3-6-2007

Multi-dimensional Classification Algorithm for Automatic Modulation Recognition

Ouail Albairat

MULTI-DIMENSIONAL CLASSIFICATION ALGORITHM FOR AUTOMATIC MODULATION RECOGNITION

THESIS

Ouail Albairat, First Lieutenant, USAF

AFIT/GE/ENG/07-01

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the United States Government.

AFIT/GE/ENG/07-01

MULTI-DIMENSIONAL CLASSIFICATION ALGORITHM FOR AUTOMATIC MODULATION RECOGNITION

THESIS

Presented to the Faculty

Department of Electrical and Computer Engineering

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Science in Electrical Engineering

Ouail Albairat, B.S.E.E.

First Lieutenant, USAF

March 2007

AFIT/GE/ENG/07-01

MULTI-DIMENSIONAL CLASSIFICATION ALGORITHM FOR AUTOMATIC MODULATION RECOGNITION

Ouail Albairat, B.S.E.E.
First Lieutenant, USAF

Approved:

/signed/	6 Mar 2007
Dr. Richard K. Martin (Chair)	date
/signed/	6 Mar 2007
LtCol Stewart L. Devilbiss, PhD (Member)	date
/signed/	6 Mar 2007
Dr. Steven C. Gustafson (Member)	date

Abstract

Modulation recognition is the process to which an intercepted signal is subjected for the purpose of extracting a signature that reveals the modulation identity of the signal. This technology has both military and commercial applications, but most of the motivation for its development comes from the signal intelligence community. Indeed, techniques that improve capabilities for intercepting and exploiting communication signals automatically are of increasing importance due to the rapid growth in communication technologies.

This thesis proposes an approach for modulation classification using existing features in a more efficient way. The Multi-Dimensional Classification Algorithm (MDCA) treats features extracted from signals of interest as elements with irrelevant identities, hence eliminating any dependence of the classifier on any particular feature. This design enables the use of any number of features, and the MDCA algorithm provides the capability to classify modulations in higher dimensions. The use of multiple features requires an equal number of data dimensions, and thus classification in as high a dimensional space as possible can improve final classification results. Finally, the MDCA algorithm uses a relatively small number of simple operations, which leads to a fast processing time.

Simulation results for the MDCA algorithm demonstrate good potential. In particular, the MDCA consistently performed well (at SNR levels down to -10dB in some cases) and in identifying more modulation types.

Acknowledgements

I am very grateful to my parents for their support and encouragements. To Dr. Martin and Dr. Gustafson, thank you both for your help and guidance.

Ouail Albairat

Table of Contents

Page

Page

List of Figures

List of Tables

List of Abbreviations

MULTI-DIMENSIONAL CLASSIFICATION ALGORITHM

FOR AUTOMATIC MODULATION RECOGNITION

I. Introduction

1.1 Background

Modulation recognition is any signal processing or statistical analysis technique that allows correct identification of the modulation type used in a communication signal. Such capability is desired by both military and commercial communities. Indeed, the efficient use of available frequency bandwidth is increasingly needed due to the ever tightening spectrum resources. To achieve high levels of efficiency, however, precise characterization of the channels and detailed information about signal presence and its modulation type are paramount. Thus, Automatic Modulation Recognition (AMR) allows the electronic warfare (EW) and signal intelligence (SIGINT) communities to better monitor the frequency spectrum environment. Several approaches to produce a fully automated modulation recognition classifier have been devised, and most are either neural network based or rely on some form of branching tree algorithms. Despite the variety of design approaches already tried, none have resulted in a robust engine that produces good classification results for low signal-to noise ratios (SNR).

1.2 Problem Statement

Pattern recognition problems are best tackled by extracting good features from the available variables. In modulation recognition the variables are the intercepted signals, and the features are any statistical or spectral measurements made on the signals. These features are then processed, based on their observed behavior for known modulations, to estimate the modulation type. Unfortunately, no set of features proposed in the AMR related literature has been shown to adequately characterize

all sets of modulation schemes. Furthermore, all the proposed AMR algorithms are feature dependent, in that their processing workflow is designed for a specific set of features. Devising new features is largely a product of time consuming trial and error, and hence creating a new algorithm for existing features is not an optimal approach once newer and better features are discovered.

All the existing AMR algorithms use multiple features, but their processing approach uses one feature at a time in a branching tree pattern. Such a one dimensional design usually tends to fail whenever there is an overlap of measured features from different modulations. This tendency increases as the signal to noise ratio decreases. The goal of this research is to produce a feature independent and multi-dimensional AMR algorithm.

1.3 Research Assumptions

This research makes the following assumptions:

1. The channel noise is modeled as additive white Gaussian noise (AWGN).
2. The AWGN power is adjusted to achieve the desired SNR level.
3. Only one signal is present in the radio frequency environment, unless stated otherwise.
4. No multi-path reflections exist in the channel, i.e., direct line-of-sight path between transmitter and receiver is assumed.
5. The SNR level is determined at the receiver input

1.4 Research Scope

This research proposes an algorithm that works with any set of two or more features. For a demonstration of the validity of the approach, the five features proposed by Azzouz and Nandi in their Digitally Modulated Signal Recognition Algorithm (DMRA) [1] are considered for the classification process. Eight digital modulation schemes are used to measure the performance of the proposed algorithm:

- Binary Phase Shift Keying (BPSK)
- Quadrature Phase Shift Keying (QPSK)
- Binary Amplitude Shift Keying (ASK2)
- 4-ary Amplitude Shift Keying (ASK4)
- Binary Frequency Shift Keying (FSK2)
- 4-ary Frequency Shift Keying (FSK4)
- Quadrature Amplitude Modulation, square 16 (QAM16)
- Quadrature Amplitude Modulation, square 64 (QAM64)

1.5 Research Approach

This research is a follow-on to previous work conducted at AFIT and AFRL/SR. Since a comprehensive evaluation of three existing commercial MRA algorithms [2] and a modification of the DMRA algorithm to extend its range of classification [21] were two major efforts performed at AFIT, the next logical next step is a new algorithm. A difficulty was in determining if a totally new approach using signal processing techniques never before tried in MRA, such as the Mellin transform as used in speech recognition, would be appropriate or if some statistical pattern recognition tools would be a better option. A decision was made to develop a statistical pattern recognizer using the proven Fisher discriminant and Bayesian decision making concepts. With an open architecture the proposed algorithm can use any features, statistical or signal processing based, that produce a real measurement or even a statistical distribution. Hence the research is focused on developing an intelligent way to maximize positive classification by using AMR features already proposed in literature while remaining flexible to incorporate any features that might be introduced in the future.

1.6 Materials and Equipment

All the simulated work was performed using `Matlab`® Version 7.0 with the Signal Processing and Communication Toolboxes. The `Matlab`® simulations were run on a

home PC with a 3.0GHz Pentium D platform using either Windows XP or Centos 4.3 operating systems.

1.7 Thesis Organization

Chapter II provides a brief review of background information and previous MRA approaches. The architecture of the proposed algorithm is described in Chapter III along with its basic mathematical concepts. It also covers features previously introduced by Azzouz and Nandi, since they constitute the main input to the proposed algorithm. Chapter IV presents the design for the simulated testing of the algorithm as well as the investigated modulations used in most of the simulations. Finally, the results of the simulations are covered in Chaper V, where they are analyzed and compared to the performance of the DMRA algorithm.

II. Background

Modulation recognition is but one step that comes after detection and isolation of a communication signal. In military applications AMR allows the exploitation of uncooperative signals for which no or little prior knowledge is available. Ideally, the AMR algorithm continuously tracks any changes in the modulation used by the signal of interest. Commercially, one product concept that can take advantage of an AMR capability is the software defined radio. Such a concept would not have the traditional communication overhead that comes with scheduling and handshaking, but only if the radio can identify the parameters of the signal of interest correctly. Multiple algorithms with different processing approaches have been proposed to achieve a modulation recognition capability. So far the goal of a robust algorithm or a package of algorithms offering the desired capability is still elusive.

Over almost three decades dozens of papers on the subject of the recognition of digitally modulated communication signals have been published. Numerous approaches have been tried to develop fast and accurate algorithms for producing acceptable results in real world applications. However, an algorithm robust enough for real world cases has not been demonstrated so far. Some of the published algorithms are presented in this chapter to provide a historical overview, and briefly describe the different approaches so far attempted.

2.1 The Liedtke Algorithm

As one of the first to address the modulation recognition problem, Liedtke [14] focuses his effort on the classification of ASK2, FSK2, the Continuous Wave (CW), and the Phase-Shift Keying modulations PSK2, PSK4, and PSK8. Liedtke's approach assumes an approximate knowledge of the center frequency and symbol rate of the communication signal and also that sampling rate is an integer multiple of the symbol rate. First, the bandwidth of the unknown signal is approximated using a bank of Finite Impulse Response (FIR) filters. The signal is processed in parallel through all the filters which have different bandwidths at the same center frequency. Then

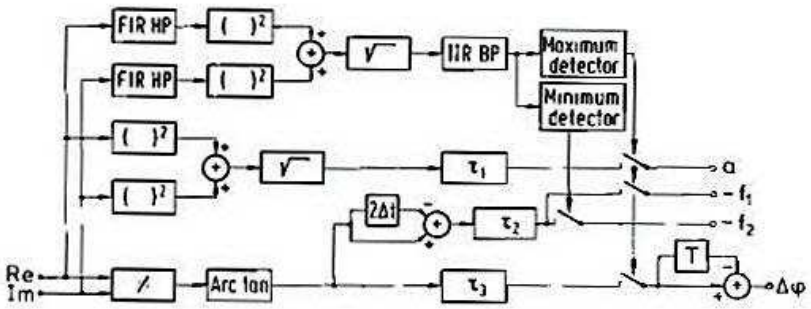


Figure 2.1: Diagram of Liedtke's universal demodulator [14] with timing recovery and feature extraction circuit.

the Universal Demodulator, a feature extractor, removes the carrier component of the signal, thus converting it to near base-band, and extracts the amplitude, instantaneous phase, and frequency features. Liedtke's algorithm uses a branching tree classifier that depends on five main features extracted from the signal:

- Amplitude variance of the extracted waveform
- Frequency variance of the extracted waveform
- Delta-phase histograms: the histogram of the phase differences between adjacent symbols
- The amplitude histogram
- The instantaneous frequency histogram

Figure 2.1 shows a diagram of the universal demodulator circuit, which processes the unknown signal as complex and automatically adjusts to the estimated bandwidth of the signal to produce a timing recovery vector for synchronization purposes, thus extracting the features. The complete classification flow is shown is Figure 2.2.

2.2 The Kim and Polydoros Algorithm

Kim and Polydoros [13] focus primarily on phase-based modulations such as BPSK and QPSK and later even extended their approach to quadrature-amplitude modulated signals QAM. Their method falls under the decision-theoretic approach,

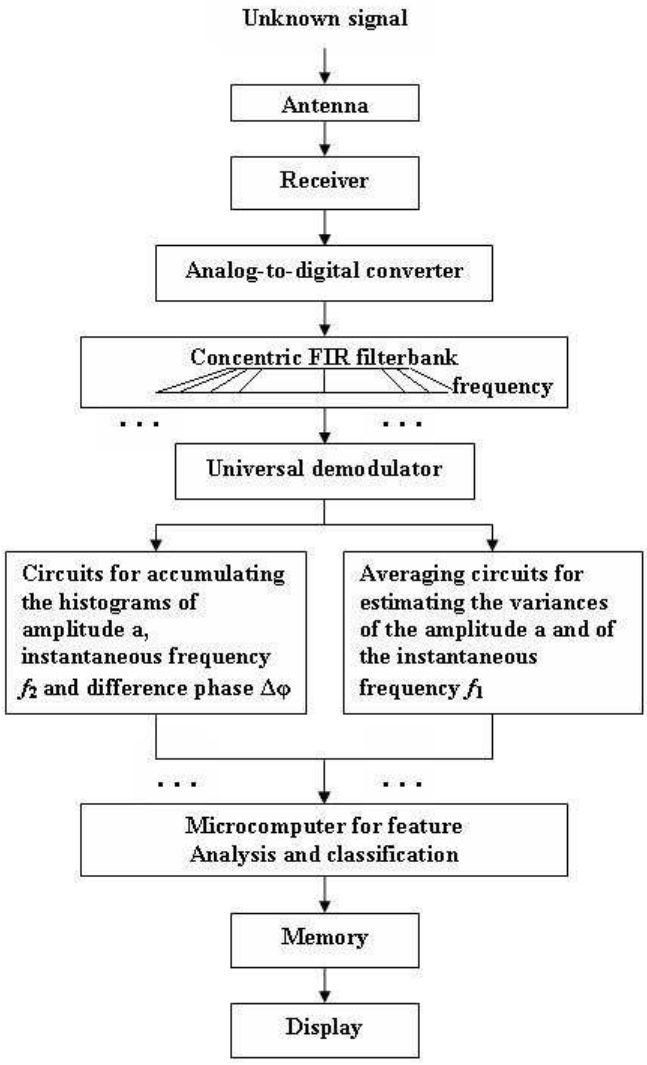


Figure 2.2: Flow Chart of Liedtke's classification approach [14].

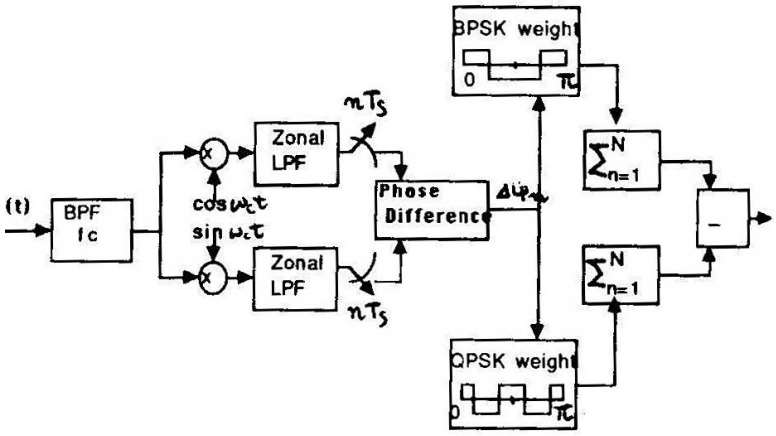


Fig.1 A phase-based BPSK/QPSK classifier structure

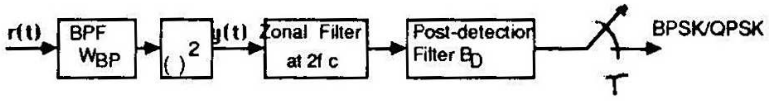


Fig.2 The square-law classifier structure

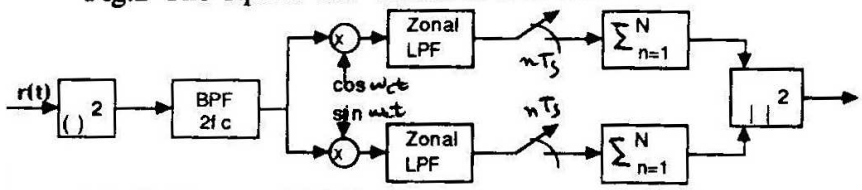


Fig.3 The quasi-LLR classifier structure

Figure 2.3: Flow Chart of the Kim and Polydoros algorithm [13].

which differs from pattern recognition schemes in that the classification rule is based on a hypothesis (a discriminating test of a statistical description of the signal). Kim and Polydoros craft three classifiers that use the complex envelop of the baseband signals.

- Phase-based classifier: This classifier uses the histogram of phase differences between adjacent symbols. In a high SNR level setup it is expected that the number of lobes in the histogram equals the number of phases in the modulation. Hence, the histogram generated from a BPSK has two lobes, and the histogram of an MPSK has M lobes.

- Square-law classifier: A special case of the Power-Law technique introduced by DeSimio and Prescott [7], the square-law property ($2\cos^2 x = \cos 2x + 1$), it generates a squared output signal with half the number of phases of the original and can be used to isolate BPSK, as its square loses the phase signature. This classifier is primarily used to differentiate between BPSK and QPSK modulated signals.
- Quasi Log-Likelihood Ratio classifier: This classifier uses a Maximum Likelihood (ML) ratio test developed from likelihood functionals of M-ary PSKs in an AWGN environment.

2.3 The DeSimio and Prescott Algorithm

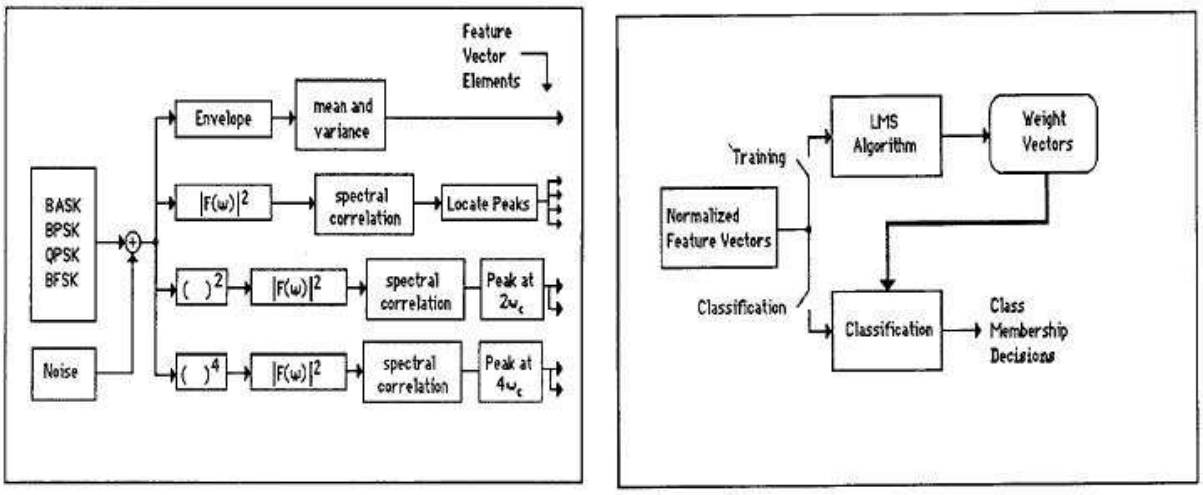
Prescott and DeSimio [7] introduce an interesting approach which consists of multiplying feature vectors of the signal to be classified by predetermined weight vectors of the modulation types the algorithm is capable of handling. The features selected by DeSimio and Prescott are

- The mean and variance of the signal envelop
- The magnitude and spectral location of the two largest peaks in the spectrum of the signal, which also yields information about the carrier frequency
- The spectrum magnitude of the squared and quadrupled signal at twice and four times the carrier frequency

Using these extracted features in an Adaptive Linear Combiner (ALC) processor, which is a Least Mean Square (LMS) based algorithm, a weighted vector descriptive of each modulation is generated. Classification is determined by the weight vector producing the largest product with the feature vector.

2.4 The Hsue and Soliman Algorithm

Hsue and Soliman [11] use the concept of zero-crossing [12] sampling to build a modulation classifier for BPSK, QPSK, PSK8, FSK2, FSK4, and FSK8. First



(a) (b)

Figure 2.4: DeSimio and Prescott's algorithm [7] (a)The feature extraction process (b) Top level diagram of the classification process.

demonstrated in modulation recognition by T.G. Callaghan, *et al.* [6] a zero-crossing sampler records the time when the input signal crosses the zero volt level. The amount of information about carrier frequency and phase transitions that a sequence of zero-crossing points holds is substantial. But the zero-crossing method is sensitive to noise. In an approach identical to Liedtke's, Hsue and Soliman classify the received signal by first separating FSK from single-tone (CW or MPSK), then making a final decision using histograms of phase variations. Figure 2.5 shows a flow diagram of the algorithm proposed by Hsue and Soliman.

2.5 The Azzouz and Nandi Algorithm

Azzouz and Nandi [1] introduce a feature based decision tree algorithm to classify ASK2, ASK4, FSK2, FSK4, BPSK and QPSK digitally modulated signals. The features used in the algorithm are based on the standard deviations of instantaneous phase, frequency, and amplitude of the modulated signal along with the power spectral density amplitude. Azzouz and Nandi's approach is a textbook example of feature-based classifiers. A detailed description of the features they propose is reviewed in Section 3.4.

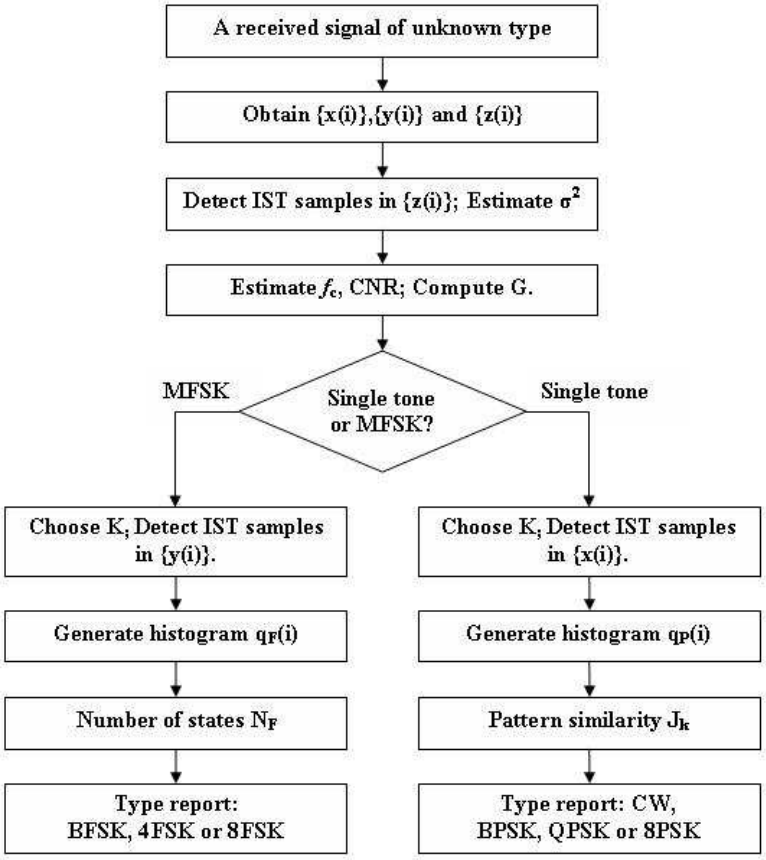


Figure 2.5: Flow Chart of Hsue and Soliman algorithm.

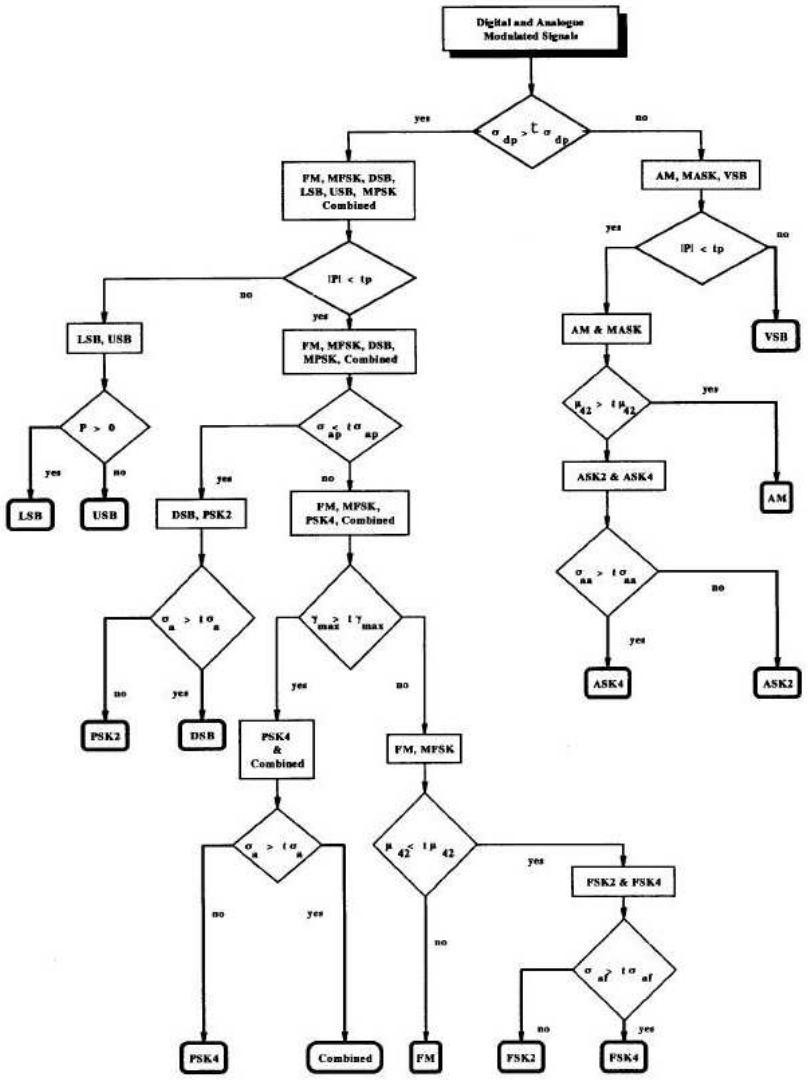


Figure 2.6: Flow Chart of Azzouz and Nandy algorithm.

One of the weak points of the algorithm is that the nature of the decision-tree, as shown in Figure 2.6 requires fixed threshold values, and since the features proposed by the authors are SNR sensitive, the threshold values can only be valid for small ranges of SNR. The architecture of the algorithm as shown in Figure 2.6, which is designed for a SNR larger than 10dB, may not be appropriate for a smaller SNR even if the threshold values are updated.

Nandi and Wong [22] proposed a Multi-Layer Perceptron (MLP) based classifier using the same features introduced by Azzouz and Nandi along with statistical

based features. The use of an Artificial Neural Network (ANN) platform proved very successful. Indeed, from the published results [22] reported for convenience in Table 2.1, the new classifier has impressive performance. The range of SNR levels where classification is possible is much wider for an increased number of modulation types (10 instead of 6). The MLP classifier used by Wong and Nandi is a two-layer, unidirectional feed-forward network that is fed normalized features.

Table 2.1: Performance of Wong and Nandi's ANN Classifier at different SNR values [22]

Performance (%)	Signal to Noise Ratio				
	-5dB	0dB	5dB	10dB	20dB
Training	90.4	98.1	99.3	99.9	100
Validation	89.4	98.0	99.3	99.8	100
Testing	89.4	97.9	99.2	99.9	100
Overall	89.7	98.0	99.3	99.9	100

2.6 The CRC Algorithm

Boudreau, *et al.* [5, 8] from the Communication Research Center (CRC) extend the algorithm proposed by Azzouz and Nandi while using the same decision-tree approach with features that are statistics of the instantaneous phase, frequency and amplitude. However, for added accuracy and additional numbers of identifiable modulations, they incorporated power-law tests in their classifier. Figure 2.7 clearly show the steps, where the number of peaks in the PSD of the raised signal to the powers two or four are used in the decision process.

2.7 The CuHBC Algorithm

The Custom Higher-Order-Cyclostationarity-Based Classifier (CuHBC) [20] is a commercial off the shelf (COTS) product. The software package is a hierarchical classifier composed of several smaller algorithms each designed to classify a limited set of modulations. Working in a sequential fashion, the software uses various proprietary classification techniques, but it seems to extensively use spectral line classification [16].

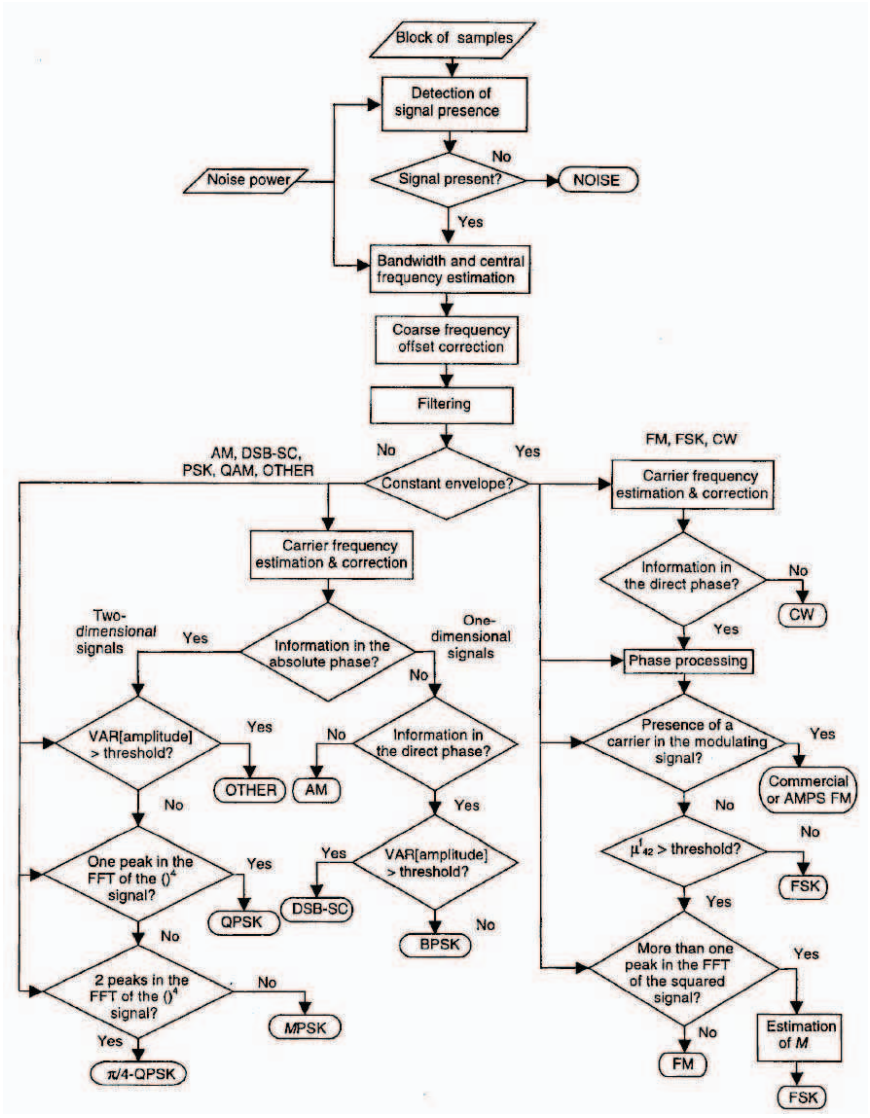


Figure 2.7: Flow Chart of the CRC algorithm [5].

