ [28].

A data skipping parameter L can be introduced in (2.47) such that $n_j = pL$ for $p = 1, 2, \dots, P$, where $P = N/L$. The parameter L is chosen such that P is an integer. This in turn means that the output of each channel in the demodulates is decimated by a factor of L in addition to being filtered. In this case the demodulates matrix \mathbf{X} will be size $N' \times P$. Additionally, the conjugate version of the complex demodulates, \mathbf{X}_* can be obtained by replacing $x(n_j - r)$ with $x^*(n_j - r)$ in (2.47).

In order to compute the matrix \mathbf{X} we start with an intermediate matrix \mathbf{G} , which is a reorganization of $x(n)$ such that the columns are N' length subsets of the original signal (in this case it is assumed that $L = 1$). This leads to

$$\mathbf{G} = \begin{bmatrix} x(1) & x(2) & \cdots & x(N) \\ x(2) & x(3) & \cdots & x(N+1) \\ \vdots & \vdots & \ddots & \vdots \\ x(N') & x(N'+1) & \cdots & x(N+N'-1) \end{bmatrix} \quad (2.48)$$

$$= [\mathbf{x}_1 \quad \mathbf{x}_2 \quad \cdots \quad \mathbf{x}_N] \quad (2.49)$$

The columns of this matrix must then be windowed with an N' -length data tapering window $a(n)$. Then, if a size $N' \times N$ matrix \mathbf{A} is defined such that each column is the data tapering window $\mathbf{a}(n) = [a(1) \quad a(2) \quad \cdots \quad a(N')]^T$, then a windowed version of \mathbf{G} can be defined as

$$\mathbf{G}_w = \mathbf{G} \odot \mathbf{A} \quad (2.50)$$

where \odot denotes the Hadamard product, that is, element-by-element multiplication. Now if we let $\text{FFT}_{N'}\{\cdot\}$ denote an N' -point fast Fourier transform of a column vector, with the lowest frequency bin being the uppermost row, then the demodulates matrix \mathbf{X} can then be calculated as

$$\mathbf{X} = \text{FFT}_{N'} \{ \mathbf{G}_w \} \quad (2.51)$$

$$= \left[\text{FFT}_{N'} \{ \mathbf{x}_1 \odot \mathbf{a}(n) \} \cdots \text{FFT}_{N'} \{ \mathbf{x}_N \odot \mathbf{a}(n) \} \right]. \quad (2.52)$$

This can be interpreted as windowing and taking an FFT of each column of the matrix \mathbf{G} .

Now recall, also, that the time-smoothed cyclic periodogram of the same signal $x(n)$ is defined as [27]

$$S_{xx_T}^\alpha(n, f)_{\Delta t} = \langle X_T(n, f + \alpha/2) X_T^*(n, f - \alpha/2) \rangle_{\Delta t} \quad (2.53)$$

$$= \frac{1}{\Delta t} \sum_{m=n-\frac{\Delta t}{2}}^{n+\frac{\Delta t}{2}} X_T(m, f + \alpha/2) X_T^*(m, f - \alpha/2) \quad (2.54)$$

It can be seen by inspection of (2.54) that if the demodulates $X_T(n, f)$ are in matrix form that this definition can also be modified to define a time-smoothed estimate of spectral correlation over the entire bifrequency plane as a matrix. The spectral correlation matrix then simplifies to

$$\mathbf{S}_{\mathbf{X}\mathbf{X}} = \mathbf{X}\mathbf{X}^H. \quad (2.55)$$

The conjugate version of this matrix is

$$\mathbf{S}_{\mathbf{X}\mathbf{X}^*} = \mathbf{X}\mathbf{X}_*^T. \quad (2.56)$$

Notice that $\mathbf{S}_{\mathbf{X}\mathbf{X}^*}$, an $N' \times N'$ matrix, is no longer a function of f or α . This is because $\mathbf{S}_{\mathbf{X}\mathbf{X}^*}$ is a matrix which contains a time-smoothed estimate of the spectral correlation function of $x(n)$ over the entire bifrequency plane. However, the indices i and j of $\mathbf{S}_{\mathbf{X}\mathbf{X}^*}$, where i is the row and j is the column, correspond to frequencies f_i and f_j where f_i is the frequency of the i^{th} row of the first demodulate matrix and f_j is the frequency of the j^{th} row of the second demodulate matrix. The frequencies f_i and f_j can be used to map elements of the spectral correlation matrix back to the bifrequency plane, which uses discrete parameters f and α , according to

$$\alpha = f_j - f_i, \quad (2.57)$$

$$f = \frac{f_i + f_j}{2}. \quad (2.58)$$

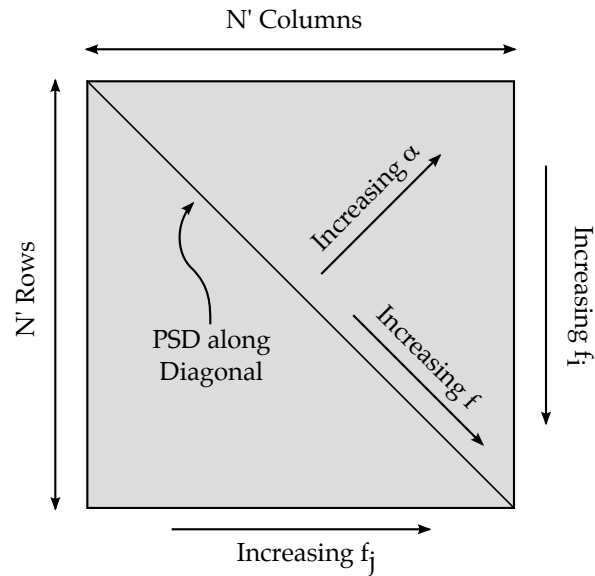
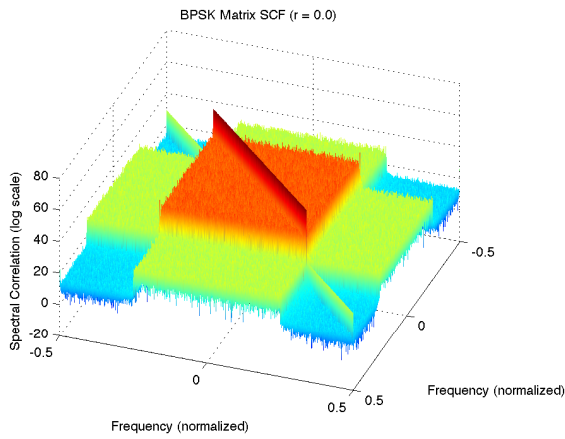


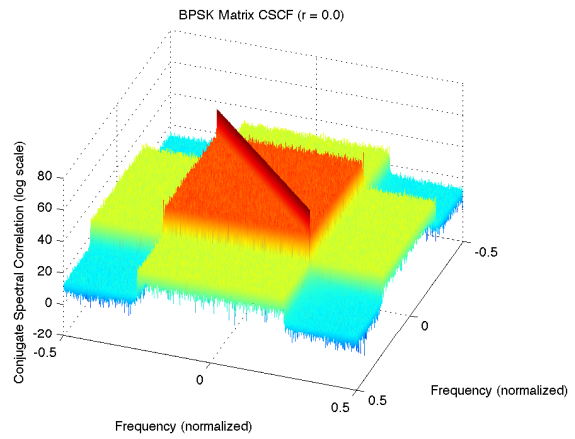
Figure 2.5: Diagram of the SCM and its attributes.

Figure 2.5 details the attributes of the spectral correlation matrix as they pertain to the bifrequency plane and its parameters.

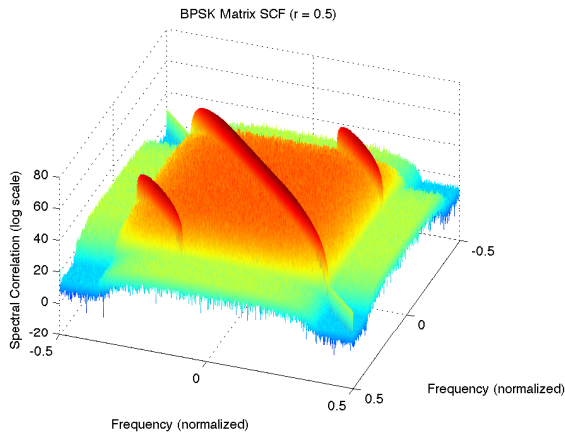
Plots of the spectral correlation matrix computed for BPSK and QPSK signals with various values for the pulse shaping filter roll-off parameter r are shown in Figures 2.6 and 2.7. These are in line with expected results when using traditional methods of computing the cyclic spectrum of BPSK and QPSK signals as seen in [2].



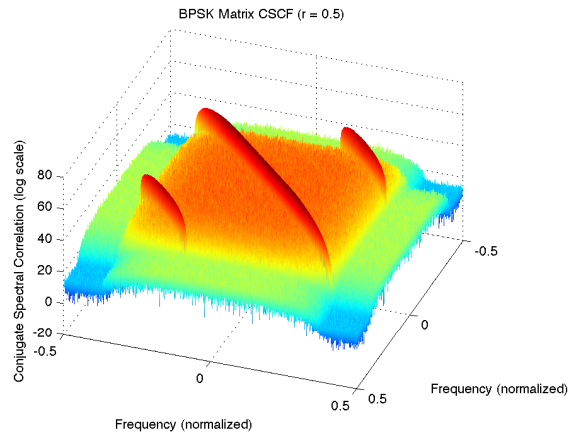
(a) BPSK SCM for $r = 0$



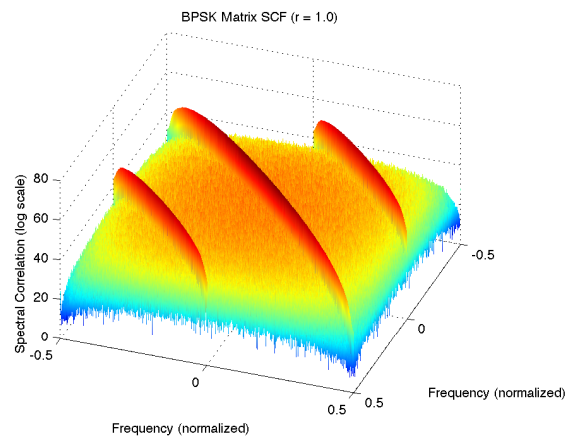
(b) BPSK CSCM for $r = 0$



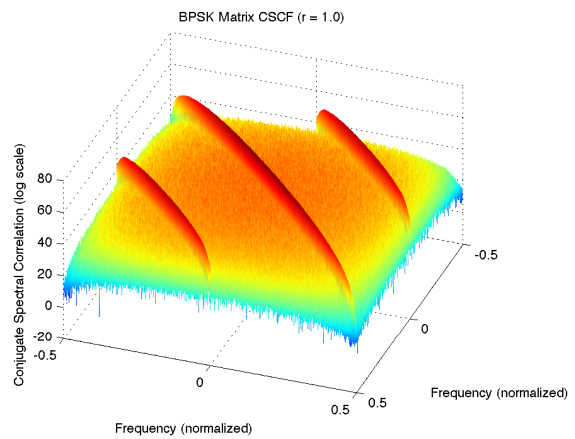
(c) BPSK SCM for $r = 0.5$



(d) BPSK CSCM for $r = 0.5$



(e) BPSK SCM for $r = 1$



(f) BPSK CSCM for $r = 1$

Figure 2.6: Plots of the SCM and CSCM for BPSK using square-root raised cosine pulse shaping with values for roll-off parameter r of 0, 0.5 and 1, averaged over 500,000 symbols.

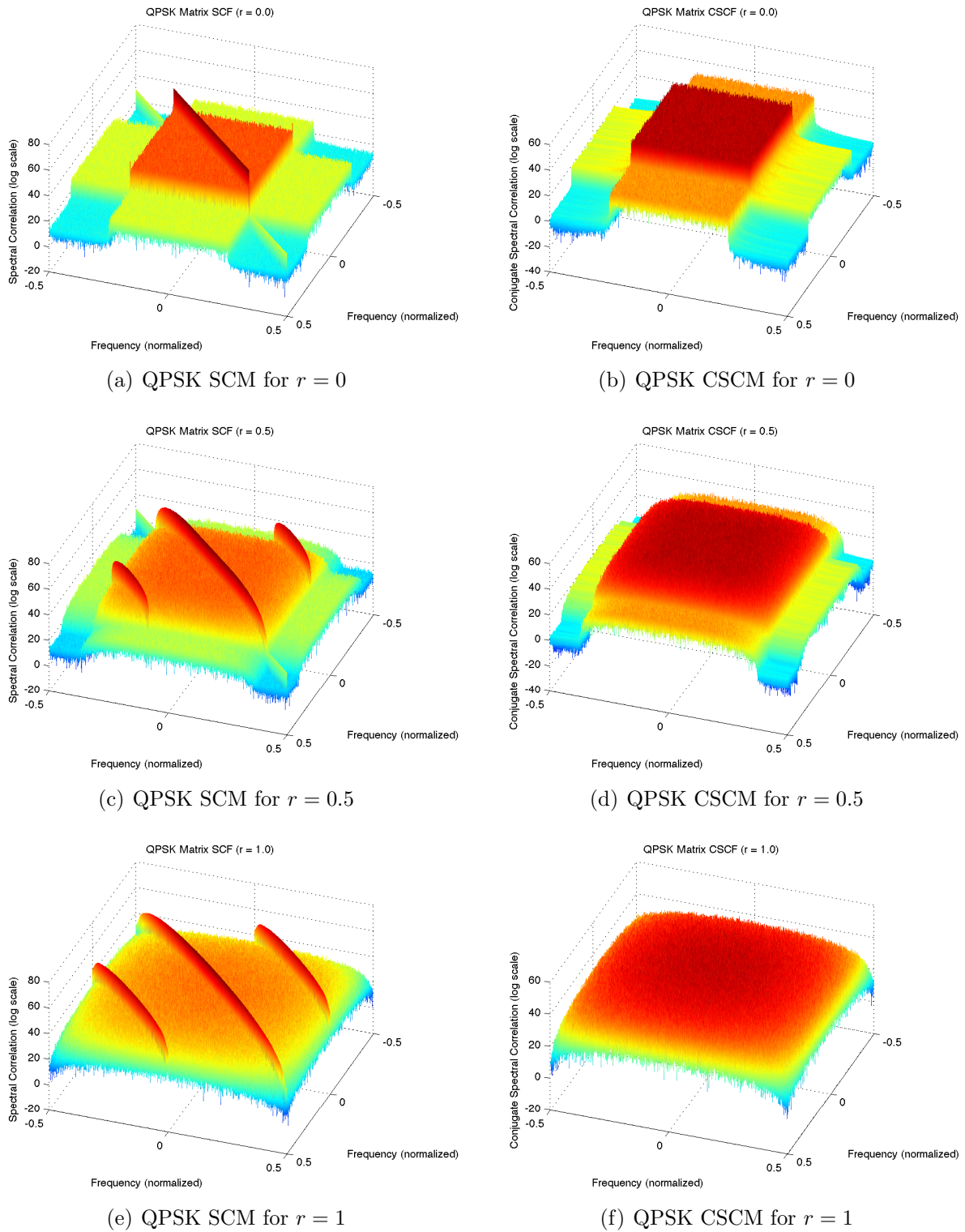


Figure 2.7: Plots of the SCM and CSCM for QPSK using square-root raised cosine pulse shaping with values for roll-off parameter r of 0, 0.5 and 1, averaged over 500,000 symbols.

Chapter 3

Hidden Markov Models for Pattern Recognition

3.1 Introduction

Pattern recognition as it pertains to communications and signal processing is the process of making a decision about the state of a signal based exclusively on observations of the signal itself. Generally this requires the signal to be examined or manipulated in such a way that a feature vector can be extracted which has minimal statistical variation for some identifying characteristic or set of characteristics. This feature vector is typically then compared against a set of known feature vectors and, using some prescribed pattern recognition algorithm, the most likely candidate is chosen. Thus, some pattern recognition algorithms require information about the set of possible choices to be known a priori, and only one of these may be selected. However, other algorithms may compare the unknown feature vector against the known set and decide that the unknown feature vector should be added to the known set, creating a new class which could be selected in future trials.

Some real world examples of pattern recognition in signal processing are speech recognition, facial recognition, and handwriting recognition, while some examples that are more exclusive to communications systems are modulation classification, signal classification, and emitter identification. Being pertinent to this work, specific emitter identification is a form pattern recognition wherein the set of known feature vectors represent wireless transmitters. While the process of extracting feature vectors from the known and unknown signals may be unique to SEI, any applicable pattern recognition algorithm can be still applied in the same way it would be applied to a general pattern recognition or pattern classification problem. For the purposes of this thesis, hidden Markov models were chosen as the pattern recognition algorithm to be paired with the eigenspace-based feature vector extraction algorithm. This is due in part to the fact that HMMs have been successfully used in the past for a host of

pattern recognition problems [2, 29].

This chapter does not attempt to give an in depth overview of the field of pattern recognition and all the possible techniques contained therein. It does, however, give an overview of how the eigenvector/eigenvalue decomposition of the spectral correlation matrix can be used to extract a feature vector from certain types of communications signals. It also provides a discussion for how HMMs can be used as a particular algorithm for pattern recognition. An introduction to the EVD and a discussion of why the eigenspace approach was chosen as a method for feature vector extraction are given in Section 3.2. Additionally, the topic of using HMMs for pattern recognition is introduced in Section 3.3. This section introduces HMM theory and talks about using an open source MATLAB library, the Hidden Markov Model Toolbox, for HMM pattern recognition.

3.2 Eigenspace Approach to Feature Vector Extraction

This work proposes using the eigenspace of the spectral correlation matrix in order to perform specific emitter identification for OFDM transmitters. The rationale for the eigenspace approach stems from the observation that the spectral correlation function is somewhat analogous to the spatial correlation matrix used commonly in antenna array processing. The eigenspace decomposition is often used to separate the spatial characteristics of various signals from the noise space in a spatial correlation matrix. Because an estimate of the spectral correlation function of an OFDM signal can contain sparsely separated cyclic components amongst noise, the eigendecomposition is potentially capable of extracting those cyclic spectral correlation components from the uncorrelated noise. Generally the subspace that is the output from the EVD can be separated into two distinct subspaces: the noise subspace, and the spectral correlation subspace. It is this spectral correlation subspace, or a subset thereof, which can contain useful information of the underlying transmitter that is both statistically consistent and unique.

3.2.1 Mathematical Basis

Assume \mathbf{A} is a square ($N \times N$) matrix. The N eigenvalues λ_i of the matrix \mathbf{A} are the solutions to

$$\mathbf{A}\mathbf{e} = \lambda\mathbf{e}, \quad (3.1)$$

where \mathbf{e} is column vector of length N . The eigenvalues can be found [30] by taking

$$\det[\mathbf{A} - \lambda\mathbf{I}] = 0, \quad (3.2)$$

which leads to setting the characteristic polynomial $p(\lambda)$ of \mathbf{A} to

$$p(\lambda) = \alpha_n \lambda^n + \alpha_{n-1} \lambda^{n-1} + \dots + \alpha_1 \lambda + \alpha_0 = 0. \quad (3.3)$$

The roots of (3.3), which is called the characteristic equation of \mathbf{A} , are the eigenvalues $\{\lambda_1, \lambda_2, \dots, \lambda_N\}$.

The eigenvectors associated with the eigenvalues of \mathbf{A} are found using the formula

$$[\mathbf{A} - \lambda_i \mathbf{I}] \mathbf{e}_i = 0, \quad (3.4)$$

where $i = 1, 2, \dots, N$. Each eigenvalue corresponds directly to a single eigenvector. The eigenvectors can be arranged in descending order such that $\lambda_{\max} \geq \lambda_i \geq \dots \geq \lambda_{\min}$, and then placed along the diagonal of matrix $\mathbf{\Lambda}$ (which is otherwise everywhere zero) such that λ_{\max} is in the upper left corner and λ_{\min} is in the lower right corner. The corresponding eigenvectors can be sorted using the same ordering applied to the eigenvalues, then placed in the matrix \mathbf{E} . The matrices $\mathbf{\Lambda}$ and \mathbf{E} contain the sorted eigenvalues and eigenvectors, respectively, of \mathbf{A} . It should be noted that \mathbf{A} can be reconstructed from the matrices using

$$\mathbf{A} = \mathbf{E} \mathbf{\Lambda} \mathbf{E}^H. \quad (3.5)$$

Now let $\mathbf{e}_i(n)$ denote the n^{th} element of the i^{th} sorted eigenvectors, and let $\mathbf{e}_i(\mathbf{n})$ denote a subset of elements indexing into the i^{th} sorted eigenvector. For example, if \mathbf{e}_i is of length 512, and if the set \mathbf{n} is chosen such that $\mathbf{n} = \{112, 244, 424, 488\}$, then $\mathbf{e}_i(\mathbf{n})$ would indicate the set consisting of the 112^{th} , 224^{th} , 424^{th} and 448^{th} elements of \mathbf{e}_i respectively, or simply $\{\mathbf{e}_i(112), \mathbf{e}_i(244), \mathbf{e}_i(424), \mathbf{e}_i(488)\}$. Using this notation, a feature vector $\mathbf{E}_M(\mathbf{n})$, retained in matrix form, can be selected from the eigenvectors \mathbf{E} by isolating the first M vectors such that

$$\mathbf{E}_M(\mathbf{n}) = [\mathbf{e}_1(\mathbf{n}) \quad \mathbf{e}_2(\mathbf{n}) \quad \dots \quad \mathbf{e}_M(\mathbf{n})], \quad (3.6)$$

where the set \mathbf{n} is chosen to be associated with elements of the eigenvectors which contain some identifying features. If \mathbf{n} contains F indices, then the size of $\mathbf{E}_M(\mathbf{n})$ would be $F \times M$. In this case, since the eigenvalues were sorted in descending order, the first M eigenvectors are associated with the M largest eigenvalues, and those are called the dominant eigenvectors of \mathbf{A} . These eigenvectors contribute more to the reconstruction of the original matrix \mathbf{A} , as per (3.5), and as such are associated with the stronger components of the original matrix.