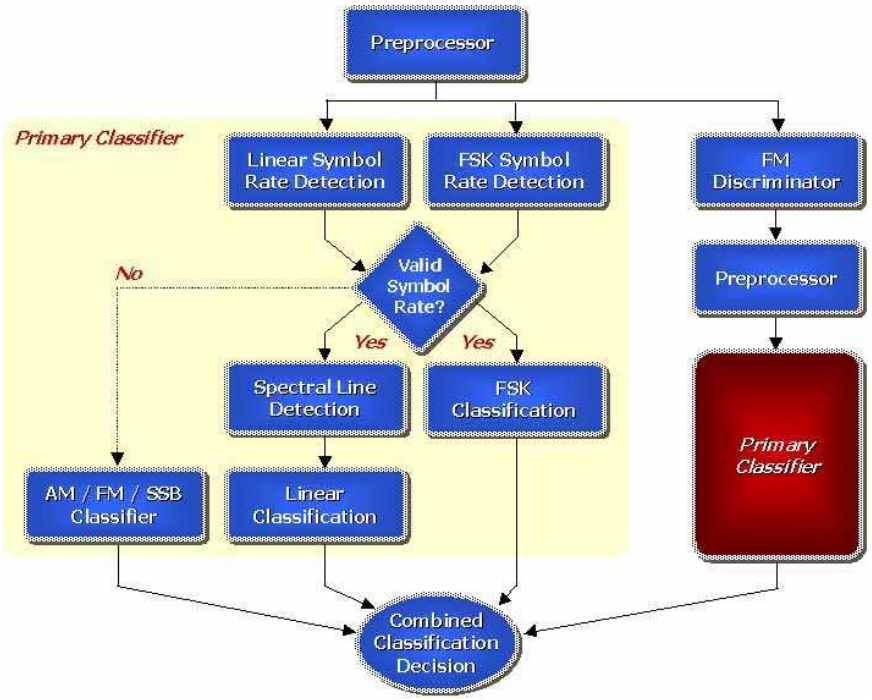
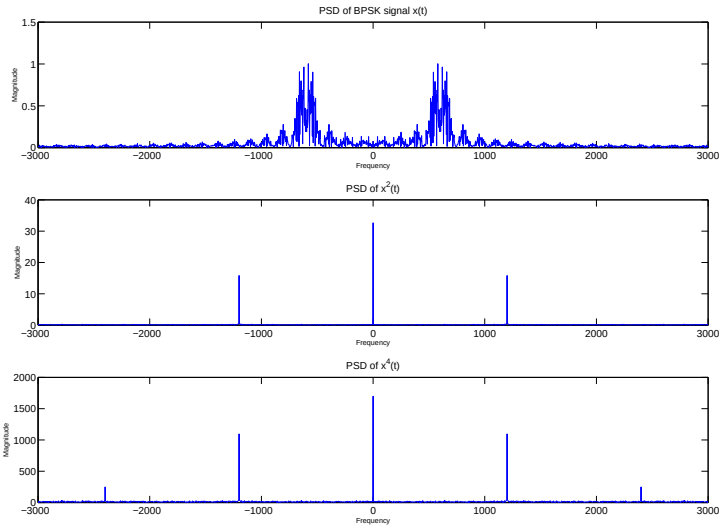


Figure 2.8: Flow Chart of the CuHBC algorithm.

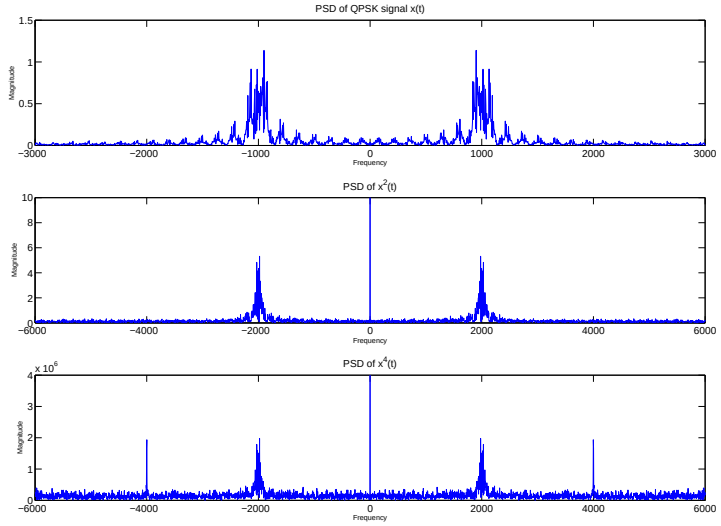
This method, which is a power-law technique, takes advantage of the fact that phase modulated signals produce spectral lines at even integer multiples of their carrier frequency depending on the number of symbols (number of phases) when the signals are raised to even powers. For example, a BPSK produces a spectral line at twice its carrier frequency when squared and a line at four times the carrier frequency when raised to the power of four. However, a QPSK signal will not produce a line until it is quadrupled. Figure 2.9 shows examples of the PSD of a BPSK signal and a QPSK signal.

2.8 The MSSA Algorithm

The Multiple-Signal Scene Analyzer(MSSA) [19] algorithm is also a commercial product developed to detect and classify LPI radar waveforms. It is more appropriate to describe the MSSA as software package that integrates multiple algorithms for the purpose of the signal detection and isolation of the band of interest (BOI) and classification of signal modulation type. The MSSA algorithm extensively uses the



(a)



(b)

Figure 2.9: PSD of a BPSK ($f_c = 600Hz$) and a QPSK ($f_c = 1000Hz$) signal, along with their squared and quadrupled versions (a) the PSD of the squared BPSK signal produces lines at twice the carrier frequency ($f = 1200Hz$.) (b) QPSK does not produce any lines until it is quadrupled, and does so at four times the carrier frequency ($f = 4000Hz$).

cumulant [17] property of cyclostationary signals, or simply the Fourier properties of sine wave when raised to the n^{th} order. Figure 2.9 shows examples of that property at $n = 2$ and $n = 4$. The MSSA algorithm uses a higher power of $n = 6$ [18] in the pursuit of a better signal classifier. Figure 2.10 shows a top level diagram of the MSSA.

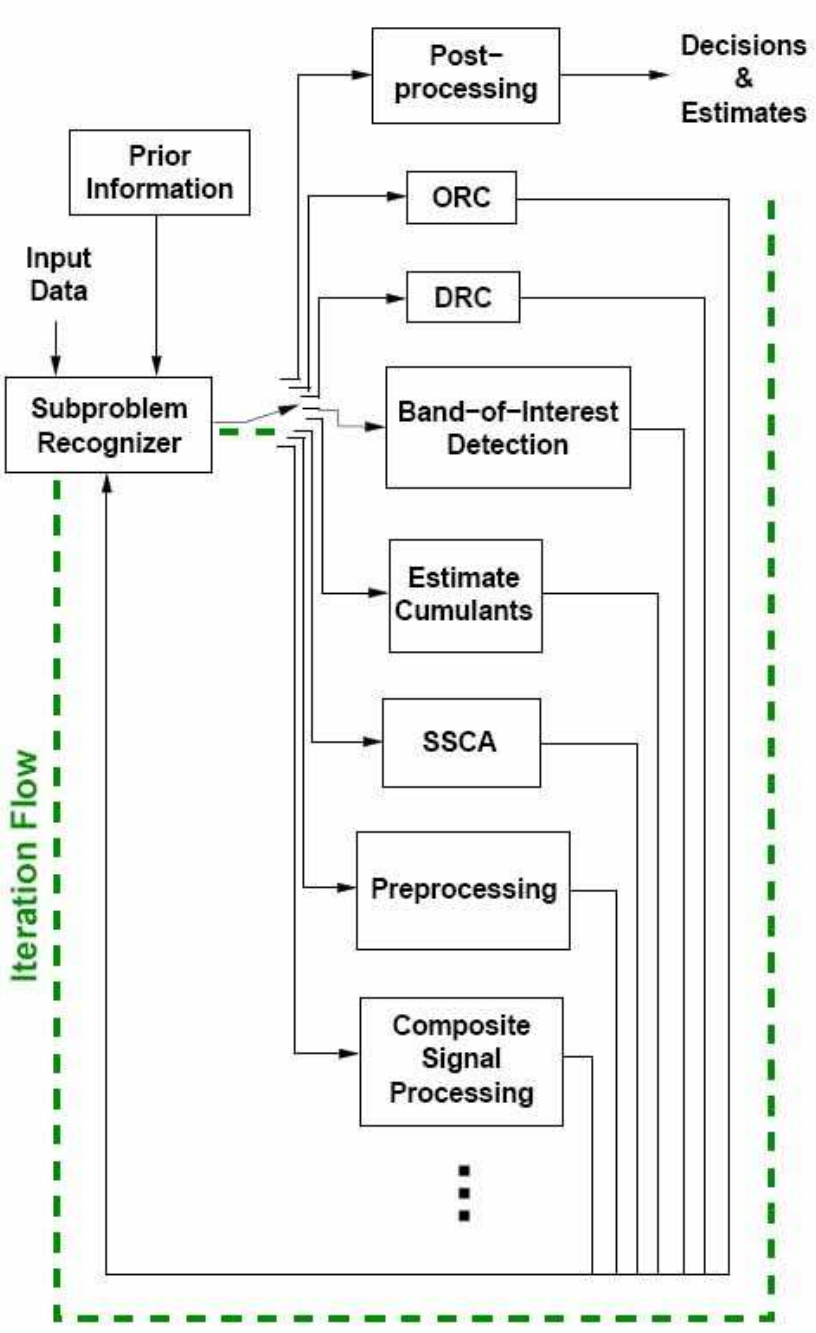


Figure 2.10: Flow Chart of the MSSA algorithm.

III. The MDCA Algorithm

3.1 Introduction

Very few classification techniques can outperform simple linear classification when good features are used. For this reason, one of the best methods to improve the performance of Automatic Modulation Recognition AMR schemes is identifying features that possess distinct signatures for the different modulations. Furthermore, classification can be improved when it is performed in the space created by the considered features. Since producing new features is challenging and also beyond the scope of this thesis work, the focus here is on the complementary approach of optimizing the computational flow between the features extraction step and the classification decision point.

The Multi-Dimensional Classification Algorithm (MDCA), the proposed algorithm for this thesis, is an open architecture approach that uses multi-dimensional classification for the AMR problem, and it is inherently independent of the features selected to perform classification. The algorithm can use whatever features selected by the user, and newer ones can later be added whenever they are available. Since the processing flow is sequential for implementation (coding) purposes but parallel in logic with an intrinsic back flow, there are no hard points that are feature dependent and which steer the process of classification in one or another direction. Rather, the parallelism of the algorithm produces a set of results that are objectively combined for a final classification. In sharp contrast, the rigid one-dimensional branching tree models presented by some researchers [1, 14, 20] are sequential decision points of feature-dependent thresholds that are in specific order. Any alteration of the features used requires an overhaul of these algorithms to cater to the new features.

3.2 Algorithm Core

Starting with a signal that is preprocessed, i.e., selected features are extracted and its signal to noise ratio (SNR) is approximated, the MDCA algorithm uses the signal features as coordinates of a point $\mathbf{x}$ in d-dimensional space (d = number of

features) and sequentially performs three simple operations: projection, Mahalanobis distance measure, and classification, as shown in Figure 3.1.

3.2.1 Projection. The d-coordinates of the signal of interest, where the values along each dimension correspond to the values of the d features, are projected onto one of the Fisher hyper-planes which are stored in the training database (the training database is covered in detail in Chapter 4). The projection yields a new point $\mathbf{y}$ with coordinates in either 2D or 3D space.

Projection is needed since the classification process is performed in a 2D or 3D space, which is generally of lower dimension than the number of features. Five features ($d = 5$), introduced by Azzouz and Nandi [1], which are discussed in Chapter 3.4, are used throughout this thesis. Using these features stems from the fact that they are used in Azzouz's one-dimensional algorithm [1], which has a limited classification capability in the SNR range, and hence a comparison is justified. Limiting classification to 2D and 3D spaces only is motivated by the fact that higher dimensions cannot be represented graphically.

3.2.2 Distance measurement. Once a projection onto a hyper-plane is performed, the Mahalanobis distances [3, 9] between the projection point and the means of all the modulations represented in that hyper-plane are measured. The approach for calculating the Mahalanobis distance is detailed in Section 3.3.4.

3.2.3 Classification and Feedback. The magnitudes of the Mahalanobis distances, measured in a hyper-plane, are ranked in ascending order. The ranking of magnitudes produces a ranking of modulation likelihoods from the most likely to the least likely. Since projections are first started in a hyper-plane that contains all the possible modulations, the distributions of these modulations are tight and hence the margins of errors are large, especially when the SNR level is low. Therefore, the accuracy of classification which depends on Mahalanobis distance in a crowded hyper-space is questionable. To minimize this uncertainty, the process is repeated from the

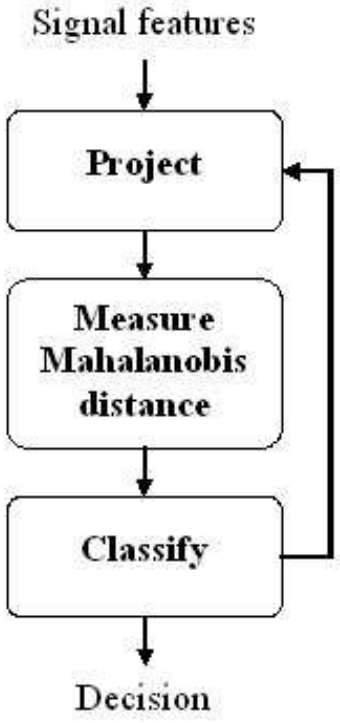


Figure 3.1: Block diagram of the main processes used in the MDCA algorithm. First, the preprocessed data in the form of extracted features from the signal of interest are projected onto predetermined Fisher hyper-planes. These planes are part of the database generated in the training phase. The database is composed of the planes representing every combination of possible modulations and SNR levels between -10dB and 20dB. Then the Mahalanobis distances [9] between the projection point and the means of the modulations that define the considered hyper-plane are measured. Finally, the Mahalanobis distances and their corresponding probabilities are stored and used to classify the signal in the Fisher hyper-space. Depending on the number of iterations selected by the user, the process is either stopped for a final decision or repeated for further focus on specific modulations, which is achieved using hyper-planes with fewer modulations.

projection step, but with the projections performed on hyper-planes that have fewer modulations while taking into consideration the ranking generated by the previous projection(s). The ranking from the first projection dictates the modulations on which attention is focused as the process proceeds to a second iteration. For example, only the three most likely modulations from the first iteration are scrutinized further, which is achieved by performing the second iteration of projections onto hyper-planes defined by a smaller group of modulations (3 or 4 as opposed to the hyper-planes defined by 8 modulations from the first iteration) with the condition that each group contain at least one of the three modulations identified in the first projection. If there is a need for more precision to distinguish between a number of modulations, the process can be repeated and the projection performed onto hyper-planes with fewer modulations. A classification example, with projection figures from each iteration, is given in Chapter 4.3.2.

3.3 Multi-Dimensionality and MDCA Underlying Mathematical Concepts

The multi-dimensionality of the MDCA algorithm stems from the number of features used in the classification process. Unlike other AMR algorithms, where each extracted feature is used independently at each decision point [1, 7, 11], MDCA considers all the features at every decision point. There is only one classification decision point at the very end of the process, and all other decision turns are soft in nature and carry only a partial weight in the overall decision making process. Furthermore, considering a multidimensional space that contains all the information about the signal of interest increases the possibility of correct classification. Traditional algorithms allow for a reduction in dimensionality, however, such an advantage comes at the cost of discarding sections of the main space in a perpendicular fashion along each dimension. The slicing of a space in such a fashion creates a multidimensional section of the space with rectangular sides. Such a geometrical shape can, sometimes, box in the distributions the of signal class, but can never tightly wrap its hyperquadric contours.

For the purpose of demonstrating the functioning of the MDCA algorithm, only classification in 2D and 3D spaces are presented in this thesis. The results of the mathematical operations in such an environment can be displayed graphically. The reduction in the dimensions of the problem are achieved through the use of the Fisher Linear Discriminant (FLD) [10]. However, due to the wide variation in input magnitudes of the features extracted from the signals of interest, the input data sets are normalized to zero mean and unit variance.

3.3.1 Normalization. Before performing any processing of the features extracted from the input signals, they are normalized to zero mean and unit variance, which is achieved by finding the means of all the points independently on each dimension, where every dimension corresponds to one of the i extracted features:

$$\mu_i = \frac{1}{N} \sum_{t=1}^{N} \mathbf{x}_{t,i}, \tag{3.1}$$

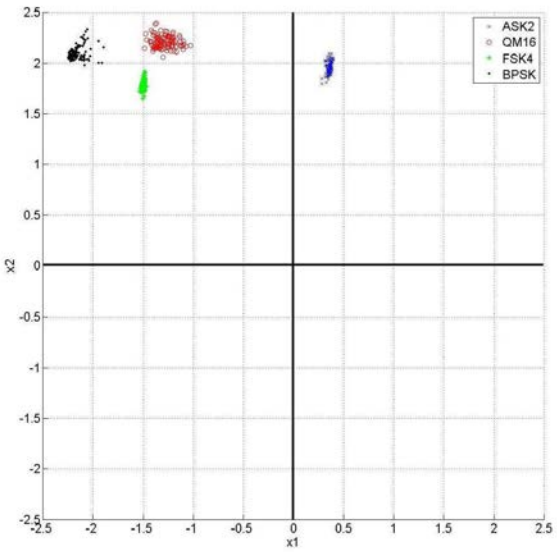
where, N denotes the total number of input signals. The new normalized coordinates are

$$\hat{\mathbf{x}}_{t,i} = \frac{\mathbf{x}_{t,i} - \mu_i}{\sigma_i}, \tag{3.2}$$

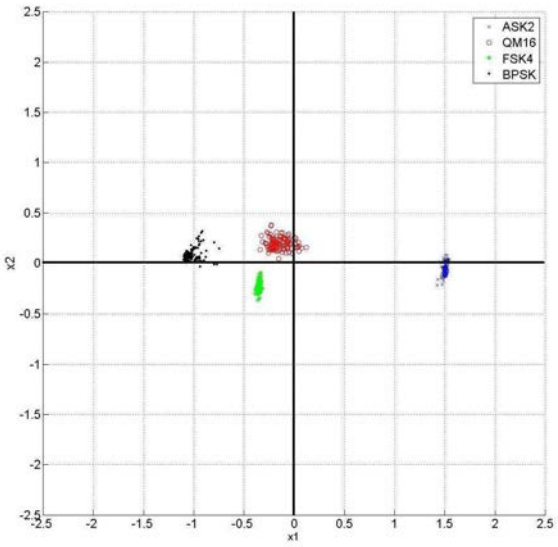
where $\mathbf{x}_{t,i}$ is the t^{th} component of one of the original extracted feature vectors, and $\hat{\mathbf{x}}_{t,i}$ is its normalized version. The standard deviation σi is defined

$$\sigma_i = \sqrt{\frac{1}{N} \sum_{t=1}^{N} \left(\mathbf{x}_{t,i} - \mu_i\right)^2}. \tag{3.3}$$

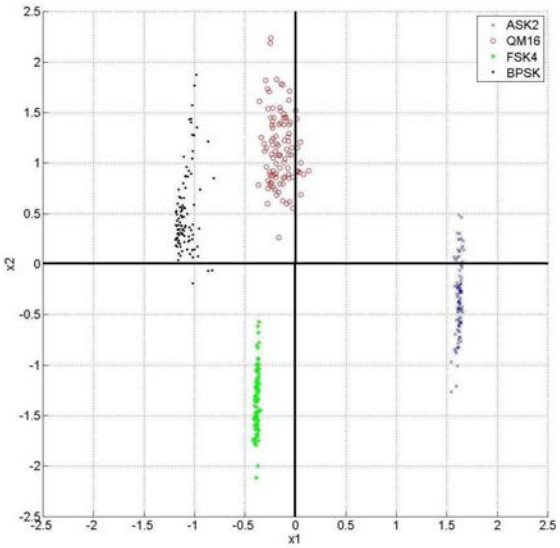
Normalizing the input data to zero mean is equivalent to shifting the data points and centering them around the axes origin, and the reduction to the unit variance is a scaling transformation which reduces the spread of the input data. Figure 3.2 shows the effect of normalizing raw input data to zero mean and a unit variance.



(a)



(b)



(c)

Figure 3.2: The effects of normalizing the input data to zero mean and unit variance. (a) the input data of four classes with their unmodified coordinates. (b) the data points are shifted so that the mean of the data equal to zero. (c) the magnitudes of the original input data are scaled to achieve a variance of one.

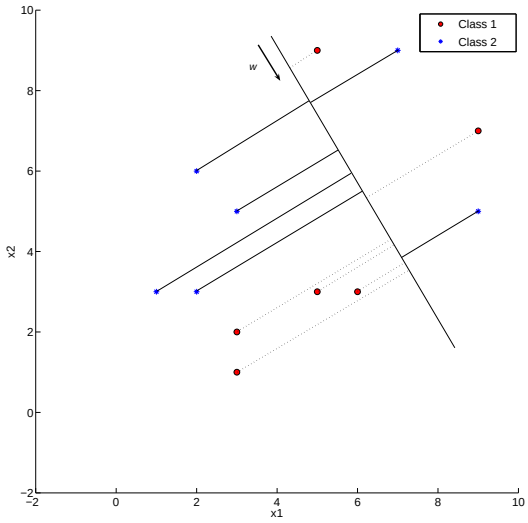


Figure 3.3: Projections of two-dimensional points from two classes onto the Fisher line which best separate the two classes.

3.3.2 Fisher Linear Discriminant(FLD). The FLD is widely used for pattern recognition. The usefulness of FLD stems from its use of a linear discriminant that yields maximum separation between two or more classes when class elements are projected onto the Fisher space from their original higher dimensional space. Figure 3.3 shows an example of projections of points from two classes in a two-dimensional space onto the one-dimensional Fisher line. Projections from each class onto the Fisher line fall in as a tight a group as possible. The level of discrimination is measured by the Fisher ratio

$$J(\mathbf{w}) = \frac{\mathbf{w}^T S_B \mathbf{w}}{\mathbf{w}^T S_W \mathbf{w}} \tag{3.4}$$

the transformation matrix $\mathbf{w}$, for which the FLD analysis finds a solution, maximizes the ratio between the "between classes scatter matrix" S_B and the "within classes scatter matrix" S_W. These scatter matrices in a c-class problem are

$$S_B = \sum_{i=1}^{c} N_i(\mu_i - \mathbf{m})(\mu_i - \mathbf{m})^T \tag{3.5}$$

and

$$S_W = \sum_{i=1}^{c} S_i \tag{3.6}$$

where N_i is the number of input per class C_i. Also, the scatter matrix S_i for each class, the mean vector μ_i of each class, and the total mean vector $\mathbf{m}$ are [4]

$$S_i = \sum_{\mathbf{x} \in C_i} (\mathbf{x} - \mu_i)(\mathbf{x} - \mu_i)^T, \tag{3.7}$$

$$\mu_i = \frac{1}{N_i} \sum_{\mathbf{x} \in D_i} \mathbf{x}, \tag{3.8}$$

and

$$\mathbf{m} = \frac{1}{N_c} \sum \mathbf{x} = \frac{1}{N_c} \sum_{i=1}^{c} N_i \mu_i, \tag{3.9}$$

where, the total number N_c of samples $\mathbf{x}$ is the sum of the number N_i of samples in each class. Maximizing the Fisher ratio follows the logic that the desired solution is one where the means of the classes are well separated. The optimal $\mathbf{w}$ has columns equal to the eigenvectors that correspond to the largest eigenvalues in the generalized eigen-problem

$$S_B \mathbf{w} = \lambda S_W \mathbf{w}. \tag{3.10}$$

Once $\mathbf{w}$ is determined the projections from the original D-dimensional space to lower d-dimensional space, where $d < c$ and $d < D$, are accomplished by

$$\mathbf{y} = \mathbf{w}^T \mathbf{x}, \tag{3.11}$$

where $\mathbf{y}$ is the projection vector.

Overall, the objective is to create a linear model that is trained with several sets, each corresponding to the modulation to be classified, of hundreds of signal sequences of known modulations. The Fisher plane is then found, which by definition produces the widest spread between class constellations of projected points.

For example, let $X_{n,1}, X_{n,2}, X_{n,3}, X_{n,4}, X_{n,5}$ be inputs to the model, where each vector $X_{n,i}$ is one of the five measured features for each of the $n = 300$ known signals S_n:

$$\begin{array}{cccccc} X_{1,1} & X_{1,2} & X_{1,3} & X_{1,4} & X_{1,5} & C_1 \\ X_{2,1} & X_{2,2} & X_{2,3} & X_{2,4} & X_{2,5} & C_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ X_{101,1} & X_{101,2} & X_{101,3} & X_{101,4} & X_{101,5} & C_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ X_{300,1} & X_{300,2} & X_{300,3} & X_{300,4} & X_{300,5} & C_3. \end{array}$$

Here C_i is the class to which the signal belongs. The covariance matrix for each class is

$$\Sigma_i = \frac{1}{N_i} \begin{pmatrix} X_{1,1} - m_{i,1} & \dots & X_{1,5} - m_{i,5} \\ \vdots & \ddots & \vdots \\ X_{100,1} - m_{i,1} & \dots & X_{100,5} - m_{i,5} \end{pmatrix}^T \cdot \begin{pmatrix} X_{1,1} - m_{i,1} & \dots & X_{1,5} - m_{i,5} \\ \vdots & \ddots & \vdots \\ X_{100,1} - m_{i,1} & \dots & X_{100,5} - m_{i,5} \end{pmatrix}.$$

where N_i is the number of inputs per class, in this case $N_i = 100$. The eigenvectors and eigenvalues of the 5x5 matrix H are then found, where the eigenvectors for the largest two eigenvalues define the 2D Fisher plane, and the largest three define the 3D Fisher hyper-plane.

$$H = \left(\sum_i N_i \Sigma_i \right)^{-1} \cdot \left[\sum_i N_i (\mathbf{m}_i - \mathbf{m})(\mathbf{m}_i - \mathbf{m})^T \right], \tag{3.12}$$

Once the projections of the input points in the X_1, X_2, X_3, X_4, X_5 space onto the Fisher plane are determined, a Gaussian is fit to each distribution (since they are assumed to be normal distributions):

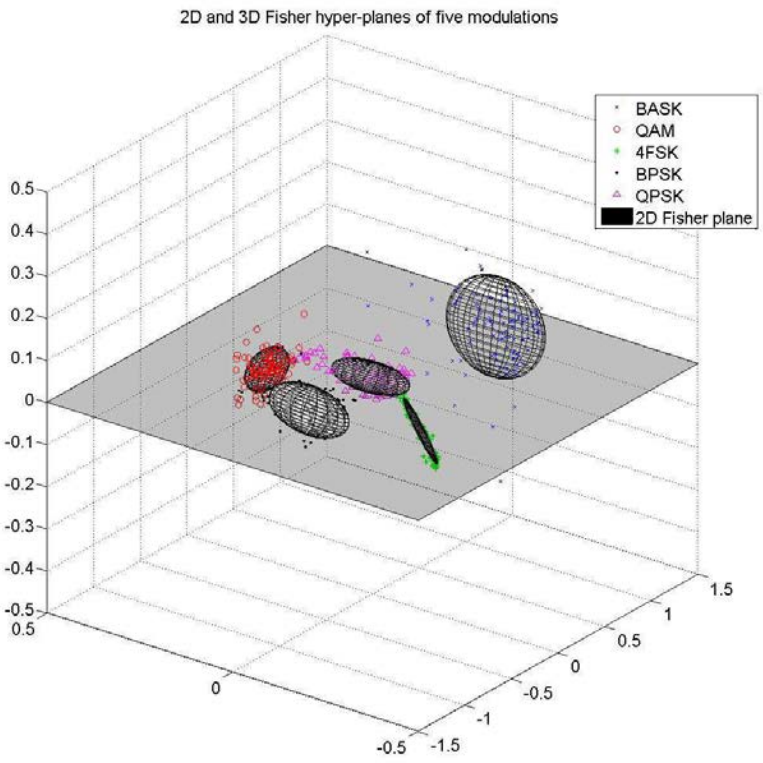


Figure 3.4: 2D and 3D Fisher planes for five classes.

$$p(\mathbf{x}) = \frac{e^{\left(-\frac{1}{2}(\mathbf{x}-\mathbf{m})^T \cdot \Sigma^{-1}(\mathbf{x}-\mathbf{m})\right)}}{(2\pi)^{5/2}\,|\Sigma|^{1/2}}. \tag{3.13}$$

Figures 3.4and 3.5 show examples of the two and three dimensional Fisher planes.

3.3.3 Multivariate normal densities. The traditional and popular multivariate normal density function for each class with d random variables y_i is

$$p(\mathbf{y}) = \frac{1}{(2\pi)^{d/2}\,|\Sigma|^{1/2}} exp\left[-\frac{1}{2}(\mathbf{y}-\mu)^T\Sigma^{-1}(\mathbf{y}-\mu)\right], \tag{3.14}$$

where Σ is the covariance matrix and μ is the mean vector and where

$$\mu = E[\mathbf{y}], \tag{3.15}$$

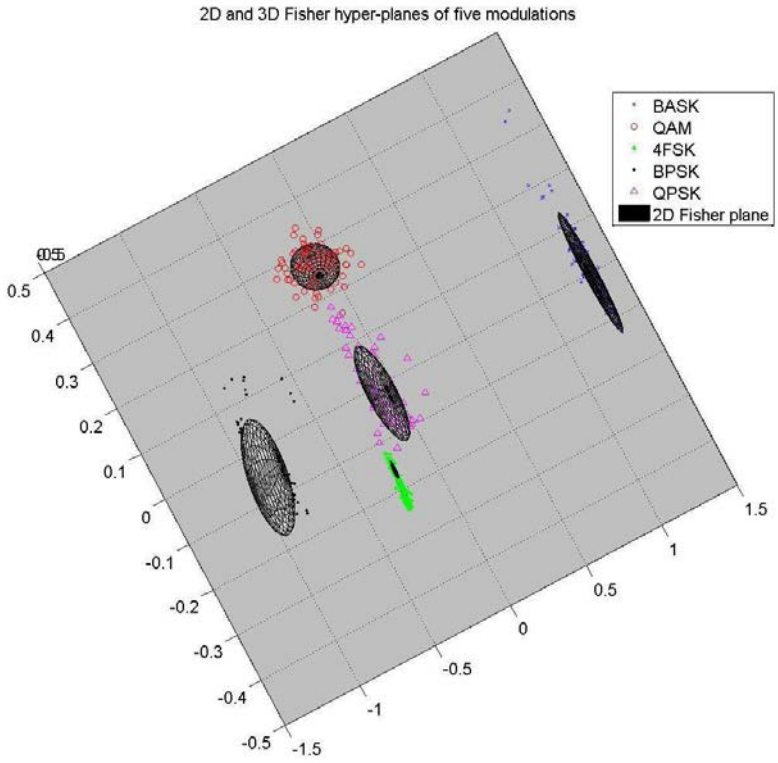


Figure 3.5: 2D and 3D Fisher planes for five classes (bird's-eye view). The view from this angle clearly shows the separation of the classes as they are projected onto the 2D Fisher plane.