When the EVD is obtained from the spectral correlation matrix, the frequency index f_i from (2.57) and (2.58), is retained in the eigenvectors. This is the same frequency used for the complex demodulates. Thus, the frequency f_n of the n^{th} element of \mathbf{e}_i is equal to f_i which is the frequency of the i^{th} row of the spectral correlation matrix.

3.2.2 Example Using OFDM

Orthogonal Frequency Division Multiplexing as a modulation scheme lends itself well to cyclostationary analysis because it tends to contain multiple cycle frequencies, especially when multiple pilots are used. This section contains plots of the eigenvectors and eigenvalues of the spectral correlation matrix of an OFDM signal based on WiMAX. For a detailed analysis on the OFDM scheme used to generate these plots, see Section 4.2. The important point to note is that the scheme employed 256 subcarriers, 8 of which were replaced with correlated pilots (that is, frequency translations of one another) using a BPSK modulation scheme, while the remaining non-guard subcarriers were modulated using QPSK. The pilots were about 2 dB higher in power than the subcarriers as per the 802.16 standard. Gaussian white noise was added to achieve an SNR of 10 dB. The SCM and CSCM, both generated using a value of $N' = 512$, were averaged over 5000 OFDM symbols (802.16 frame size is typically about 50 OFDM symbols). This equates to a little less than 0.5 seconds of integration time. Perfect frequency and symbol synchronization were assumed (but this may not be a requirement) as the OFDM snapshot was generated at baseband and synchronously sampled. Figures 3.1 and 3.2 show plots of the eigenvalues and eigenvectors of the SCM and CSCM, respectively, for the simulated OFDM signal.

It was mentioned in the previous section that the sorted EVD tends to divide the eigenspace of the target matrix into two distinct subspaces: the noise subspace and the signal subspace (or in this case, the spectral correlation subspace). In the case of OFDM this signal (or spectral correlation) subspace can be further divided into pilot subspace and subcarrier subspace. Recall that the matrix \mathbf{A} can be reconstructed from the eigenvectors and eigenvalues according to

$$\mathbf{A} = \mathbf{E}\mathbf{\Lambda}\mathbf{E}^H. \quad (3.7)$$

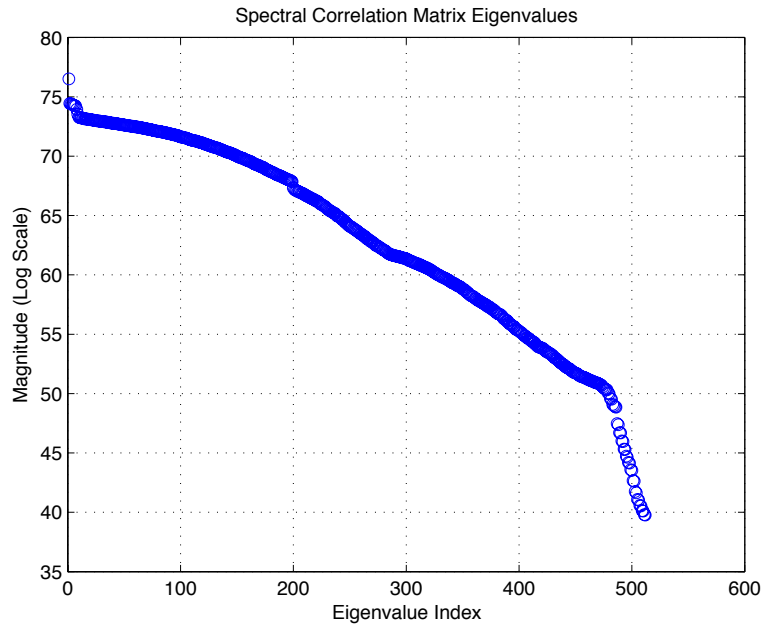
The sorted eigenvectors and eigenvalues are such that

$$\mathbf{E}^T = \begin{bmatrix} \mathbf{E}_p \\ \mathbf{E}_s \\ \mathbf{E}_n \end{bmatrix}, \quad (3.8)$$

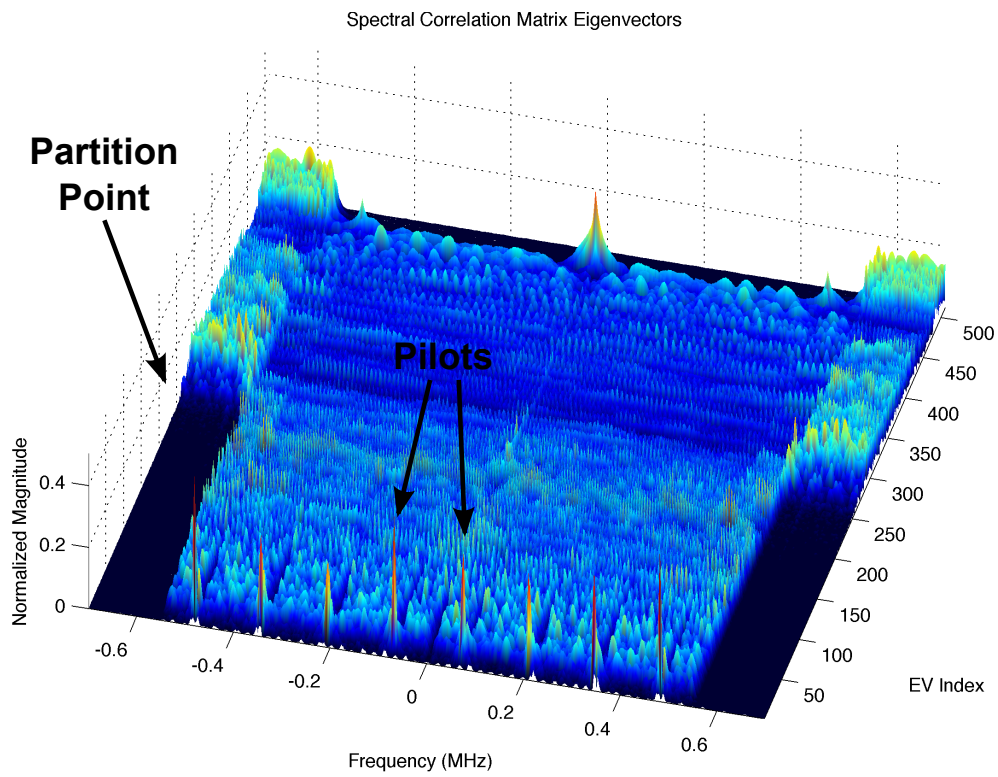
and

$$\mathbf{\Lambda} = \begin{bmatrix} \mathbf{\Lambda}_p & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{\Lambda}_s & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{\Lambda}_n \end{bmatrix}, \quad (3.9)$$

where \mathbf{E}_p , \mathbf{E}_s and \mathbf{E}_n are the pilot, subcarrier and noise eigenvectors, respectively, and along the diagonals of the otherwise everywhere zero matrices $\mathbf{\Lambda}_p$, $\mathbf{\Lambda}_s$ and $\mathbf{\Lambda}_n$ are the pilot,

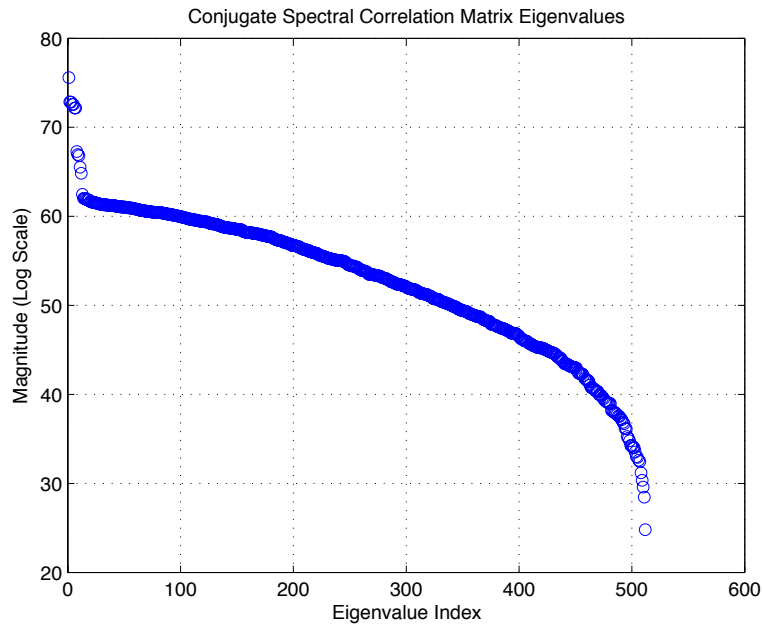


(a) SCM Eigenvalues

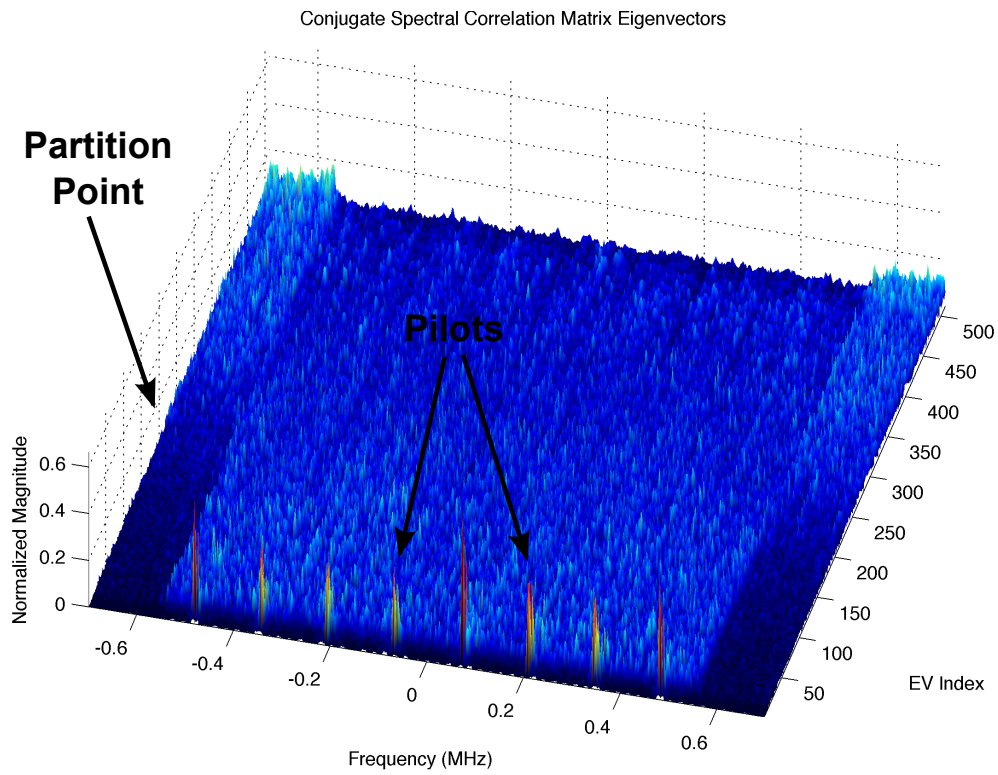


(b) SCM Eigenvectors

Figure 3.1: Spectral correlation matrix eigenvalues and eigenvectors for OFDM with 8 pilots.



(a) CSCM Eigenvalues



(b) CSCM Eigenvectors

Figure 3.2: Conjugate spectral correlation matrix eigenvalues and eigenvectors for OFDM with 8 pilots.

subcarrier and noise eigenvalues. Thus, (3.5) can be rewritten such that

$$\mathbf{A} = \begin{bmatrix} \mathbf{E}_p \\ \mathbf{E}_s \\ \mathbf{E}_n \end{bmatrix}^T \begin{bmatrix} \Lambda_p & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \Lambda_s & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \Lambda_n \end{bmatrix} \begin{bmatrix} \mathbf{E}_p \\ \mathbf{E}_s \\ \mathbf{E}_n \end{bmatrix}^* \quad (3.10)$$

This partitioning can be seen in the plots in Figures 3.1 and 3.2, although less so in the latter. In the plot of the SCM eigenvectors, the first 10 (approximately) eigenvectors are clearly seen to be associated with the 8 pilots which are embedded, evenly spaced, between the subcarriers. This is reinforced by the SCM eigenvalues, where approximately the first 10 eigenvalues, and especially the 1st eigenvalue, are partitioned separately from the remaining values. At approximately eigenvector 260, a transition point can be observed from subcarrier eigenvectors to noise eigenvectors. In the plot of the eigenvalues, this subcarrier to noise eigenvalue transition is not immediately apparent, however the magnitude of the pilot eigenvalues, in the approximate range of 1 to 10, are clearly a few dB higher in magnitude than the subcarrier eigenvalues. In the CSCM plots, the partitioning between noise and subcarrier eigenvectors is less abrupt, however the partitioning of the pilot eigenvectors is greater.

3.3 Hidden Markov Models

Hidden Markov models (HMMs) have been successfully used for speech recognition and pattern classification problems [29]. HMMs deal with the problem of a hidden stochastic model underlying an observable output of a system. For instance, if a single coin is tossed repeatedly, the output will consist of a series of either "heads" (H) or "tails" (T), each occurring with equal probability. However, if there are 2 coins available for flipping, the output will still remain a series of H or T , but there may be some underlying Markov process which determines whether one or the other coin is flipped. This model which determines the output is hidden from the observer, and therefore is a hidden Markov model.

3.3.1 Three Urns Example

A classic example of an HMM is the case of three urns each containing varying numbers of different colored balls. The balls can be either red (R), blue (B) or green (G), and each urn contains a different number of each type. One of the urns is selected at random according to some prior probability distribution, and a ball is selected at random from that urn, the color is noted and the ball is replaced. Another urn is chosen according to some Markov model. This process is then repeated indefinitely. A trellis diagram of this process is shown in Figure 3.4 [2,29]. The observed output of this process is simply a series of colors

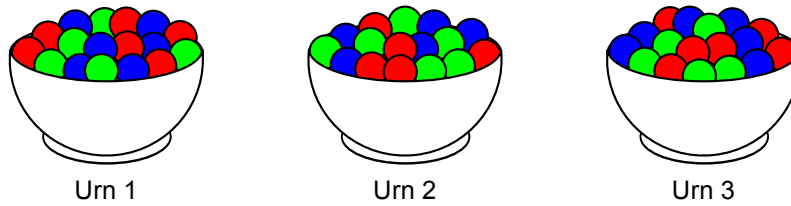


Figure 3.3: Three urns example.

$\{R, G, R, B, G, R, B, B, G, \dots\}$. The output appears to be a series of randomly selected balls, but since there is an underlying Markov process determining which urn a ball is selected from at each trial, the random process must be characterized using an HMM.

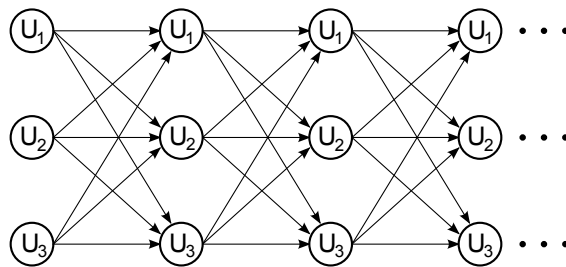


Figure 3.4: Trellis diagram for the three urns example.

For the three urn example above, we can define a state transition matrix which defines the probability of transitioning to any of the urns given that a particular urn was selected at the last trial. The state transition matrix can be defined as

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \quad (3.11)$$

where $a_{ij} = Pr(U_i|U_j)$, is the probability of transitioning from U_i to U_j . Each row of the matrix must sum to 1. In addition to a state transition matrix, we can also define an observation probability matrix \mathbf{B} as

$$\mathbf{B} = \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{bmatrix} \quad (3.12)$$

where $b_{ij} = Pr(i|U_j)$ is the probability of selecting a ball of a particular color from urn U_j .

In this case we may let the color "red" correspond to 1, the color "green" correspond to 2, and "blue" correspond to 3. Thus the probability of selecting a green ball from the third urn would be $Pr(2|U_3)$. The last piece of information that we can define about this example is the prior probability vector π . In this example the vector π will be a 1×3 vector containing the probability of selecting each of the urns at the beginning of the output sequence. Thus all the necessary information about the model which generates an output sequence using this example can be denoted using $\lambda = \{\mathbf{A}, \mathbf{B}, \pi\}$.

3.3.2 General Discrete Case

This model can be extended to a more general case [2,29] if we let the number of states in the underlying model (equivalent to the number of urns in the previous example) be denoted as N , and let the number of outcomes at each state (equivalent to the number of balls in each urn) be denoted as K . If we define two discrete-time stochastic processes $x(t)$ and $y(t)$, for $t = \{1, 2, \dots, T\}$, we can let $x(t)$ be the sequence of states in the underlying Markov chain, and $y(t)$ can be the output sequence after choosing from one of the possible outputs. The sequence $y(t)$ is highly dependent on the underlying sequence $x(t)$, but the state sequence $x(t)$ is hidden from the observer. In the general case the state transition matrix becomes

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1N} \\ a_{21} & a_{22} & \dots & a_{23} \\ \vdots & \vdots & \ddots & \vdots \\ a_{N1} & a_{N2} & \dots & a_{NN} \end{bmatrix} \quad (3.13)$$

where $a_{ij} = Pr(x(t) = i | x(t-1) = j)$ is the probability of transitioning to state i given that the previous state is state j . In addition, the matrix \mathbf{B} can be generalized as

$$\mathbf{B} = \begin{bmatrix} b_{11} & b_{12} & \dots & b_{1N} \\ b_{21} & b_{22} & \dots & b_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ b_{K1} & b_{K2} & \dots & b_{KN} \end{bmatrix} \quad (3.14)$$

where $b_{jk} = Pr(y(t) = j | x(t) = k)$ is the probability of a particular outcome given that particular state has been transitioned to. Thus, in this generalized case, the observation probability vector $\mathbf{p}(t)$ can be defined as

$$\mathbf{p}(t) = \begin{bmatrix} Pr(y(t) = 1) \\ Pr(y(t) = 2) \\ \vdots \\ Pr(y(t) = K) \end{bmatrix}. \quad (3.15)$$

This can be expressed in terms of the state transition matrix, the observation probability matrix and the prior probability vector as

$$\mathbf{p}(t) = \mathbf{B}\mathbf{A}^{t-1}\boldsymbol{\pi}. \quad (3.16)$$

Here, the prior probability vector is simply

$$\boldsymbol{\pi} = \begin{bmatrix} Pr(x(1) = 1) \\ Pr(x(1) = 2) \\ \vdots \\ Pr(x(1) = N) \end{bmatrix}. \quad (3.17)$$

Again, this generalized model can be completely described using the nomenclature $\lambda = \{\mathbf{A}, \mathbf{B}, \boldsymbol{\pi}\}$.

3.3.3 Gaussian Mixtures

It is often the case in signal processing that observations and transformations of a given signal will have a continuous, or near-continuous (that is, sampled but with low quantization noise) output. The implication for HMMs is that having a discrete set of outputs for each state in the Markov process may not adequately model the underlying signal or the observations of that signal. Using a Gaussian mixture to model the output probability distribution of each state of a Markov chain can more accurately model an HMM whose underlying process is continuous [31, 32]. Thus, an HMM that uses a continuous probability distribution for each state is sometimes called a continuous density HMM.

In order to use continuous density HMMs for pattern recognition it is necessary to replace the observation probability matrix \mathbf{B} with the probability distribution

$$b_j(x) = \sum_{k=1}^{N_j} c_{jk} \mathcal{N}(x|\boldsymbol{\mu}_{jk}, \boldsymbol{\Sigma}_{jk}), \quad (3.18)$$

where j is a possible state in the Markov chain (which is hidden). In this equation, $\mathcal{N}(x|\boldsymbol{\mu}_{jk}, \boldsymbol{\Sigma}_{jk})$ is a Gaussian probability distribution, where $\boldsymbol{\mu}_{jk}$ is the mean vector, $\boldsymbol{\Sigma}_{jk}$ is the covariance matrix and c_{jk} are the N_j mixture coefficients with the constraints $c_{jk} \geq 0$ and $\sum c_{jk} = 1$. The Gaussian mixture is a general probability distribution function which is constituted as the sum of scaled, translated and dilated Gaussian distribution functions, and as such is quite flexible. Thus using a Gaussian mixture for the probability distribution of each of the hidden states allows the generic discrete model to be extended to the continuous case.