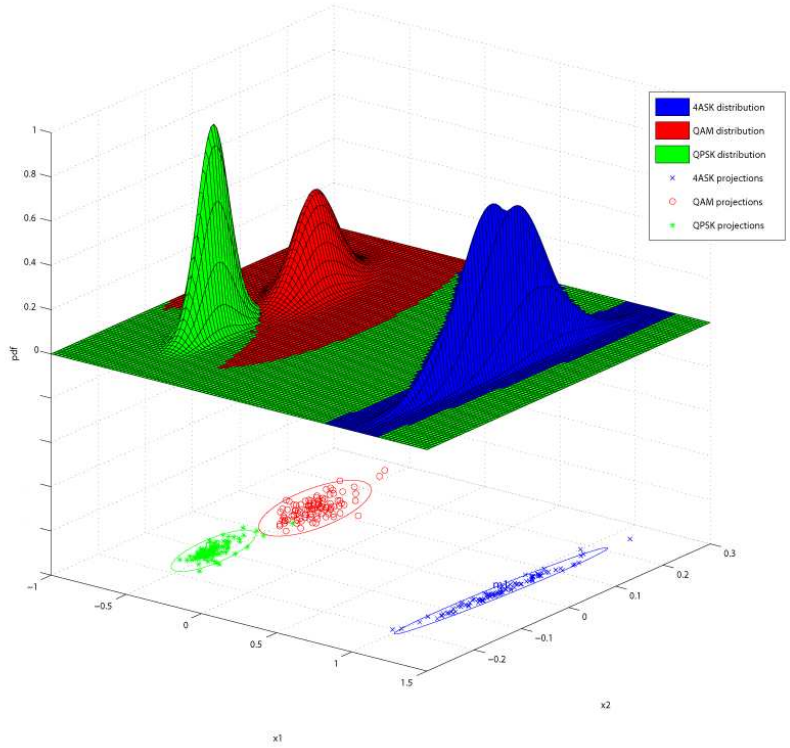


Figure 3.6: Each data class is projected onto the 2D Fisher plane and then is modeled as a Gaussian density to better describe the class distribution. The colored surfaces denote the ranges where each distribution is dominant. Bayesian classification uses this pattern to delineate the boundaries of maximum likelihood regions. A test signal is classified to be of the type of the class in which region the signal projection occurs.

$$\Sigma = E\left[(\mathbf{y} - \mu)(\mathbf{y} - \mu)^T\right]. \tag{3.16}$$

Figure 3.6 shows an example of a Gaussian fit to three classes that are projected onto their two dimensional Fisher plane.

3.3.4 Mahalanobis Distance. The measure from a point $\mathbf{x}$ to the mean μ of a multivariate density distribution with covariance $\mathbf{\Sigma}$ is

$$\Delta = \sqrt{(\mathbf{x} - \mu)^T \mathbf{\Sigma}^{-1} (\mathbf{x} - \mu)} \tag{3.17}$$

which is known as the Mahalanobis distance. This distance is taken from the exponent of the general multivariate normal density (Equation 3.14), where in this case $\mathbf{x}$ and μ are d-dimensional vectors, and $\mathbf{\Sigma}$ is a $(d \times d)$ matrix. Unlike the Euclidian distance, the Mahalanobis distance takes into account the correlations of the data set, which is important since in a multivariate distribution the densities expand in an uneven fashion along different directions, thus having ellipsoidal shapes. The Mahalanobis distance is simply the distance between the point $\mathbf{x}$ and the mean μ of a distribution divided by the width of the ellipsoid in the direction of $\mathbf{x}$.

3.4 Features

The five features introduced by Azzouz and Nandi [1] are chosen for the pattern recognition linear model that interfaces with the classification algorithm proposed here. These features provide a baseline for assessing the performance of the proposed algorithm.

3.4.1 Maximum Power Spectral Density (MPSD). MPSD is the maximum of the squared Fourier transform of the normalized amplitudes of the observed signals:

$$\gamma_{max} = \frac{max\,|DFT(a_{cn})|^2}{N_s} \tag{3.18}$$

where DFT( ) is the Discrete Fourier Transform, N_s is the number of samples, f is the frequency, and a_{cn} is the amplitude normalized to zero mean and unit variance, i.e.,

$$a_{cn} = \frac{|\mathbf{x}|}{m_a} - \mathbf{1}, \tag{3.19}$$

where $\mathbf{x}$ is the observed signal vector and m_a is the average value of the instantaneous amplitudes.

3.4.2 Absolute Amplitude. This feature is the standard deviation of of the "... absolute values of the normalized-centered instantaneous amplitude" for a segment of the signal [1] and is

$$\sigma_{AA} = \sqrt{\frac{1}{N_s}\left(\sum_{i=1}^{N_s} a_{cn}^2(i)\right) - \left(\frac{1}{N_s}\sum_{i=1}^{N_s} |a_{cn}(i)|\right)^2}. \tag{3.20}$$

3.4.3 Absolute Phase. This feature is the standard deviation of the absolute values of the " centered non-linear component of the instantaneous phase evaluated over the non-weak intervals of a signal segment" [1] and is

$$\sigma_{AP} = \sqrt{\frac{1}{C}\left(\sum_{x(i)>a_t} \phi_{NL}^2(i)\right) - \left(\frac{1}{C}\sum_{x(i)>a_t} |\phi_{NL}(i)|\right)^2}, \tag{3.21}$$

where C is the number of samples that have non-linear components ϕ_{NL} of the instantaneous phase ϕ. The threshold a_t eliminates signal segments most affected by noise.

3.4.4 Direct phase. This feature is the standard deviation of the direct phase also evaluated over the non-weak segment portion of the signal, and it is

$$\sigma_{DP} = \sqrt{\frac{1}{C}\left(\sum_{x(i)>a_t} \phi_{NL}^2(i)\right) - \left(\frac{1}{C}\sum_{x(i)>a_t} \phi_{NL}(i)\right)^2}. \tag{3.22}$$

3.4.5 Absolute Frequency. This feature is the "absolute value of the normalized-centered instantaneous frequency evaluated over the non-weak intervals of a signal segment" [1] and is

$$\sigma_{AF} = \sqrt{\frac{1}{C}\left(\sum_{x>a_t} f_{NL}^2\right) - \left(\frac{1}{C}\sum_{x>a_t} |f_{NL}|\right)^2}, \tag{3.23}$$

where

$$f_N = \frac{f - m_f}{r_s} \tag{3.24}$$

and f is the instantaneous frequency, m_f is the instantaneous frequency average, and r_s is the sequence symbol rate. The symbols are the communication segments produced by the modulation process which transforms binary block of zeros and ones to analog waveforms. Each observed signal is composed of a sequence of communication symbols.

3.5 The Algorithm

As discussed in section 3.2 the main processes of the MDCA algorithm happen within its core, yet the flow information within the algorithm depends on other essential modules.

3.5.1 The Database. All classifications require comparison of some sort. The MDCA algorithm is no different, as it requires an extended library of information about Fisher hyper-planes and the distributions they contain. The library is generated during the training process, where thousands of sample signals with different instances of noise and modulations are used. The information about each hyper-plane is stored in a ".dat" file, and each file is referred to as a "page". A page contains the **w** matrix that defines the Fisher hyper-plane of one specific combination of modulations (anywhere from two to eight modulations) as well as the mean vectors and covariance matrices of distributions of the modulations. Furthermore, one SNR level is used when generating a page. Therefore, there are seven pages for every combination at the specific SNR levels of -10dB, -5dB, 0dB, 5dB, 10dB, 15dB, and 20dB. Ideally, more SNR levels provide better accuracy, but since the testing of the algorithm is performed at these levels, there is no need for a larger database. Finally, the database itself is divided into a number of levels, and grouped into each level are pages with the same number of modulations. The levels are stacked in such a way that the first level always contains the pages with all modulations and each level after that has

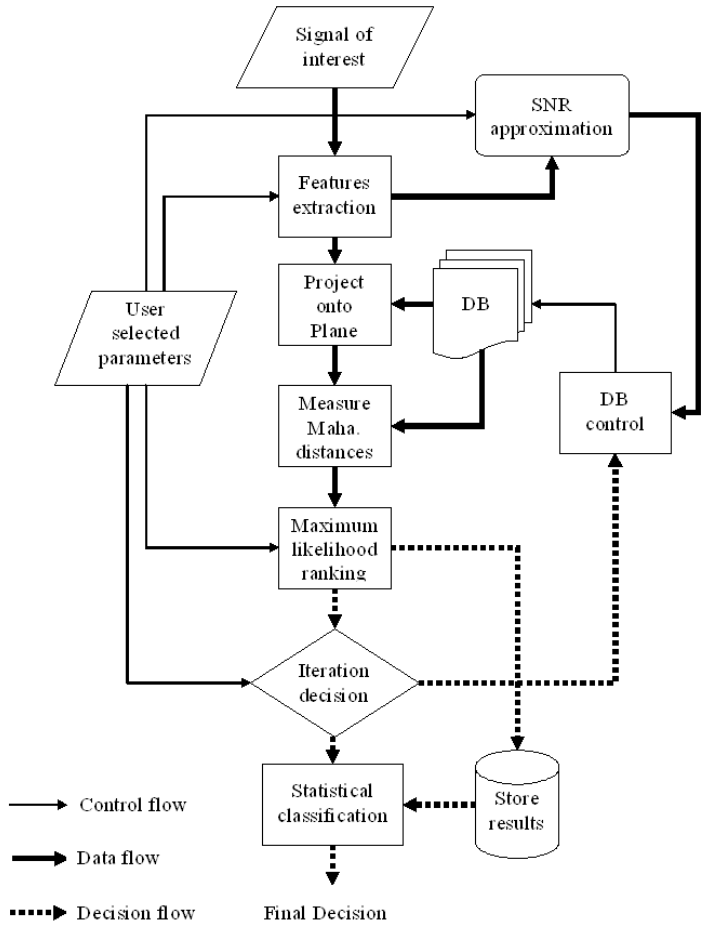


Figure 3.7: Flow diagram of the MDCA algorithm. Three types of information pass between modules. First, the data flow, which consists of hard values (SNR, extracted features, coordinates, directional and covariance matrices, etc.) that are passed from one module to another for the sole purpose of using them in mathematical operations. Next, the decision information flow, which is mainly composed of modulation categories. This information is used in steering the data flow to decide on the next comparisons. Finally, the control flow is simply information containing instructions to either do what the user specifically requests (use 3D classification instead of 2D, have the algorithm approximate SNR instead of the user providing a value, etc.) or direct instructions from control modules that decide what vectors and matrices to expedite to the algorithm core for each iteration.

pages with one fewer modulation each time. Hence, if the first level contains pages with eight modulations, the fourth level contains pages with five modulations. Not all levels are needed for the algorithm to work and produce good results. In this thesis only levels one and six are used to test the performance of the MDCA algorithm.

3.5.2 SNR Approximation. The extracted features from a signal are empirical data that also provide information about the SNR levels. Thus, an option is added to automatically approximate the level of SNR. Such an option is useful since it allows the algorithm to employ the best fitting pages in the classification process.

3.5.3 User Selected Paramaters. The MDCA takes a number of input parameters from the user in the form of both choice and value inputs. The choices the user can make are:

- The features to be used in the classification process
- Whether to allow the algorithm to approximate SNR level.

The types of values the user might add are

- The prior probabilities of modulations
- The SNR level
- The number of iterations the algorithm performs
- The levels from the database for each iteration

IV. Methodology

4.1 Introduction

This thesis verifies the concept of the MDCA algorithm. Therefore, a complete system is simulated (a system refers to a setup of a working algorithm in a life-like environment) as shown in Figure 4.1. Two steps are necessary to generate final statistics for correct classifications. First, a number of modulated signals are generated and then used to create the database of hyper-planes, which is then used in the MDCA algorithm as the reference DB. Second, new sets of signals are generated then run through the MDCA algorithm for classification. The performance of the algorithm is a tally of correct classifications to incorrect ones.

4.2 Experiment Design and Parameters

Simulating a working MDCA in a virtual environment is a necessary step for demonstrating the multi-dimensional classification concept. Two sets of signals with similar parameters are used. During the training phase the first set produces the database of likely signals, while the second set is later used during the testing phase to gauge the performance of the algorithm. The values of several signal parameters are chosen within constrained intervals. These choices are made to mimic previous work [1, 2] and also to keep the simulation work reasonably manageable.

Several parameters were considered during the generation of both the training and test signals.

4.2.1 Modulation. Only eight communication modulation types are used, since the main objective of this research is to show the usefulness of classification in high dimensions (not thorough testing of the concept in every possible scenario). Also, the choice of these modulations is made consistent with previous work [1] for comparison purposes. The chosen modulations include BPSK, QPSK, ASK2, ASK4, FSK2, FSK4, QAM16, and QAM64. With the exception of the QAM family of

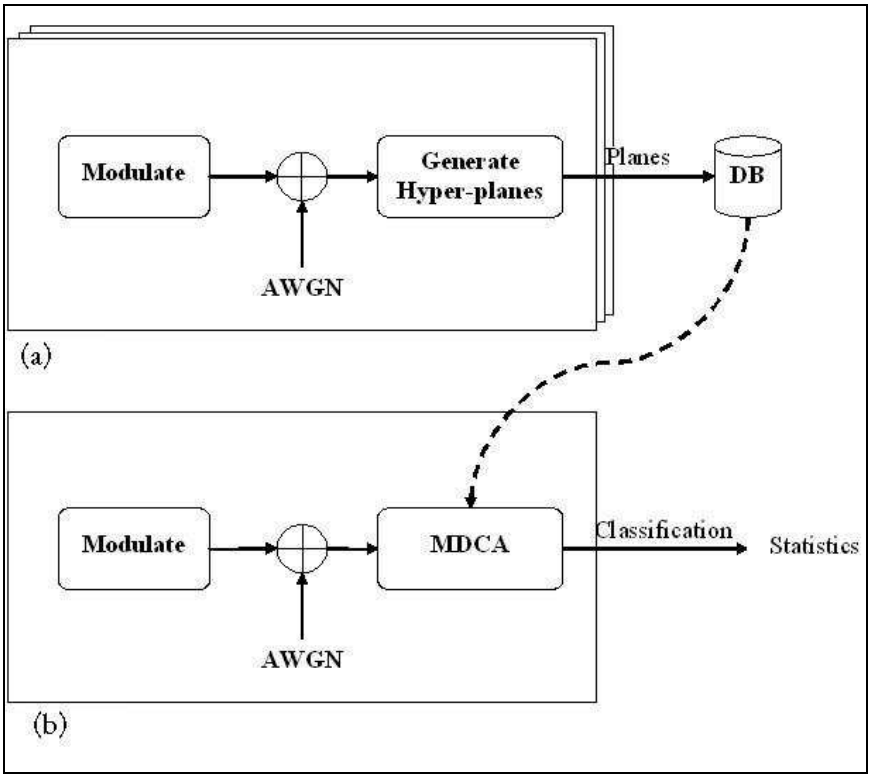


Figure 4.1: Experiment setup showing the two steps of (a) generating the database used in the MDCA algorithm and (b) testing of the algorithm.

modulations, all can be represented by

$$S_A(t) = A_c \cdot A(t) \cdot cos(2\pi(f_c + f(t))t + \phi(t) + \theta_0), \tag{4.1}$$

where A_c denotes the carrier power, f_c is the carrier frequency, θ_0 is the initial phase, and $A(t)$, $f(t)$ and $\phi(t)$ are the signal baseband amplitude, frequency, and phase respectively. Using Euler's formula

$$e^{jx} = \cos x + j\sin x \tag{4.2}$$

and ignoring the initial phase term, the modulated signal can also be represented by

$$S_A(t) = \mathrm{Re}\{A(t)e^{j\phi(t)}e^{j(2\pi(f(t)+f_c)t)}\} \tag{4.3}$$

For the first type of modulation, phase shift keying or PSK, the information is encoded within the phase of the signal, where each set of *2k*-bit values in the digital stream to be transmitted is represented by one of the k possible signals, each with a specific phase:

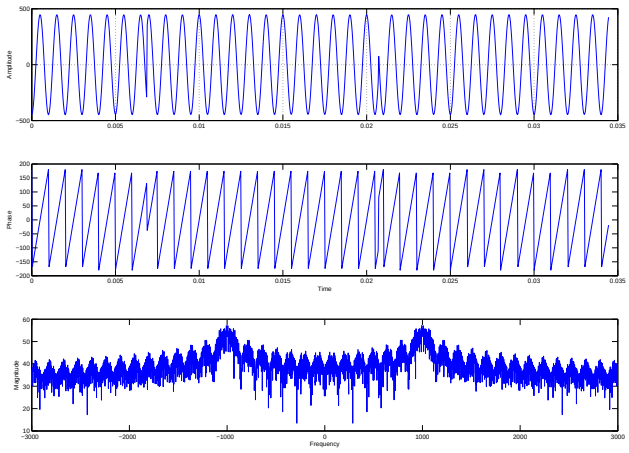
$$S_A(t) = A_c \cdot cos(2\pi f_c t + \phi(t)) \tag{4.4}$$

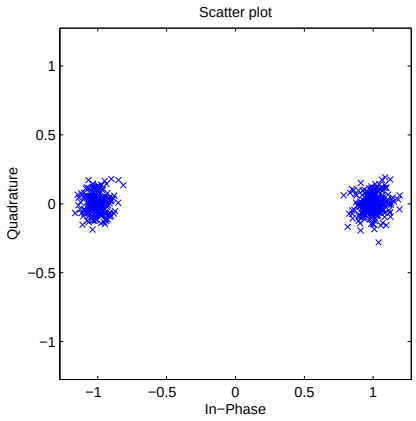
Figure 4.2 and Fig. 4.3 show one example each of the phase shift keying modulations BPSK and QPSK.

For amplitude shift keying or ASK, the information is amplitude encoded during modulation, and therefore the amplitude term $A(t)$ toggles between predetermined constant values, but no phase information is encoded in the signal. Function (4.1) then reduces to

$$S_A(t) = A_c \cdot A(t) \cdot cos(2\pi f_c t). \tag{4.5}$$

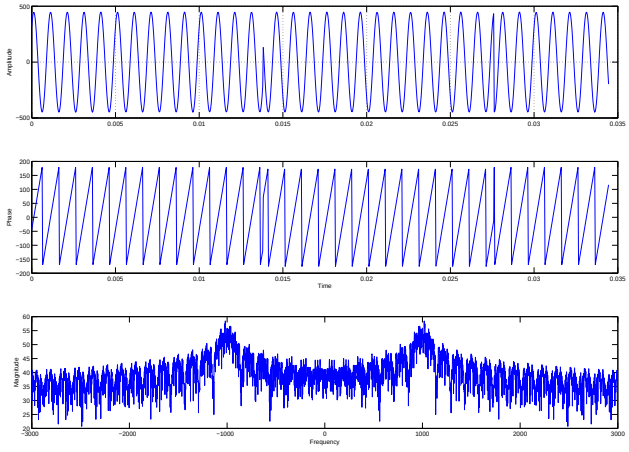
Examples of the two amplitude shift keying modulations (ASK2 and ASK4) are shown in Figure 4.2 and Figure 4.3.

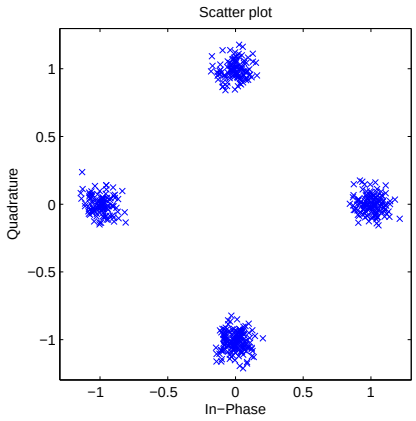




(a) (b)

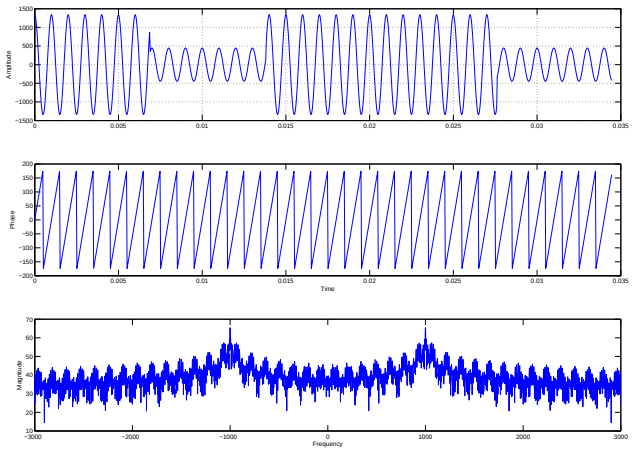
Figure 4.2: BPSK modulation (a) amplitude, phase, and spectrum representations. (b) scatter plot.



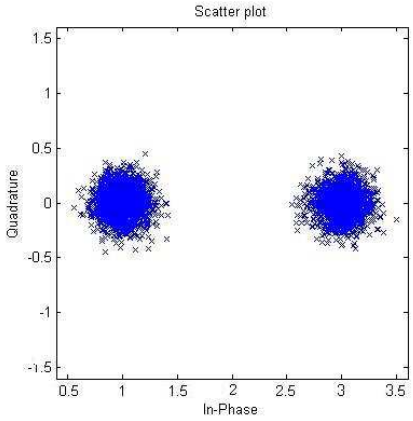


(a) (b)

Figure 4.3: QPSK modulation(a) amplitude, phase, and spectrum representations. (b) scatter plot.

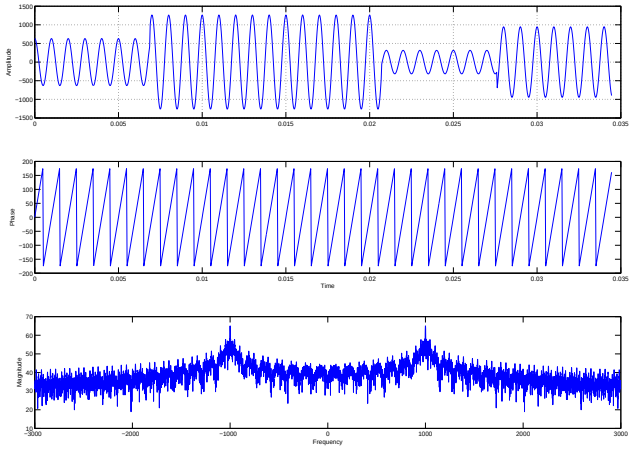


(a)

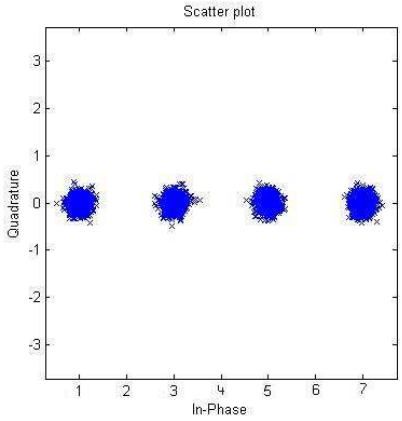


(b)

Figure 4.4: ASK2 modulation (a) amplitude, phase, and spectrum representations. (b) scatter plot.



(a)



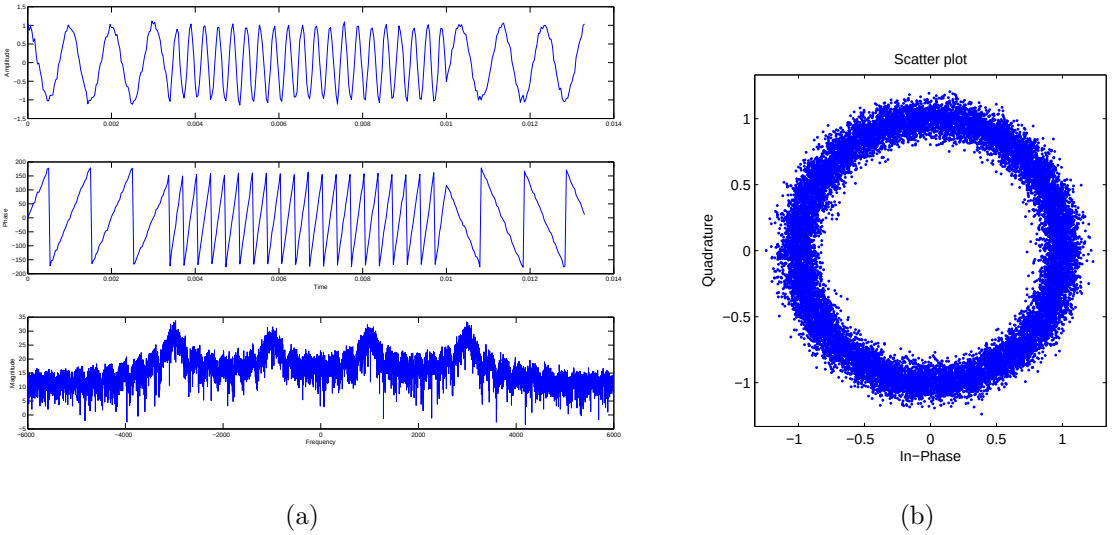
(b)

Figure 4.5: ASK4 modulation (a) amplitude, phase, and spectrum representations. (b) scatter plot.

The frequency shift keying FSK modulation, as the its name suggests, uses different frequencies for each symbol representation:

$$S_A(t) = A_c \cdot cos(2\pi(f_c + f(t))t + \theta_0) \tag{4.6}$$

Examples of FSK modulated signals are shown in Figure 4.6 and Fig. 4.7. Finally,



(a) (b)

Figure 4.6: FSK2 modulation (a) amplitude, phase, and spectrum representations. (b) scatter plot.

QAM is coded both in phase and amplitude, but not in frequency so that $f(t) = 0$:

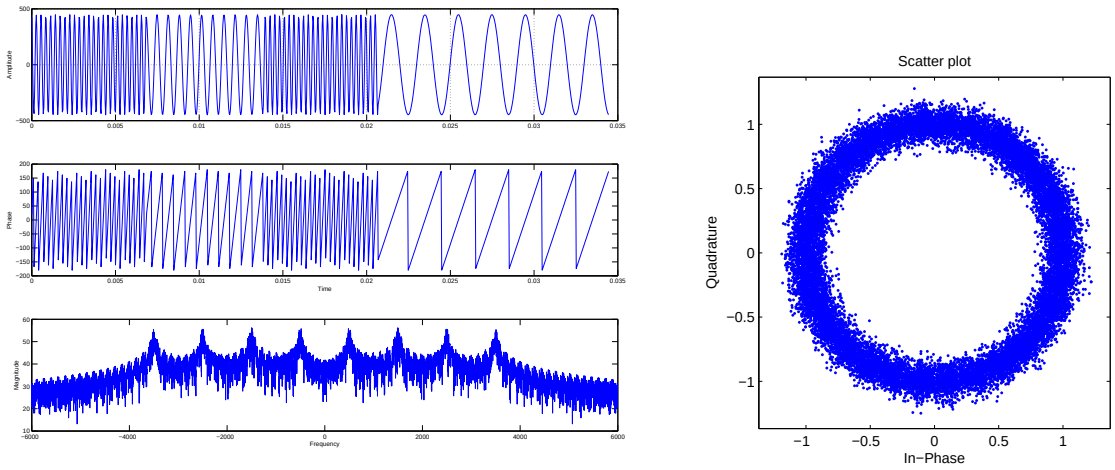
$$S_A(t) = I(t) \cdot cos(2\pi f_c t) + Q(t) \cdot sin(2\pi f_c t). \tag{4.7}$$

This equation is an alternate representation of (Equation 4.1) and is derived from (Equation 4.3), which is also

$$S_A(t) = [I(t) + jQ(t)]\, e^{j(2\pi f_c t)}, \tag{4.8}$$

where

$$I(t) + jQ(t) = A(t)e^{j\phi(t)} \tag{4.9}$$



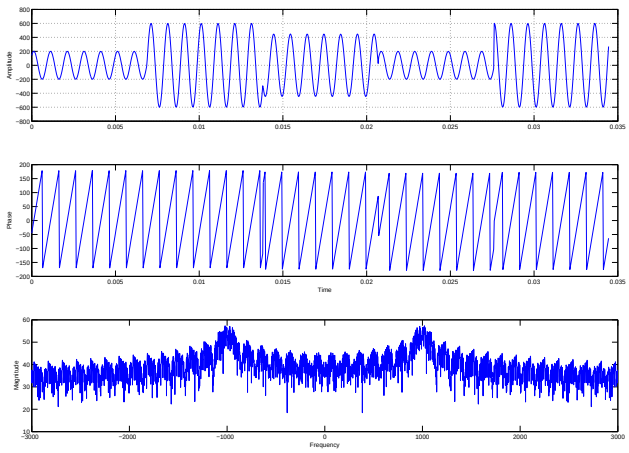
(a) (b)

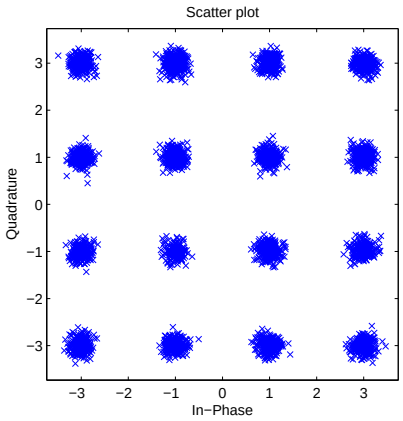
Figure 4.7: FSK4 modulation (a) amplitude, phase, and spectrum representations. (b) scatter plot.