3.3.4 Configuring HMMs for Pattern Recognition

In order to configure continuous density HMMs for pattern recognition it is first necessary to obtain a limited set of observable system outputs. Each system output must be a series of observations following a Gaussian mixture probability distribution and governed by an HMM with unique model parameters—that is, with a unique prior probability vector π and state transition matrix \mathbf{A} . In this case the observation probability matrix \mathbf{B} is replaced by N Gaussian mixture distributions, where N is the number of states in the Markov model. An HMM for each of the systems is extracted using the Baum-Welch algorithm [33]. Typically when the system output is a time series of sample data, as is often the case in signal processing implementations of pattern recognitions systems, it must first be manipulated into a succinct feature vector prior to extraction of HMMs. This set of HMMs comprise the known dataset.

Once a set of known HMMs has been established, the pattern recognition problem of classifying unknown observable systems can be undertaken. Assuming that any unclassified system is identical to one of the previously modeled systems, an HMM can be extracted from the unclassified system and compared with the HMM associated with each of the known systems. The log-likelihood probability that the unknown system has the HMM of each of the previously extracted known systems can then be computed, and the maximum of these determines which of the known systems the unknown system is classified as. In this fashion a series of unknown systems can be classified as long as the unknown systems are of the same type as one of the known systems.

3.3.5 HMM Toolbox

The Hidden Markov Model Toolbox [34] is a collection of MATLAB functions which can be used for training HMMs to build a set of known feature vectors, and testing unknown feature vectors against the set of known feature vectors. A list of the functions from the toolbox that were used in the simulations contained in this work are listed in Table 3.1, and a more detailed tutorial can be found in [34]. The collection of functions found in the toolbox allow HMMs to be used as a pattern recognition tool for a variety of applications without the heavy cost associated with coding the necessary capabilities from scratch.

The basic methodology (discussed in greater detail in Chapter 5) employed when implementing the functions discussed in this toolbox is to divide the simulation algorithm into two main sections: the training section and the testing section. In the training section, an HMM is computed from the feature vectors extracted from the known signal classes. This is accomplished by performing multiple trials with the function `mhmm_em.m`, which uses the Baum-Welch algorithm, for each of the signal classes in the known dataset. The input to the function `mhmm_em.m` is the matrix $\mathbf{E}_M(\mathbf{n})$, from above, for the case when only the SCM is used for training. When both the SCM and CSCM are used, the input is a matrix constructed by joining together the matrices $\mathbf{E}_{M,SCM}(\mathbf{n})$ and $\mathbf{E}_{M,CSCM}(\mathbf{n})$ extracted from the

Table 3.1: Pertinent HMM Toolbox Functions

| Function | Description |
|------------------------------|--|
| <code>normalise.m</code> | Helper utility used to ensure that the entries of a matrix sum to 1 |
| <code>mk_stochastic.m</code> | Helper utility which ensures that the rows of a 2-dimensional matrix sum to 1 |
| <code>mixgauss_init.m</code> | Generates parameters for a Gaussian mixture probability distribution |
| <code>mhmm_em.m</code> | Computes the ML parameters of an HMM with Gaussian mixtures, using the Baum-Welch algorithm, from a known feature vector in order to construct an a priori dataset |
| <code>mhmm_logprob.m</code> | Computes the log-likelihood probability that an unknown feature vector has the HMM (with Gaussian mixture) of a previously trained HMM from a known dataset |

SCM and CSCM respectively. The output of `mhmm_em.m` is the HMM which best fits the input data, consisting of a state transition matrix, a prior probability vector and a set of Gaussian mixture probability distributions.

In the testing section, ideally a different set of data for each of the classes may be used to test against each of the known HMMs computed from the a priori dataset. For each unknown feature vector, the function `mhmm_logprob.m` must be called against each HMM in the known dataset. Thus, if there are N known classes, `mhmm_logprob.m` must be called N^2 times. A confusion matrix can be generated by choosing the largest value returned by `mhmm_logprob.m` for all known HMMs. The inputs to the function `mhmm_logprob.m` are the matrix of features as described in the above paragraph, and the HMM (state transition matrix, prior probability vector and set of Gaussian mixture distributions) of one of the trained emitters. The output is the log-likelihood probability that the input feature vector has the HMM of the previously trained emitter.

Chapter 4

Wireless Transmitter Simulation

4.1 Introduction

In order to evaluate the performance of a previously untested SEI algorithm, in the absence of captured data from real world devices, it is desirable, during the early stages of research, to simulate a set of wireless transmitters in such a way that discerning between is occasionally possible. Emitter identification relies on the principle that differences between devices in a set are great enough to be measurable, but not so great as to be obvious or to destroy the ability of the underlying transmission scheme to transmit data. Additionally, aspects of the simulation which cause the underlying modulation scheme to become uniquely identifiable should be a realistic approximation of the type of effects that would be encountered when scrutinizing real world devices, otherwise the SEI algorithm would not be useful. For this reason, as many aspects of wireless transmitters as are possible must be taken into account when implementing an algorithm for accurately simulating a wireless transmitter.

This work uses simulated data in place of real world data mainly because simulation provides more control over the experimental process at this early stage of research. Orthogonal frequency division multiplexing, the underlying modulation scheme of WiMAX, is a relatively simple to simulate digital transmission scheme, thus making the 802.16 specification an appealing resource from which to draw simulation specifics. Other aspects, such as blocks of the RF component chain, were chosen in light of evidence in previous work [2,3], which tested against digitized Wi-Fi bursts, that these components greatly contribute to the outcome of SEI tests performed therein. Additionally, all wireless transmissions must at some point undergo some channel effects; a fact which cannot be avoided. Since a multipath channel may effect the outcome of an SEI algorithm differently than an AWGN channel, it is useful test the performance of said algorithm both with and without multipath.

This chapter gives an overview of the concept of OFDM, and then provides specifics about how it is implemented by real world specifications such as WiMAX. It also provides details

on how aspects of a wireless transmitter pertinent to SEI can be simulated, as well as give an overview of simulating channel effects. Section 4.2 provides an overview of OFDM and OFDMA, how they are generated, and how the 802.16 specification (WiMAX) uses OFDM for a selection of access schemes. In the case of 802.16, an overview of pilot placement is provided for these access schemes. Section 4.3 details methods and algorithms used in the simulations contained in this thesis for modeling RF components which affect classification of transmitters. A high level block diagram of the simulation process as a whole is depicted in Figure 4.1.

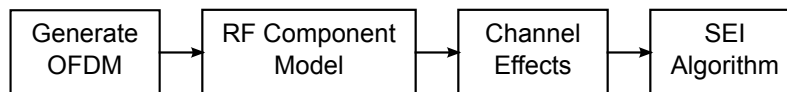


Figure 4.1: Block diagram of simulation.

4.2 Orthogonal Frequency Division Multiplexing

Orthogonal frequency division multiplexing is a method of modulation which has been in existence for several years [35], but has not gained popularity until relatively recently. This is primarily due to the fact that implementing OFDM required enough computational power and complexity as to prevent its widespread use prior to the adaptation of compact computing devices seen today. Modern modern wireless devices, such as smartphones, are equipped with the computational power necessary to implement OFDM-based transmission schemes reliably a on small scale, and hence the modulation is currently becoming one of the most widely adopted communication schemes for new wireless infrastructures.

OFDM differs from single carrier modulation in that it transmits a block of symbols in parallel using an entire bank of orthogonal subcarriers at evenly spaced frequencies. Thus a long series of single-carrier symbols are combined into single longer symbol that is time-invariant over the duration of that symbol. Multipath effects are mitigated by inserting a guard interval—called a cyclic prefix—between successive OFDM symbols. While no real spectral efficiency is gained by transmitting data using this scheme, it does mean that a considerable tolerance to multipath can be achieved with some additional computational complexity [36]. One of the primary advantages to using OFDM is its inherent resistance to multipath. OFDM regains some computational efficiency since the modulation can be easily implemented using both a DFT and IDFT, and hence implemented using an FFT and IFFT.

Baseband generation of OFDM can be described mathematically with a single equation [37, 38]. Assuming we have some bit stream b_k which are mapped to symbols a_k , a single OFDM symbol with N subcarriers can be generated by multiplying the symbols $\{a_1, a_2, \dots, a_N\}$ by a series of orthogonal complex sinusoids. Thus we have

$$x(t) = \sum_{k=0}^{N-1} a_k e^{j2\pi k \frac{t}{T}}, \quad (4.1)$$

where T is the duration of the OFDM symbol. This is simply an inverse DFT, making OFDM computationally efficient to generate and demodulate using the IFFT and FFT algorithms. However, in practical implementations, certain subcarriers are reserved as guard bands and for the insertion of a pilot signal. In addition, a cyclic prefix is inserted between OFDM symbols to mitigate the effects of a multipath channel delay spread. This is accomplished by taking the last G samples of an OFDM symbol and prepending them to the beginning of the symbol. A block diagram of OFDM generation is shown in Figure 4.2.

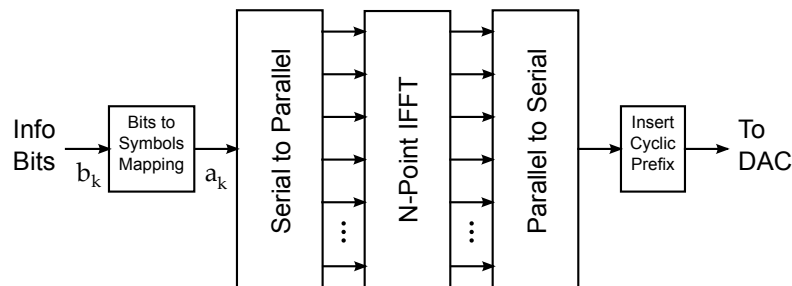


Figure 4.2: Block diagram of an OFDM transmitter

Orthogonal Frequency Division Multiple Access (OFDMA) is a variation of OFDM wherein the subcarriers are shared by multiple users, but is fundamentally identical to OFDM. The number of subcarriers allocated to each user may be dynamically reallocated each symbol and the allocation scheme may vary between downlink (DL) and uplink (UL).

In order to ensure that the information contained in an OFDM signal can be fully recovered to the extent that the environment allows, pilot sequences are inserted into the OFDM/OFDMA signal, taking the place of some of the subcarriers. The number and location of the pilots depends largely on the bandwidth of the signal and whether or not it is OFDM or OFDMA. However, the pilots are completely necessary for symbol and frequency synchronization.

4.2.1 WiMAX

WiMAX is the technology described by the IEEE 802.16-series standard. It is categorized as a wireless MAN (Metropolitan Area Network) system. The air interface for 802.16 uses OFDM and OFDMA exclusively for both the uplink and downlink. Early versions of the 802.16 specification were intended for non-mobile use only, but mobile functionality was eventually appended in subsequent iterations. More recent versions of the specification have

also added MIMO (Multiple Input-Multiple Output) functionality. An overview of some of the global parameters governing the 802.16 specification [39] is provided in Table 4.1.

Table 4.1: Global WiMAX Parameters

| | |
|---|------------------------------|
| Access Scheme (UL and DL) | OFDM, OFDMA |
| Bandwidth (MHz) | 1.25, 2.5, 5, 10, 20 |
| Sampling Frequency (MHz) | 1.4, 2.8, 5.6, 11.2, 22.4 |
| Number of Subcarriers | 128, 256, 512, 1024, 2048 |
| Subcarrier Spacing | 10.938 kHz |
| Frame Size (Typically) | 5 μ s |
| Cyclic Prefix (Fraction of Symbol Duration) | 1/32, 1/16, 1/8, 1/4 |
| Modulation Types | BPSK, QPSK, 16QAM, 64QAM |

There are many aspects of WiMAX that are not taken into consideration in this work, such as frame structure, preambles and control channels. The aspect of the specification which plays the most important role in terms of the ability to perform SEI is the method in which pilots are allocated into the OFDM/OFDMA subcarriers. These pilots are generally placed more sporadically for OFDM (approximately 25 subcarrier spacing), and are more closely spaced for OFDMA (anywhere from 3 to 12). In the case of OFDMA, the pilot allocations may change between successive symbols. The specification calls for pilots which are in some cases slightly higher power than the subcarriers, and in others the same power as the subcarriers, as well as pilots which are correlated (that is, frequency translations of one another) and uncorrelated. For the purposes of this simulation all pilots are taken to be slightly higher power (2.5 dB) and correlated, and as such the 802.16 specification is used as a guide for pilot placement and not followed verbatim. The OFDM and OFDMA generated for these simulations are by no means a complete and accurate simulation of the 802.16 specification.

The simulated OFDM/OFDMA snapshots contained in later sections are either 5000 or 3000 symbols in duration, as specified. This equates to slightly less than 0.5 seconds and slightly more than 0.25 seconds of integration time respectively.

Four downlink types (one OFDM and 3 OFDMA) were chosen from the specification for use in the simulations: OFDM, Partial Usage of Subcarriers (PUSC), Full Usage of Subcarriers (FUSC), and Optional Full Usage of Subcarriers (OFUSC). These are also outlined in Table 4.2. In the table, the term constant pilots refers to whether the pilot allocation changes between successive symbols. The OFDMA downlink types specify how subcarriers are divided among users and the placement of pilots in each symbol. The OFDM downlink

is defined only for the 256 subcarrier (2.5 MHz) case, while all other OFDMA types are defined for the 128, 512, 1024 and 2048 (1.25, 5, 10 and 20 MHz respectively) cases.

Table 4.2: Selection of 802.16 Downlink Types

| Sub-channelization | Access Scheme | Constant Pilots |
|--|---------------|-----------------|
| None | OFDM | Yes |
| Partial Usage of Subcarriers (PUSC) | OFDMA | No |
| Full Usage of Subcarriers (FUSC) | OFDMA | Some |
| Optional Full Usage of Subcarriers (OFUSC) | OFDMA | No |

The indices of the pilot subcarriers for downlink OFDM are the set $\{-88, -63, -38, -13, 13, 38, 63, 88\}$, where -128 is the index of the lowest frequency subcarrier and 127 is the index of the highest frequency subcarrier.

For PUSC the subcarriers are divided into clusters consisting of 14 subcarriers. The allocation of pilots in each cluster alternates between two states, depending on whether the symbol is an even symbol or an odd symbol. The pilot allocation for a single cluster is shown in Figure 4.3 for an even and odd symbol.

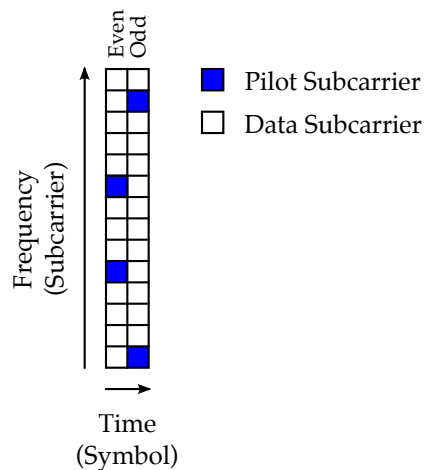


Figure 4.3: Pilot mapping for a PUSC cluster.

For FUSC, every 12th subcarrier is allocated as a pilot, and this allocation shifts by a factor +6 every odd symbol. Thus if the indices $\{0, 12, 24\}$ constitute the set of pilots for an even symbol, for an odd symbol these indices would become $\{6, 18, 30\}$. Additionally, FUSC designates a set of constant pilots, spaced 36 subcarriers apart (but whose mapping depends on the bandwidth), the allocation of which does not change between symbols.

Finally, OFUSC allocates its pilots according to

$$9k + 3m + 1, \quad (4.2)$$

where $k = 0, 1, \dots, N_{pilots} - 1$ and $m = [\text{symbol index}] \bmod 3$. Thus the pilots are separated by 9 subcarriers, the allocation indices shift by 3 every symbol, and the allocation sequence repeats every 3rd symbol. See Figure 4.4 for pilot-only spectral plots of the four selected 802.16 downlink types.

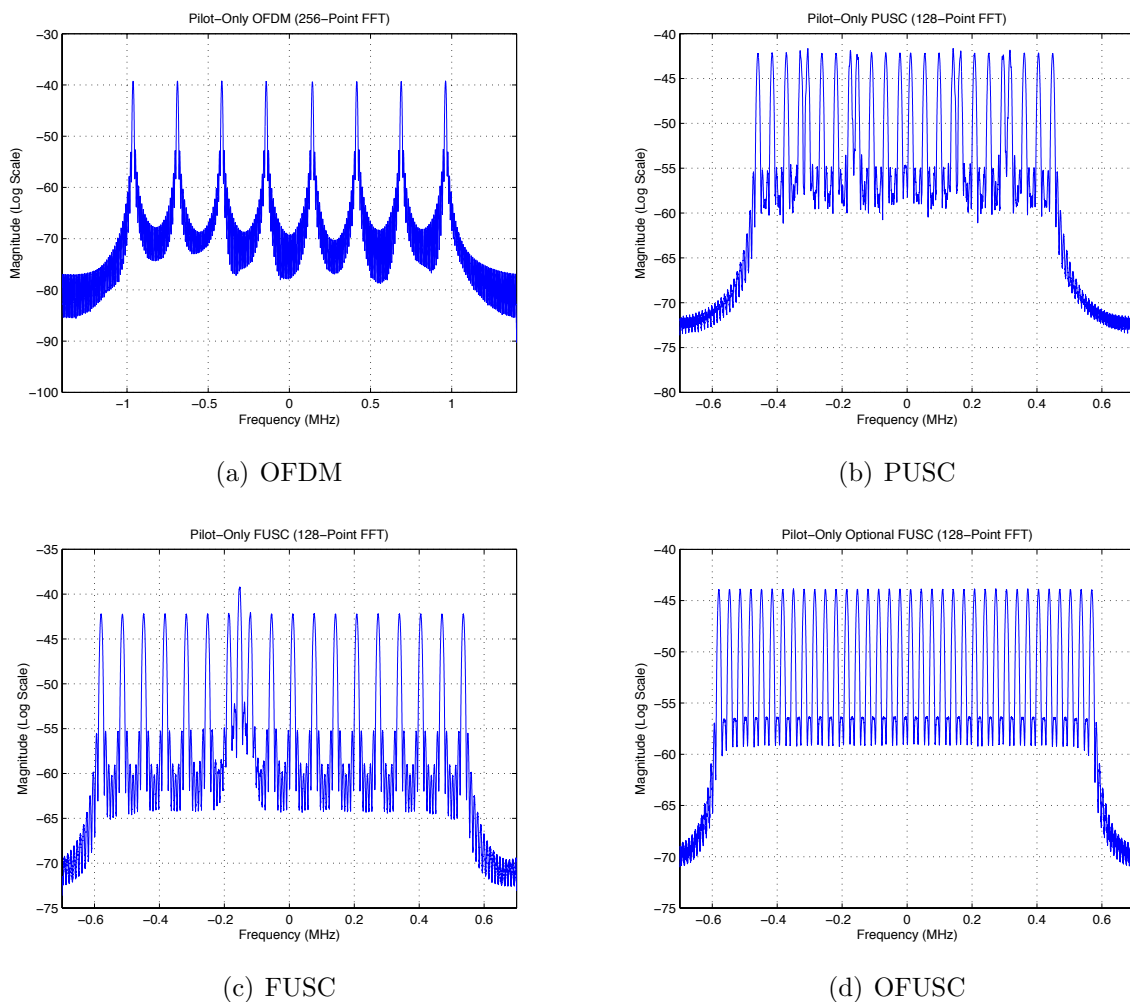


Figure 4.4: Pilot-only spectral plots of the selected 802.16 downlink types. The constant pilot in the FUSC spectrum appears higher powered than the variable pilots due to averaging.

The pseudorandom binary sequence (PRBS) generator used to generate the pilot sequence for 802.16 [39] is depicted in Figure 4.5. The polynomial for this generator is $X^{11} + X^9 + 1$. The initial value of the bit register is determined by several specified factors, however for

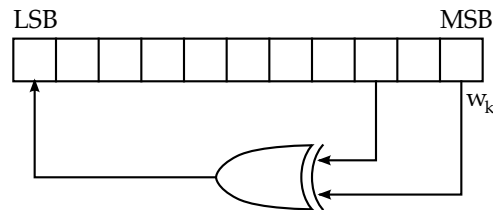


Figure 4.5: Pseudorandom binary sequence generator used for 802.16 pilots.

simulation purposes the pilot register was initialized with all ones. The output bit sequence w_k is then mapped to the pilot sequence c_k using

$$\operatorname{Re}\{c_k\} = \frac{8}{3} \left(\frac{1}{2} - w_k \right), \quad (4.3)$$

$$\operatorname{Im}\{c_k\} = 0. \quad (4.4)$$

4.3 Transmitter Component Modeling

In order to simulate the effect of various filters, power amplifiers and other RF components, it is necessary to apply some model, or set of models, which transforms the data in some fashion which mimics the behavior of these components. This transformation should apply some alterations to the underlying transmitted waveform, but not to a degree where the information contained in the modulating signal is corrupted beyond recovery. For the purposes of this simulation several filter models, intended to mimic the composite frequency response of several RF components, are presented in the following section.

4.3.1 Filter Modeling

One component which must be a part of the RF transmitter chain of every emitter is some type of filter. The filters may be of the lowpass or bandpass variety, but they must all accommodate a relatively wide bandwidth of up to 20MHz in the case of WiMAX. Because filters may be designed with cost in mind as opposed to fidelity, the filters in some cheaper devices may not be perfectly flat across the passband. In addition to filters, other RF components may also have some spectral imperfections. The composite frequency response of all components in a transmitter, due to the unique combination of those components, and to some degree due to tolerances in the manufacturing process, may be unique to a specific vendor or model, if not to the device itself. The possibility that the composite frequency response is unique for a single wireless device is made even more probable if lower quality

components are used where manufacturing costs are at a premium. Evidence in [2,3] suggests that this is indeed true for Wi-Fi wireless devices.