This representation is refered to as the in-phase *(I)* and quadrature-phase *(Q)* components of the signal.

4.2.2 Number of Symbols N_{sym}. Based on previous research, the number of communication symbols in the classified signal is a critical parameter that defines the performance of most existing AMR algorithms. This dependence stems from the relation between the number of symbols and the sampling time of the signal. Although several algorithms use thousands of communication symbols per classified signal, in this research only 40 symbols per signal are used. Indeed, experimenting with different sampling rates and number of symbols reveal that fewer symbols are needed if a signal is over-sampled (used 29kHz sampling rate which equates to 15 times Nyquist rate or more.) This property is dependent on the features considered for the MDCA algorithm.

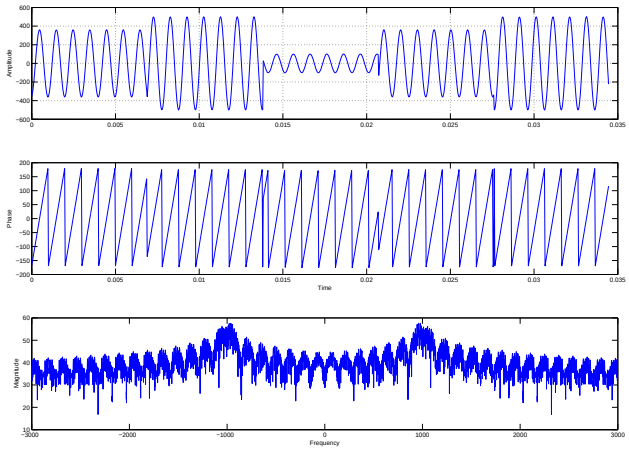
4.2.3 Carrier Frequency f_c. All the frequencies are in the near-baseband range from 200Hz to 2000Hz in 100Hz increments.

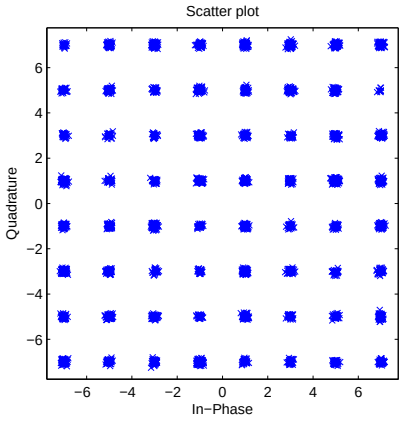




(a) (b)

Figure 4.8: QAM16 modulation (a) amplitude, phase, and spectrum representations. (b) scatter plot.





(a) (b)

Figure 4.9: QAM64 (a) amplitude, phase, and spectrum representations. (b) scatter plot.

4.2.4 Signal-to-Noise Ratio(SNR). The SNR level is by far the most critical of all parameters, as step variations in its values can cause significant changes in the values of the features extracted from any modulated signal. The signal-to-noise ratio is defined as the ratio of the signal power to the noise power

$$\mathbf{SNR} = \frac{\mathbf{P}_{signal}}{\mathbf{P}_{noise}} = \left(\frac{\mathbf{A}_{signal}}{\mathbf{A}_{noise}}\right)^2, \tag{4.10}$$

where A is the root mean square (RMS) amplitude. Due to the wide dynamic range of the considered signals, the SNR levels are measured in decibels at the receiver input.

$$\mathbf{SNR}[dB] = 10\log_{10}\left(\frac{\mathbf{P}_{signal}}{\mathbf{P}_{noise}}\right) \tag{4.11}$$

In this research only a limited number of SNR values are considered. Additive white Gaussian noise (AWGN), with varying power levels is then added to the generated signals to achieve SNR levels equal to **-10dB, -5dB, 0dB, 5dB, 10dB, 15dB, and 20dB** are the targeted values. However, due to the randomness of the added noise the actual measured SNR values randomly hover around $\pm 1dB$ from the desired values. Furthermore, every signal used in generating the database or during the testing phase is composed of a modulated sequence to which a unique noise instance is added. Table 4.1 summarizes the parameters in the signals used to generate the database.

Table 4.1: Signals used to generate the Database

Modulations	f_c(Hz)	SNR(dB)	Total
BPSK			1400
QPSK			1400
FSK2	100	-10	1400
FSK4	to	to	1400
ASK2	2000	20	1400
ASK4	(100Hz increments)	(5dB increments)	1400
QAM16			1400
QAM64			1400
Grand Total			**11200**

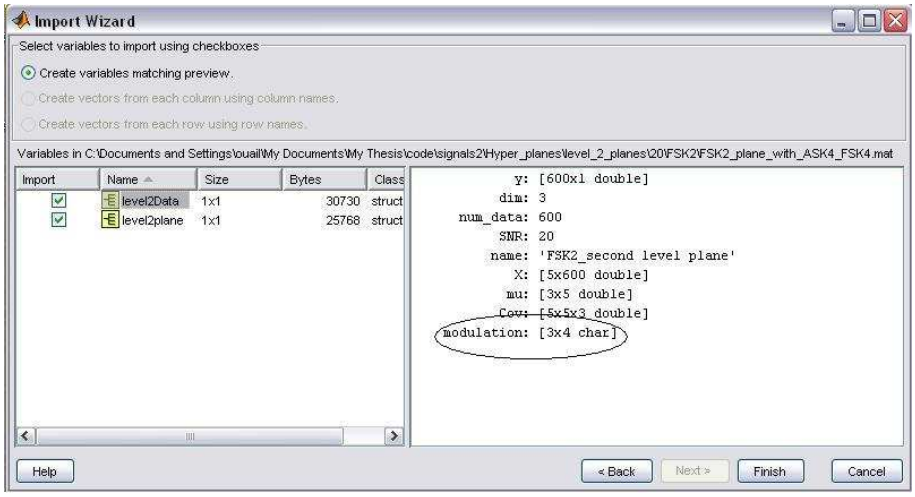
4.3 Simulation Overview

The simulation process is comprised of two main steps, database generation and classification. These steps and performance measurement are overviewed here.

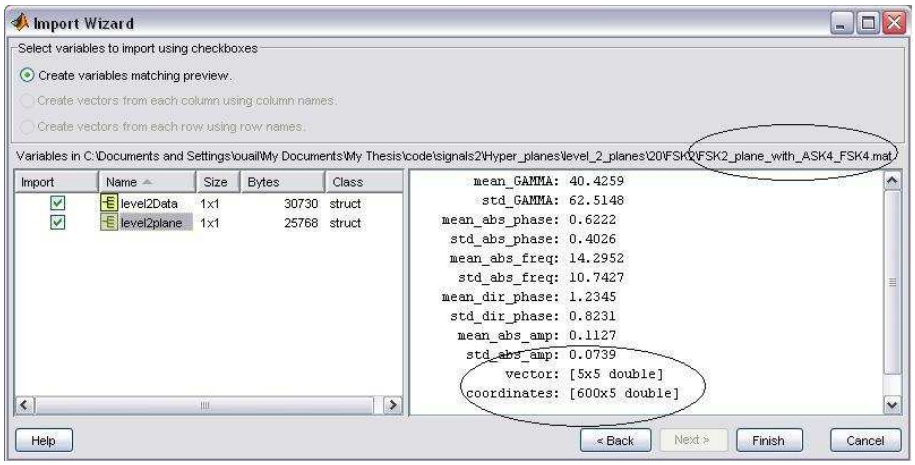
4.3.1 Database Generation. The MDCA algorithm is a primitive pattern recognition concept, for it requires training in all the environments in which it might be used as a classification tool. The product of MDCA training process is a database containing information on the behavior of modulated signals in specific environments. The environment can be a simple AWGN channel with a single signal present or, for example, a fading/multi-path channel with third party interferers, but for the purpose of this research only the first case is considered. The database is therefore a collection of user defined scenarios in which classification is to take place. The central element of the MDCA database is a "page", which is simply a ".dat" file (".mat" for MATLAB) containing the following information:

- **Modulation:** The name of the modulations used to generate the page
- **SNR:** The level of the modulated signals used to generate the page. Only one SNR level is allowed per page.
- **Projection matrix:** The matrix of the Fisher plane defined by the signals considered for the page
- **Mean vector:** The mean of the normal distribution of each modulation in the Fisher plane
- **Covariance matrix:** The covariance of the distribution of each modulation in the Fisher plane

Since each page is defined by its "rank", or the number of modulations, the database is organized by grouping in "levels" all pages of equal rank. An MDCA algorithm requires a database of at least two levels. The user decides on the number and type of levels and therefore which to generate for the training phase. Here only two levels are used. The first consists of seven eight-modulation pages, one for each



(a)



(b)

Figure 4.10: Example of the content of a page. Here a MATLAB import wizard shows the content of the page "FSK2 page with ASK4 FSK4.mat." From the descriptive file title, this page contains the Fisher plane described by the three modulations FSK2, FSK4, and ASK4. Here (a) containsmatrix "modulation", along with the SNR level and other information used in generating the page, (b) contains the "vector", which is the projection matrix of the plane as well as the projection "coordinates" of the data used in generating the plane.

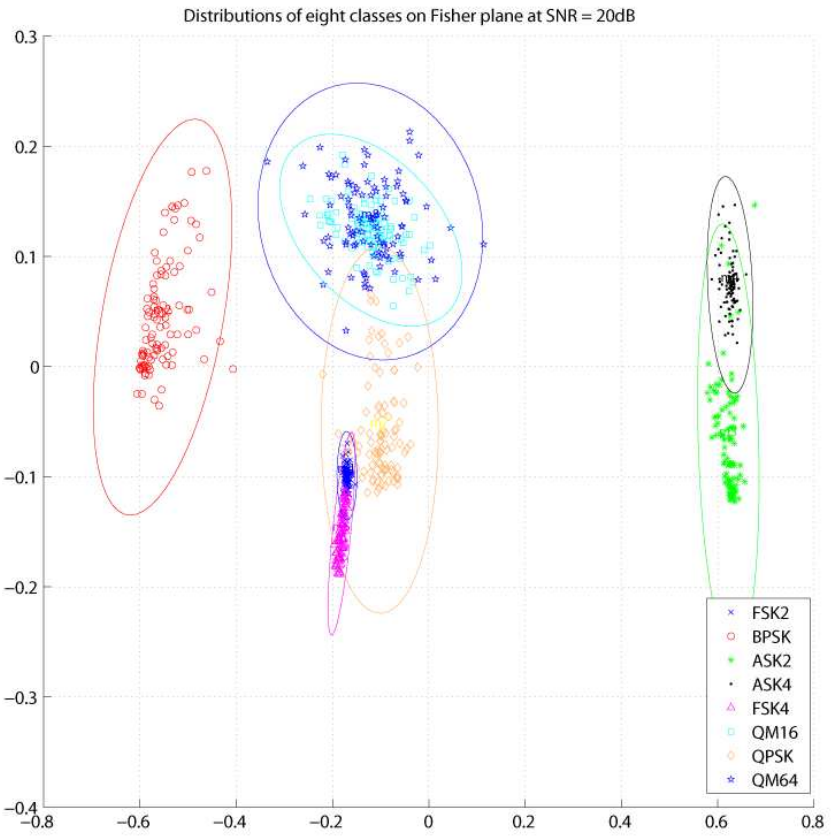


Figure 4.11: Example of a 2D first level page of rank = 8 with SNR = 20dB. The projection matrix defines the plane, and covariance matrices and mean vectors define the ellipses. Also shown are the projections of training data used in generating this 2D Fisher hyper-plane. The ellipses correspond to the equal probability contours for each distribution that enclose, 87.5% of the normal distribution density.

SNR value, and the second consists of all possible combinations of three-modulations at every SNR value. Figure 4.11 and Figure 4.12 show visual representations of the information contained in a first level "page", plus the projections of the signals (each point represents a signal) used in its generation. The two representations are in the 2D and 3D Fisher hyper-planes generated by the data. The MDCA algorithm is an intelligent classifier capable of choosing the best information from its database while classifying a signal, reviewing its choices, and adjusting for discrepancies.

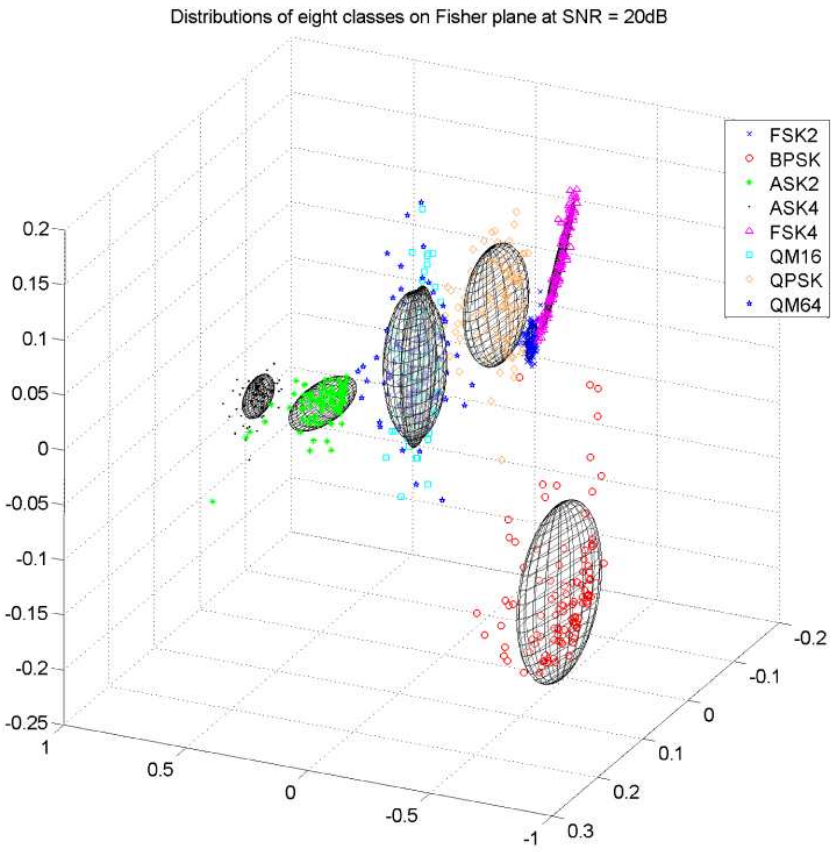


Figure 4.12: Example of a 3D first level page, with SNR = 20dB and rank = 8, along with projections of training data used to generate this 3D Fisher hyper-plane.

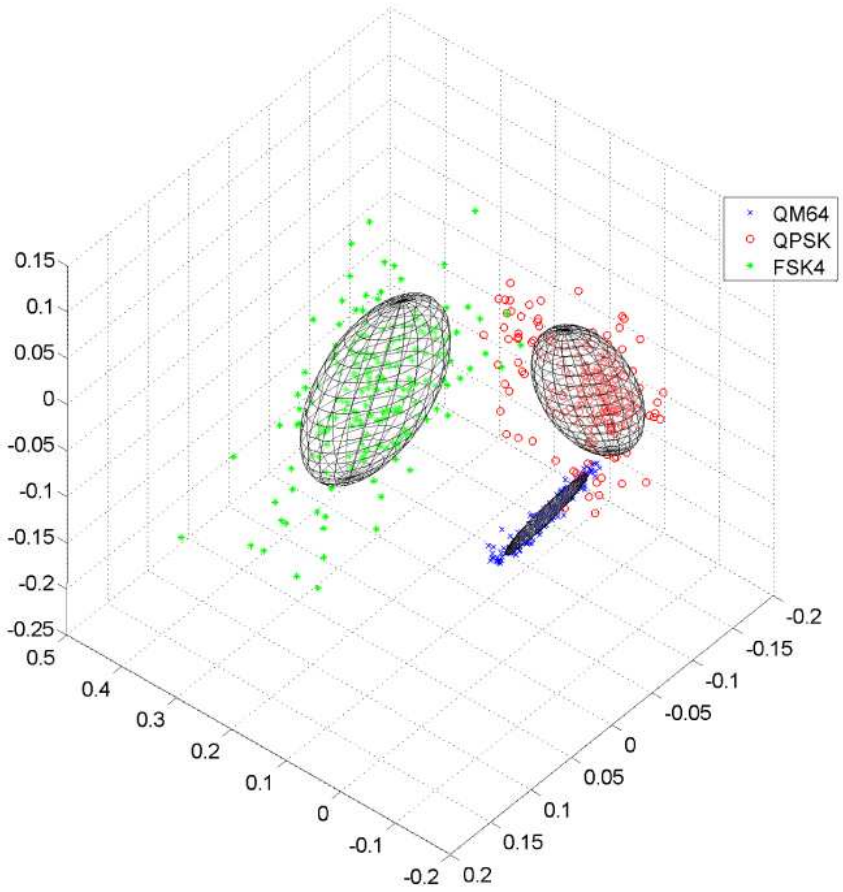


Figure 4.13: Example of a 3D second level page with SNR = 20dB and rank = 3, along with the projections of training data used to generate this 3D Fisher hyper-plane.

4.3.2 Classification Process and Performance Measurement. The simulation of a working MDCA algorithm in a virtual environment is conducted using two hundred signals per modulation per SNR level for a total of 11,200 signals. Classification is performed on one signal at a time. First the five features described in Section 2.2 are extracted from the signal. These features are used as coordinates of a point, which is then projected onto the first level Fisher hyper-plane with the most appropriate SNR value with dimension (2D or 3D) chosen by the user. Figure 4.14 shows the projection point of an example test signal (measured SNR = 15.3 dB) onto the most appropriate first level page (8 modulations Fisher plane and SNR = 15dB.) Next, the Mahalanobis distances between the projection point and the mean of each of the eight distributions are measured. The smallest three Mahalanobis Distances correspond to the three most likely modulations to which one of the test signal might belong. The results of the projection on the first level page are thus, three ranked modulations.

In the second level projection step the same features extracted from the test signal are now projected onto three groups of pages. Each group consists of pages containing at least one of the three most likely modulations identified in the previous step, and each page is a combination of three modulations from the eight. In contrast to the first level step in which only one page is used, the second level step involves projections onto 63 pages (21 per group) because for every SNR level there is only one page of rank eight and $C_3^8 = 56$ pages of rank 3, of which only 21 consistently include a given modulation. Figures 4.15, 4.16 and 4.17 show six examples of the 63 projections. The purpose of the second level step is to compare the test signal to all modulations in a restricted and biased environment. The bias is in favor of the three most likely modulations from step one, and the restriction is to the presence of fewer (only 3) modulations per page. These limitations produce a "zoom in effect" which, due to the better separation between the normal distributions in the 3-modulations Fisher planes, produces a better approximation of the test signal. Since all the pages in each group contain one of the three modulations identified in the first step, projecting the test signal on the pages of a group is similar to comparing the modulation defining

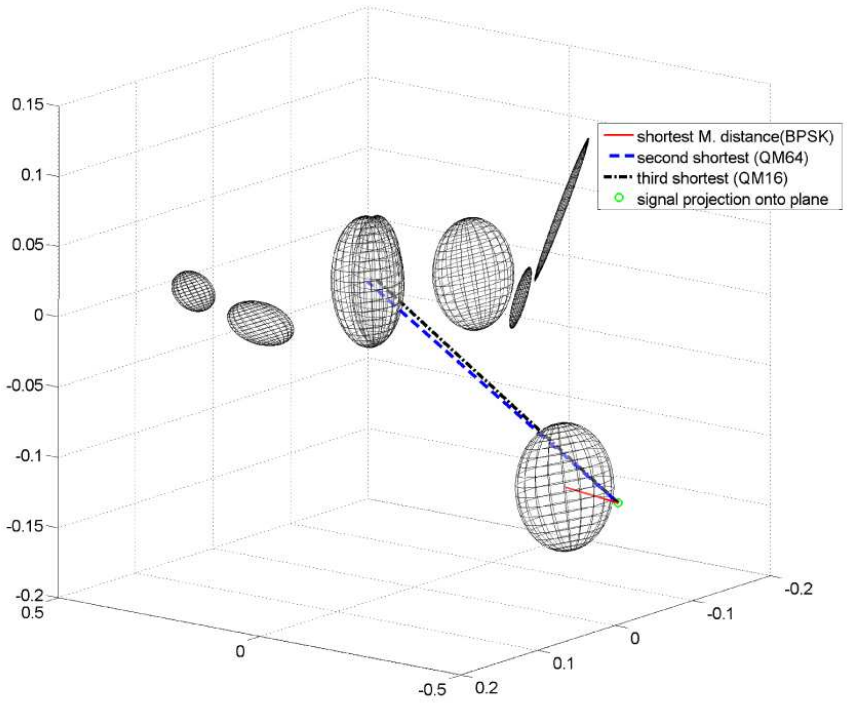
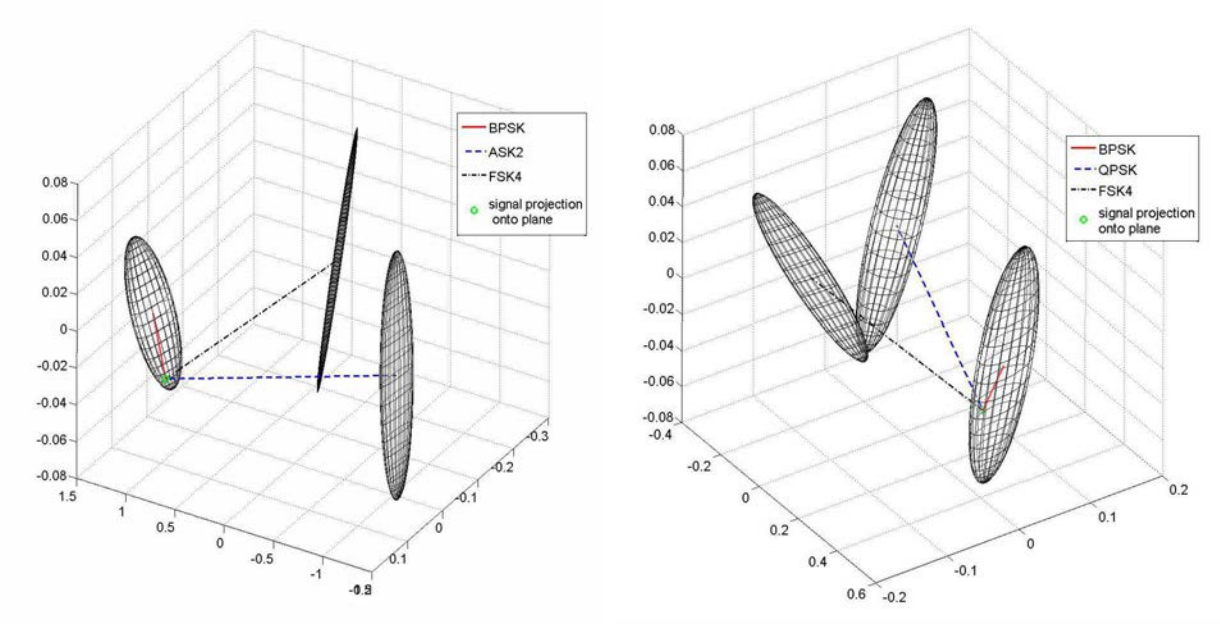


Figure 4.14: Projection of a test signal onto the 3D Fisher hyper-plane with all eight modulations and the corresponding estimated SNR level.

the group to all other modulations versus the actual modulation of the test signal. Thus, the Mahalanobis distances are measured and ranked for each projection. The modulation that performs best within its group is declared the "winner". In the case of a tie between two modulations, the pages containing both are used for a final arbitration. Figure 4.18 shows a flow diagram overview of the MDCA algorithm as implemented here.

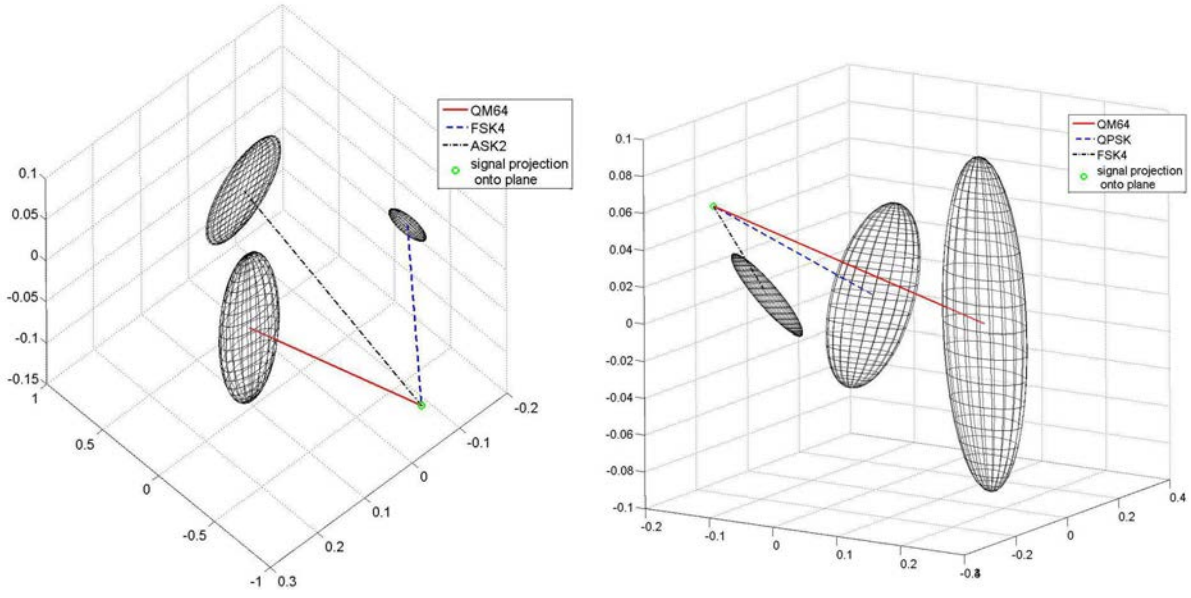
Once the MDCA algorithm produces a result, the classification generated for each signal is compared to its actual identity and the answer is recorded as

- Hit: for a correct classification
- Miss: for an incorrect classification
- Near Hit: if the algorithm declares the correct class as second most likely



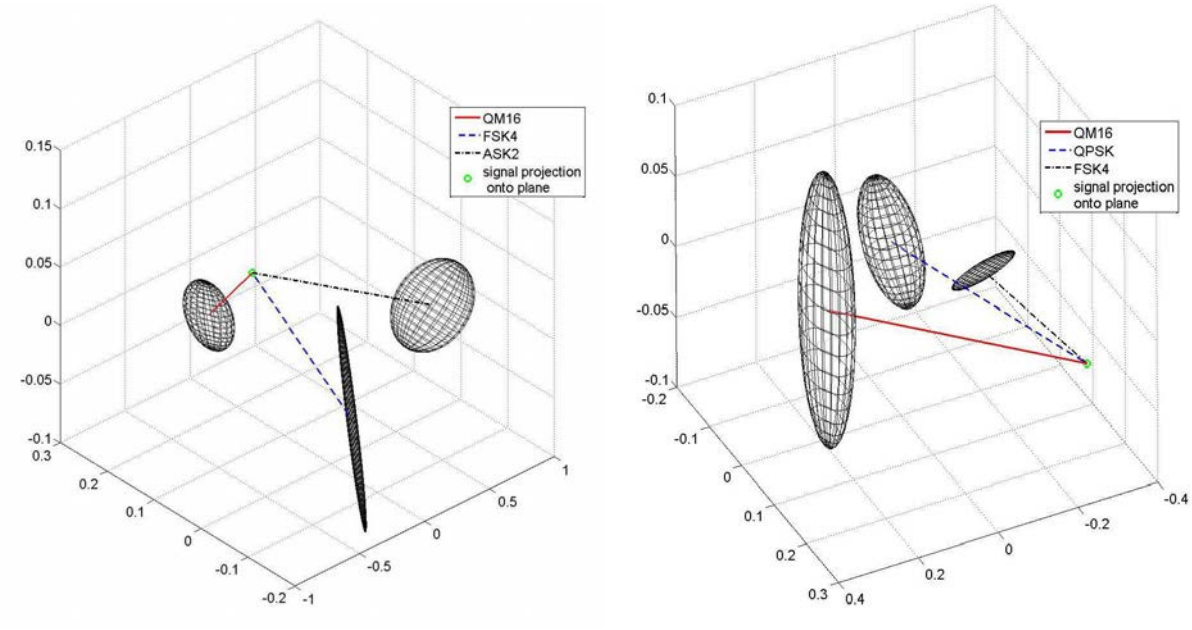
(a) (b)

Figure 4.15: Two examples of the projection of a test signal onto second level 3D Fisher hyper-planes of rank 3. The planes are defined by the first choice "BPSK" as defined in the first level projection.



(a) (b)

Figure 4.16: Two examples of the projection of a test signal onto second level 3D Fisher hyper-planes of rank 3. The planes are defined by the second choice "QAM64" as defined in the first level projection.



(a) (b)

Figure 4.17: Two examples of the projection of a test signal onto second level 3D Fisher hyper-planes of rank 3. The planes are defined by the third choice "QAM16" as defined in the first level projection.