Because frequency selective fading is likely to be present in metropolitan environments, where technologies such as WiMAX are commonly deployed, designers of OFDM-based wireless devices may feel that the cost-cutting measures of incorporating lower quality components into the design of the RF chain is justified, despite those components having a frequency response which may not be perfectly flat across the passband. A frequency domain equalizer can be easily incorporated into an OFDM receiver since frequency domain processing is inherent to the nature of the modulation. Such an equalizer will be required to mitigate the effects of a frequency selective fading environment, and in so doing will also mitigate the effects of any RF components with a non-flat frequency response. In essence the effects of a suboptimal component are removed through equalization required for channel effects.

For the purposes of simulation, it is desirable to have a set of filter taps $h(n)$ which model the composite frequency response of the RF components in a transmitter. A simple method for designing such a filter is to create a spectral mask in the frequency domain, and use the inverse Fourier transform (in this case the inverse FFT) to obtain the complex filter taps. This mask is an allpass filter with variable gain across the bandwidth. The point of the mask is not to accept or reject different portions of the bandwidth, but rather to apply a small increase or decrease in gain at different frequencies, thus mimicking the RF output of a unique wireless transmitter.

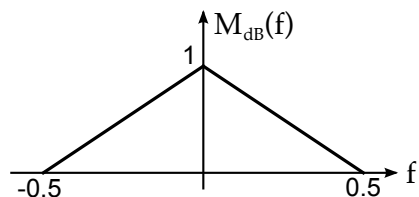


Figure 4.6: Example of a spectral mask.

The first step is to define a function $M_{\text{dB}}(f)$ in dB (normalized in frequency) which is nominally 0 dB, but deviates slightly from this value across the bandwidth. For instance, a simple mask may have a value of 0 dB at $f = -0.5$, increase to 1 dB at $f = 0$ and then decrease back to 0 dB at $f = 0.5$ (see Figure 4.6). Mathematically this function would be defined as

$$M_{\text{dB}}(f) = \begin{cases} 2f + 1, & -0.5 \leq f < 0 \\ 1 - 2f, & 0 \leq f \leq 0.5. \end{cases} \quad (4.5)$$

The function $M_{\text{dB}}(f)$ is then smoothed using a unit area filter $g(f)$ such that

$$M_{\text{dB,smooth}}(f) = g(f) \otimes M_{\text{dB}}(f), \quad (4.6)$$

where \otimes denotes convolution. Finally, the complex filter taps $h(n)$ for the spectral mask in this example is computed using

$$h(n) = \text{IFFT} \{10^{M_{\text{dB,smooth}}(f)/10}\}, \quad (4.7)$$

where IFFT denotes the inverse fast Fourier transform. The filter $h(n)$ is then used to apply the spectral mask defined in $M_{\text{dB}}(f)$ to a signal $x(n)$ with

$$y(n) = x(n) \otimes h(n). \quad (4.8)$$

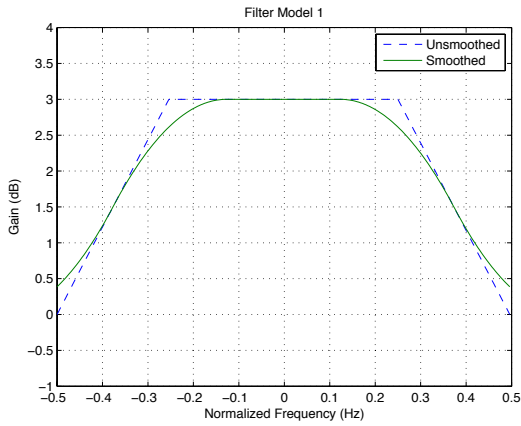
In practice, $y(n)$ is calculated by transforming $x(n)$ and $h(n)$ back to the frequency domain and performing a multiplication, so it may be necessary to also retain $M_{\text{dB,smooth}}(f)$.

In order to simulate the effect that components of various tolerances and degrees of quality have on OFDM, several different spectral masks were created to simulate emitters unique wireless devices. These masks were designed to provide some degree of discernability. That is, they were chosen to be only subtly different, but different enough such that discernment is possible. Parameters used to generate the masks are shown in Table 4.3 while plots of the unsmoothed and smoothed masks are shown in Figure 4.7. These models represent six unique emitters or classes of emitters, embodied by variations that affect their RF transmissions (in this case OFDM) such that they can be uniquely identified. They are referred to in the remainder of this simulation as emitters, unless otherwise specified.

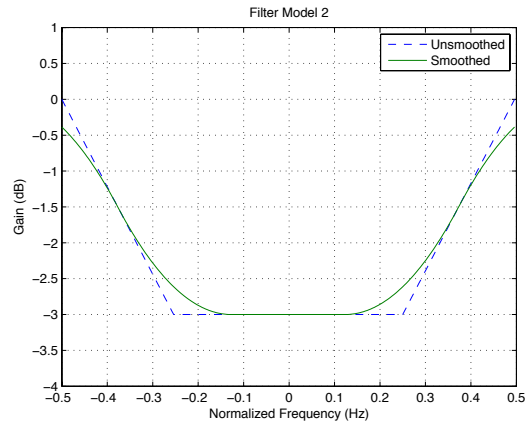
Table 4.3: Spectral Mask Parameters (Normalized Frequency)

| Name | Shape | $M_{\text{dB}}(-0.5)$ | $M_{\text{dB}}(-0.25)$ | $M_{\text{dB}}(0.25)$ | $M_{\text{dB}}(0.5)$ |
|--------|-------------|-----------------------|------------------------|-----------------------|----------------------|
| Mask 1 | Convex | 0 dB | 3 dB | 3 dB | 0 dB |
| Mask 2 | Concave | 0 dB | -3 dB | -3 dB | 0 dB |
| Mask 3 | Right Peak | 0 dB | 1 dB | 3 dB | 0 dB |
| Mask 4 | Left Peak | 0 dB | 3 dB | 1 dB | 0 dB |
| Mask 5 | Peak-Valley | 0 dB | 3 dB | -3 dB | 0 dB |
| Mask 6 | Valley-Peak | 0 dB | -3 dB | 3 dB | 0 dB |

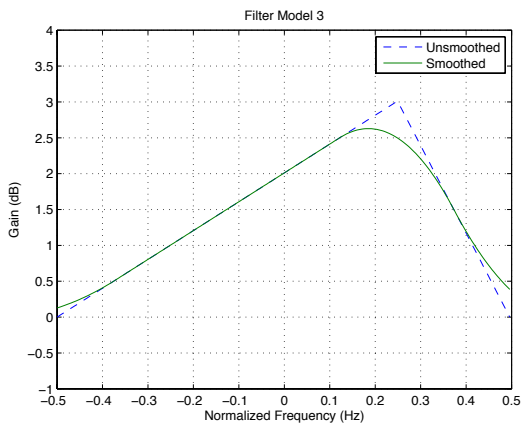
While the 3 dB variations in these models may appear to be outside acceptable technical specifications, there is evidence that commercial integrated circuit SAW filters contain frequency response variations of up to 1 to 2 dB [40]. It is therefore possible that the composite frequency response of all components in an RF chain of an emitter using such components has variations of up to 3 or more dBs. The above spectral masks aim to simulate, for the purposes of demonstrating the utility of a new SEI algorithm, the case of several emitters



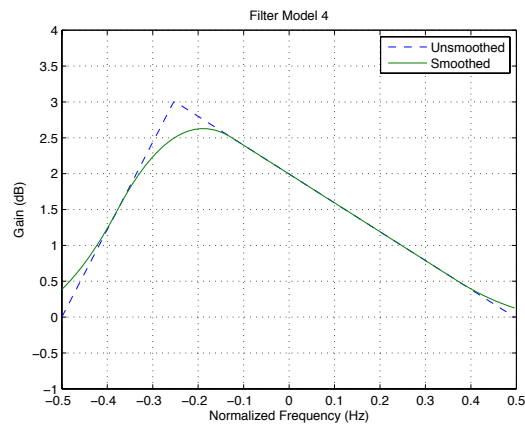
(a) Mask 1



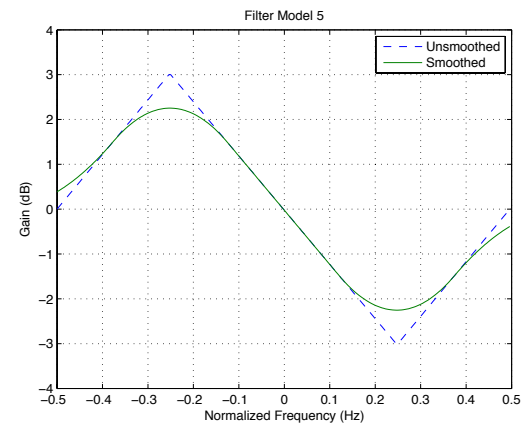
(b) Mask 2



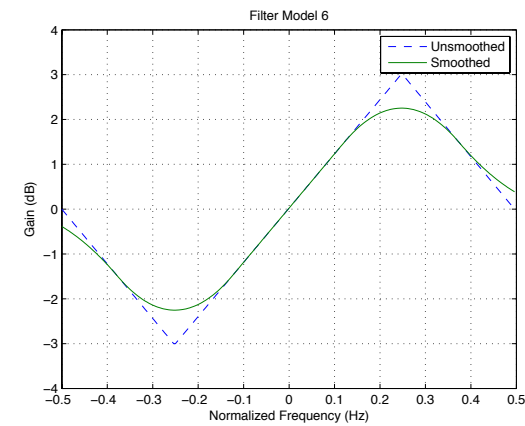
(c) Mask 3



(d) Mask 4



(e) Mask 5



(f) Mask 6

Figure 4.7: Spectral masks used for modeling spectrally non-flat RF components.

each with slightly larger than nominal, but unique, variations in its composite RF chain frequency response.

Chapter 5

Specific Emitter Identification

5.1 Introduction

Specific emitter identification is the process of analyzing the RF output of a wireless transmitter in such a way as to ascertain that the device generating the transmission is different from another device which has undergone a similar analysis. It is fundamentally a pattern recognition, or pattern classification, problem. An analogous pattern recognition problem may be to identify the individual that uttered a phrase or sentence, as opposed to determining the words that were spoken. These techniques rely on characteristics which uniquely identify one possible class out of a set of known possible classes. There may or not be a limited "alphabet" with which SEI is performed. That is, it may be necessary to have information on all devices a priori. In some cases it may also be possible to identify a "new" device, one which cannot be adequately classified by the known set, and add it to the set of known devices, however this aspect of SEI is not covered in this work.

This chapter describes the algorithm used to perform SEI of a set of 6 simulated OFDM (802.16-based) devices. Section 5.2 covers the extraction of feature vectors from eigenvectors of the spectral correlation matrix. Statistics of the feature vectors (which use the pilot eigenvectors) are shown for each of the four selected link types, all for the case of no RF component models or channel models applied. Section 5.3 details the training algorithm used to extract HMMs for the 6 known emitters, as well as the testing algorithm from which the main classification simulation results are obtained. This section also contains plots of the feature vectors with the RF component models of the 6 known emitters applied.

5.2 Feature Vector Extraction

When performing pattern recognition, it is important to extract from the data a feature vector which carries the most unique information about that data set. It is for this reason that a feature vector is chosen as the subset of the eigenvectors which allows each emitter to be identified with the least amount of ambiguity. The pilot eigenvectors, which are the subset of eigenvectors associated with the pilots (that is, at the frequency of the pilots) were chosen for use as the feature vector for each of the link types, since they can be easily separated from the subcarrier and noise eigenvectors.

In this section, an analysis of the eigenvectors are given for each of the chosen 802.16 downlink types as well as detail for how the corresponding feature vector is extracted from the data for their subsequent use in the remaining simulations contained in this chapter. Each feature vector is mapped from the pilot eigenvectors differently depending on the on the location of the pilots. This mapping is also shown for each downlink type, and plots of the statistics of the resulting feature vector are shown for each link type.

For the feature vectors in this section, an FFT size of $N' = 512$, with a skipping parameters of $L = 64$, was used to channelize the data when calculating the SCM and CSCM for each of the subsequent downlink types. Each FFT used 256 points of data padded to 512. Additionally, all simulated 802.16 snapshots in this chapter use the cyclic prefix option that is 1/8 the OFDM symbol duration as well as QPSK for data subcarrier modulation. Recall that (3.6) is used to extract a feature vector from the eigenvectors of the SCM and CSCM. In this case, $M = 6$ dominant eigenvectors are used and \mathbf{n} is chosen for each downlink type according to Table 5.1. The values for the parameter M and vector of indices \mathbf{n} were chosen observationally. Utilizing 6 eigenvectors provided a feature vector which generally contained only pilot eigenvectors, while \mathbf{n} was chosen for each downlink type by observing the indices of the eigenvectors that peaked due to the pilots.

5.2.1 No Models Applied

The following plots display the eigenvectors, and their corresponding feature vectors, extracted from the four simulated 802.16 downlink types for the case when no RF models or propagation channel models are applied. This provides a baseline against which they can be compared when both RF models and propagation channel models are applied. Each plot is the average over 20 iterations, where for each iteration 5000 symbols (a little under 0.5 seconds of integration time) of the selected 802.16 downlink type were generated from which the SCM or CSCM was calculated, eigenvector/eigenvalue decomposition were computed, and a feature vector extracted from the pilot eigenvectors. The plots of the feature vectors contain the standard deviation, in addition to the mean, of all trials.

Figure 5.2 contains plots of the average eigenvectors and feature vectors for 802.16 OFDM

Table 5.1: Eigenvector to Feature Vector Mapping

| Downlink | Matrix | Eigenvector Indices (\mathbf{n}) |
|---------------|--------|--|
| OFDM | SCM | $\mathbf{n} = \{81, 131, 181, 231, 283, 333, 383, 433\}$ |
| | CSCM | $\mathbf{n} = \{81, 131, 181, 231, 283, 333, 383, 433\}$ |
| PUSC | SCM | $\mathbf{n} = \{89, 105, 121, 137, 145, 163, 179, 193, 201, 217, 233, 249, 261, 277, 293, 309, 317, 331, 347, 365, 373, 389, 405, 421\}$ |
| | CSCM | $\mathbf{n} = \{89, 105, 121, 137, 145, 161, 177, 193, 201, 217, 233, 249, 260, 276, 292, 308, 317, 333, 349, 365, 373, 388, 404, 420\}$ |
| FUSC | SCM | $\mathbf{n} = \{45, 69, 93, 117, 141, 165, 189, 201, 213, 237, 261, 285, 309, 333, 357, 381, 405, 429, 453\}$ |
| | CSCM | $\mathbf{n} = \{46, 70, 94, 118, 142, 166, 190, 199, 214, 238, 262, 286, 310, 334, 358, 382, 406, 430, 454\}$ |
| Optional FUSC | SCM | $\mathbf{n} = \{45, 57, 69, 81, 93, 105, 117, 129, 141, 153, 165, 177, 190, 202, 214, 226, 237, 249, 261, 273, 285, 296, 308, 320, 333, 345, 357, 369, 381, 393, 405, 417, 429, 441, 453, 465\}$ |
| | CSCM | $\mathbf{n} = \{45, 57, 70, 82, 93, 106, 117, 129, 140, 153, 165, 176, 189, 201, 213, 225, 237, 248, 260, 273, 285, 297, 309, 322, 333, 345, 358, 370, 381, 394, 405, 417, 428, 441, 453, 464\}$ |

downlink. Additionally, Figure 5.3 show the same plots for OFDMA downlink PUSC, Figure 5.4 contains these plots for OFDMA downlink FUSC, and Figure 5.5 contains these plots for OFDMA downlink optional FUSC. Figure 5.1 contains a legend for the plots in this section as well as for any plots of feature vectors in the remaining sections of this chapter.

Note that in the eigenvector plots for each downlink type the pilot placement follows the placements demonstrated in the plots in Figure 4.4. This is because the dominant eigenvectors in each case are the pilot eigenvectors from (3.10), which are also visible in the example in Figure 3.1. The pilot eigenvectors can be thought of as alternate two dimensional representation of the pilot-only OFDM spectrum. The power difference of the pilots, as seen in the pilot eigenvectors, is due to varying levels of spectral correlation in the SCM (or CSCM) from which the eigenvectors are extracted. Also note that the feature vectors follow the peaks of eigenvectors, and are extracted from the eigenvectors according to (3.6) and Table 5.1. Again, the purpose of these plots is to provide a baseline against which to compare the effect of the spectral masks on each of the downlink types.



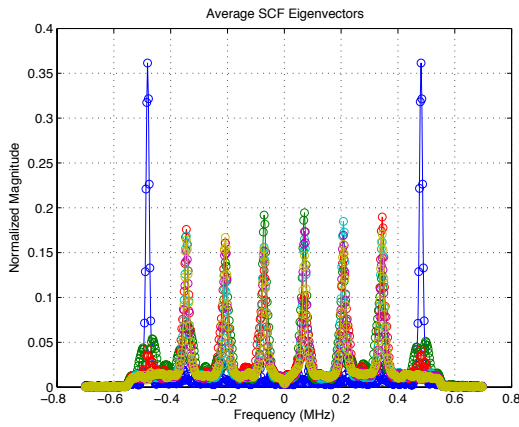
Figure 5.1: Legend for eigenvector and feature vector plots.

5.2.2 Models Applied

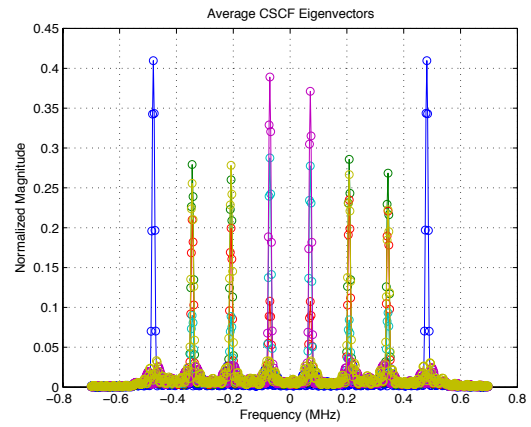
The following plots demonstrate the statistics of the feature vectors for the six RF component models specified in Chapter 4 with WGN added to achieve 10 dB SNR. This shows the effect that the different RF models have on the feature vectors for each simulated 802.16 downlink type. Each plot contains the statistics (mean and standard deviation) of a feature vector averaged over 20 snapshots, each of which is generated with 3000 symbols per snapshot (a little over 0.25 seconds of integration time).

Figures 5.6 and 5.7 contain plots of the feature vectors of the SCM and CSCM respectively for each of the six RF models applied to 802.16 OFDM downlink. Figures 5.8 and 5.9 contain plots of the feature vectors of the SCM and CSCM for OFDMA downlink PUSC with the six models applied. Figures 5.10 and 5.11 plot the FVs for the six RF models for OFDMA downlink FUSC. And lastly, Figures 5.12 and 5.13 contain the same plots for OFDMA downlink optional FUSC.

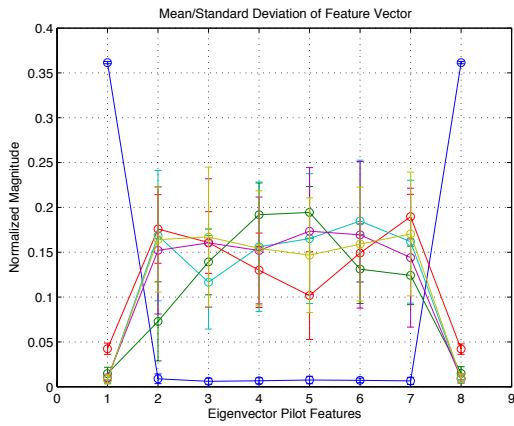
These plots are meant to demonstrate the uniqueness of the feature vectors of the 6 different emitters, each simulated using a unique spectral mask shown in Figure 4.7, for each of the four downlink types. It can be gleaned from the plots that variability in the spectral masks used to simulated each emitter have a greater effect on the uniqueness of the feature vectors extracted from the SCM than those extracted from the CSCM. This is possibly due to fact that the conjugate spectrum is computed by cross multiplying an upright and inverted DFT, thus along the diagonal of the CSCM the effects of any asymmetrical power distribution of the pilots would be cancelled. That is, the stronger pilots are correlated with weaker pilots and vice-versa. This is apparent when comparing Figures 5.6 and 5.7, where the CSCM plots generally symmetric whereas the SCM plots are generally asymmetric. The lack of uniqueness in the CSCM plots is also possibly due to the fact that the CSCM has no energy in the PSD for the QPSK modulated subcarriers. It only has energy for the BPSK pilots.



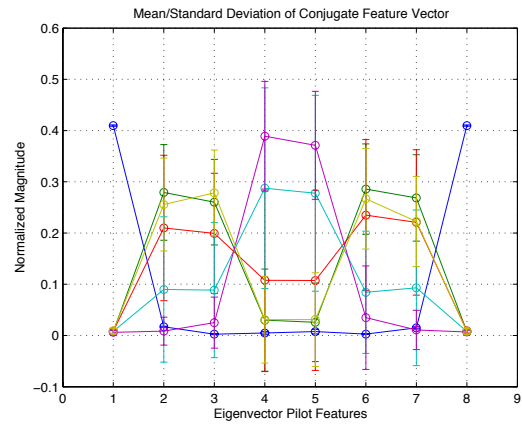
(a) Average SCM Eigenvalues



(b) Average CSCM Eigenvalues

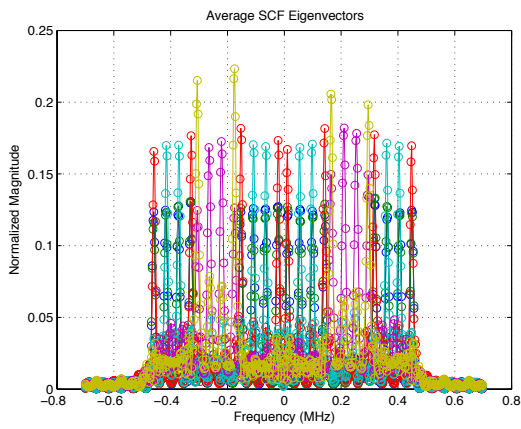


(c) FV Statistics

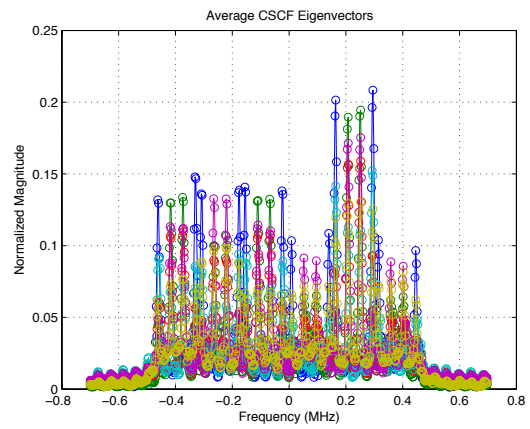


(d) Conjugate FV Statistics

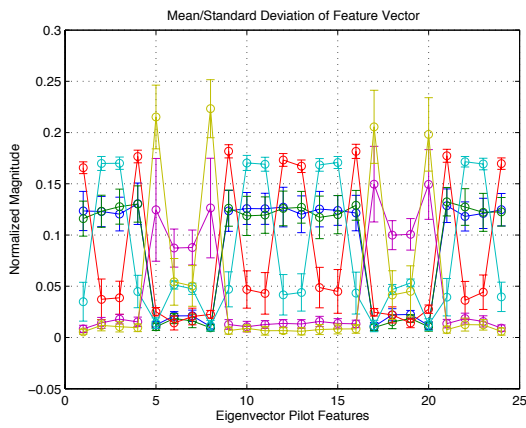
Figure 5.2: (a) and (b): Average of the dominant 6 (pilot) eigenvectors of the SCM and CSCM for OFDM downlink using 2.5 MHz bandwidth (256 subcarriers). (c) and (d): Mean and standard deviation of the feature vectors of the SCM and CSCM for OFDM downlink using 2.5 MHz bandwidth. Note that the FV of the CSCM initially appears more unique, and stronger, than that of the SCM.



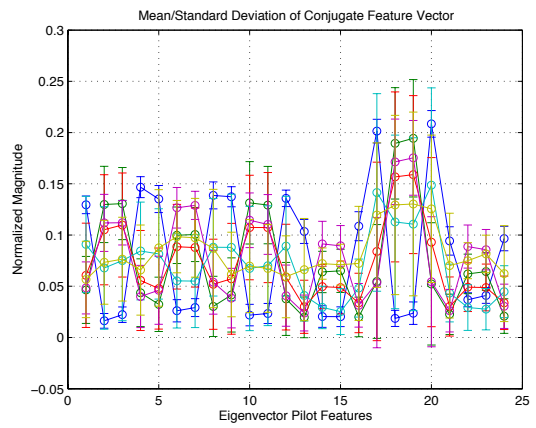
(a) Average SCM Eigenvectors



(b) Average CSCM Eigenvectors

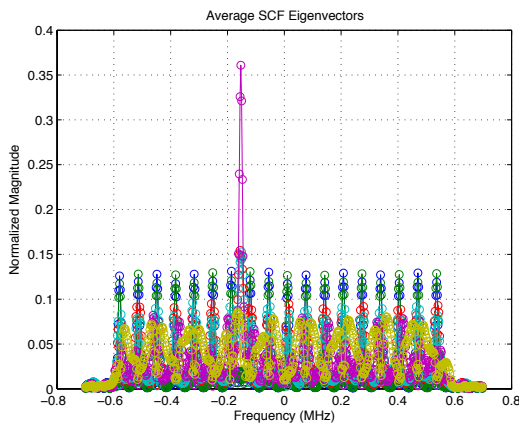


(c) FV Statistics

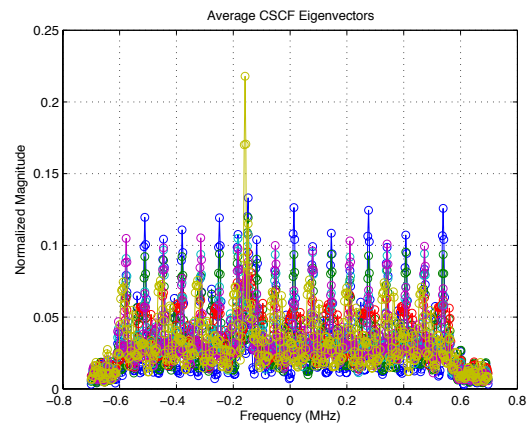


(d) Conjugate FV Statistics

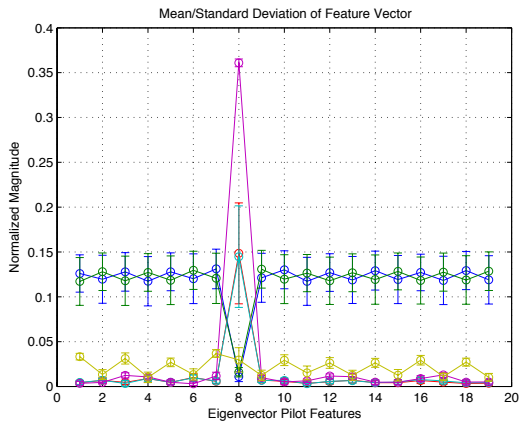
Figure 5.3: (a) and (b): Average of the dominant 6 (pilot) eigenvectors of the SCM and CSCM for OFDMA downlink PUSC using 1.25 MHz bandwidth (128 subcarriers). (c) and (d): Mean and standard deviation of the feature vectors of the SCM and CSCM for OFDMA downlink PUSC using 1.25 MHz bandwidth. Here the multi-dimensional effect of the pilot eigenvectors is clearly seen as the pilot spectral energy is distributed across all 6 eigenvectors.



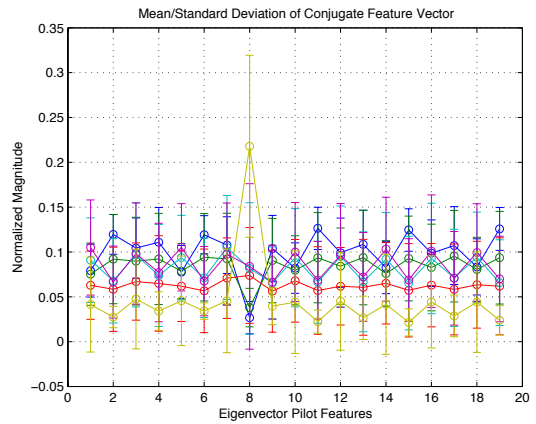
(a) Average SCM Eigenvectors



(b) Average CSCM Eigenvectors

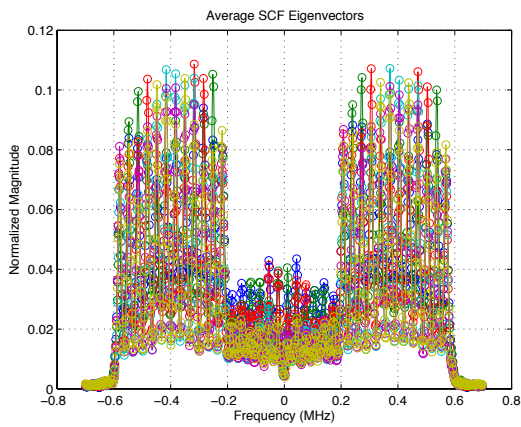


(c) FV Statistics

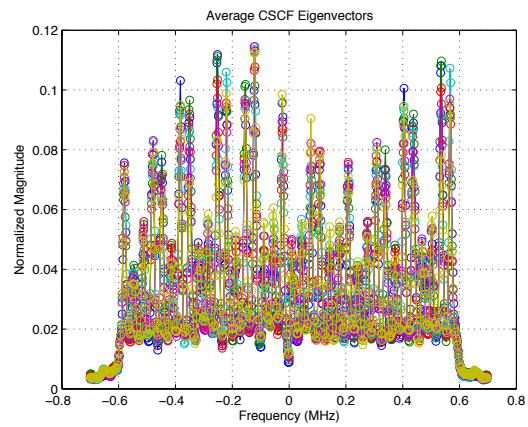


(d) Conjugate FV Statistics

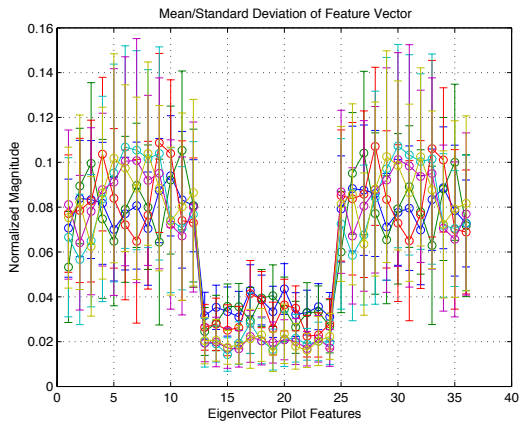
Figure 5.4: (a) and (b): Average of the dominant 6 (pilot) eigenvectors of the SCM and CSCM for OFDMA downlink FUSC using 1.25 MHz bandwidth (128 subcarriers). (c) and (d): Mean and standard deviation of the feature vectors of the SCM and CSCM for OFDMA downlink FUSC using 1.25 MHz bandwidth. Note, in the case of the SCM, the single constant pilot in the 5th dominant eigenvector and the even distribution of the remaining pilots in the 1st and 2nd dominant eigenvectors.



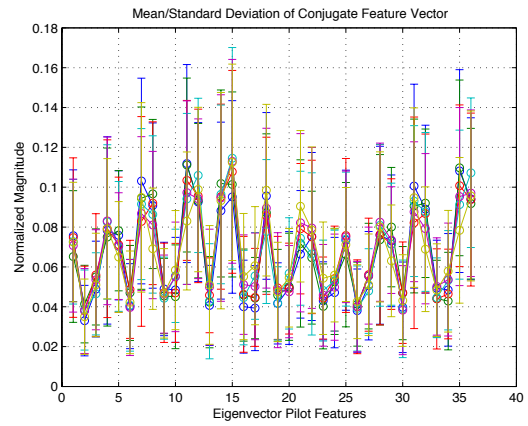
(a) Average SCM Eigenvectors



(b) Average CSCM Eigenvectors

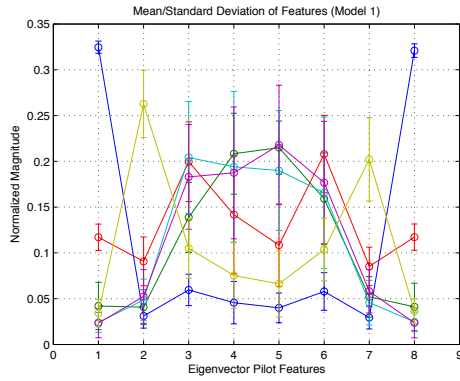


(c) FV Statistics

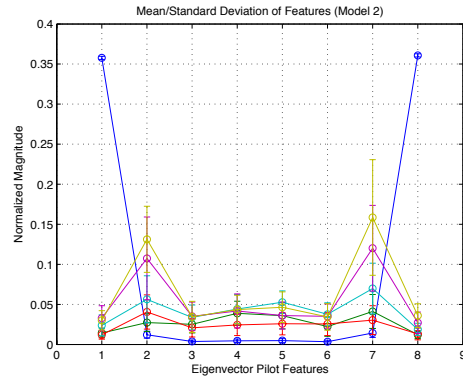


(d) Conjugate FV Statistics

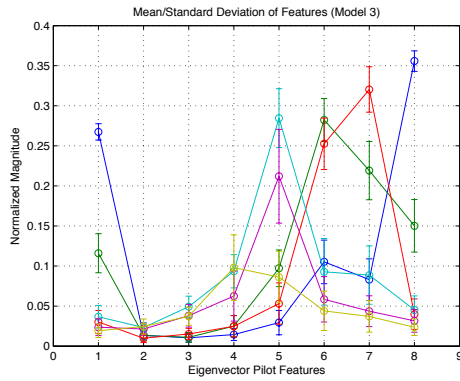
Figure 5.5: (a) and (b): Average of the dominant 6 (pilot) eigenvectors of the SCM and CSCM for OFDMA downlink optional FUSC using 1.25 MHz bandwidth (128 subcarriers). (c) and (d): Mean and standard deviation of the feature vectors of the SCM and CSCM for OFDMA downlink optional FUSC using 1.25 MHz bandwidth. Here the pilot spectral energy is distributed evenly across all 6 eigenvectors, in contrast to the previous 3 downlink types.



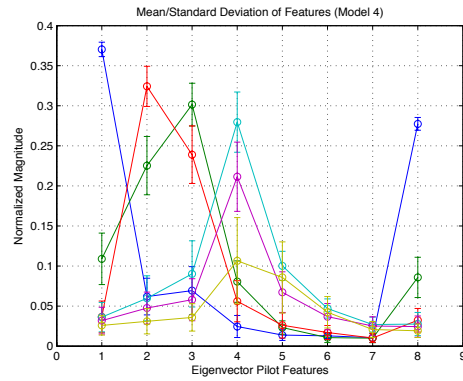
(a) Emitter 1



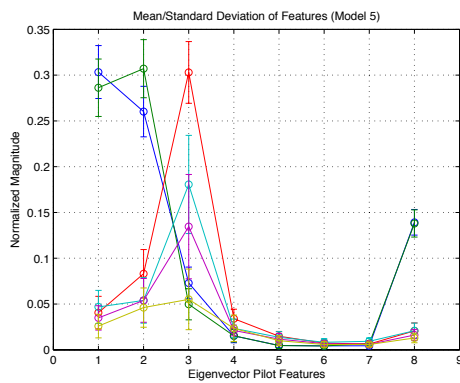
(b) Emitter 2



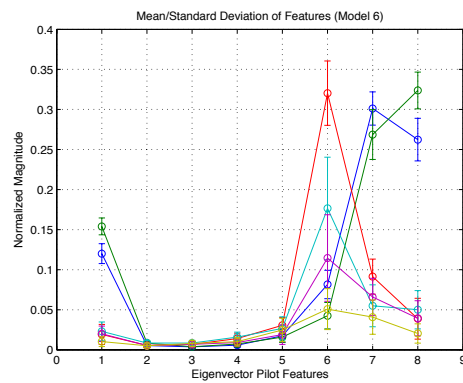
(c) Emitter 3



(d) Emitter 4

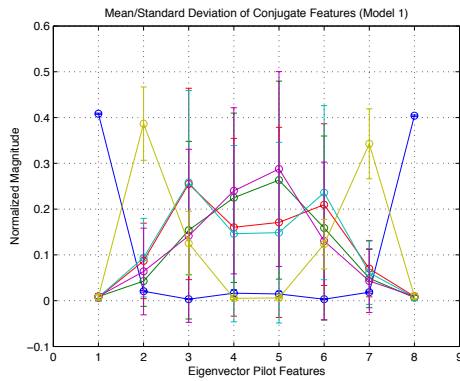


(e) Emitter 5

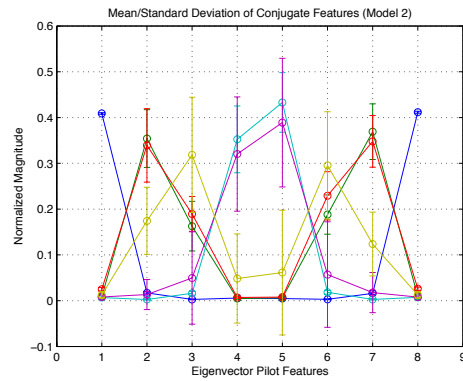


(f) Emitter 6

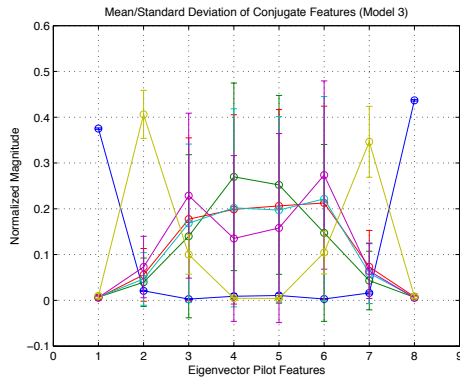
Figure 5.6: Mean and standard deviation of the SCM feature vectors of the 6 simulated emitters for OFDM downlink (2.5 MHz bandwidth). Notice that the statistics are unique for each emitter, and they closely follow the spectral masks in Figure 4.7.



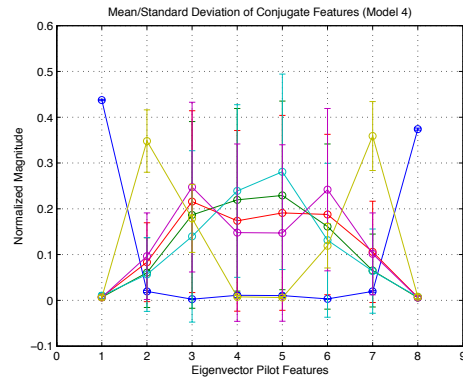
(a) Emitter 1



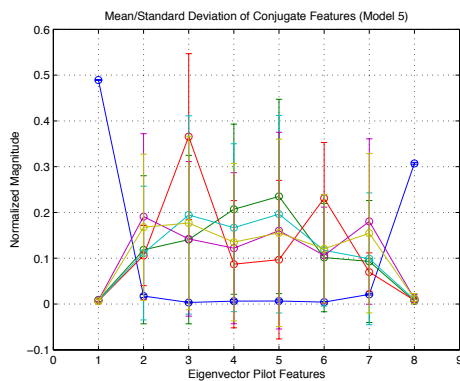
(b) Emitter 2



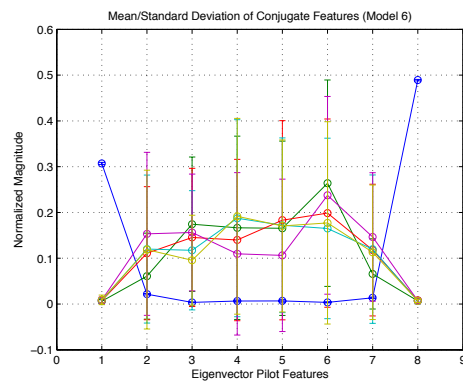
(c) Emitter 3



(d) Emitter 4

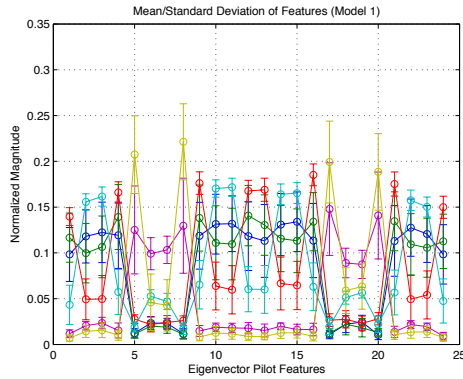


(e) Emitter 5

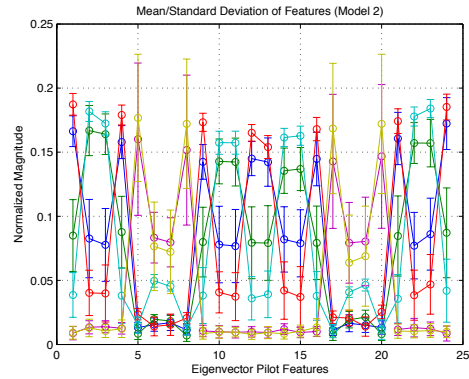


(f) Emitter 6

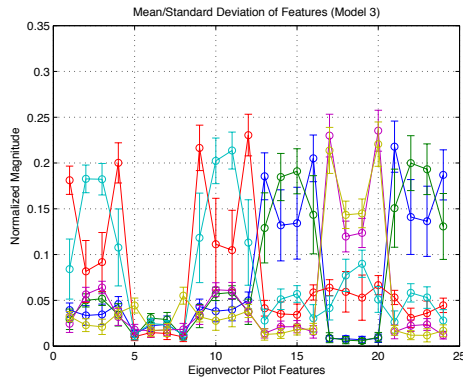
Figure 5.7: Mean and standard deviation of the CSCM feature vectors of the 6 simulated emitters for OFDM downlink (2.5 MHz bandwidth). Here it can be seen that the statistics have greater deviation and lower uniqueness than those of the SCM for OFDM downlink.