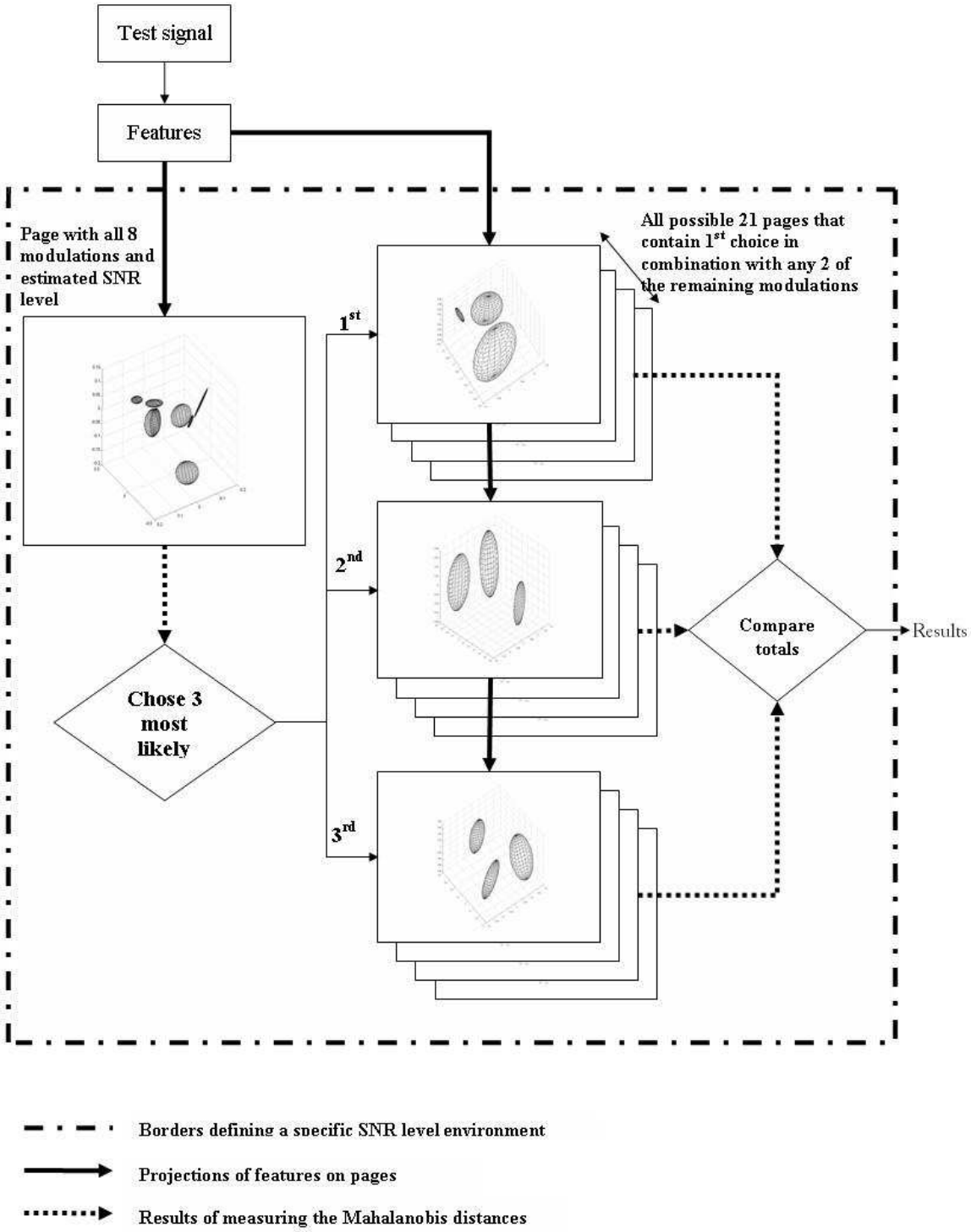


Figure 4.18: Flow diagram of classification of a test signal as performed in a fixed SNR level environment.

A complete tally of the results of all simulation runs is used for assessing the performance of the MCDA algorithm.

4.4 SNR Estimation

As shown in the previous sections, the MDCA algorithm requires a good database of well organized (SNR/Modulations) libraries. However, the proper use of the database requires the knowledge of the SNR level of the signal to be classified. This requirement stems from the fact that the noise level affects the values of the features extracted from a modulated signal. Figures 4.19 to Fig. 4.22 show changes in the values of four features with respect to SNR level.

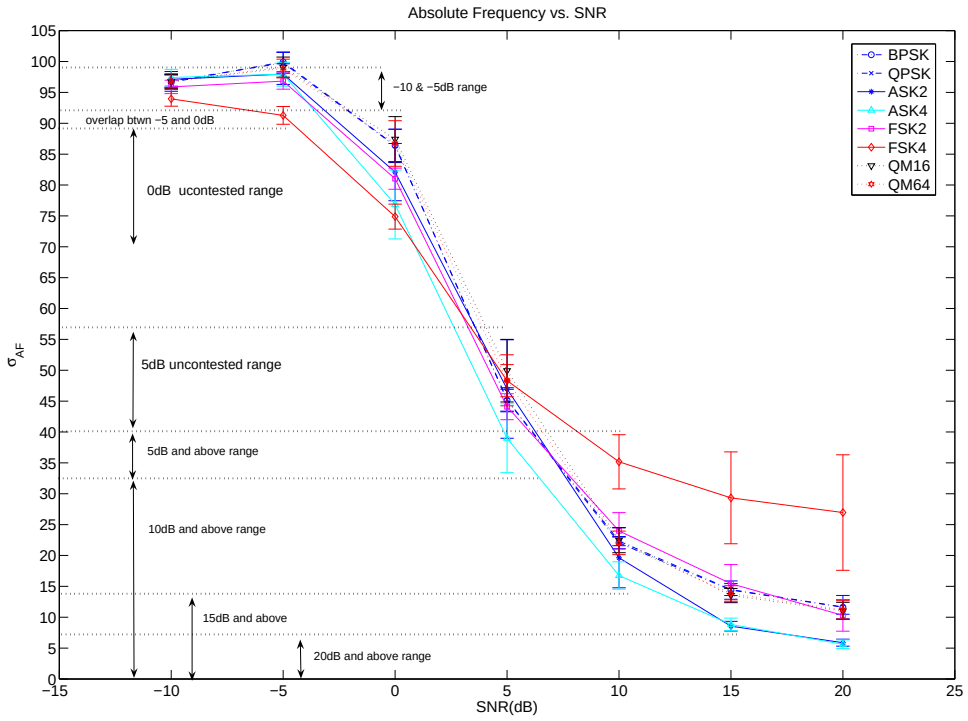


Figure 4.19: Changes in the Absolute Frequency feature with respect to SNR values.

Thus, the pages describing an identical group of modulations at different SNR levels can be dramatically different from one another. Figure 4.23 contains four vi-

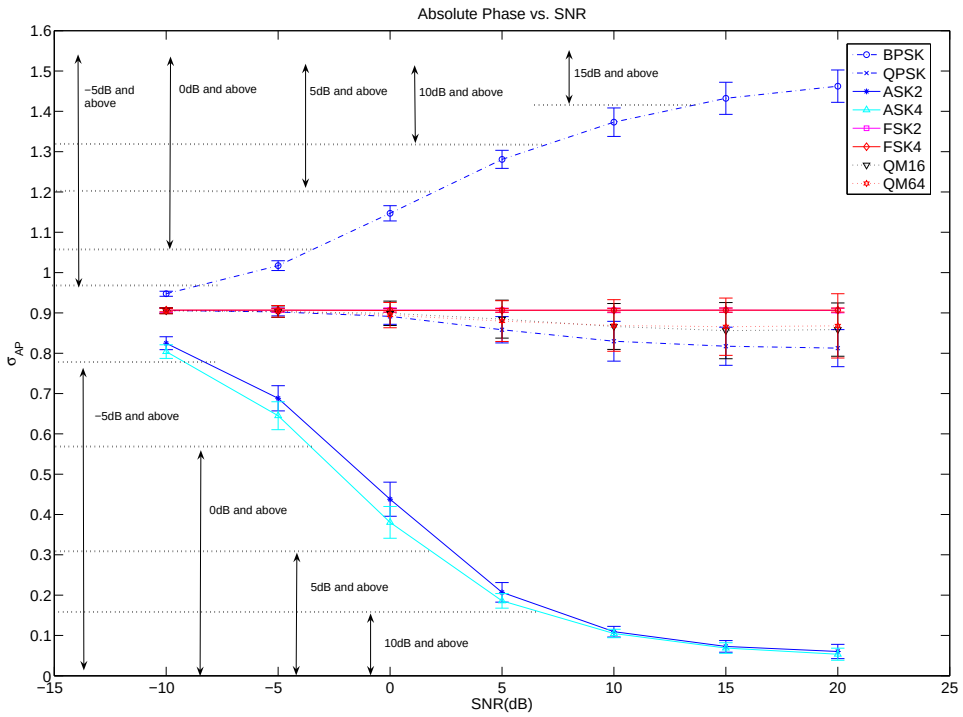


Figure 4.20: Changes in the Absolute Phase feature with respect to SNR values.

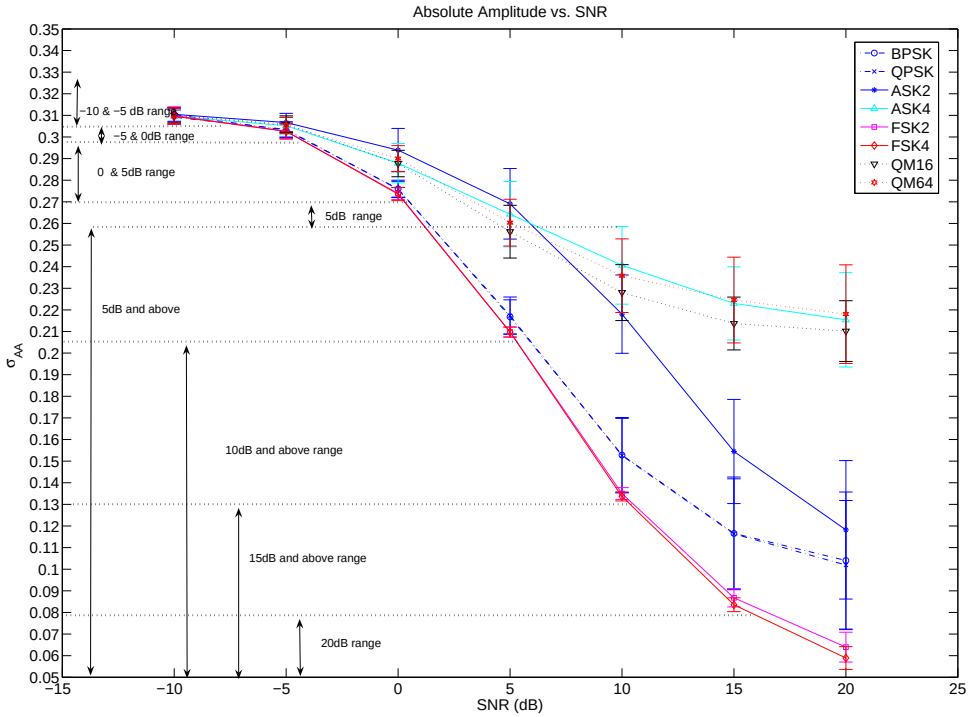


Figure 4.21: Changes in the Absolute Amplitude feature with respect to SNR values.

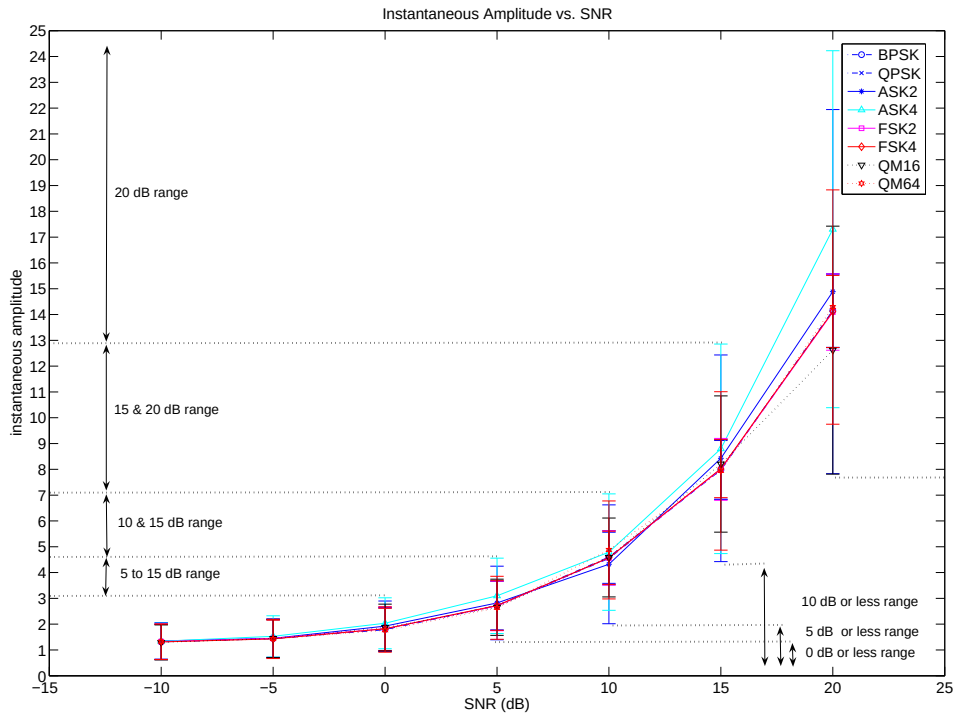


Figure 4.22: Changes in the Instantaneous Amplitude with respect to SNR values.

sualizations of pages defined by the same group of four modulations (ASK2, ASK4, FSK4 and QM16) at four different SNR levels 15dB, 10dB, 5dB, and -5dB.

Since the knowledge of SNR is not always available to the user, a simple method using some of the same features extracted for classification purposes can produce an adequate approximation of the SNR level. Each extracted feature is compared to a series of threshold values which determines the SNR interval to which it belongs. By stacking the intervals produced from each feature, a final and narrower interval is produced. The range of the final interval may hold no more than one of the SNR levels considered in the simulation. If not, the lowest SNR level that fits in the interval range is used for the classification that follows.

Only Absolute Frequency, Absolute Phase, Absolute Amplitude and Direct Phase features are used in the SNR estimator. The processing of each feature produces an SNR likelihood interval. Stacking all the likelihood intervals reveals the estimated SNR value of the signal being tested. Figure 4.24 shows a top level diagram of the approach implemented to estimate the SNR level of the test signal. As an example, the feature values extracted from a test signal, which happen to be a BPSK signal at $SNR = -5dB$, are:

```
Gamma max          =  3.284354e+000
Absolute Phase     =  1.030133e+000
Direct Phase       =  1.874272e+000
Absolute Amplitude =  3.030919e-001
Absolute Frequency =  9.842693e+001
```

Table 4.2 shows the ranges within which each of the four extracted feature fall, and the overlap of all ranges.

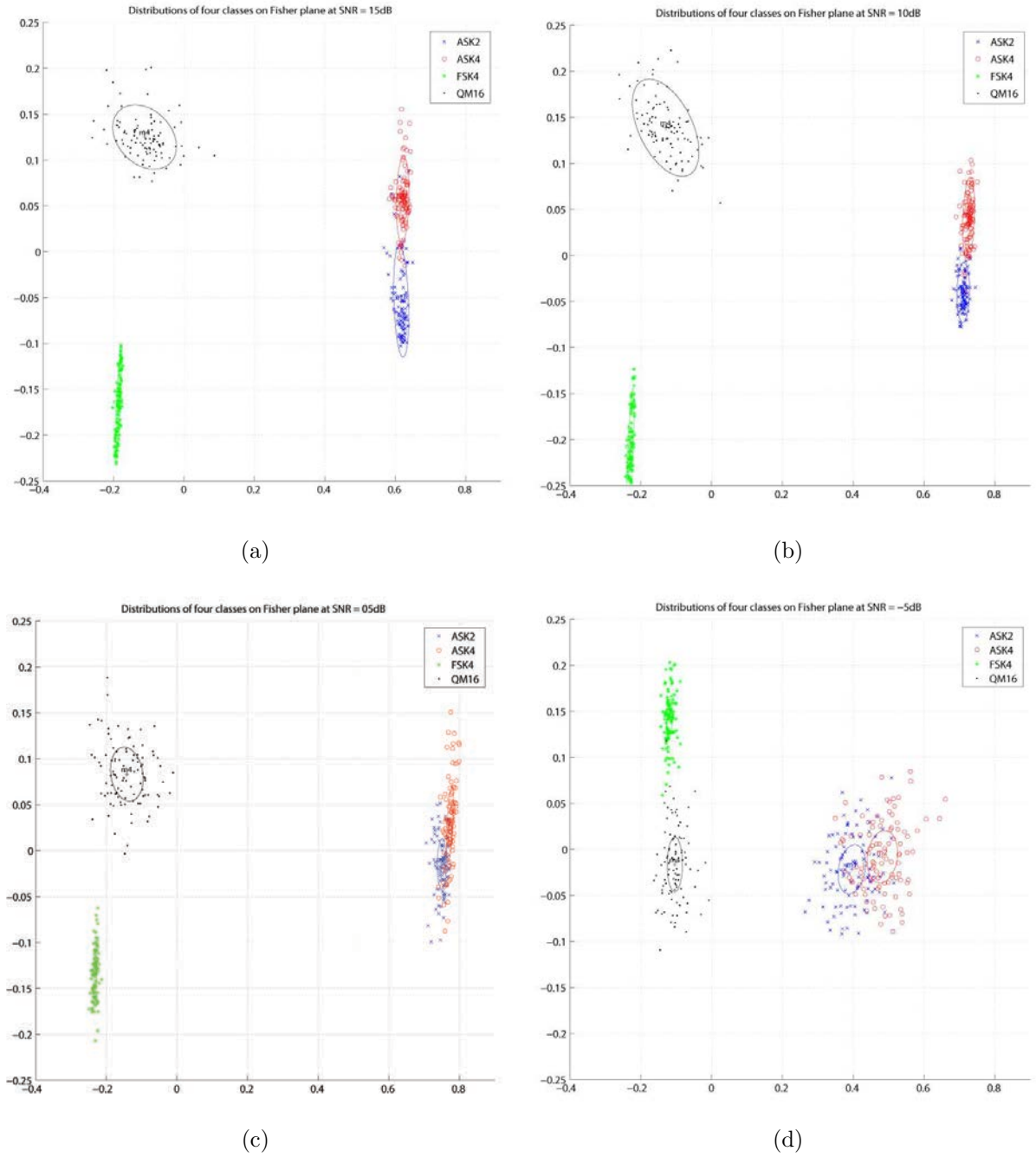


Figure 4.23: The SNR level of signals affect the values of extracted features, and hence the Fisher planes of a set of modulations changes as SNR changes. Here the Fisher planes of a group of four modulations are shown at four SNR levels (a) 15dB. (b) 10dB. (c) 5dB. (d) -5dB.

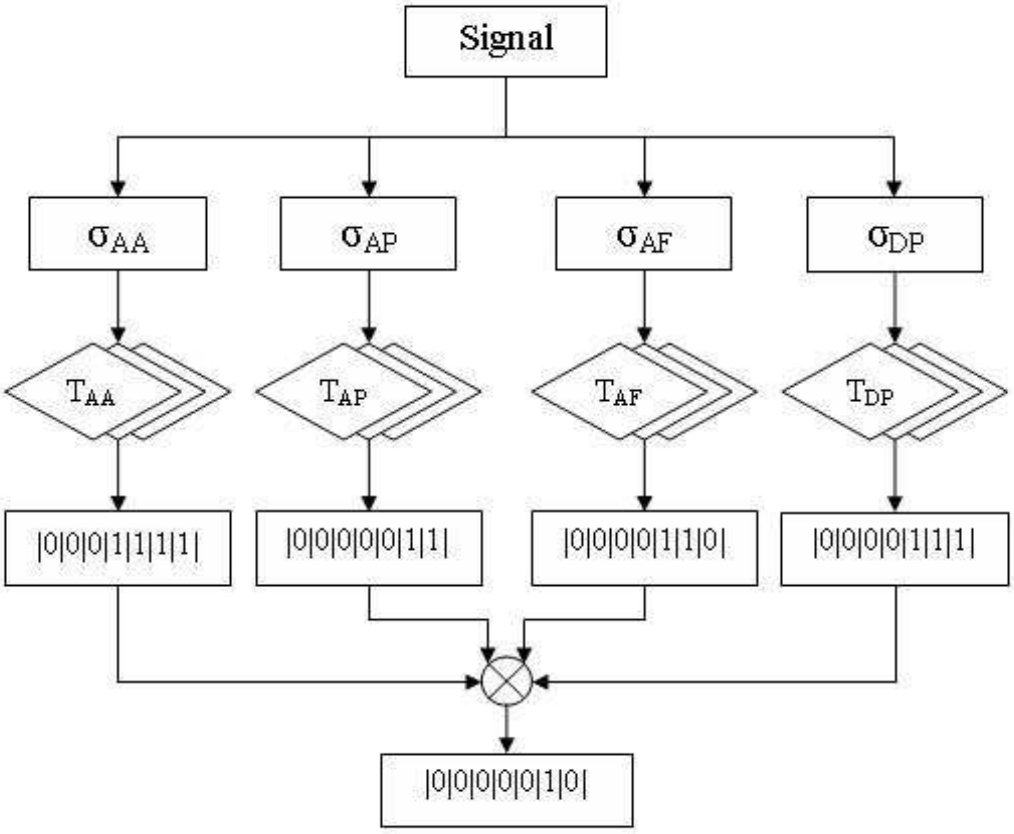


Figure 4.24: Diagram of the SNR estimation method. Each of the four features is input to a distinct branching tree (T) which then produces a (1x7) vector of ones and zeros corresponding to the seven SNR regions. The ones indicate that the given feature value is within the region, the zeros indicate the opposite. Multiplying the four output vectors of the four branching trees produces the desired overlap. The `Matlab`® code for each of the four branching trees is provided in Appendix 2.

Table 4.2: Example of SNR level estimation of a signal

SNR range	$-10dB$	$-5dB$	$0dB$	$5dB$	$10dB$	$15dB$	$20dB$
Absolute Frequency	✓	✓	-	-	-	-	-
Absolute Phase	-	✓	✓	✓	✓	✓	✓
Absolute Amplitude	-	✓	✓	-	-	-	-
Direct Phase	-	✓	✓	✓	✓	✓	✓
Final Range	-	✓	-	-	-	-	-

V. Results and Analysis

This chapter outlines the simulation results of the MDCA test example as introduced in Chapter 4. The performances of the SNR estimator and that of the MDCA classifier are simulated separately. The simulation of the MDCA classifier is conducted with assumed knowledge of the SNR level. After Section 5.3 the results of a simulation of MDCA with SNR estimation are introduced and discussed. The verification process is considered in this chapter rather than in Chapter 4 because the review of the results produced by the MDCA algorithm is more of a subjective process than an objective one. Indeed, the decision to accept the result of the algorithm or to conduct a rerun depends entirely on the choices made by the user during the training phase. The database generated during the training phase provide detailed information on the behavior of modulations when compared to one another in a restricted environment. A good analysis of such behavior (locations, and collisions of the constellations representing the modulations) assesses the reliability of the answer provided by the MDCA algorithm, and hence if there is a need for a classification rerun.

5.1 Results

The results of the simulation of the SNR estimation module and the classification algorithm are presented in this section. First, the performance of the SNR estimation module is presented for signals of unknown modulations. Then, the results of the simulations of the MDCA algorithm with SNR levels known are given, followed by the simulation results of the algorithm where the SNR levels are estimated.

5.1.1 SNR Estimation. Tables 5.1 through 5.8 show the results of SNR level estimation for each of the eight modulations considered here.

There are two main points to observe from the results in Tables 5.1 to 5.8:

Table 5.1: SNR estimation of signals modulated with FSK2 modulation

Simulated	Deduced SNR level (dB)						
SNR level (dB)	-10	-5	0	5	10	15	20
-10	**183**	17					
-5	32	**168**					
0			**200**				
5				**190**			
10					**198**	2	
15						**200**	
20						4	**196**

Table 5.2: SNR estimation of signals modulated with FSK4 modulation

Simulated	Deduced SNR level(dB)						
SNR level (dB)	-10	-5	0	5	10	15	20
-10	**183**	17					
-5	21	**176**					
0			**200**				
5				**191**			
10					**167**	3	
15						**184**	
20							**182**

Table 5.3: SNR estimation of signals modulated with ASK2 modulation

Simulated	Deduced SNR level (dB)						
SNR level (dB)	-10	-5	0	5	10	15	20
-10	**184**	16					
-5	1	**179**					
0			**175**				
5				**166**	4		
10					**178**	7	
15						**196**	1
20						7	**192**

Table 5.4: SNR estimation of signals modulated with ASK4 modulation

Simulated	Deduced SNR level (dB)						
SNR level (dB)	-10	-5	0	5	10	15	20
-10	**178**	21					
-5		**190**					
0			**190**				
5				**144**	38		
10					**169**	9	
15						**170**	8
20						7	**182**

Table 5.5: SNR estimation of signals modulated with BPSK modulation

Simulated	Deduced SNR level (dB)						
SNR level (dB)	-10	-5	0	5	10	15	20
-10	**176**	24					
-5		**197**					
0			**192**				
5				**200**			
10					**200**		
15					1	**197**	2
20						46	**154**

Table 5.6: SNR estimation of signals modulated with QPSK modulation

Simulated	Deduced SNR level (dB)						
SNR level (dB)	-10	-5	0	5	10	15	20
-10	**173**	27					
-5	20	**179**					
0			**187**				
5				**200**			
10					**200**		
15					42	**158**	
20						144	**56**

Table 5.7: SNR estimation of signals modulated with QM16 modulation

Simulated	Deduced SNR level (dB)						
SNR level (dB)	-10	-5	0	5	10	15	20
-10	**180**	20					
-5	62	**132**					
0		11	**176**				
5			2	**198**			
10					**197**		
15					145	**53**	
20					15	180	**0**

Table 5.8: SNR estimation of signals modulated with QM64 modulation

Simulated	Deduced SNR level(dB)						
SNR level (dB)	-10	-5	0	5	10	15	20
-10	**179**	20					
-5	73	**123**					
0		21	**169**				
5			1	**198**			
10					**187**		
15					145	**36**	
20					24	157	**0**

- The sum of the estimated signals per row does not always add up to the total trials, which means that the estimator fails to assign an SNR level to a number of test signals.
- In most of the cases where the SNR level is not estimated correctly, the answer provided by the estimator is that of a level adjacent to the actual SNR level. Also, most incorrect estimations are of a lower value rather than a higher one. This is no coincidence, since the estimator always chooses the lowest SNR level when more than one level is possible.

These failures are primarily due to structural weaknesses, non-considered cases, and imperfect conditional statements in the four branching trees used in the estimator. Therefore, further adjustments to the branching trees are needed, and corrections could lead to a better estimator performance.

Overall, the performance of the SNR estimator is satisfactory in that the estimation does not assume any knowledge of the nature or existence of a signal in the investigated AWGN channel. The estimator introduced here is unique in that most estimators introduced in literature are application specific models [15], and therefore most of them require some knowledge of the signals present in the considered channels.

A weakness of the proposed estimator is that it is channel dependent model. Any variations in the type of noise (colored vs. AWGN), the number of signals present in the channel, and the type of possible modulated signals require an overhaul of all four branching trees.

5.1.2 Modulation classification. Since the features introduced by Azzouz and Nandy [1] in their DMRA algorithm are the same ones used here for simulation of the MDCA algorithm, it is pertinent to first present the performance achieved by Azzouz and Nandy not for comparison purposes but for use as a reference. Tables 5.9 and 5.10 show the performance of the DMRA algorithm at SNR levels of 15dB and 10dB.

Table 5.9: Confusion matrix of the DMRA algorithm at SNR = 15dB, as published by Azzouz and Nandy [1]

Simulated	Deduced Modulation Type							
Modulation Type	FSK2	FSK4	ASK2	ASK4	BPSK	QPSK	QAM16	QAM64
FSK2	**199**	1						
FSK4	1	**197**				2		
ASK2			**198**	2				
ASK4				**200**				
BPSK	1				**199**			
QPSK	2					**198**		
QAM16							**NA**	
QAM64								**NA**

Table 5.10: Confusion matrix of the DMRA algorithm at SNR = 10dB, as published by Azzouz and Nandy [1]

Simulated	Deduced Modulation Type							
Modulation Type	FSK2	FSK4	ASK2	ASK4	BPSK	QPSK	QAM16	QAM64
FSK2	**199**	1						
FSK4	1	**197**				2		
ASK2			**198**	2				
ASK4				**200**				
BPSK	1				**199**			
QPSK	2					**198**		
QAM16							**NA**	
QAM64								**NA**

Tables 5.11 through 5.17 show the results of the MDCA algorithm as simulated in a two layer setup (8/3), where 8 refers to the rank of the pages in the first layer and 3 refers to the rank of the pages in the second layer.

Table 5.11: Confusion matrix of the MDCA (8/3) at SNR = 20dB

Simulated	Deduced Modulation Type							
Modulation Type	FSK2	FSK4	ASK2	ASK4	BPSK	QPSK	QAM16	QAM64
FSK2	**184**	3				13		
FSK4	6	**194**						
ASK2			**193**					
ASK4			1	**199**				
BPSK					**200**			
QPSK						**186**		14
QAM16						1	**44**	155
QAM64							11	**189**

Table 5.12: Confusion matrix of the MDCA (8/3) at SNR = 15dB

Simulated	Deduced Modulation Type							
Modulation Type	FSK2	FSK4	ASK2	ASK4	BPSK	QPSK	QAM16	QAM64
FSK2	**184**	6				10		
FSK4	6	**194**						
ASK2			**194**	6				
ASK4			3	**197**				
BPSK					**200**			
QPSK						**197**		3
QAM16							**115**	85
QAM64							45	**155**

5.2 Observations and Analysis

Unlike the the performance of the Azzouz and Nandi DMRA algorithm, which is particularly strong and consistent, the results from the MDCA algorithm varied widely across the SNR ranges as well as between modulations types.