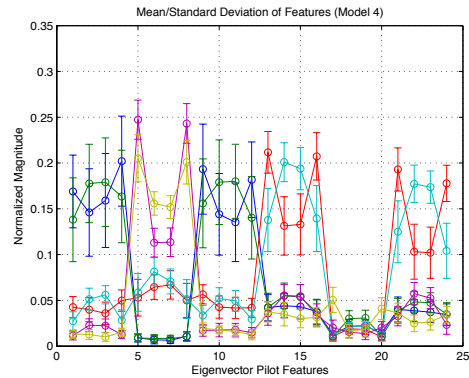
(a) Emitter 1



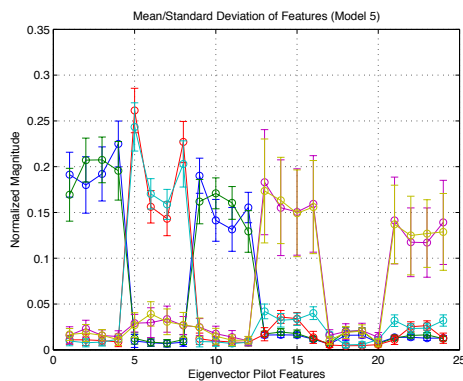
(b) Emitter 2



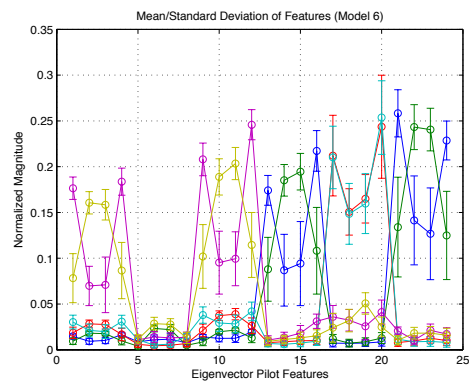
(c) Emitter 3



(d) Emitter 4

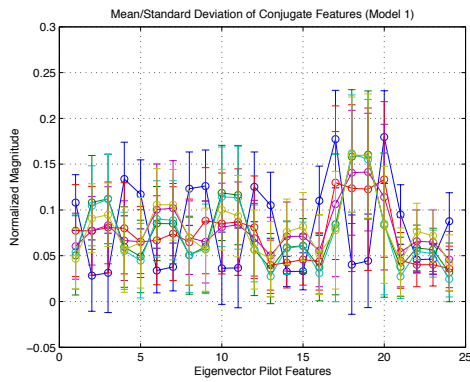


(e) Emitter 5

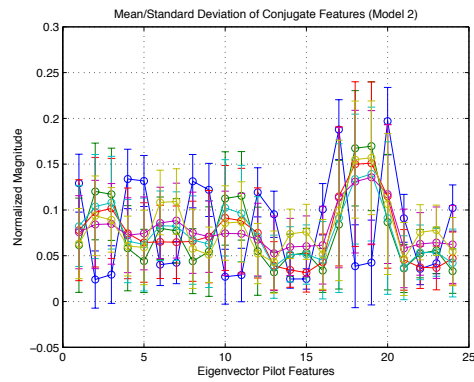


(f) Emitter 6

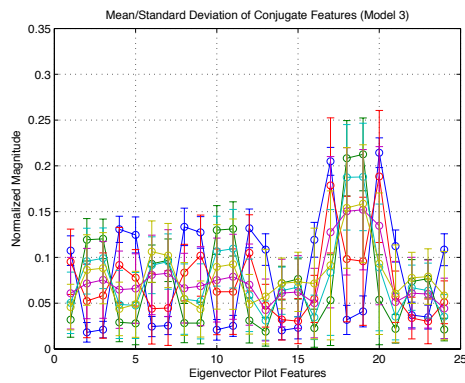
Figure 5.8: Mean and standard deviation of the SCM feature vectors of the 6 simulated emitters for OFDMA downlink PUSC (1.25 MHz bandwidth). Notice the features appear in different eigenvectors depending on the shape of the spectral mask of each emitter.



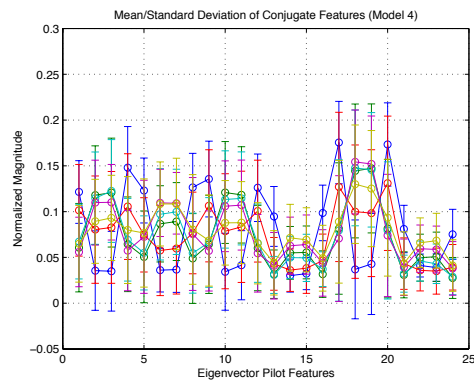
(a) Emitter 1



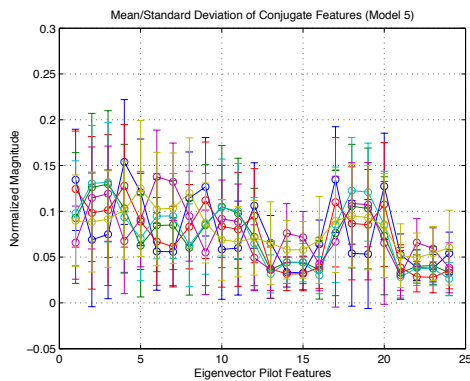
(b) Emitter 2



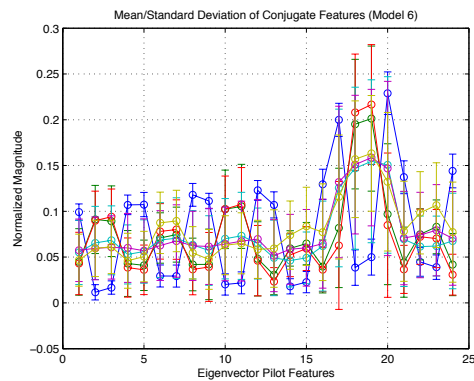
(c) Emitter 3



(d) Emitter 4

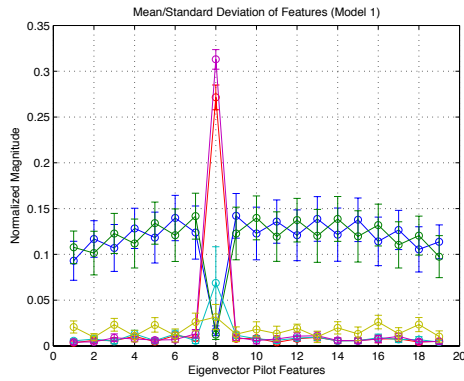


(e) Emitter 5

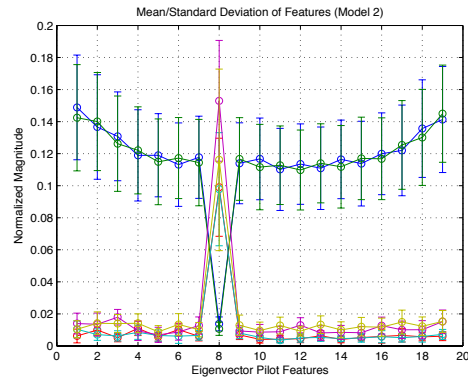


(f) Emitter 6

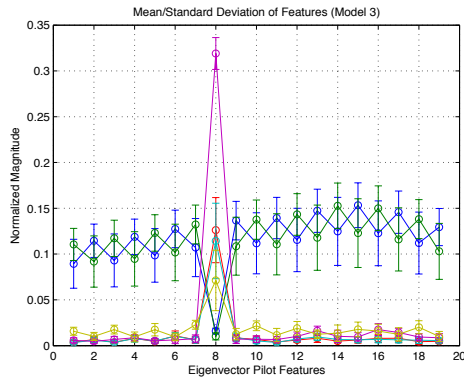
Figure 5.9: Mean and standard deviation of the CSCM feature vectors of the 6 simulated emitters for OFDMA downlink PUSC (1.25 MHz bandwidth). Again, uniqueness is seen between the statistics of the emitters, but it is far less than for the case of the SCM for OFDMA downlink PUSC.



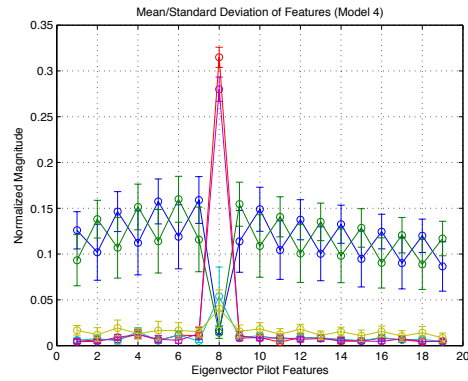
(a) Emitter 1



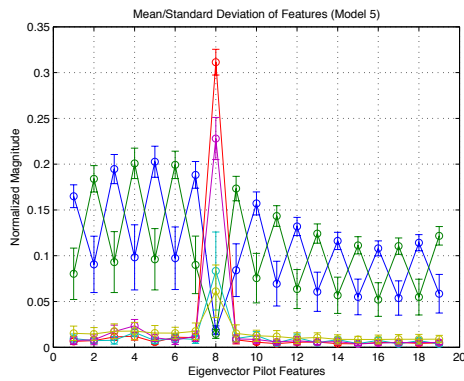
(b) Emitter 2



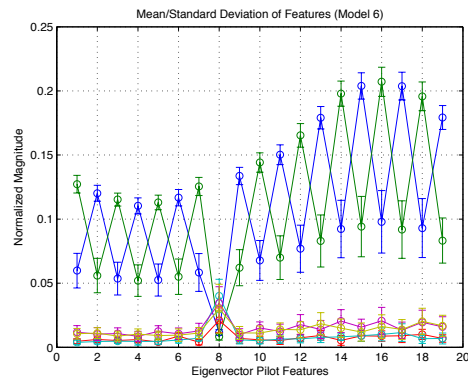
(c) Emitter 3



(d) Emitter 4

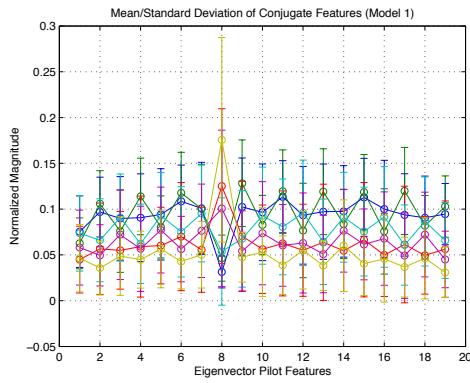


(e) Emitter 5

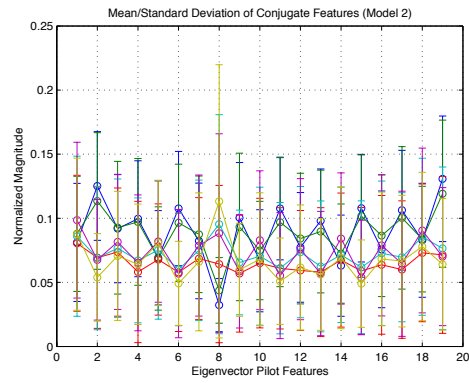


(f) Emitter 6

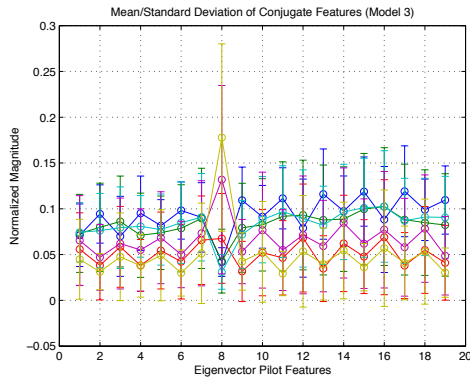
Figure 5.10: Mean and standard deviation of the SCM feature vectors of the 6 simulated emitters for OFDMA downlink FUSC (1.25 MHz bandwidth). Since the majority of the features are constricted to the 2 dominant eigenvectors, the uniqueness of the statistics is diminished compared to the case when the features are distributed throughout all of the eigenvectors.



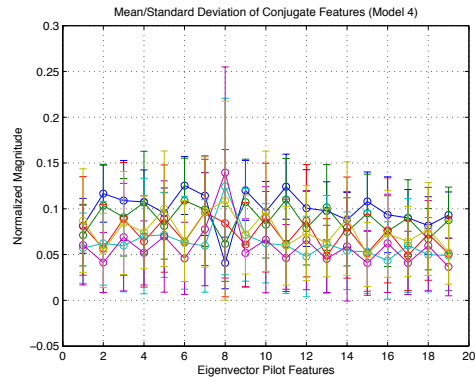
(a) Emitter 1



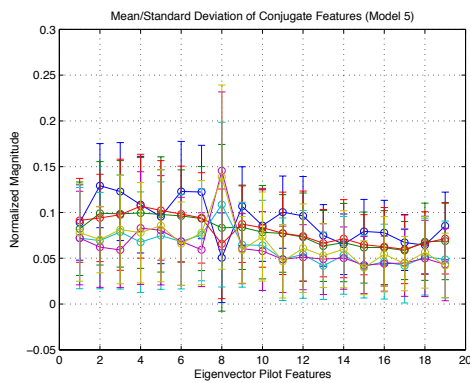
(b) Emitter 2



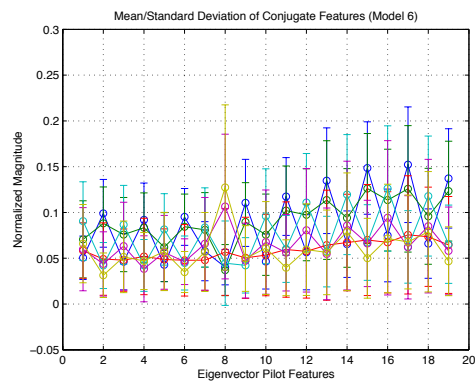
(c) Emitter 3



(d) Emitter 4

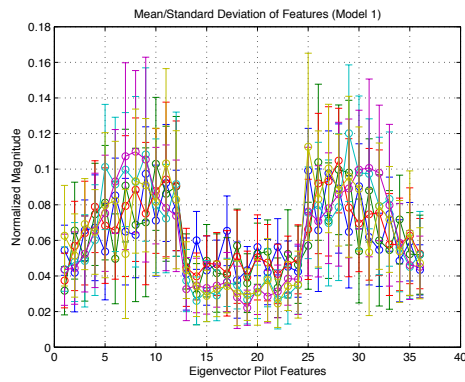


(e) Emitter 5

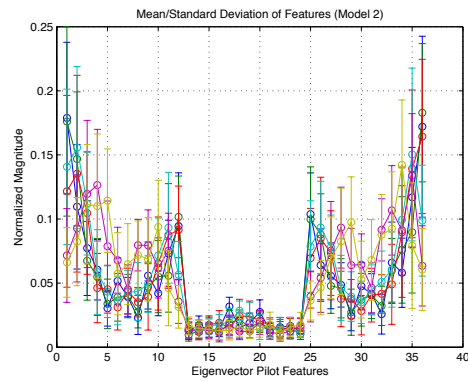


(f) Emitter 6

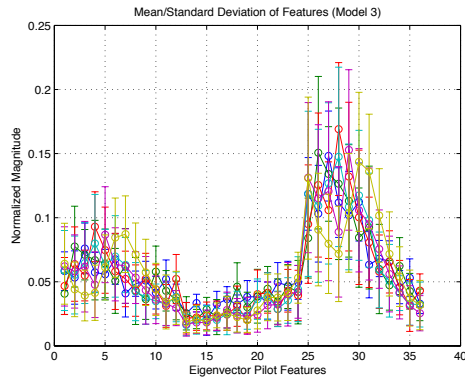
Figure 5.11: Mean and standard deviation of the CSCM feature vectors of the 6 simulated emitters for OFDMA downlink FUSC (1.25 MHz bandwidth). The deviation is higher than the SCM case, however the pilot features are distributed through the eigenvectors.



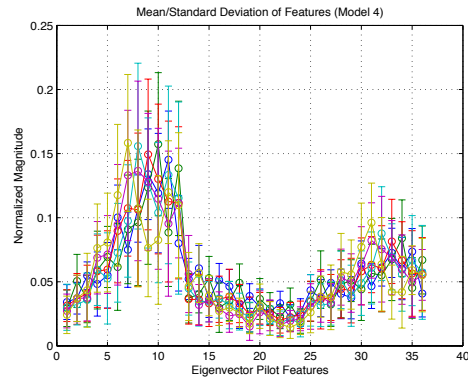
(a) Emitter 1



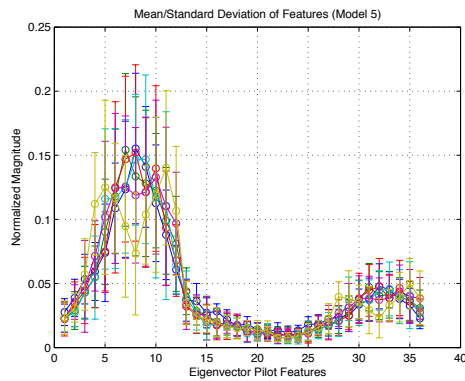
(b) Emitter 2



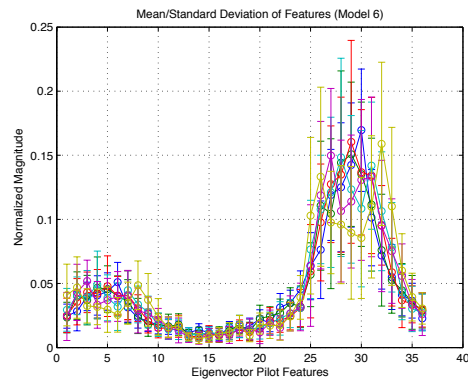
(c) Emitter 3



(d) Emitter 4

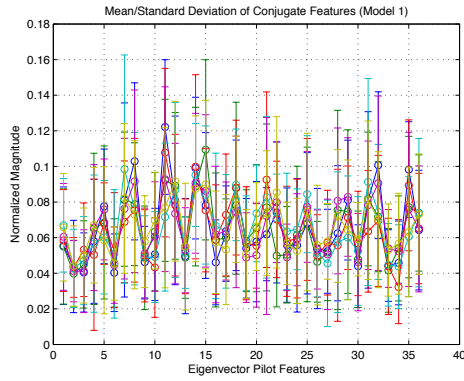


(e) Emitter 5

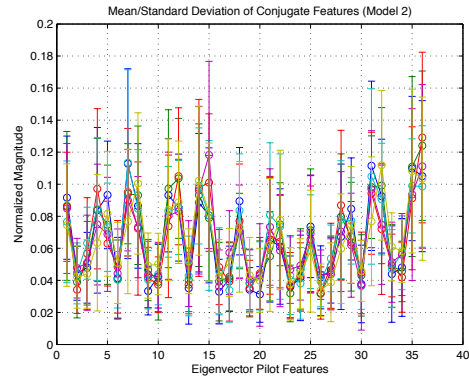


(f) Emitter 6

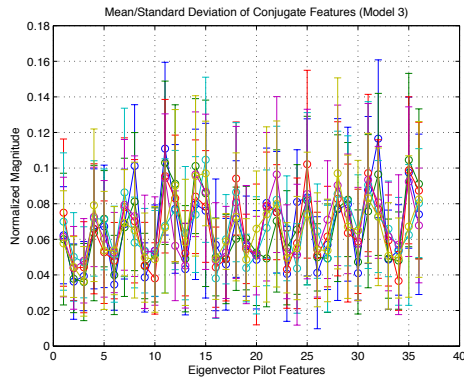
Figure 5.12: Mean and standard deviation of the SCM feature vectors of the 6 simulated emitters for OFDMA downlink optional FUSC (1.25 MHz bandwidth). A much high deviation is seen in the statistics of the feature vectors, however they are unique for each emitter and distributed evenly throughout the eigenvectors.



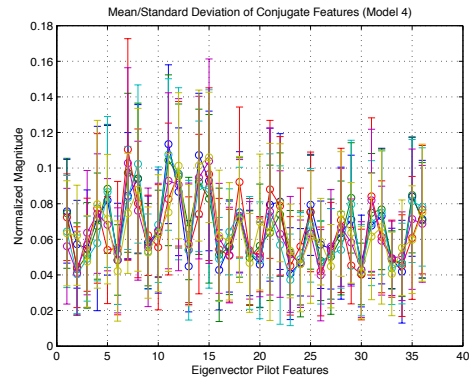
(a) Emitter 1



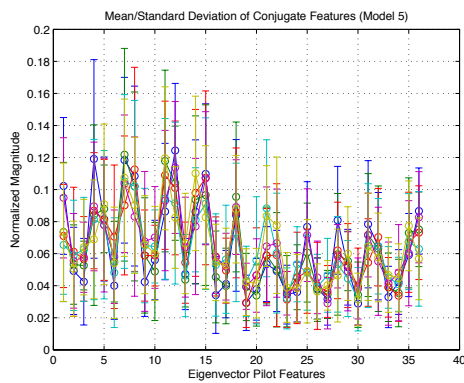
(b) Emitter 2



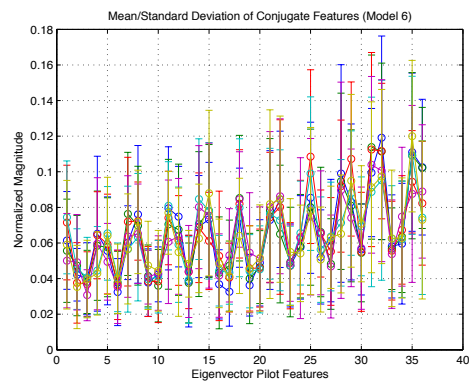
(c) Emitter 3



(d) Emitter 4



(e) Emitter 5



(f) Emitter 6

Figure 5.13: Mean and standard deviation of the CSCM feature vectors of the 6 simulated emitters for OFDMA downlink optional FUSC (1.25 MHz bandwidth). These statistics have high deviation and low uniqueness.