Several observations can be made from Tables 5.11 to 5.17:

Table 5.13: Confusion matrix of the MDCA (8/3) at SNR = 10dB

Simulated	Deduced Modulation Type							
Modulation Type	FSK2	FSK4	ASK2	ASK4	BPSK	QPSK	QAM16	QAM64
FSK2	**183**	10					7	
FSK4	4	**196**						
ASK2			**192**	8				
ASK4			4	**196**				
BPSK					**200**			
QPSK						**189**	3	8
QAM16							**106**	94
QAM64							54	**146**

Table 5.14: Confusion matrix of the MDCA (8/3) at SNR = 5dB

Simulated	Deduced Modulation Type							
Modulation Type	FSK2	FSK4	ASK2	ASK4	BPSK	QPSK	QAM16	QAM64
FSK2	**139**	30					31	
FSK4	25	**174**						
ASK2			**160**	40				
ASK4			21	**179**				
BPSK					**200**			
QPSK						**195**	5	
QAM16							**148**	52
QAM64							77	**123**

Table 5.15: Confusion matrix of the MDCA (8/3) at SNR = 0dB

Simulated	Deduced Modulation Type							
Modulation Type	FSK2	FSK4	ASK2	ASK4	BPSK	QPSK	QAM16	QAM64
FSK2	**142**	9				45	4	
FSK4	14	**184**					2	
ASK2			**154**	46				
ASK4			21	**179**				
BPSK					**200**			
QPSK	1					**192**	7	
QAM16							**114**	86
QAM64							59	**141**

Table 5.16: Confusion matrix of the MDCA (8/3) at SNR = -5dB

Simulated	Deduced Modulation Type							
Modulation Type	FSK2	FSK4	ASK2	ASK4	BPSK	QPSK	QAM16	QAM64
FSK2	**123**	6				34	25	12
FSK4	3	**197**					2	
ASK2			**146**	54				
ASK4			38	**162**				
BPSK					**199**		1	
QPSK	4					**143**	46	7
QAM16	1					20	**113**	66
QAM64	1					13	88	**98**

Table 5.17: Confusion matrix of the MDCA (8/3) at SNR = -10dB

Simulated	Deduced Modulation Type							
Modulation Type	FSK2	FSK4	ASK2	ASK4	BPSK	QPSK	QAM16	QAM64
FSK2	**44**	36				42	46	32
FSK4	28	**147**				2	14	9
ASK2			**147**	53				
ASK4			54	**146**				
BPSK					**199**			1
QPSK	6	2				**68**	43	81
QAM16	8	5				47	**49**	91
QAM64	9	2				43	52	**94**

- The performance of the classifier in distinguishing between QAM16 and QAM64 is not good even at high SNR levels.
- Classification is performed below the 10dB limit of the DMRA algorithm. The performance of the classifier remains good for the PSK family to 0dB SNR, and it degrads to an average of 80% for the other modulations.
- Most classification errors occurr within modulation families. With the exception of the FSK2, all other modulations almost remained free of cross-modulation errors to -5dB SNR.

Some achievements using no features other than the ones used in the DMRA are

- Classification is performed below the 10dB limit of the DMRA algorithm. Here, FSK4 and QPSK are largely well classified even at 0dB , while performance for the BPSK case remains almost perfect to -10dB.
- Two modulations, QAM16 and QAM64, which were not considered by Azzouz [1] are classified. Although not always perfectly identified as independent modulations, they are rarely confused for other than QAM modulations. Table 5.18 shows the instances where each of these modulations is identified to belong to the QAM family (i.e. either QAM16 or QAM64.)

Table 5.18: QAM64 and QAM16 classified as QAM (200 trials per modulation per SNR level)

	Signals deduced as a QAM						
SNR (dB)	-10	-5	0	5	10	15	20
QAM16	**140**	**179**	**200**	**200**	**200**	**200**	**199**
QAM64	**146**	**186**	**200**	**200**	**200**	**200**	**200**

5.3 Using only six modulation

It is difficult to compare the MDCA algorithm to DMRA, because the later is tailored specifically for the six modulations it classifies and for SNR levels larger than 10dB. However, the performance of the MDCA in a more restricted environment, such

as one that ignores the QAM modulation family, is investigated. Tables 5.19 and 5.20 show the results of the aforementioned implementation for the 15dB and 10dB SNR levels. The simulation results for the other SNR levels are included in Appendix B.

Table 5.19: Confusion matrix of the MDCA (6/3)at SNR = 15dB

Simulated	Deduced Modulation Type					
Modulation Type	FSK2	FSK4	ASK2	ASK4	BPSK	QPSK
FSK2	**179**	9				12
FSK4	13	**187**				
ASK2			**193**	7		
ASK4			3	**197**		
BPSK					**200**	
QPSK						**200**

Table 5.20: Confusion matrix of the MDCA (6/3) at SNR = 10dB

Simulated	Deduced Modulation Type					
Modulation Type	FSK2	FSK4	ASK2	ASK4	BPSK	QPSK
FSK2	**171**	17				12
FSK4	5	**195**				
ASK2			**193**	7		
ASK4			9	**191**		
BPSK					**200**	
QPSK						**200**

It is clear that the results provided in Tables 5.19 and 5.20 show an overall improvement and a minor setback in the case of FSK2.

5.4 Classification at different Dimensions

One of the objectives of this research is to show the advantages of classifying modulated signals in high dimensional spaces. In theory, the higher the classification space the better the results. Unfortunately, this statement only holds true if every used dimension adds a degree of distinction between classes. In our case, the use of five features allows classification in a space of a maximum of four dimensions, and hence, a simulated classification in one, two, three and four dimensional spaces is possible and also warranted. Table 5.21 provides the simulation results of a set of test

signals classified in two, three and four dimensional spaces at 10dB SNR. The results of the remaining SNR levels are included in Appendix B.

Table 5.21: Confusion matrix of the MDCA (6/3) at SNR = 10dB

Simulated	Deduced Modulation Type (2D classification)					
Modulation Type	FSK2	FSK4	ASK2	ASK4	BPSK	QPSK
FSK2	**181**	16				3
FSK4	10	**190**				
ASK2			**191**	9		
ASK4			8	**192**		
BPSK					**200**	
QPSK	2					**198**
	(3D classification)					
FSK2	**171**	17				12
FSK4	5	**195**				
ASK2			**193**	7		
ASK4			9	**191**		
BPSK					**200**	
QPSK						**200**
	(4D classification)					
FSK2	**177**	8				15
FSK4	5	**195**				
ASK2			**197**			
ASK4			12	**188**		
BPSK					**200**	
QPSK						**200**

Overall the results are quite similar across all three spaces, but in specific cases there is fluctuation in performance. In some cases classifications in 2D outperform those in 3D and 4D for the same modulation type. Consequently, one might be tempted to dismiss multi-dimensionality as a useful tool. However, to understand this unintuitive behavior the information used to generate the classification hyper-planes must be examined. Indeed, reviewing the eigenvalues of the H matrices (Equation 3.12), for which the eigenvectors that define the Fisher hyper-planes where classification is performed provides several clues. Knowing that the eigenvalues of the H matrix provide a relative estimate of the spread of input data points along their corresponding eigenvectors, one can conclude the following from Table 5.22:

Table 5.22: The eigenvalues of the 7 H matrices that provide the eigenvectors which in turn define the 7 level-one Fisher hyper-planes used in the (8/3) set of modulations

Ranked	The eigenvalues at SNR levels (dB)						
eigenvalues	-10	-5	0	5	10	15	20
λ_1	0.3394	0.6260	1.4785	1.9587	1.7598	1.3389	1.3546
λ_2	0.0193	0.0354	0.0253	0.0685	0.1264	0.0945	0.0923
λ_3	0.0047	0.0151	0.0189	0.0269	0.0371	0.0454	0.0490
λ_4	0.0001	0.0019	0.0092	0.0067	0.0102	0.0038	0.0059

- The first two eigenvalues are most significant in magnitude, while the third and the fourth eigenvalues are almost always marginal, which means that most of the separation between modulations is provided in the first two dimensions, while the last two provide less separation.
- In general the eigenvalues decrease in magnitude as the SNR level decreases, which explains the increasing collisions between modulations for low SNR

Thus the eigenvalues of the H matrices constructed from the given five dimensional data, which in turn correspond to the five features used, reveal that in most cases there is little to gain from performing classification in a space with more than two axes.

5.5 Top three possible modulations

The MDCA algorithm, as it is implemented in all simulations, provides not only the most likely modulation to match the signal of interest but also a list of all the modulations in its database in a ranked order of possible correct answers. Table 5.23 shows the frequency with which the signal of interest is identified within the top three possible answers.

Table 5.23: Confusion matrix of the MDCA with ranks(6/3) showing the frequency with which the actual modulation is classified within the top three choices.

Simulated Modulation Type	Ranked first						Ranked second						Ranked third						TOTAL
	FSK2	FSK4	ASK2	ASK4	BPSK	QPSK	FSK2	FSK4	ASK2	ASK4	BPSK	QPSK	FSK2	FSK4	ASK2	ASK4	BPSK	QPSK	
	SNR = 20dB																		
FSK2	**178**						**8**						**14**						**200**
FSK4		**193**						**7**						**0**					**200**
ASK2			**192**						**8**						**0**				**200**
ASK4				**198**						**2**						**0**			**200**
BPSK					**200**						**0**						**0**		**200**
QPSK						**197**						**2**						**1**	**200**
	SNR = 15dB																		
FSK2	**177**						**14**						**0**						**200**
FSK4		**181**						**19**						**0**					**200**
ASK2			**192**						**8**						**1**				**200**
ASK4				**191**						**7**						**0**			**200**
BPSK					**200**						**0**						**0**		**200**
QPSK						**199**						**0**						**1**	**200**
	SNR = 10dB																		
FSK2	**181**						**19**						**0**						**200**
FSK4		**190**						**10**						**0**					**200**
ASK2			**191**						**8**						**1**				**200**
ASK4				**192**						**8**						**0**			**200**
BPSK					**200**						**0**						**0**		**200**
QPSK						**198**						**2**						**0**	**200**
	SNR = 5dB																		
FSK2	**152**						**44**						**4**						**200**
FSK4		**168**						**24**						**8**					**200**
ASK2			**156**						**44**						**0**				**200**
ASK4				**174**						**26**						**0**			**200**
BPSK					**200**						**0**						**0**		**200**
QPSK						**197**						**2**						**1**	**200**
	SNR = 0dB																		
FSK2	**158**						**38**						**2**						**198**
FSK4		**185**						**13**						**2**					**200**
ASK2			**151**						**49**						**0**				**200**
ASK4				**176**						**0**						**0**			**200**
BPSK					**200**						**0**						**0**		**200**
QPSK						**193**						**0**						**0**	**200**
	SNR = -5dB																		
FSK2	**149**						**0**						**0**						**188**
FSK4		**199**						**1**						**0**					**200**
ASK2			**145**						**45**						**10**				**200**
ASK4				**163**						**37**						**0**			**200**
BPSK					**199**						**1**						**0**		**200**
QPSK						**185**						**0**						**0**	**200**
	SNR = -10dB																		
FSK2	**72**						**34**						**10**						**116**
FSK4		**143**						**33**						**14**					**190**
ASK2			**145**						**50**						**0**				**195**
ASK4				**151**						**49**						**0**			**200**
BPSK					**199**						**0**						**1**		**200**
QPSK						**177**						**0**						**22**	**199**

VI. Conclusion

6.1 Summary

Automatic modulation recognition has concerned a number of researchers for almost three decades, but the quest for a reliable and fast algorithm remains elusive. Work done so far has been limited in both scope and the diversity of approaches. Though the traditional signal processing and pattern recognition techniques have not yet been exhaustively evaluated, they also haven not been shown to work in environments other than controlled AWGN channels, which usually contain only one signal (the one being evaluated).

This research attempts to develop a platform that has the potential to perform as well as most of the work so far achieved while staying flexible for adaption to future breakthroughs in the field of modulation recognition. This objective is addressed by considering the nature of classification in a pattern recognition mode and focusing the working mechanisms of the platform to accommodate this nature. Classification of modulated signals with unknown parameters using pattern recognition schemes can not be conducted separately from the environment of the signals. Thus, for each possible environment one must expect a different signature and small or large deviations from the norm of the signals in consideration, so a thorough investigation of the behavior of signals in each environment is required. Thus the objective of the this effort is building unique and representative models of the signals in each plausible environment. None of the techniques proposed in the literature can characterize every possible modulation. Therefore, the capabilities and limitations of all features that can be used in a global classifier must be studied and integrated carefully in the classification process.

The MDCA algorithm answers these challenges with very simple approaches. First, the database of the algorithm is in reality a reference library of environments with different parameters which happen to contain modulated signals. Second, the MDCA's does not use any features directly in its core processes, but rather uses available features first in building the database and then in reference for classification.

The results of the implementations of the MDCA algorithm, are quite satisfactory. Indeed, with only five features which previously had a limited range of use, the MDCA extended useful classification to lower SNR levels and added to the modulation classes that are classifiable.

6.2 Recommendation for Future Research

Much more work is needed to improve the MDCA, and even more efforts are needed to improve the quality and size of its database. Some of the step that may be useful to achieve this goal are

- The existing features need to be surveyed and included as potential inputs to the MDCA, which can increase in the classification accuracy
- The quality of the surveyed features need to be assessed versus channel parameters for the considered modulations. The sensitivity of the features to changes in the channel parameters, and the robustness of the features (i.e., the consistency of the signatures they produce for the modulations as parameters change) is valuable information in need of integration in the classification process.
- The capability of the MDCA to classify modulations not available in its database as "unknown" saves time and effort. This can be achieved by declaring the empty areas in the Fisher hyper-planes as zones for unknown modulations, or by setting threshold Mahalanobis distances below which the projections can no longer be considered as possible modulations.

Appendix A. Special Case

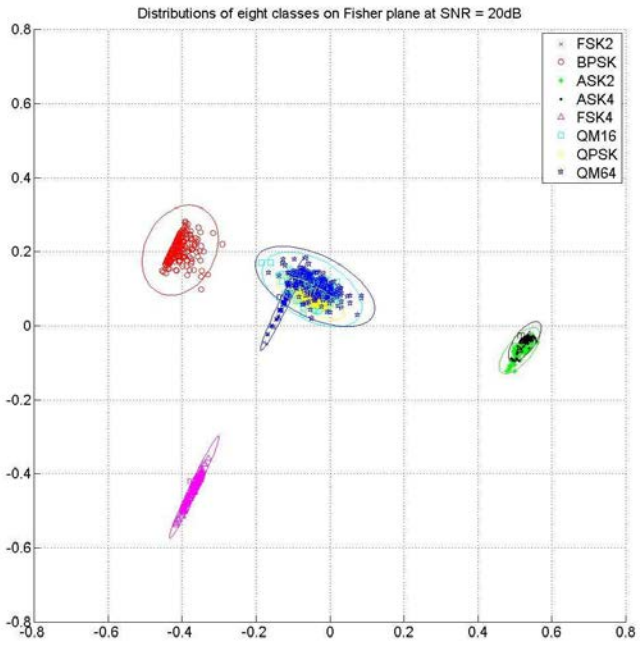
This Appendix covers the results of a simulation run of 12,800 test signals using a (8/3) model. There is a major difference between this set of simulations and the ones presented previously in that both the training and test signals are generated using the single carrier frequency of 5kHz. This procedure is used because it became apparent that the extracted features are sensitive to the frequency of the carrier signal, and hence it became necessary to consider the carrier frequency as one of the parameters to fix when generating the pages of the database. The results of this simulation are perfect for all the modulations, except ASK, at SNR levels down to 5dB. The relatively poorer performance of ASK is because it is one of the modulation types that lacks an implementation standard. In a simulation rerun where the ASK signals are implemented differently (magnitudes tripled) the results are perfect down to -10dB. The results of the second simulation are not presented because the changes applied to the ASK modulated signals fundamentally modified their signatures. Hence, it is inappropriate to compare the results of the second simulation to previous simulations.

The figures in this appendix are of the first level pages of all the SNR levels previously considered plus a -20dB implementation. The -20dB case is added because even at that level the ASK modulated signals exhibit a different behavior than the rest. While all constellation merge toward the center (i.e., behaving like pure noise and becoming undistinguishable from one another) the convergence of the ASK constellations is delayed and does not resolve until the -30dB SNR level is reached. All the 2D planes are shown in a fixed reference plane to highlight the convergence process toward the center, while the 3D hyper-planes are shown from angles that best present the separation between classes. Furthermore, one can clearly see improvements as the classification space is increased from two to four dimensions.

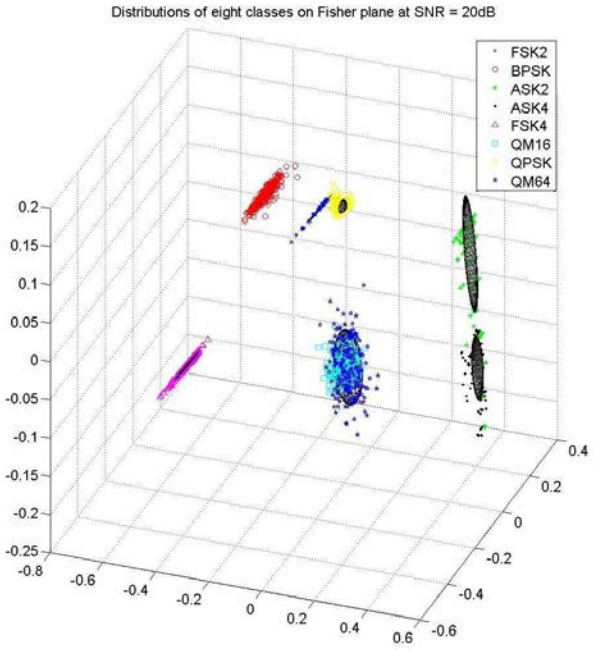
Overall, the results from this set of simulations are better than previous experiments.

Table A.1: Confusion matrix of the MDCA (8/3), single carrier model, at SNR = 20dB

Simulated	Deduced Modulation Type (2D classification)						
Modulation Type	FSK2	FSK4	ASK2	ASK4	BPSK	QPSK	QAM
FSK2	**199**					1	
FSK4		**200**					
ASK2			**190**	10			
ASK4			1	**199**			
BPSK					**200**		
QPSK	4					**195**	1
QAM16							**200**
QAM64							**200**
	(3D classification)						
FSK2	**200**						
FSK4		**200**					
ASK2			**190**	10			
ASK4			2	**198**			
BPSK					**200**		
QPSK						**200**	
QAM16							**200**
QAM64							**200**
	(4D classification)						
FSK2	**200**						
FSK4		**200**					
ASK2			**191**	9			
ASK4			2	**198**			
BPSK					**200**		
QPSK						**200**	
QAM16							**200**
QAM64							**200**



(a)

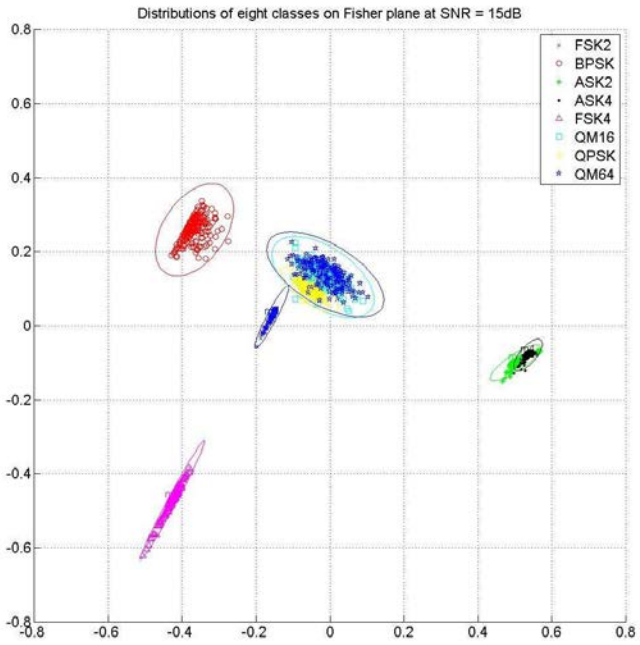


(b)

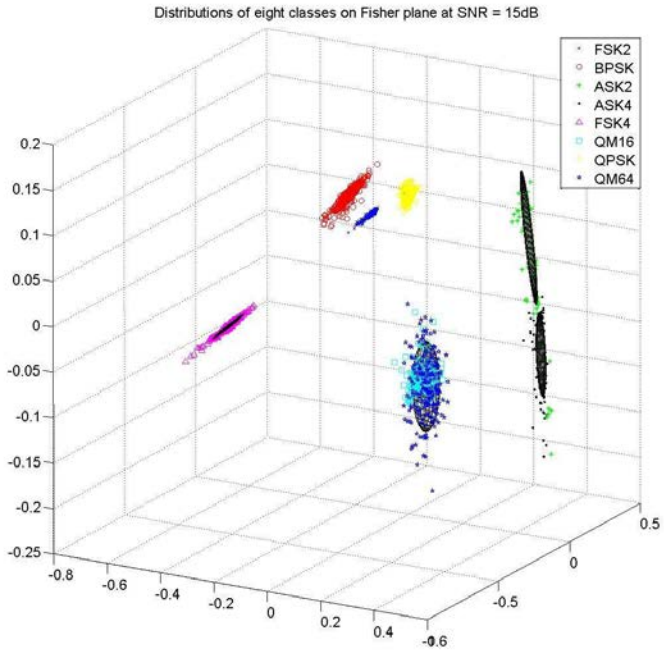
Figure A.1: First level page with SNR = 20dB

Table A.2: Confusion matrix of the MDCA (8/3), single carrier model, at SNR = 15dB

Simulated	Deduced Modulation Type (2D classification)						
Modulation Type	FSK2	FSK4	ASK2	ASK4	BPSK	QPSK	QAM
FSK2	**200**						
FSK4		**200**					
ASK2			**191**	9			
ASK4			1	**199**			
BPSK					**200**		
QPSK						**200**	
QAM16							**200**
QAM64							**200**
	(3D classification)						
FSK2	**200**						
FSK4		**200**					
ASK2			**192**	8			
ASK4			3	**197**			
BPSK					**200**		
QPSK						**200**	
QAM16							**200**
QAM64							**200**
	(4D classification)						
FSK2	**200**						
FSK4		**200**					
ASK2			**193**	7			
ASK4			2	**198**			
BPSK					**200**		
QPSK						**200**	
QAM16							**200**
QAM64							**200**



(a)

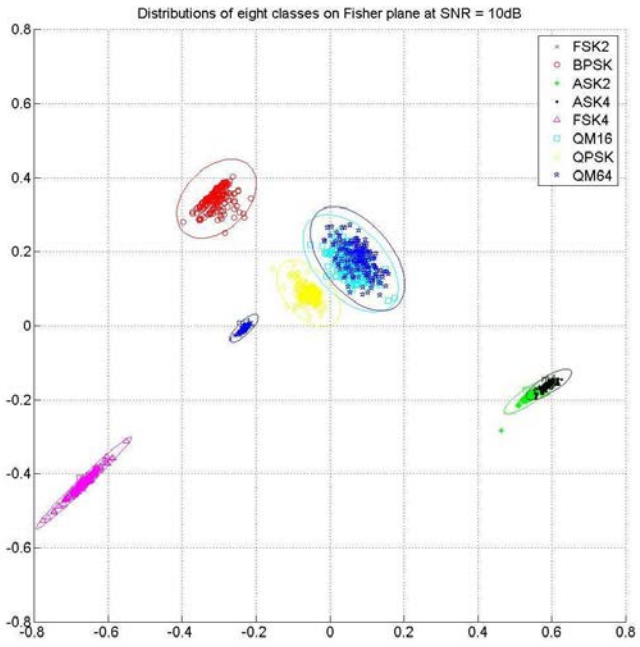


(b)

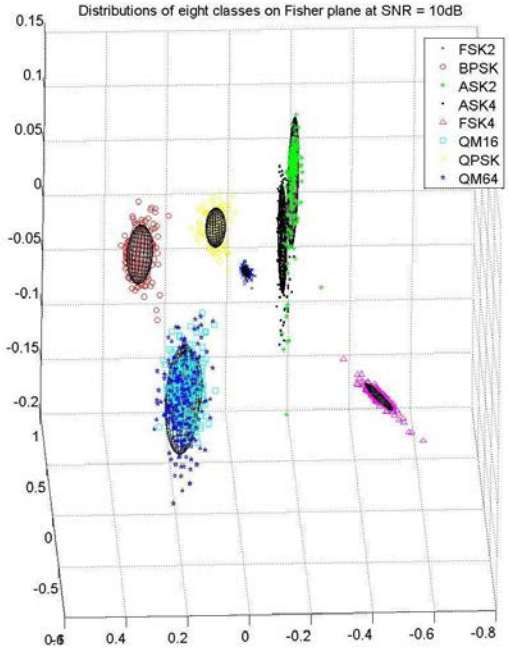
Figure A.2: First level page with SNR = 15dB

Table A.3: Confusion matrix of the MDCA (8/3), single carrier model, at SNR = 10dB

Simulated	Deduced Modulation Type (2D classification)						
Modulation Type	FSK2	FSK4	ASK2	ASK4	BPSK	QPSK	QAM
FSK2	**200**						
FSK4		**200**					
ASK2			**187**	13			
ASK4			11	**189**			
BPSK					**200**		
QPSK						**200**	
QAM16							**200**
QAM64							**200**
	(3D classification)						
FSK2	**200**						
FSK4		**200**					
ASK2			**191**	9			
ASK4			10	**190**			
BPSK					**200**		
QPSK						**200**	
QAM16							**200**
QAM64							**200**
	(4D classification)						
FSK2	**200**						
FSK4		**200**					
ASK2			**191**	9			
ASK4			8	**192**			
BPSK					**200**		
QPSK						**200**	
QAM16							**200**
QAM64							**200**



(a)

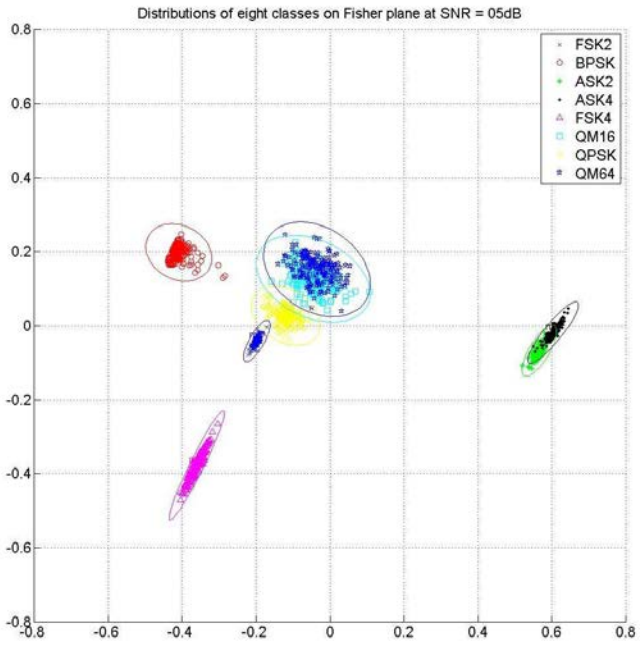


(b)

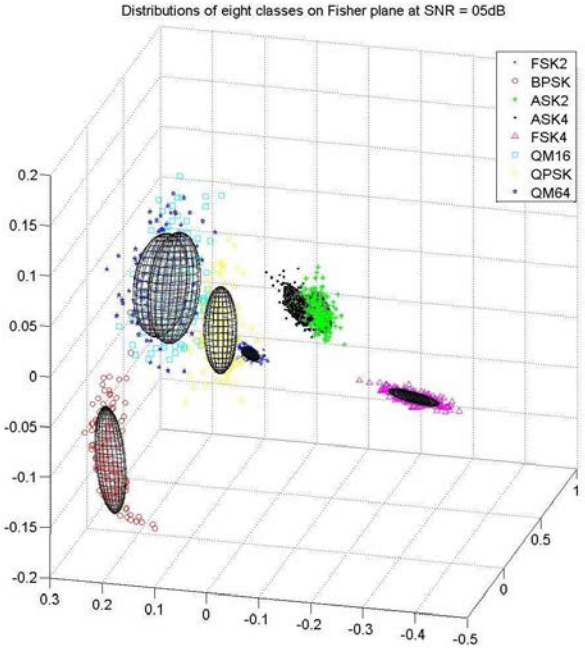
Figure A.3: First level page with SNR = 10dB

Table A.4: Confusion matrix of the MDCA (8/3), single carrier model, at SNR = 5dB

Simulated	Deduced Modulation Type (2D classification)						
Modulation Type	FSK2	FSK4	ASK2	ASK4	BPSK	QPSK	QAM
FSK2	**200**						
FSK4		**200**					
ASK2			**187**	13			
ASK4			9	**191**			
BPSK					**200**		
QPSK						**200**	
QAM16							**200**
QAM64							**200**
	(3D classification)						
FSK2	**200**						
FSK4		**200**					
ASK2			**187**	13			
ASK4			8	**192**			
BPSK					**200**		
QPSK						**200**	
QAM16							**200**
QAM64							**200**
	(4D classification)						
FSK2	**200**						
FSK4		**200**					
ASK2			**192**	8			
ASK4			13	**187**			
BPSK					**200**		
QPSK						**200**	
QAM16							**200**
QAM64							**200**



(a)

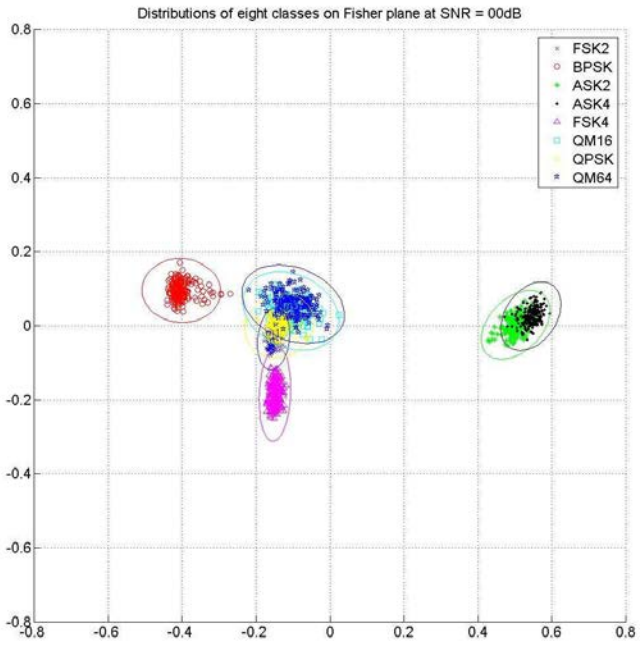


(b)