5.3 Classification Test

In order to test the utility of the EVD as a tool for SEI of OFDM-based signals, it is necessary to devise an algorithm which attempts to classify a series of unknown emitters. This is accomplished by performing a pattern recognition simulation, using HMMs as the the pattern recognition algorithm in this case. However, because HMMs as a tool for pattern recognition require a priori information about the dataset being tested, it is necessary to divide the simulation in two distinct sections. The first part is the training section, wherein HMMs are extracted from feature vectors of each of the known emitters. The second part is the testing (or classification) section, wherein the HMM extracted from the feature vector of an unknown emitter is compared against HMMs from all of the known emitters, and a single emitter is chosen from this comparison.

5.3.1 Algorithm

In the training section, depicted in Figure 5.14, 10 OFDM snapshots are generated to train an HMM for a single emitter. Since six emitters are simulated as outlined in the Chapter 4, this process is repeated six times, but it must be done for each emitter in the known dataset. This assumes that the number of emitters in the dataset after training is equal to the number of emitters in the testing simulation, and that the emitters are the same in both cases. For each OFDM snapshot generated for testing, the SCM and CSCM are computed, and each of these are decomposed into eigenvalues and corresponding eigenvectors. A feature vector is then extracted from the dominant eigenvectors of each snapshot, and is then used to train an HMM associated with that emitter. It should be noted that no noise is added to the snapshots used for training HMMs. This comes from an assumption that a dataset for SEI is built in a laboratory environment where access to the radios being tested is unrestricted, and data can be collected in an environment free from interference and noise.

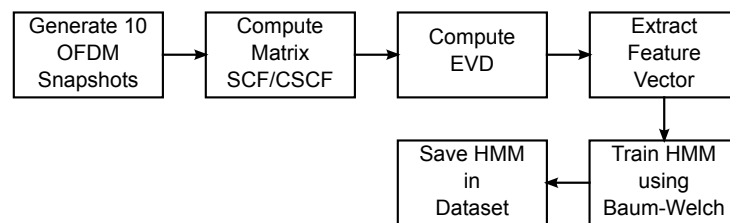


Figure 5.14: Block diagram of HMM training for a single simulated emitter (RF model).

Likewise in the testing section, depicted in Figure 5.15, 100 OFDM snapshots are generated to test the classification of each simulated emitter. As in the training section, the SCM and CSCM are computed for each snapshot, and from these the eigenvectors, and hence, a

feature vector, are extracted. For each previously trained emitter HMM in the training set, the log-likelihood probability is computed that the unknown emitter has the same HMM. The maximum of these is used to classify the unknown emitter.

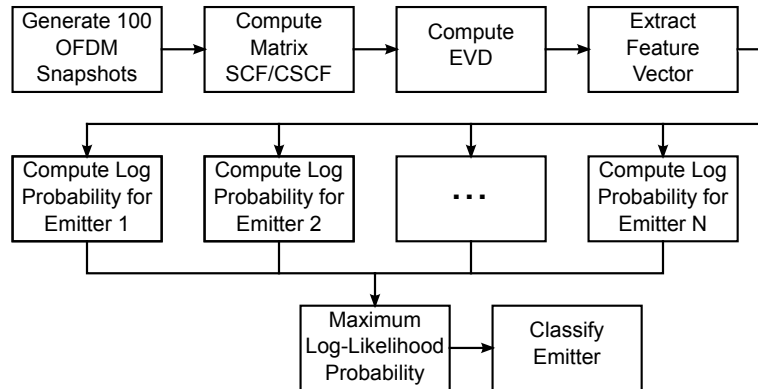


Figure 5.15: Block diagram of HMM training (emitter classification) for a single simulated emitter (RF model).

5.3.2 Simulation and Results

Using the above outlined simulation parameters, an emitter classification test was performed for each of the four 802.16 downlink types and the six simulated emitters described in Chapter 4. For each of the six emitters, 10 snapshots were used to train the HMMs, with 3000 symbols per snapshot and no noise added. For each downlink four confusion matrix plots are shown. The results shown in the first two plots, for each downlink type, are from classification tests that were performed using the SCM only for both 10 dB and 0 dB SNR. The results in the second two plots for each downlink type are from classification tests performed using both the SCM and CSCM. The results are shown in Figures 5.16 through 5.23 as confusion matrix plots.

Each confusion matrix plot is the result of a classification test performed using 100 trials. The rows are the actual emitters tested, with the top row being emitter 1 and the bottom row being emitter 6, and the columns are the detected emitter, with the left column being emitter 1 and the right column being emitter 6. In this case "emitter" refers to which spectral mask in Figure 4.7 was applied to the OFDM snapshot. The color intensity at each point in the grid indicates the number of snapshots that were classified as a particular emitter. So, for example, a confusion matrix plot of all dark red along the diagonal and blue everywhere else would indicate that all six emitters were correctly identified for all 100 trials. Each plot is, in essence, a graphical representation of a confusion matrix, which would otherwise be a numerical representation of the results of a classification test. The numbers in each square

are the number of times the unknown emitter was classified as the emitter in the that column. No number is specified in the cases where the number of classifications is zero.

Overall, the results of these tests are excellent for OFDM downlink and OFDMA downlink OFUSC, adequate for OFDMA downlink PUSC and mostly unacceptable for OFDMA downlink FUSC. For all downlink types the 0 dB SNR case produced worse results, as expected, than the 10 dB SNR case, except for the case of OFDMA downlink OFUSC which resulted in perfect or near-perfect classification in all cases. One unexpected result is that use of both the SCM and CSCM feature vectors for pattern classification produced worse results than use of the SCM only (again, except for OFDMA downlink OFUSC). One reason for this may be that QPSK was used as the subcarrier modulation in all tests. This results in a conjugate spectrum (the diagonal of the CSCM) which is not spectrally correlated, other than the for pilots which are BPSK modulated. Use of QPSK for the subcarriers leaves a CSCM with less overall spectral correlation, and this is compounded by the fact that the EVD generally performs better for the case when there is higher energy along the diagonal of the matrix than elsewhere. Also, the symmetry of the CSCM feature vectors, relative to the SCM feature vectors, may produce an overall feature vector which is less unique than for the SCM alone. Another case of notably poor results is the case of OFDMA downlink FUSC. This is likely due to the fact that the SCM feature vectors for this case had the majority of the features concentrated in the dominant 2 eigenvectors, thus the uniqueness of each feature vector overall was lower than those in which the features are distributed among all 6 eigenvectors.

5.3.3 Effect of Multipath on Discernability

One effect which can realistically be expected when transitioning this SEI algorithm to the case of real-world data is that of a fading environment due to multi path. For this reason it is desirable to have a baseline result, wherein the best case scenario of the above tests is reconfigured to simulate a fading environment. In the prior results, the best performing downlink type was OFDMA downlink OFUSC, and the best test conditions came from the 10 dB SNR case using the SCM eigenvectors only. This case was chosen for the multipath test in this section, wherein a standard channel propagation model was applied to OFDM during the channel model portion of the simulation.

A wide range of standard channel propagation models are readily available. One such standard is the Stanford University Interim (SUI) series of channel models. The SUI standard is comprised of a set of six channel models (SUI-1 through SUI-6) which cover a wide range of favorable and unfavorable transmission conditions [41]. The chosen model for use in this test is the SUI-3 model whose parameters can be seen in Table 5.2. This particular model provides a moderate amount of signal degradation, but is less severe than models 4 through 6. It has a low degree of doppler and delay spread, but is also only partially LOS (line of sight). The simulated terrain of the model is described as hilly terrain with light tree density

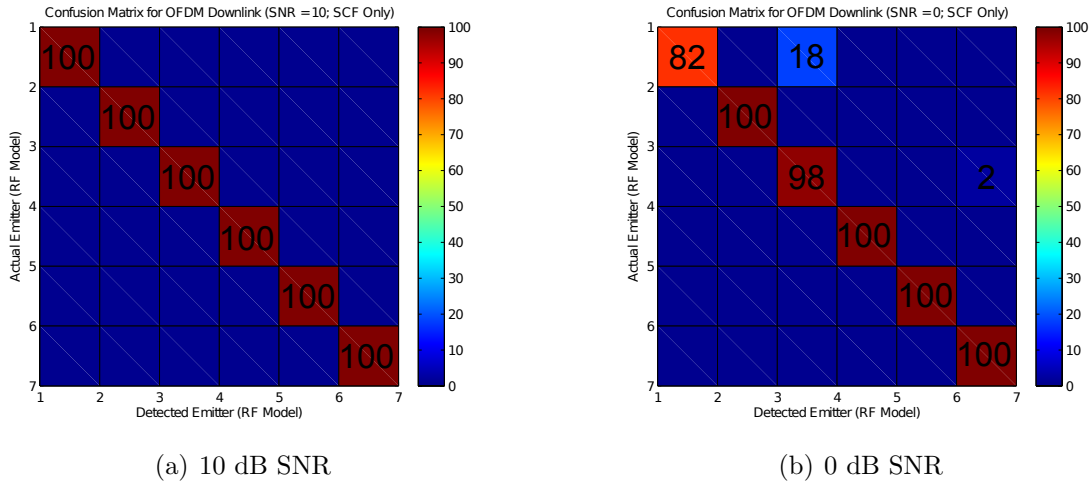


Figure 5.16: Confusion matrices for SEI classification tests of OFDM downlink with 10 dB and 0 dB SNR using SCM only. Results are excellent except for emitter 1 in the 0 dB SNR case.

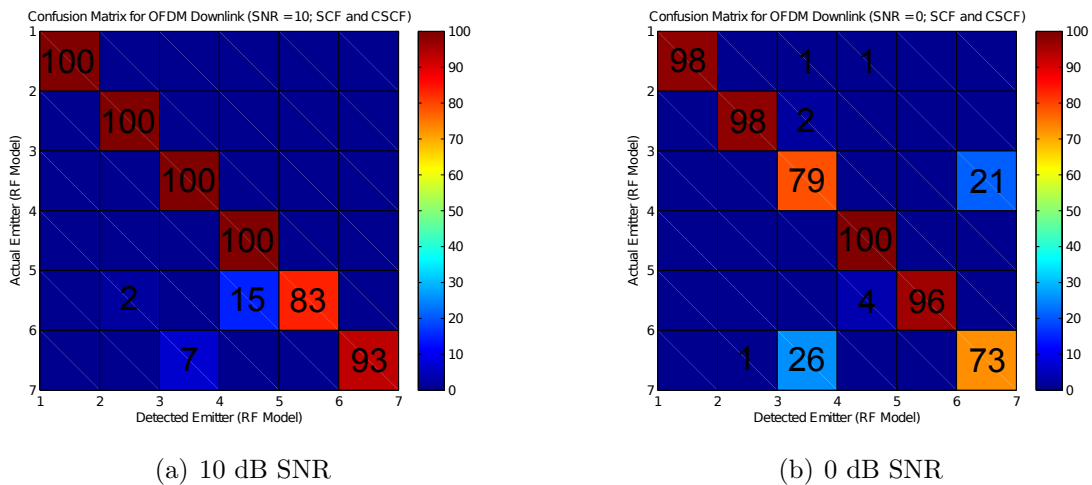


Figure 5.17: Confusion matrices for SEI classification tests of OFDM downlink with 10 dB and 0 dB SNR using SCM and CSCM. For the 0 dB SNR case there is a slight ambiguity between emitter 3 and emitter 6.

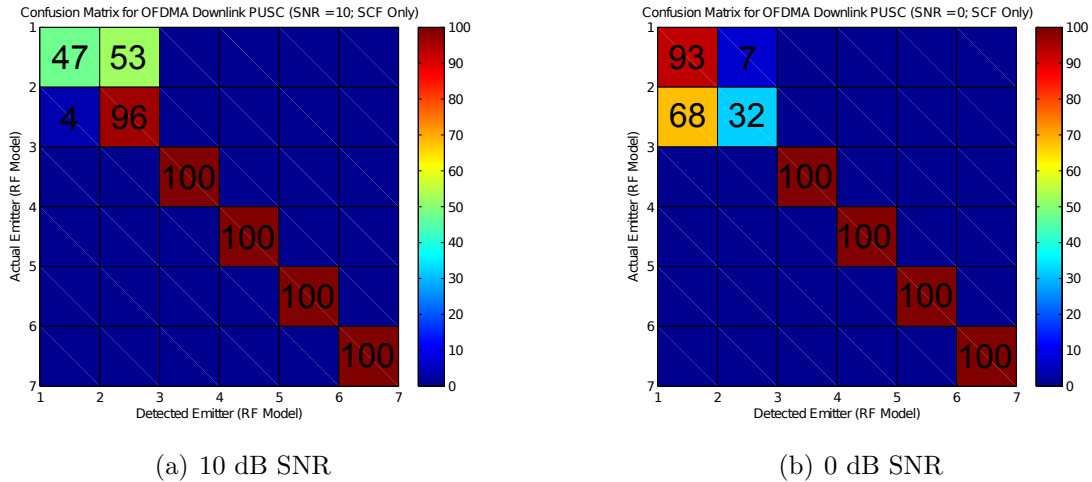


Figure 5.18: Confusion matrices for SEI classification tests of OFDMA downlink PUSC with 10 dB and 0 dB SNR using SCM only. Ambiguity between emitter 1 and emitter 2 is apparent.

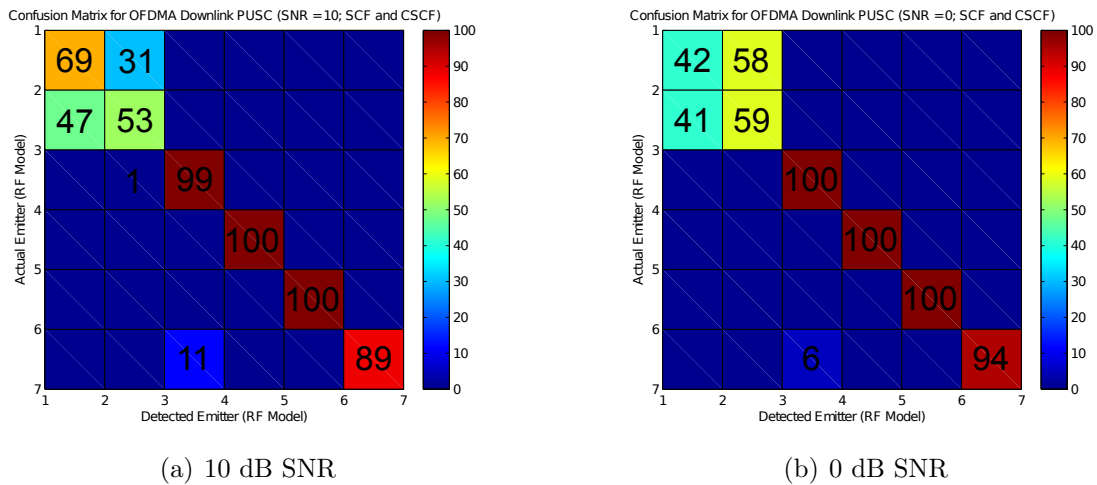


Figure 5.19: Confusion matrices for SEI classification tests of OFDMA downlink PUSC with 10 dB and 0 dB SNR using SCM and CSCF. Note that results due to the lack of uniqueness between SCM feature vectors extracted from emitters 1 and 2 were worsened by the non-uniqueness of the CSCF feature vectors.

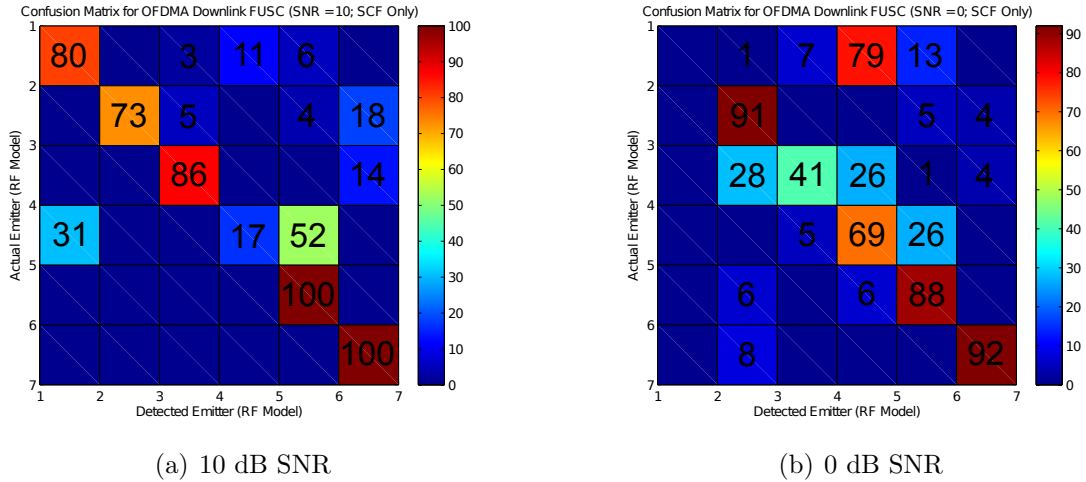


Figure 5.20: Confusion matrices for SEI classification tests of OFDMA downlink FUSC with 10 dB and 0 dB SNR using SCM only. Performance is acceptable for all but emitter 4 at 10 dB SNR. For the 0 dB SNR case performance is generally poor.

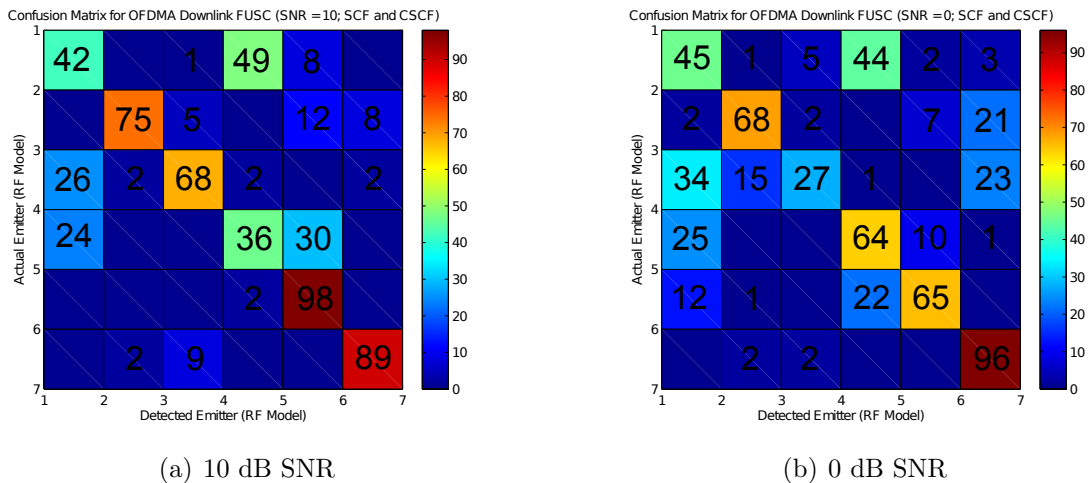


Figure 5.21: Confusion matrices for SEI classification tests of OFDMA downlink FUSC with 10 dB and 0 dB SNR using SCM and CSCM. Noting the overall poor performance, the non-uniqueness in the feature vectors of both the SCM and CSCM is clearly apparent.

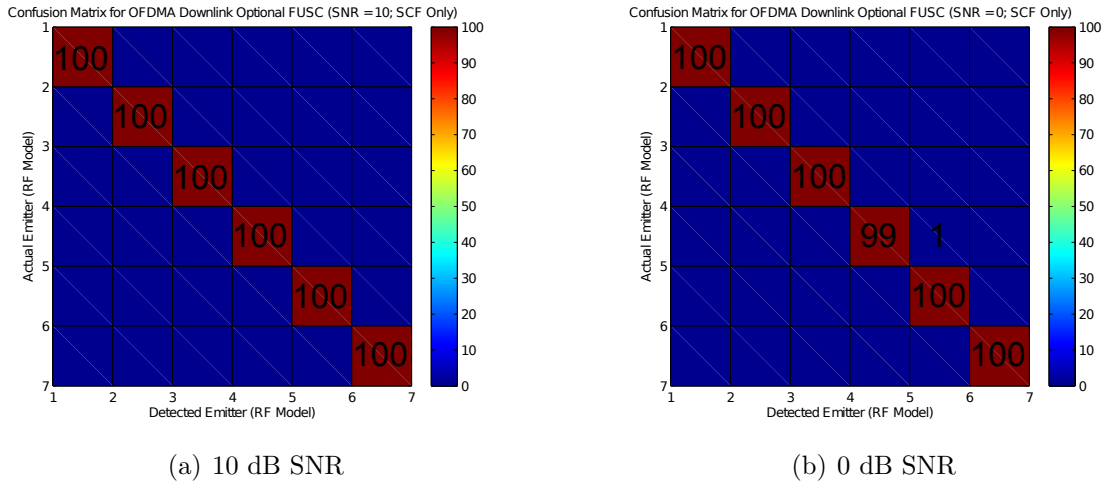


Figure 5.22: Confusion matrix for SEI classification test of OFDMA downlink optional FUSC with 10 dB and 0 dB SNR using SCM only. Nearly perfect classification was obtained in both cases.