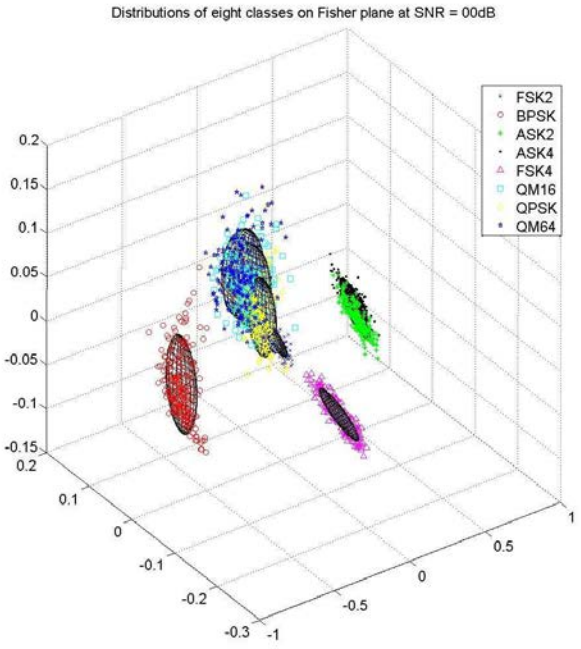
Figure A.4: First level page with SNR = 5dB

Table A.5: Confusion matrix of the MDCA (8/3), single carrier model, at SNR = 0dB

Simulated	Deduced Modulation Type (2D classification)						
Modulation Type	FSK2	FSK4	ASK2	ASK4	BPSK	QPSK	QAM
FSK2	**166**	1				33	
FSK4		**200**					
ASK2			**169**	31			
ASK4			30	**170**			
BPSK					**200**		
QPSK	14					**182**	4
QAM16							**200**
QAM64							**200**
	(3D classification)						
FSK2	**158**	1				41	
FSK4		**200**					
ASK2			**172**	28			
ASK4			28	**172**			
BPSK					**200**		
QPSK	3					**188**	9
QAM16							**200**
QAM64							**200**
	(4D classification)						
FSK2	**154**	1				44	1
FSK4		**200**					
ASK2			**171**	29			
ASK4			26	**174**			
BPSK					**200**		
QPSK	1					**188**	11
QAM16							**200**
QAM64							**200**



(a)

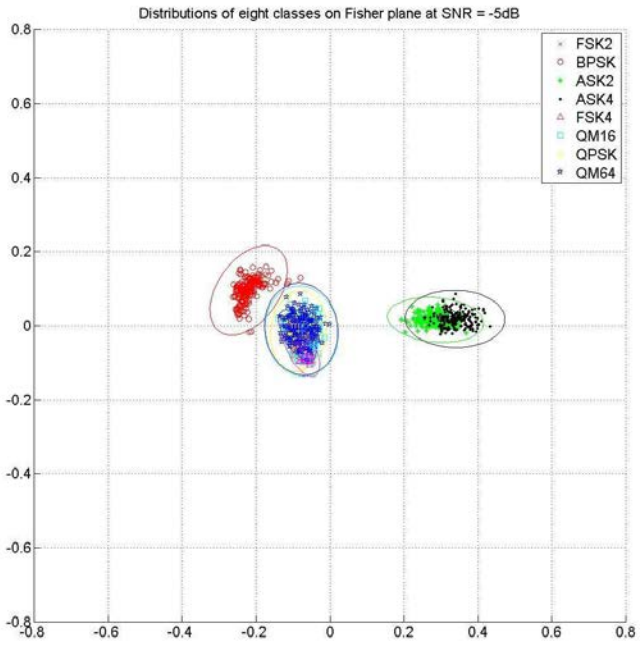


(b)

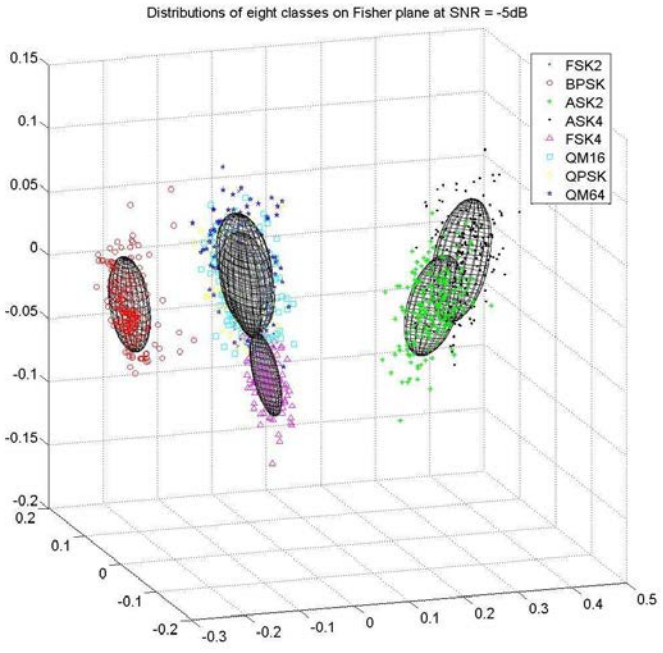
Figure A.5: First level page with SNR = 0dB

Table A.6: Confusion matrix of the MDCA (8/3), single carrier model, at SNR = -5dB

Simulated	Deduced Modulation Type (2D classification)						
Modulation Type	FSK2	FSK4	ASK2	ASK4	BPSK	QPSK	QAM
FSK2	**46**	7				95	52
FSK4	4	**159**				12	25
ASK2			**151**	49			
ASK4			39	**161**			
BPSK					**200**		
QPSK	9					**115**	76
QAM16	1	4				13	**182**
QAM64	1					4	**195**
	(3D classification)						
FSK2	**74**	11				66	49
FSK4	11	**173**				16	
ASK2			**150**	50			
ASK4			34	**166**			
BPSK					**199**		1
QPSK	9	1				**106**	84
QAM16		1				10	**189**
QAM64	1						**199**
	(4D classification)						
FSK2	**58**	11				68	63
FSK4	10	**173**				17	
ASK2			**139**	61			
ASK4			29	**171**			
BPSK					**199**		1
QPSK	7	1				**95**	97
QAM16						10	**190**
QAM64							**200**



(a)

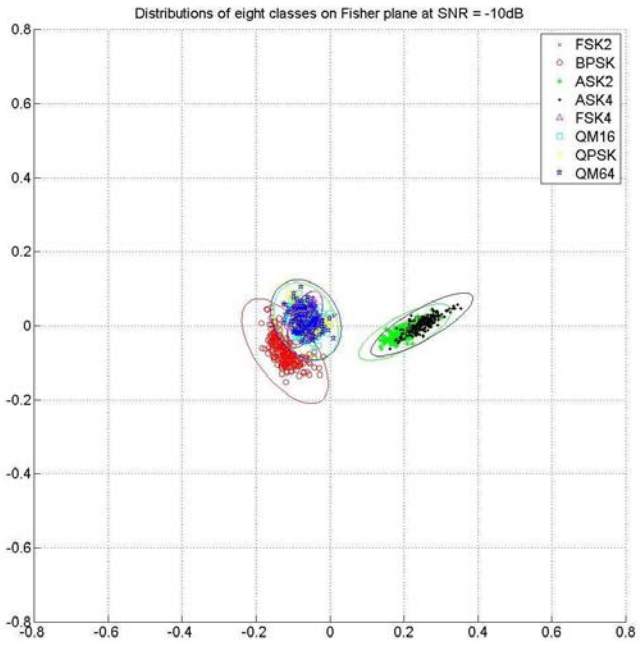


(b)

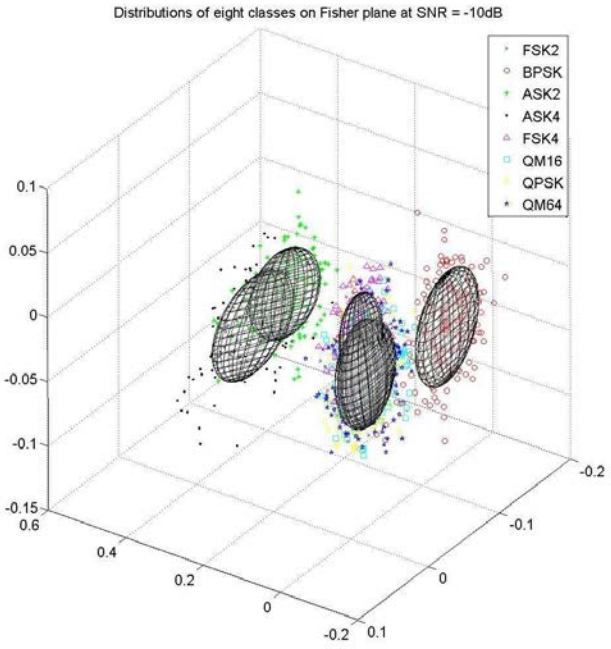
Figure A.6: First level page with SNR = -5dB

Table A.7: Confusion matrix of the MDCA (8/3), single carrier model, at SNR = -10dB

Simulated	Deduced Modulation Type (2D classification)						
Modulation Type	FSK2	FSK4	ASK2	ASK4	BPSK	QPSK	QAM
FSK2	**10**	13			1	57	119
FSK4	18	**28**				48	106
ASK2			**155**	45			
ASK4			51	**149**			
BPSK					**198**		2
QPSK	3				1	**71**	125
QAM16	1	4				51	**124**
QAM64	2	4			1	47	**146**
	(3D classification)						
FSK2	**11**	44			1	40	104
FSK4	8	**124**				13	57
ASK2			**157**	43			
ASK4			53	**147**			
BPSK					**198**		2
QPSK	6	9			1	**53**	131
QAM16	3	15				45	**137**
QAM64	1	11			1	39	**148**
	(4D classification)						
FSK2	**10**	43			1	37	109
FSK4	7	**122**				18	53
ASK2			**158**	42			
ASK4			55	**145**			
BPSK					**199**		1
QPSK	3	7				**55**	135
QAM16	2	14				45	**139**
QAM64	1	11			1	41	**146**



(a)

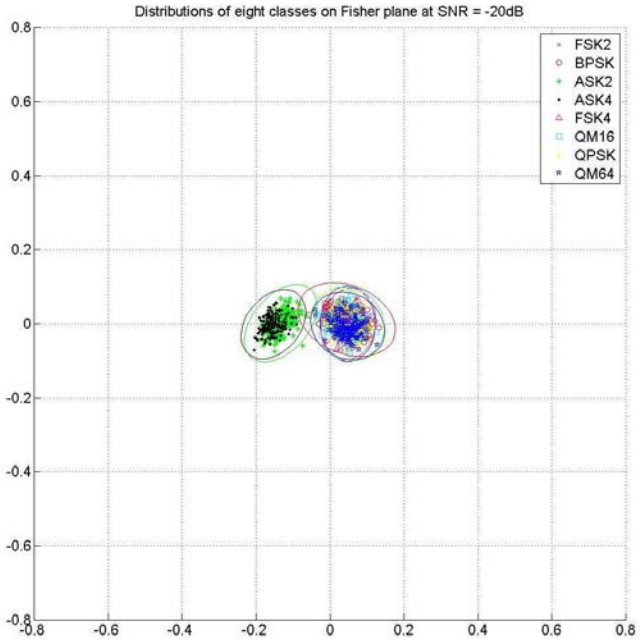


(b)

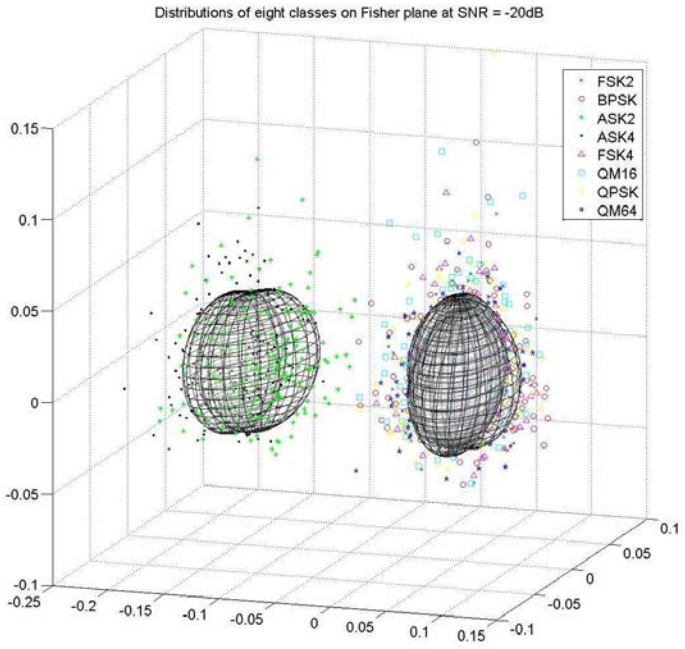
Figure A.7: First level page with SNR = -10dB

Table A.8: Confusion matrix of the MDCA (8/3), single carrier model, at SNR = -20dB

Simulated	Deduced Modulation Type (2D classification)						
Modulation Type	FSK2	FSK4	ASK2	ASK4	BPSK	QPSK	QAM
FSK2	**0**	4			74	31	91
FSK4		**2**			66	44	88
ASK2			**143**	56	1		
ASK4			84	**116**			
BPSK		2			**122**	31	45
QPSK		3			67	**44**	86
QAM16	1	3			79	36	**81**
QAM64	1	1			89	29	**74**
	(3D classification)						
FSK2	**1**	2			78	41	78
FSK4	1	**1**			70	42	86
ASK2			**141**	58	1		
ASK4			88	**112**			
BPSK		2			**132**	28	38
QPSK		1			77	**52**	70
QAM16					82	47	**71**
QAM64					95	46	**59**
	(4D classification)						
FSK2	**0**	4			83	39	74
FSK4		**1**			78	46	75
ASK2			**138**	61	1		
ASK4			84	**116**			
BPSK		1			**141**	26	32
QPSK		5			81	**52**	62
QAM16		5			93	45	**57**
QAM64		2			99	37	**62**



(a)



(b)

Figure A.8: First level page with SNR = 20dB

Appendix B. Additional Results

More of the results of the simulations introduced in Chapter 5 are provided in this appendix.

B.1 Two layer model with ranks 6 and 3.

Tables A.1 to A.5 show the results for the model built around two layers of ranks six and three.

Table B.1: Confusion matrix of the MDCA (6/3) at SNR = 20dB

Simulated	Deduced Modulation Type					
Modulation Type	FSK2	FSK4	ASK2	ASK4	BPSK	QPSK
FSK2	**174**	4				22
FSK4	6	**194**				
ASK2			**192**	8		
ASK4			3	**197**		
BPSK					**200**	
QPSK						**200**

Table B.2: Confusion matrix of the MDCA (6/3)at SNR = 5dB

Simulated	Deduced Modulation Type					
Modulation Type	FSK2	FSK4	ASK2	ASK4	BPSK	QPSK
FSK2	**134**	34				32
FSK4	22	**175**				3
ASK2			**154**	46		
ASK4			26	**174**		
BPSK					**200**	
QPSK						**200**

B.2 Two layer model with ranks 6 and 2.

This section presents the simulation results of a two layer model with ranks six and two. They are intended to show that having fewer modulation in lower levels can improve classification results. Tables B.6 to B.12 show that the substituting the 3-class pages with 2-class pages in the second layer does not significantly improve

Table B.3: Confusion matrix of the MDCA (6/3) at SNR = 0dB

Simulated	Deduced Modulation Type					
Modulation Type	FSK2	FSK4	ASK2	ASK4	BPSK	QPSK
FSK2	**145**	14				41
FSK4	12	**188**				2
ASK2			**148**	52		
ASK4			25	**175**		
BPSK					**200**	
QPSK	1					**199**

Table B.4: Confusion matrix of the MDCA (6/3) at SNR = -5dB

Simulated	Deduced Modulation Type					
Modulation Type	FSK2	FSK4	ASK2	ASK4	BPSK	QPSK
FSK2	**143**	8				49
FSK4	1	**199**				
ASK2			**135**	65		
ASK4			35	**165**		
BPSK					**199**	1
QPSK	6					**194**

Table B.5: Confusion matrix of the MDCA (6/3) at SNR = -10dB

Simulated	Deduced Modulation Type					
Modulation Type	FSK2	FSK4	ASK2	ASK4	BPSK	QPSK
FSK2	**79**	30				91
FSK4	50	**141**				9
ASK2			**142**	58		
ASK4			46	**154**		
BPSK					**199**	1
QPSK	24					**175**

Table B.6: Confusion matrix of the MDCA with two layers (6/2)at SNR = 20dB

Simulated	Deduced Modulation Type					
Modulation Type	FSK2	FSK4	ASK2	ASK4	BPSK	QPSK
FSK2	**169**	5				26
FSK4	5	**193**				2
ASK2			**193**	7		
ASK4			2	**198**		
BPSK					**200**	
QPSK						**200**

Table B.7: Confusion matrix of the MDCA with two layers (6/2)at SNR = 15dB

Simulated	Deduced Modulation Type					
Modulation Type	FSK2	FSK4	ASK2	ASK4	BPSK	QPSK
FSK2	**187**	3				10
FSK4	11	**189**				
ASK2			**196**	4		
ASK4			10	**190**		
BPSK					**200**	
QPSK	1					**199**

Table B.8: Confusion matrix of the MDCA with two layers (6/2)at SNR = 10dB

Simulated	Deduced Modulation Type					
Modulation Type	FSK2	FSK4	ASK2	ASK4	BPSK	QPSK
FSK2	**176**	8				16
FSK4	6	**194**				2
ASK2			**195**	5		
ASK4			7	**193**		
BPSK					**200**	
QPSK						**200**

Table B.9: Confusion matrix of the MDCA with two layers (6/2)at SNR = 5dB

Simulated	Deduced Modulation Type					
Modulation Type	FSK2	FSK4	ASK2	ASK4	BPSK	QPSK
FSK2	**124**	32				44
FSK4	24	**175**				1
ASK2			**154**	46		
ASK4			23	**177**		
BPSK					**200**	
QPSK	1					**199**

Table B.10: Confusion matrix of the MDCA with two layers (6/2)at SNR = 0dB

Simulated	Deduced Modulation Type					
Modulation Type	FSK2	FSK4	ASK2	ASK4	BPSK	QPSK
FSK2	**144**	10				46
FSK4	15	**185**				2
ASK2			**149**	51		
ASK4			19	**181**		
BPSK					**200**	
QPSK	3					**197**

Table B.11: Confusion matrix of the MDCA with two layers (6/2)at SNR = -5dB

Simulated	Deduced Modulation Type					
Modulation Type	FSK2	FSK4	ASK2	ASK4	BPSK	QPSK
FSK2	**139**	7				54
FSK4	1	**199**				2
ASK2			**141**	59		
ASK4			42	**158**		
BPSK					**199**	1
QPSK	4					**196**

Table B.12: Confusion matrix of the MDCA with two layers (6/2) at SNR = -10dB

Simulated	Deduced Modulation Type					
Modulation Type	FSK2	FSK4	ASK2	ASK4	BPSK	QPSK
FSK2	**69**	30				101
FSK4	47	**140**				13
ASK2			**143**	57		
ASK4			46	**154**		
BPSK					**199**	1
QPSK	20	1				**179**

results. The improvements are marginal in most cases, while the remaining cases show no improvement or even some degradation. However, there is a reduction of the within family errors (modulations of same family, for example FSK2 and FSK4.)

B.3 Classification using different dimensions

This section provides the remainder of results from Section 5.4.

Table B.13: Confusion matrix of the MDCA (6/3) at SNR = 20dB

Simulated	Deduced Modulation Type (2D classification)					
Modulation Type	FSK2	FSK4	ASK2	ASK4	BPSK	QPSK
FSK2	**178**	11				11
FSK4	7	**193**				
ASK2			**192**	8		
ASK4			2	**198**		
BPSK					**200**	
QPSK	1	2				**197**
	(3D classification)					
FSK2	**174**	4				22
FSK4	6	**194**				
ASK2			**192**	8		
ASK4			3	**197**		
BPSK					**200**	
QPSK						**200**
	(4D classification)					
FSK2	**174**	3				23
FSK4	5	**194**				1
ASK2			**192**	8		
ASK4			3	**197**		
BPSK					**200**	
QPSK						**200**

Table B.14: Confusion matrix of the MDCA (6/3) at SNR = 15dB

Simulated	Deduced Modulation Type (2D classification)					
Modulation Type	FSK2	FSK4	ASK2	ASK4	BPSK	QPSK
FSK2	**177**	17				6
FSK4	19	**181**				
ASK2			**192**	8		
ASK4			9	**191**		
BPSK					**200**	
QPSK	1					**199**
	(3D classification)					
FSK2	**179**	9				12
FSK4	13	**187**				
ASK2			**193**	7		
ASK4			3	**197**		
BPSK					**200**	
QPSK						**200**
	(4D classification)					
FSK2	**180**	4				16
FSK4	8	**192**				
ASK2			**194**	6		
ASK4			7	**193**		
BPSK					**200**	
QPSK						**200**

Table B.15: Confusion matrix of the MDCA (6/3) at SNR = 5dB

Simulated	Deduced Modulation Type (2D classification)					
Modulation Type	FSK2	FSK4	ASK2	ASK4	BPSK	QPSK
FSK2	**152**	28				20
FSK4	30	**168**				2
ASK2			**156**	44		
ASK4			26	**174**		
BPSK					**200**	
QPSK	3					**197**
	(3D classification)					
FSK2	**134**	34				32
FSK4	22	**175**				3
ASK2			**154**	46		
ASK4			26	**174**		
BPSK					**200**	
QPSK						**200**
	(4D classification)					
FSK2	**136**	31				33
FSK4	22	**175**				3
ASK2			**155**	45		
ASK4			25	**175**		
BPSK					**200**	
QPSK						**200**

Table B.16: Confusion matrix of the MDCA (6/3) at SNR = 0dB

Simulated	Deduced Modulation Type (2D classification)					
Modulation Type	FSK2	FSK4	ASK2	ASK4	BPSK	QPSK
FSK2	**158**	14				28
FSK4	15	**185**				
ASK2			**151**	49		
ASK4			24	**176**		
BPSK					**200**	
QPSK	7					**193**
	(3D classification)					
FSK2	**145**	14				41
FSK4	12	**188**				
ASK2			**148**	52		
ASK4			25	**175**		
BPSK					**200**	
QPSK	1					**199**
	(4D classification)					
FSK2	**144**	14				42
FSK4	13	**187**				
ASK2			**148**	52		
ASK4			25	**175**		
BPSK					**200**	
QPSK	1					**199**

Table B.17: Confusion matrix of the MDCA (6/3) at SNR = -5dB

Simulated	Deduced Modulation Type (2D classification)					
Modulation Type	FSK2	FSK4	ASK2	ASK4	BPSK	QPSK
FSK2	**149**	7				44
FSK4	1	**199**				
ASK2			**145**	55		
ASK4			37	**163**		
BPSK					**199**	1
QPSK	15					**185**
	(3D classification)					
FSK2	**143**	8				49
FSK4	1	**199**				
ASK2			**135**	65		
ASK4			35	**165**		
BPSK					**199**	1
QPSK	6					**194**
	(4D classification)					
FSK2	**144**	8				48
FSK4	1	**199**				
ASK2			**138**	62		
ASK4			35	**165**		
BPSK					**199**	1
QPSK	6					**194**

Table B.18: Confusion matrix of the MDCA (6/3) at SNR = -10dB

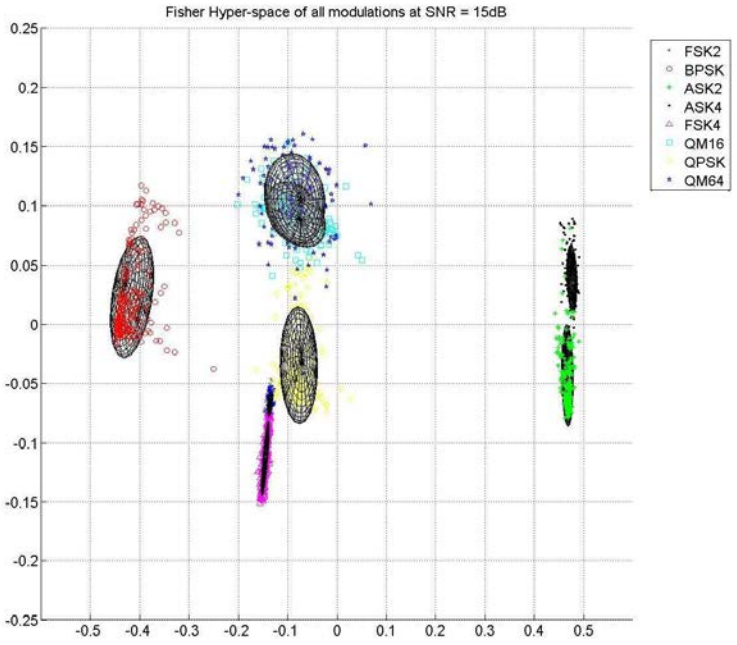
Simulated	Deduced Modulation Type (2D classification)					
Modulation Type	FSK2	FSK4	ASK2	ASK4	BPSK	QPSK
FSK2	**72**	33				95
FSK4	46	**143**				11
ASK2			**145**	55		
ASK4			49	**151**		
BPSK					**199**	1
QPSK	21	1			1	**177**
	(3D classification)					
FSK2	**79**	30				91
FSK4	50	**141**				
ASK2			**142**	58		
ASK4			46	**154**		
BPSK					**199**	1
QPSK	24	1				**175**
	(4D classification)					
FSK2	**78**	30				92
FSK4	50	**141**				
ASK2			**141**	59		
ASK4			46	**154**		
BPSK					**199**	1
QPSK	23	1				**176**

Appendix C. Additional Figures

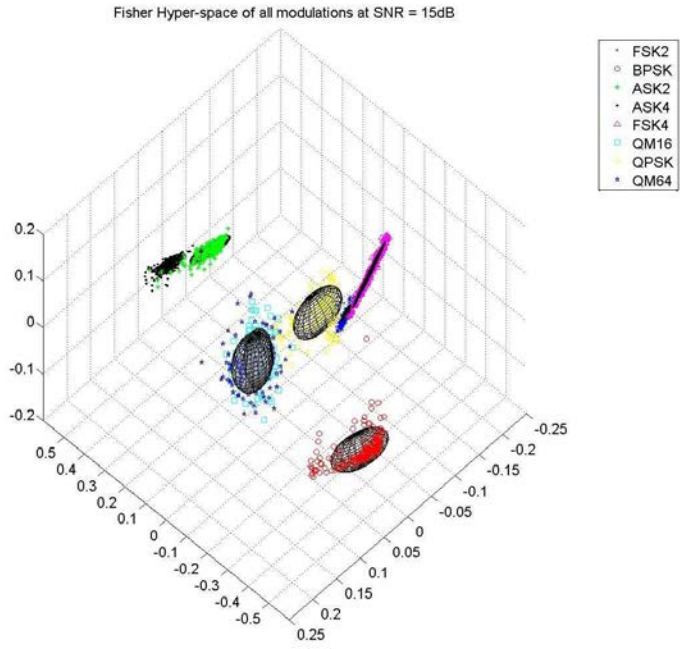
To understand the results provided in the tables of Chapter V, visualization of the model generated (.i.e, the pages of the database) can give insight into the behavior of the modulations, or more appropriately the features extracted from them as a function of the SNR levels in their respective Fisher hyper-planes. This exercise is crucial because it allows the user to identify weaknesses of the model prior to any testing or field use. The most important elements are the instances where collisions between classes occur. Identifying these overlaps assist in

- **Assessing the quality of the features used in the model**
- **Identifying when and for what modulations a different set of features are necessary for classification**
- **Predicting the reliability of the model**
- **Identifying fixes for problem cases**

Only the Fisher hyper-planes of the first level pages from each of the models simulated here are presented in this appendix.