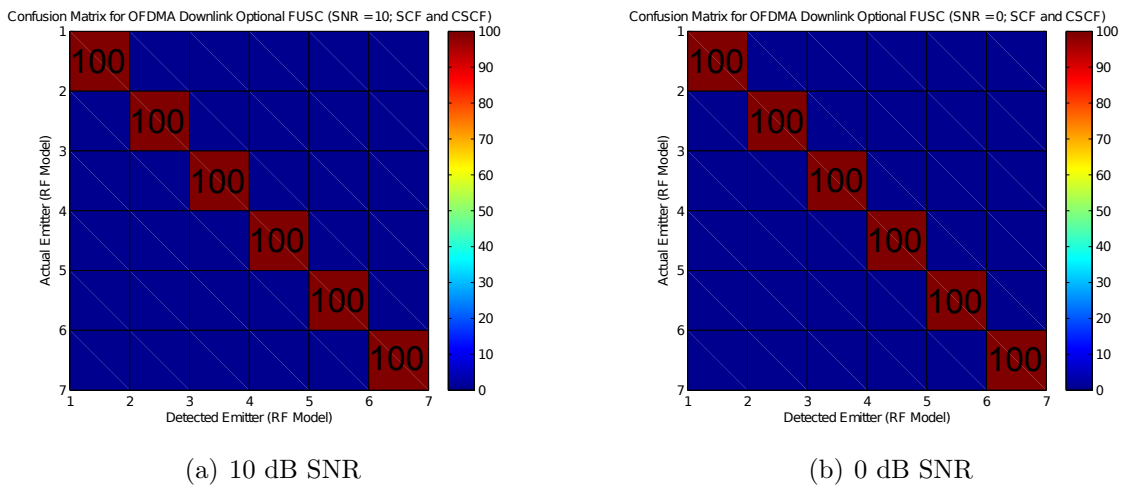


Figure 5.23: Confusion matrices for SEI classification tests of OFDMA downlink optional FUSC with 10 dB and 0 dB SNR using SCM and CSCM. Perfect classification was obtained in both cases.

or flat terrain with moderate-to-heavy tree density [41].

Table 5.2: SUI-3 Channel Model Parameters

| Parameter | Tap 1 | Tap 2 | Tap 3 | Units |
|-----------|-------|-------|-------|---------|
| Delay | 0 | 0.4 | 0.9 | μs |
| Power | 0 | -5 | -10 | dB |
| K-factor | 1 | 0 | 0 | |
| Doppler | 0.4 | 0.3 | 0.5 | Hz |

A LOS channel model with NLOS (non-line-of-sight) components can be achieved in MATLAB using the `ricianchan` function found in the communications toolbox. The ratio of LOS to NLOS components is determined by the K-factor. The simulation depicted in Figure 5.24 is another trial of the simulation in the previous section which used OFDMA downlink OFUSC at 10 dB SNR with SCM only, except that each snapshot was filtered using a channel generated by the `ricianchan` function, which was input with the model parameters listed in Table 5.2. Upon inspection, the results are far worse than those depicted in Figure 5.22, and this is clearly due to the level of multipath fading. Additional processing is necessary to retain the optimum functionality of the classification algorithm when multipath is present, however the information obtained from the algorithm may still have some usefulness when performance is suboptimal.

One method which may increase the performance of this algorithm in the presence of multipath is to perform the training portion of the SEI algorithm with the multipath channel applied to the training snapshots. Any frequency components in the training snapshots which vary due to the channel may be averaged out given a sufficiently long integration time. This technique would be especially applicable in the scenario of a femtocell implementation, wherein the propagation channel could potentially remain the same.

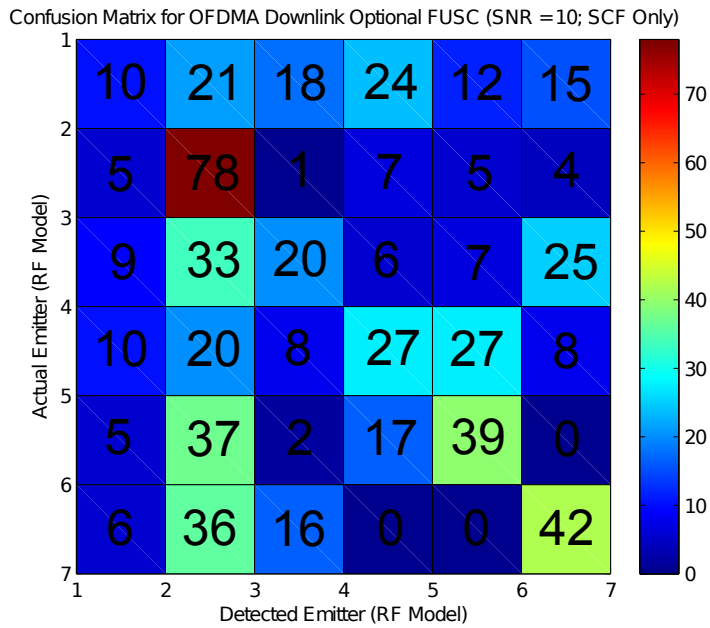


Figure 5.24: Confusion matrix for SEI classification test of OFDMA downlink optional FUSC with 10 dB and 0 dB SNR using SCM only. The SUI-3 multipath propagation channel model was applied to each snapshot prior to classification.

Chapter 6

Conclusions

Specific emitter identification is an important technology that has many applications including cognitive radios, femtocells and wireless security. By utilizing SEI, the potential exists to uniquely associate a class of wireless transmitters, if not a particular device itself, with an RF transmission. However, some drawbacks to SEI are that it requires a dataset of known emitters to be collected a priori, and it generally requires a high SNR. Additionally, SEI may only be capable of identifying a class of emitters, such as those being manufactured by a specific vendor, or a specific model manufactured by a vendor, as opposed to a unique wireless device.

In this thesis a technique for SEI of OFDM signals was presented which involves computing the cyclic spectrum in matrix form, decomposing the matrix into eigenvectors and eigenvalues, and extracting a feature vector from the dominant eigenvector that is associated with the OFDM pilots. A pattern classification test was performed using HMMs trained from feature vectors extracted from six simulated emitters for four OFDM configurations based on 802.16 specifications for downlink.

While the algorithm presented above was configured exclusively for SEI, the techniques employed may have applications in other areas. For example, the spectral masks applied to each OFDM snapshot in order to simulate a unique wireless device may be constructed in such a way as to embed a unique, non-destructive RF watermark or fingerprint in the OFDM signal. Another possible application is modulation classification of OFDM signals. Since the pilot eigenvectors contain information about the allocation of the OFDM pilots, this information could be used to successfully differentiate broader classes of OFDM modulation types using differing pilot allocations schemes.

6.1 Results

The main SEI simulation contained in this work was presented in Chapter 5. It should be noted that all OFDM signals used in this simulation were both generated and processed at baseband. Thus, all signals were processed as though perfect frequency and symbol synchronization were performed. However, this may not be a requirement for successful emitter identification. In circumstances where non-simulated data is used to perform an SEI test, it would first be necessary to perform demodulation of the the RF signal by converting it to baseband. This would entail first removing any frequency offsets and then recovering the symbol timing by utilizing the pilots inserted into the OFDM signals.

The results of the main SEI simulation are shown in Figures 5.16 through 5.23. In general, it can be seen across all results that extracting an FV from the SCM alone performs moderately better than extracting an FV from both the SCM and CSCM. One reason for this can be ascertained by examining the FV statistics plots in Figures 5.6 through 5.13. By inspection, the statistics of the FVs extracted from the CSCM appear less affected by the RF models applied to the OFDM snapshot, thus providing a slightly less statistically unique feature vector than the SCM. It can be concluded from this that use of the SCM alone for FV extraction will provide slightly better results than use of the SCM and CSCM in a pattern classification test. Additionally, as discussed in Chapter 5, use of the QPSK for the subcarriers provide lower overall spectral correlation for the CSCM than for the SCM.

It can be seen in Figures 5.16 and 5.17 that nearly perfect classification was achieved in the tests involving OFDM downlink. The worst performance comes in the 0 dB SNR test for the case of SCM and CSCM, wherein some ambiguities arise in classifying emitters 3 and 6. The tests with OFDMA downlink PUSC, seen in Figures 5.18 and 5.19, show good performance across all test, except for the ambiguity which arises between emitters 1 and 2. The case with OFDMA downlink FUSC, seen in Figures 5.20 and 5.21, reveals unreliable performance in all cases, due to lack of uniqueness in the feature vectors. Lastly, the best performance of the four OFDM types comes from the case of OFDMA downlink OFUSC, see Figures 5.22 and 5.23, where perfect classification was achieved in all four tests.

6.2 Advantages and Limitations

The main advantage of the eigenspace approach to SEI of OFDM signals is that the technique works surprisingly well at low SNR for certain pilot configurations. This can be readily seen in the results of the classification tests for OFDMA downlink OFUSC, where perfect classification was achieved over 100 trials, even at a relatively low SNR of 0 dB. This is due in part to the fact that the EVD is capable of dividing signal subspace from noise subspace, thus allowing for increasing noise levels to have a minimal affect on the ability to classify an emitter.

One limitation of this approach to SEI is that the pilots inserted in the OFDM signal must be correlated. This is due to the fact that the FVs rely on spectral correlation for cycle frequencies to appear in the spectral correlation matrix. This means that the pilots must be created from the same generating sequence, since they are created from the same PRBS generator. In the case where the pilots are uncorrelated, a version of this method may still be used if the power of each pilot is individually measured by translating and cross-correlating against a simulated version of the pilot sequence, or by performing cross-cyclic spectral analysis with a synthesized pilot-only signal. Additionally, each pilot in the OFDM signals used in the above simulations was roughly 2.5 dB higher than the subcarrier power level, and this may ultimately be a requirement for this particular algorithm to provide consistent results.

Computational speed limitations of the EVD for cases where the spectral correlation matrix was computed with $N' = 1024$ or larger prohibited simulations for any case other than $N' = 512$. For this reason simulated OFDM was limited to 1.25 MHz (128 subcarriers) for OFDMA and 2.5 MHz (256 subcarriers) OFDM in order to ensure that the channelizing FFT size N' is larger than the number of subcarriers. In order to process larger bandwidth OFDM signals, such as the 20 MHz bandwidth (2048 subcarriers) option in 802.16, it would be necessary to extract from the spectral correlation matrix only the eigenvectors associated with the largest eigenvalues. Thus, it may be necessary to find more efficient methods of extracting the dominant eigenvectors from larger matrices.

6.3 Future Work

For future work related to the topic of this thesis, it is suggested that the effects of integration time on the probability of correct classification be thoroughly explored. Since a relatively large number of OFDM symbols per snapshot was used in the simulations in Chapter 5, it is possible that snapshots consisting of only a small number of OFDM symbols may be significantly more difficult to classify correctly. This would be especially pertinent in real world scenarios where OFDM bursts may only contain a small number of symbols. Also, it is suggested that BPSK be used as the subcarrier when testing feature vectors extracted from the CSCM, due to the lack of spectral correlation features when QPSK is used.

Additionally, it is suggested that the effect of SNR, especially in the presence of varying degrees of multipath, on the probability of correct classification be explored. The environment of the typical application for SEI has the potential to be a severe multipath environment. An SEI algorithm which does not perform well under these conditions would only be marginally useful. Thus it is recommended that new processing techniques are developed for the case when multipath is severe enough to prevent reliable performance of this SEI algorithm.

The aforementioned harsh environment, especially in the case of the femtocell application, may also be rife with co-channel interference. In future work, it is suggested that the SEI

algorithm presented in this thesis be tested in the presence of co-channel interference. This could imply either another form of OFDM modulated signal (802.16 or otherwise), or some non-OFDM signal, likely narrowband relative to the OFDM signals. However, it is unlikely that this algorithm will work for co-channel interference from an OFDM signal with identical pilot placement, unless some frequency offset is present.

One alteration to the SEI algorithm that may allow for the processing of signals in the presence of multipath is the inclusion of some multi-antenna processing. This may be as simple as direct beamforming for the case when the multipath is due to an LOS component and a low number of NLOS components, or as complicated as a hybrid array processing/SEI algorithm. Inclusion of diversity combining algorithms, such as maximum ratio combining, may also be beneficial. Finally, a purely speculative approach may be to perform a hybrid equalization/SEI algorithm, one which simultaneously performs equalization and emitter identification over a large number of OFDM samples.

In above simulations, the absolute value of the dominant eigenvectors was computed prior to extracting a feature vector, thus any phase information associated with the eigenvectors was not included. It is possible that the phase of the pilot eigenvectors, especially those at indices which correspond to pilot frequencies, may provide an additional dimension of identifiability to the feature vectors. This would apply more so in the case where any filters in the RF chain of the emitter have a non-uniform group delay. In addition to ignoring phase information, the above simulations also did not take into account spectral coherence, which is related to spectral correlation. For future work it is suggested that both phase and spectral coherence information be included in the feature vectors used in the above simulations.

Finally, due to the fact that all OFDM snapshots examined in this thesis were simulated, it is recommended that future work examine the approach to SEI presented in this thesis for real world, non-simulated signals. This would require a data capture system in a lab environment and several 802.16 devices from an array of manufacturers. Since emitters conform to the uplink portion of the specification, some modification of the algorithms as outlined in this thesis would be required. This is due in part to the fact that in the case of OFDMA, multiple users maybe present on the various subcarriers of each OFDMA symbol. Thus, some filtering would be required prior to performing any analysis.

Bibliography

- [1] K. Kim, C. M. Spooner, I. Akbar, and J. H. Reed. Specific emitter identification for cognitive radio with application to IEEE 802.11. In *Global Telecommunications Conference*, New Orleans, LA, December 2008.
- [2] Kyou Woong Kim. *Exploiting Cyclostationarity for Radio Environmental Awareness in Cognitive Radios*. PhD thesis, Virginia Polytechnic Institute and State University, Blacksburg, VA, May 2008.
- [3] Chad M. Spooner. Specific emitter identification for modern communication signals. *NWRA Technical Data*, June 2008.
- [4] Jeffrey H. Reed, Juan E. Surs, Timothy Newman, and Lizdabel Morales. Using cognition in femtocell applications. White Paper, Wireless@VT, Virginia Polytechnic Institute and State University, Summer 2009.
- [5] Thomas Charles Clancy. *Dynamic Spectrum Access in Cognitive Radio Networks*. PhD thesis, University of Maryland, 2006.
- [6] A. Fehske, J. Gaeddert, and J. H. Reed. A new approach to signal classification using spectral correlation and neural networks. *New Frontiers in Dynamic Spectrum Access Networks, 2nd IEEE International Symposium on*, November 2005.
- [7] Timothy R. Newman and T. Charles Clancy. Security threats to cognitive radio signal classifiers. *Wireless @ Virginia Tech Symposium*, June 2009.
- [8] Nate Goergen, W. Sabrina Lin, K. J. Ray Liu, and T. Charles Clancy. Channel-like fingerprinting overlays for authenticating OFDM signals using channel side information. *Asia Pacific Signal and Information Processing Association*, December 2010.
- [9] L. E. Langley. Specific emitter identification (SEI) and classical parameter fusion technology. *WESCON/93. Conference Record*, September 1993.
- [10] Kenneth I. Talbot, Paul R. Duley, and Martin H. Hyatt. Specific emitter identification and verification. *Northrop Grumman Mission Systems, Technology Review Journal*, Spring/Summer 2003.

- [11] Abdulla Mufarrah Abdulla. Identification of push-to-talk transmitters using wavelets and spectral correlation. Master's thesis, Naval Postgraduate School, September 1996.
- [12] K.A. Remley, C.A. Grosvenor, R.T. Johnk, D.R. Novotny, P.D. Hale, M.D. McKinley, A. Karygiannis, and E. Antonakakis. Electromagnetic signatures of wlan cards and network security. *Signal Processing and Information Technology, 2005. Proceedings of the Fifth IEEE International Symposium on*, pages 484 – 488, 21 December 2005.
- [13] William A. Gardner, Antonio Napolitano, and Luigi Paura. Cyclostationarity: half a century of research. *Signal Process.*, 86(4):639–697, 2006.
- [14] W. T. Bennett. Statistics of regenerative digital transmission. *Bell Systems Technical Journal*, 37:1501–1542, 1958.
- [15] L. E. Franks. *Signal Theory*. Prentice Hall, Englewood Hall, NJ, 1969.
- [16] William A. Gardner, editor. *Cyclostationarity in Communications and Signal Processing*. IEEE Press, Piscataway, NJ, 1994.
- [17] Kyouwoong Kim, Ihsan A. Akbar, Kyung K. Bae, Jung-sun Um, Chad M. Spooner, and Jeffrey H. Reed. Cyclostationary approaches to signal detection and classification in cognitive radio. *New Frontiers in Dynamic Spectrum Access Networks, 2nd IEEE International Symposium on*, April 2007.
- [18] William A. Gardner. An introduction to cyclostationary signals. In William A. Gardner, editor, *Cyclostationarity in Communications and Signal Processing*. IEEE Press, 1994.
- [19] William A. Gardner. *Representation and Estimation of Cyclostationary Processes*. PhD thesis, University of Massachusetts, 1972.
- [20] Jon Altuna. *Cyclostationary Blind Equalisation in Mobile Communications*. PhD thesis, The University of Edinburgh, 1997.
- [21] Chad M. Spooner. The frequency smoothing method of spectral correlation measurement. NWRA Technical Report, 2006.
- [22] Chad M. Spooner. The strip spectral correlation analyzer method of spectral correlation measurement. *NWRA Technical Data*, December 2006.
- [23] Randy S. Roberts, William A. Brown, and Herschel H. Loomis Jr. Computationally efficient algorithms for cyclic spectral analysis. *IEEE Signal Processing Magazine*, April 1991.
- [24] William A. Brown and Herschel H. Loomis Jr. Digital implementations of spectral correlation analyzers. *IEEE Transactions on Signal Processing*, 41(2):703–720, Feb. 1993.

- [25] Randy S. Roberts, William A. Brown, and Herschel H. Loomis Jr. A review of digital spectral correlation analysis: Theory and implementation. In William A. Gardner, editor, *Cyclostationarity in Communications and Signal Processing*. IEEE Press, 1994.
- [26] David A. Streight. Application of cyclostationary signal selectivity to the carry-on multi-platform gps assisted time difference of arrival system. Master's thesis, Naval Postgraduate School, March 1997.
- [27] Eric April. On the implementation of the strip spectral correlation algorithm for cyclic spectrum estimation. Technical Report 94-2, Defence Research Establishment Ottawa, February 1994.
- [28] Chad M. Spooner. Filterbanks for adaptive transmit filtering. *Military Communications Conference*, October 2007.
- [29] Lawrence M. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, February 1989.
- [30] Allan D. Kraus. *Matrices for Engineers*. Hemisphere Publishing Corporation, 1987.
- [31] Jeff Bilmes. What hmms can do. Technical Report UWEETR-2002-0003, Department of Electrical Engineering, University of Washington, January 2002.
- [32] C. J. Leggetter and P. C. Woodland. Speaker adaptation of hmms using linear regression. Technical Report CUED/F-INFENG/TR. 181, Cambridge University Engineering Department, June 1994.
- [33] Leonard E. Baum, Ted Petrie, George Soules, and Norman Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains. *The Annals of Mathematical Statistics*, 41(1):164–171, 1970.
- [34] Kevin Murphy. Hidden markov model (hmm) toolbox for matlab. <http://www.cs.ubc.ca/~murphyk/Software/HMM/hmm.html>. Last checked July 2011.
- [35] Zhendao Wang and G.B. Giannakis. Wireless multicarrier communications. *Signal Processing Magazine, IEEE*, 17(3):29 – 48, May 2000.
- [36] Leonard J. Cimini. Analysis and simulation of a digital mobile channel using orthogonal frequency division multiplexing. *Communications, IEEE Transactions on*, 33(7):665 – 675, July 1985.
- [37] Christoph Rapp. Effects of hpa-nonlinearity on a 4-dpsk/ofdm-signal for a digital sound broadcasting system. *Proceedings of the Second European Conference on Satellite Communications*, October 1991.
- [38] S. Sesia, I. Toufik, and M. Baker, editors. *LTE-The UMTS Long Term Evolution: From Theory to Practice*. Wiley, 2009.

- [39] IEEE. Ieee standard for local and metropolitan area networks, part 16: Air interface for broadband wireless access systems. *IEEE Std 802.16-2009*, May 2009.
- [40] Taiyo Yuden Co. LTD. F6ka-series wlan integrated circuit saw filter technical datasheets. www.yuden.co.jp, October 2011.
- [41] Raj Jain. Channel models: A tutorial. Class Notes, CSE547S - Wireless and Mobile Networking, Washington University in St. Louis, 2007.

Vita

Peter Sahmel attended Georgia Tech from 1999 through 2004, receiving a Bachelor of Science in Electrical Engineering. He has been an employee of Zeta Associates since 2004. His research interests include emitter identification, signal classification, geolocation, adaptive antenna array processing and interference mitigation. A sampling of his eclectic range of hobbies includes rock climbing, digital photography, homebrewing, and electronic music composition.