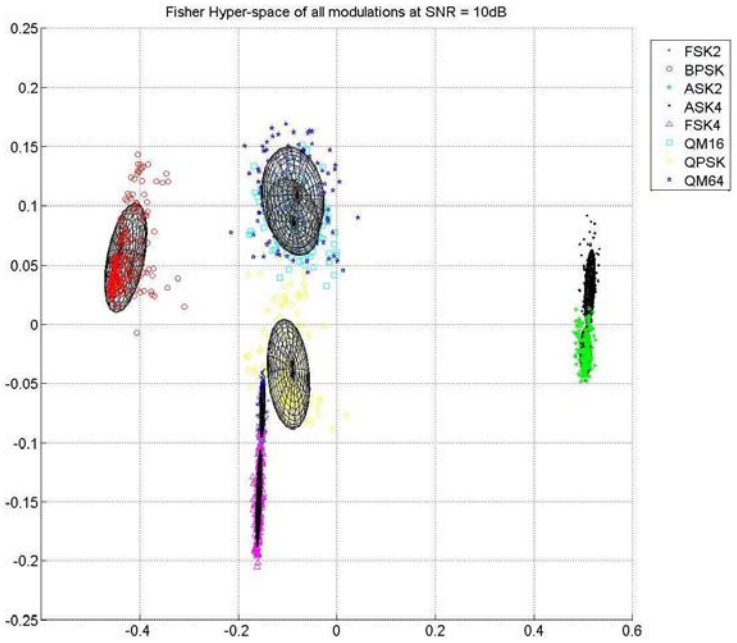


(a)

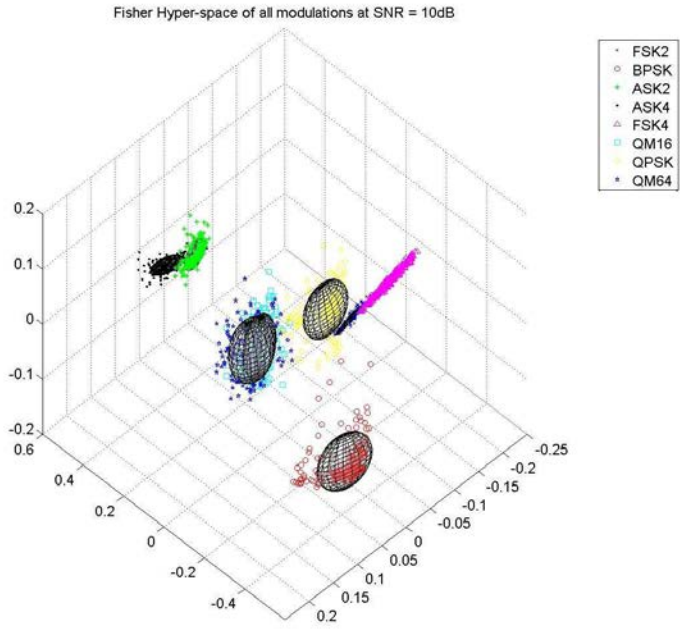


(b)

Figure C.1: First level page with SNR = 15dB

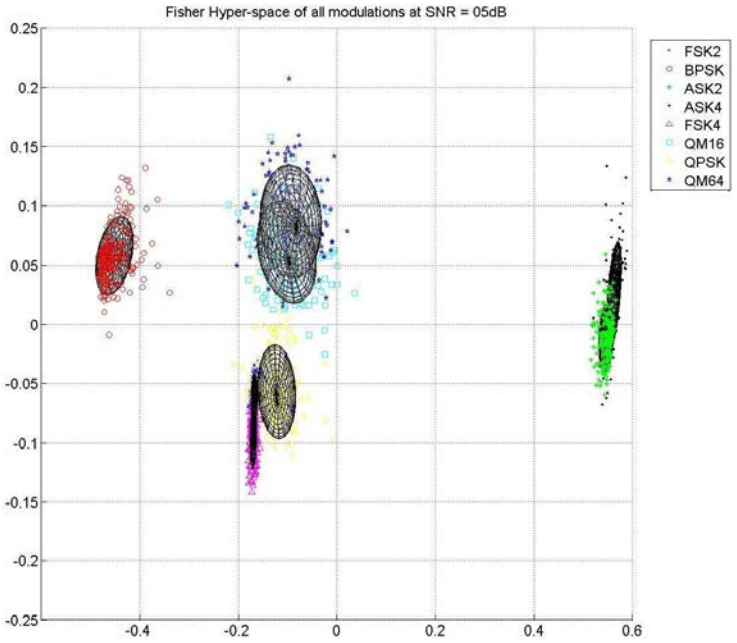


(a)

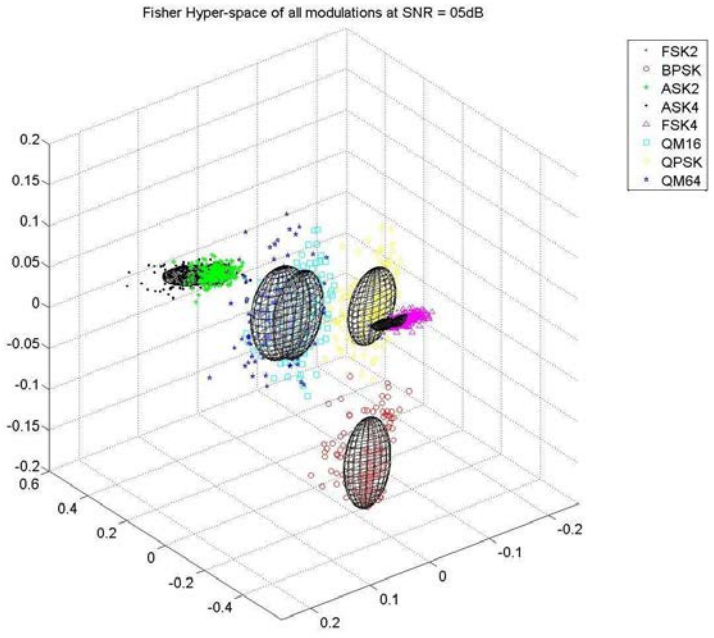


(b)

Figure C.2: First level page with SNR = 10dB

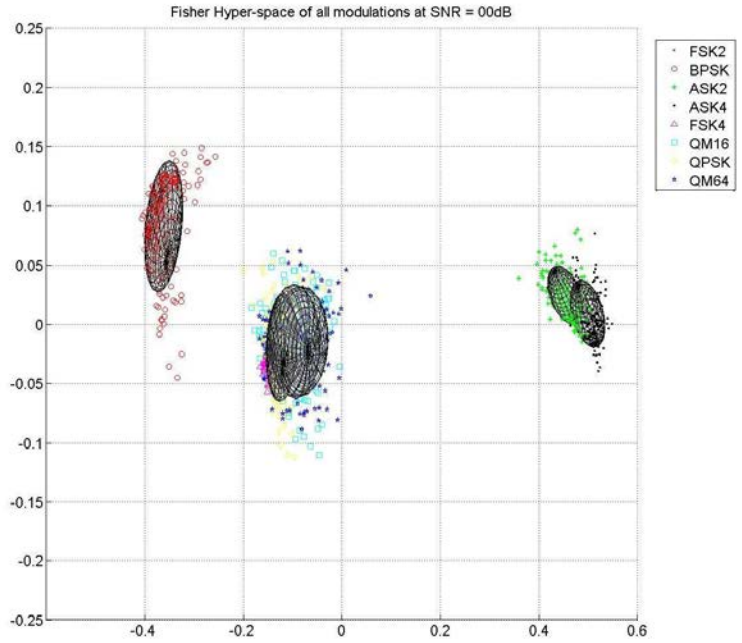


(a)

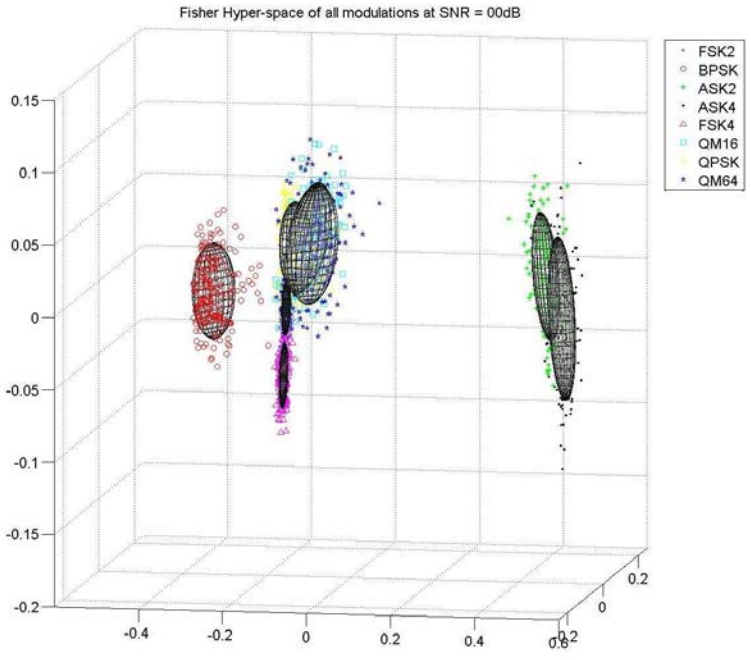


(b)

Figure C.3: First level page with SNR = 05dB

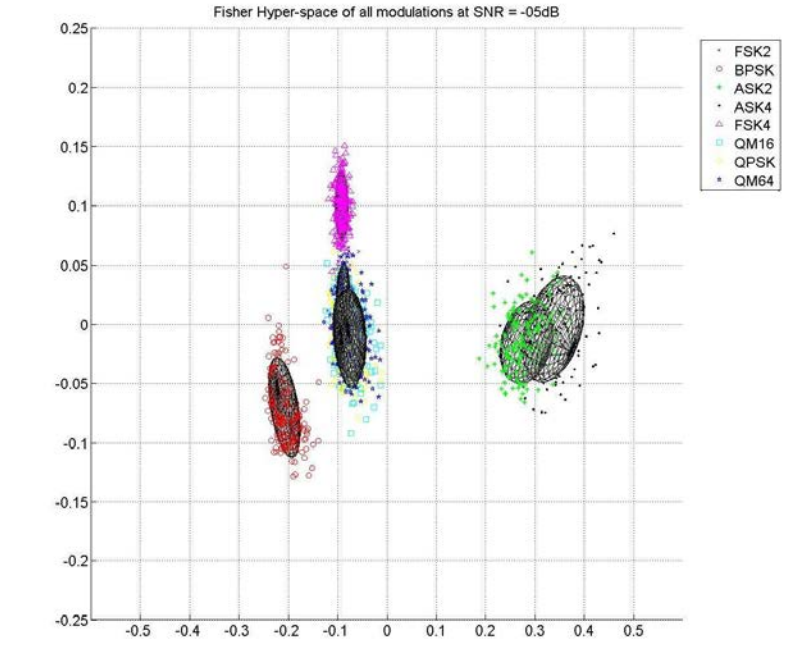


(a)

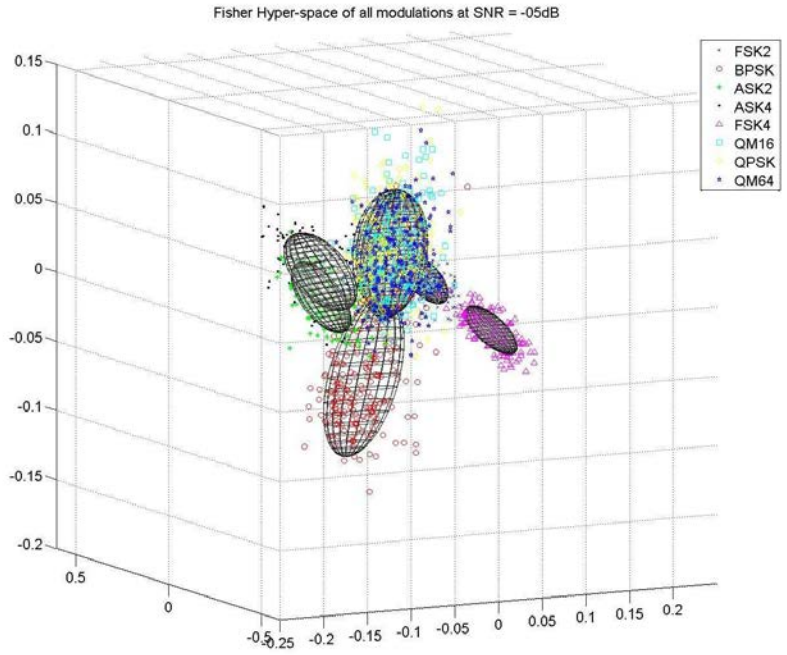


(b)

Figure C.4: First level page with SNR = 0dB

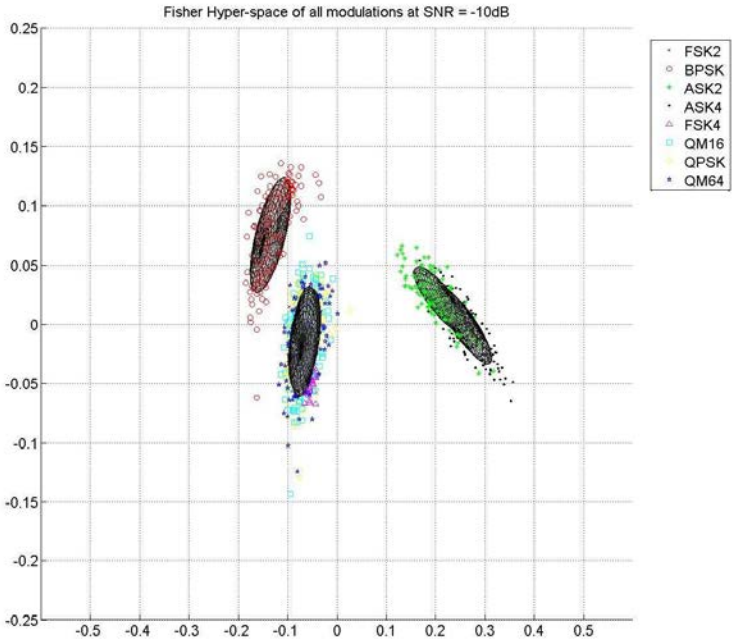


(a)

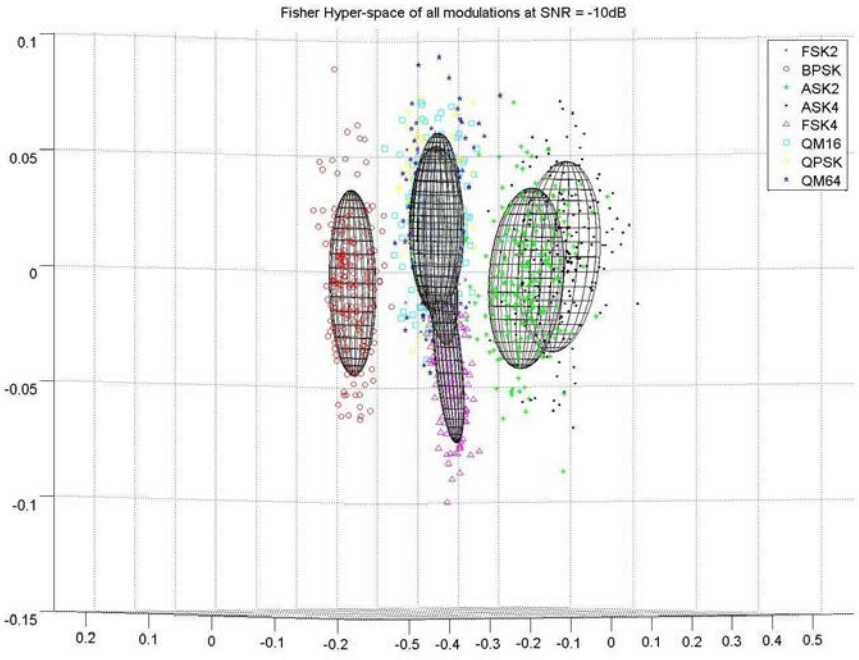


(b)

Figure C.5: First level page with SNR = -5dB



(a)



(b)

Figure C.6: First level page with SNR = -10dB

This appendix includes two examples of the Matlab® code used in this thesis.

D.1 First Example

The SNR estimation module is a set of four simple branching tree processes. Due to the simplicity of the process, it is more efficient to provide the function code than to graphically describe each branching tree.

Listing D.1: The Matlab® implementation of the SNR estimation process. (appendix4/estimateSNR.m)

```
1
function [X] = estimate_SNR(features)
%+-------------------------------------------------------+
%| function estimates the SNR level ranges of a signal for |
%| each input feature.                                   |
6 %| input: "features" struct contains                     |
%|          .absolute amplitude feature                  |
%|          .absolute frequency feature                  |
%|          .absolute phase feature                      |
%|          .direct phase feature                        |
11 %| output: "X" struct contains four vectors each is (1x7)  |
%|         corresponding to the 7 SNR ranges (1: possible, |
%|         0: not possible) as derived from each of the    |
%|         given features.                               |
%| version: 1.7                                          |
16 %+-------------------------------------------------------+

%==================== Absolute phase range =================
if ( 1.47 < features.abs_phase) || (features.abs_phase < 0.055)
    X.AP = [0 0 0 0 0 0 1];
21 elseif ( 1.415 < features.abs_phase) || (features.abs_phase < ...
   0.09)
    X.AP = [0 0 0 0 0 1 1];
elseif ( 1.315 < features.abs_phase) || (features.abs_phase < ...
   0.17)
    X.AP = [0 0 0 0 1 1 1];
elseif ( 1.2 < features.abs_phase) || (features.abs_phase < 0.3)
26     X.AP = [0 0 0 1 1 1 1];
elseif ( 1.06 < features.abs_phase) || (features.abs_phase < 0.56)
    X.AP = [0 0 1 1 1 1 1];
elseif ( 0.98 < features.abs_phase) || (features.abs_phase < 0.76)
    X.AP = [0 1 1 1 1 1 1];
31 else
    X.AP = [1 1 1 1 1 1 1];
end

%================= Absolute amplitude range =================
```

```
if ( 0.3115 < features.abs_amp)
    X.AA = [1 0 0 0 0 0 0];
elseif ( 0.304 < features.abs_amp)
    X.AA = [1 1 0 0 0 0 0];
    if ( 0.306 > features.abs_amp)
        X.AA = [0 1 0 0 0 0 0];
    end
elseif ( 0.287 < features.abs_amp)
    X.AA = [1 1 1 0 0 0 0];
    if ( 0.304 > features.abs_amp)
        X.AA = [0 1 1 0 0 0 0];
    elseif ( 0.298 > features.abs_amp)
        X.AA = [0 0 1 0 0 0 0];
    end
elseif ( 0.26 < features.abs_amp)
    X.AA = [1 1 1 1 0 0 0];
    if ( 0.287 > features.abs_amp)
        X.AA = [0 0 1 1 0 0 0];
    elseif ( 0.27 > features.abs_amp)
        X.AA = [0 0 0 1 0 0 0];
    end
elseif ( 0.206 < features.abs_amp)
    X.AA = [1 1 1 1 1 1 1];
    if ( 0.26 > features.abs_amp)
        X.AA = [0 0 0 1 1 1 1];
        if ( 0.243 < features.abs_amp)
            X.AA = [0 0 0 1 1 0 0];
        end
    end
elseif ( 0.206 > features.abs_amp)
    X.AA = [0 0 0 0 1 1 1];
    if ( 0.13 > features.abs_amp)
        X.AA = [0 0 0 0 0 1 1];
    elseif ( 0.08 > features.abs_amp)
        X.AA = [0 0 0 0 0 0 1];
    end

end

%==================== Direct phase range ====================
if ( 2.2 < features.dir_phase) || (features.dir_phase < 0.2)
    X.DP = [0 0 0 0 1 1 1];
elseif ( 2.12 < features.dir_phase) || (features.dir_phase < 0.4)
    X.DP = [0 0 0 1 1 1 1];
elseif ( 2 < features.dir_phase) || (features.dir_phase < 0.75)
    X.DP = [0 0 1 1 1 1 1];
elseif ( 1.86 < features.dir_phase) || (features.dir_phase < 1.2)
    X.DP = [0 1 1 1 1 1 1];
else
    X.DP = [1 1 1 1 1 1 1];
end
```

```
%=============== Absolute afrequency range ==================
if  (features.abs_freq < 7)
    X.AF = [0 0 0 0 0 0 1];
elseif  (features.abs_freq < 13)
    X.AF = [0 0 0 0 0 1 1];
elseif  (features.abs_freq < 32)
    X.AF = [0 0 0 0 1 1 1];
elseif (features.abs_freq < 60)
    X.AF = [0 0 0 1 1 1 1];
    if ( 40 < features.abs_freq)
        X.AF = [0 0 0 1 0 0 0];
    end

elseif (features.abs_freq > 60)
    X.AF = [1 1 1 0 0 0 0];
    if (features.abs_freq < 88)
        X.AF = [0 0 1 0 0 0 0];
    elseif (features.abs_freq < 91)
        X.AF = [0 1 1 0 0 0 0];
    elseif (features.abs_freq > 91)
        X.AF = [1 1 0 0 0 0 0];
        if ( 100 < features.abs_freq)
            X.AF = [0 1 0 0 0 0 0];
        end
    end

end
```

D.2 Second Example

To implement the simulation of the 12800 test signals, which were generated at the fixed carrier frequency 5kHz, SNR levels (-20dB, -10dB, -5dB, 0dB, 5dB, 10dB, 15dB, 20dB,) the DMRA algorithm was modified where the algorithm's core fixed for two layers then put in the concatenated loops necessary to test every signals.

Listing D.2: The Matlab® code to simulate the working of a two layer (8/3) classifier which generate the confusion matrices presented in Appendix C.
(appendix4/SimulationBISoneFC.m)

```

%+-----------------------------------------------------------+
%| Simulation routine to generate the confusion matrices     |
%| for the eight modulations using a two level (8/3)         |
%| classifier                                                |
%+-----------------------------------------------------------+

clear all
clc
```

```
%=============================================================%
% all the possible combinations of 3 modulations out of 8
combinations(:,:,1) =[1 2 3; 1 2 4; 1 2 5; 1 2 6; 1 2 7; 1 2 8;...
                      1 3 4; 1 3 5; 1 3 6; 1 3 7; 1 3 8; 1 4 5;...
                      1 4 6; 1 4 7; 1 4 8; 1 5 6; 1 5 7; 1 5 8;...
                      1 6 7; 1 6 8; 1 7 8];
combinations(:,:,2) =[2 1 3; 2 1 4; 2 1 5; 2 1 6; 2 1 7; 2 1 8;...
                      2 3 4; 2 3 5; 2 3 6; 2 3 7; 2 3 8; 2 4 5;...
                      2 4 6; 2 4 7; 2 4 8; 2 5 6; 2 5 7; 2 5 8;...
                      2 6 7; 2 6 8; 2 7 8];
combinations(:,:,3) =[3 2 1; 3 2 4; 3 2 5; 3 2 6; 3 2 7; 3 2 8;...
                      3 1 4; 3 1 5; 3 1 6; 3 1 7; 3 1 8; 3 4 5;...
                      3 4 6; 3 4 7; 3 4 8; 3 5 6; 3 5 7; 3 5 8;...
                      3 6 7; 3 6 8; 3 7 8];
combinations(:,:,4) =[4 2 3; 4 2 1; 4 2 5; 4 2 6; 4 2 7; 4 2 8;...
                      4 3 1; 4 3 5; 4 3 6; 4 3 7; 4 3 8; 4 1 5;...
                      4 1 6; 4 1 7; 4 1 8; 4 5 6; 4 5 7; 4 5 8;...
                      4 6 7; 4 6 8; 4 7 8];
combinations(:,:,5) =[5 2 3; 5 2 4; 5 2 1; 5 2 6; 5 2 7; 5 2 8;...
                      5 3 4; 5 3 1; 5 3 6; 5 3 7; 5 3 8; 5 4 1;...
                      5 4 6; 5 4 7; 5 4 8; 5 1 6; 5 1 7; 5 1 8;...
                      5 6 7; 5 6 8; 5 7 8];
combinations(:,:,6) =[6 2 3; 6 2 4; 6 2 5; 6 2 1; 6 2 7; 6 2 8;...
                      6 3 4; 6 3 5; 6 3 1; 6 3 7; 6 3 8; 6 4 5;...
                      6 4 1; 6 4 7; 6 4 8; 6 5 1; 6 5 7; 6 5 8;...
                      6 1 7; 6 1 8; 6 7 8];
combinations(:,:,7) =[7 2 3; 7 2 4; 7 2 5; 7 2 6; 7 2 1; 7 2 8;...
                      7 3 4; 7 3 5; 7 3 6; 7 3 1; 7 3 8; 7 4 5;...
                      7 4 6; 7 4 1; 7 4 8; 7 5 6; 7 5 1; 7 5 8;...
                      7 6 1; 7 6 8; 7 1 8];
combinations(:,:,8) =[8 2 3; 8 2 4; 8 2 5; 8 2 6; 8 2 7; 8 2 1;...
                      8 3 4; 8 3 5; 8 3 6; 8 3 7; 8 3 1; 8 4 5;...
                      8 4 6; 8 4 7; 8 4 1; 8 5 6; 8 5 7; 8 5 1;...
                      8 6 7; 8 6 1; 8 7 1];
%=============================================================%
rootpath = ['C:\Documents and Settings\ouail\My Documents\'...
            'My Thesis\code\signals2\one_FC'];

possible_modulations = ['FSK2'; 'FSK4';'ASK2'; 'ASK4';'BPSK';...
                        'QPSK';'QM16'; 'QM64'];
parent_dir = [ '\FSK2\'; '\FSK4\';'\ASK2\'; '\ASK4\';...
              '\BPSK\';'\QPSK\';'\QM16\'; '\QM64\'];

envir_dir = ['SNR_20dB\'; 'SNR_15dB\'; 'SNR_10dB\';...
             'SNR_05dB\';'SNR_00dB\'; 'SNR_-5dB\';...
             'SNR-10dB\'; 'SNR-20dB\'];

snr = [20 ; 15 ; 10 ; 05 ; 0  ; -5 ; -10; -20];


DB_path = ['C:\Documents and Settings\ouail\My Documents\'...
           'My Thesis\code\test_signals\one_FC\Hyper_planes'];
```

```
dimenssions = [2,3,4];

tic
for k = 1:8
    for n = 1:8
        path = [rootpath parent_dir(k,:) 'OUT_files\' envir_dir(n...
            ,:)];
        test_signals_file_name = ['OUT_' possible_modulations(k,:)...
            ...
             '_' num2str(snr(n))];
        load([path test_signals_file_name]);

        DB_name = ['Main_plane\' num2str(snr(n),'%02d')  '\...
            main_plane'];

        %load the first level page
        load([DB_path '\' DB_name]);

        Tally = zeros(3,8);          %Re-Intializations

        for i = 1:200                %
            for y = 1:3              %for each one of the ...
                dimensions 2,3,4
                features.GAMMA = gammamax(i);
                features.abs_phase = AP(i);
                features.dir_phase = DP(i);
                features.abs_amp = AA(i);
                features.abs_freq = AF(i);
                projection_coordinates = ...
                    project_onto_plane(features, plane, ...
                        dimenssions(y));
                Dt.mu = Data.mu(:,1:dimenssions(y))';
                Dt.Cov = Data.Cov(1:dimenssions(y),1:dimenssions(y...
                    ),:);
                [modulation_rank, maha_distances] = ...
                    Mahalanobis(Dt, projection_coordinates);

                for p = 1:3          %3 modulations with smallest
                                     %Mahalanobis distance to ...
                                        projection
                    first = combinations(1,1,modulation_rank(p));

                    for z = 1:21

                        second = combinations(z,2,first);
                        third = combinations(z,3,first);
                        level_2_DB_dir = ['level_2_planes\' ...
                            num2str(snr(n),'%02d') '\'...
                            num2str(possible_modulations(first,:),...
                                '%02d')];
```

```
106                         level_2_DB_name = [possible_modulations(...
                               first,:)...
                                '_plane_with_' possible_modulations(...
                                   second,:)...
                                '_' possible_modulations(third,:)];

                            %load the appropriate second level page
111                         load([DB_path '\' level_2_DB_dir '\' ...
                               level_2_DB_name]);

                            %project the signal on level 2 hyperplanes
                            level2_proj_coordinates = ...
                              project_onto_plane(features, level2plane...
                                 , dimenssions(y));
116
                            %get the Mahalanobis distances
                            Dt2.mu = level2Data.mu(:,1:dimenssions(y))...
                               ';
                            Dt2.Cov = level2Data.Cov(1:dimenssions(y)...
                               ,1:dimenssions(y),:);
                            [level2_mod_rank, level2_maha_distances] =...
                                ...
121                           Mahalanobis(Dt2, level2_proj_coordinates...
                                 );

                            %define the actual set of modulations and ...
                               store in
                            %RANK matrix
                            RANK(z,:,p) = [combinations(z,...
                               level2_mod_rank(1),modulation_rank(p))...
                               ,...
126                             combinations(z,level2_mod_rank(2),...
                                   modulation_rank(p)),...
                                combinations(z,level2_mod_rank(3),...
                                   modulation_rank(p))];

                        end
                    end
131
                    [Decision(i,y)] = class_decision2(modulation_rank,...
                        RANK);
                end

                Tally(1,Decision(i,1)) = Tally(1,Decision(i,1)) + 1;
136             Tally(2,Decision(i,2)) = Tally(2,Decision(i,2)) + 1;
                Tally(3,Decision(i,3)) = Tally(3,Decision(i,3)) + 1;

            end
            %compile the totals of each set of tallies
141         ALL_TALLY(k,:,n) = Tally(1,:);  %2D classification results
            ALL_TALLY2(k,:,n) = Tally(2,:); %3D
            ALL_TALLY3(k,:,n) = Tally(3,:); %4D
```

```
        end
    end
146
    toc
```

Bibliography

1. Azzouz, E. and A. Nandi. *Automatic Modulation Recognition of Communication Signals*. Kluwer Academic Publishers, 1996.

2. Badgett, Christopher D. *Performance Evaluation of Automated Digital Modulation Recognition Algorithms*. Master's thesis, AFIT, Wright-Patterson Air Force Base, Ohio, March 2006.

3. Bishop, Christopher M. *Neural Networks for Pattern Recognition*, 35. Oxford University Press, 13 edition, 1995.

4. Bishop, Christopher M. *Neural Networks for Pattern Recognition*, 105–112, 227. Oxford University Press, 13 edition, 1995.

5. Boudreau, D., C. Dubuc, F. Patenaude, M. Dufour, J. Lodge, and R. Inkol. "A fast automatic modulation recognition algorithm and its implementation in a spectrum monitoring application". volume 2, 732–736 vol.2. 2000.

6. Callaghan, T.G., J. L. Perry, and J.K. Tjho. "Sampling and algorithms aid modulation recognition". *Microwaves RF*, pp.117–121, September 1985.

7. DeSimio, Martin P. and Glenn E. Prescott. "Adaptive generation of decision functions for classification of digitally modulated signals". Aerospace and Electronic Conference, 1010–1014. IEEE National Conf, 1988.

8. Dubuc, C., D. Boudreau, F. Patenaude, and R. Inkol. "An automatic modulation recognition algorithm for spectrum monitoring applications". *IEEE International Conference on Communications*, volume 1, 570–574 vol.1. 1999.

9. Duda, Richard O., Peter E. Hart, and David G.Strok. *Pattern Classification*, 36, 107, 622, 626. Wiley-Interscience, second edition, 2001.

10. Duda, Richard O., Peter E. Hart, and David G.Strok. *Pattern Classification*, 85, 117–124. Wiley-Interscience, second edition, 2001.

11. Hsue, Shue-Zen and Samir S. Soliman. "Automatic Modulation Recognition of Digitally Modulated Signals". *Proc. of MILCOM'89*, 1989.

12. Hsue, Shue-Zen and Samir S. Soliman. "Automatic Modulation Recognition using zero-crossing". *IEEE Proceedings*, vol.137(No.6):pp.459–464, December 1990.

13. Kim, K. and A. Polydoris. "Digital Modulation Classification: The BPSK versus QPSK case". pp.431–436. IEEE Conf., 1988.

14. Liedtke, F.F. "Computer Simulation of an Automatic classification procedure for digitally modulated communication signals with unknown parameters". *Signal Processing*, vol.6(No.4):311–323, August 1984.

15. Pauluzzi, David R. and Norman C. Beaulieu. "A Comparison of SNR Estimation Techniques for the AWGN Channel". *IEEE Tansactions on Communications*, 48(10), October 2000.

16. Reichert, J. "Automatic classification of communication signals using higher order statistics". *IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 5, 221–224 vol.5. 1992.

17. Spooner, C. M. "Classification of co-channel communication signals using cyclic cumulants". *Conference Record of the Twenty-Ninth Asilomar Conference on Signals, Systems and Computers*, volume 1, 531–536 vol.1. 1995.

18. Spooner, C. M. "On the utility of sixth-order cyclic cumulants for RF signal classification". *Conference Record of the Thirty-Fifth Asilomar Conference on Signals, Systems and Computers*, volume 1, 890–897 vol.1. 2001.

19. Spooner, C. M. "Automated Processing of Conventional and LPI Radar Signals Using ATK's Multiple-Signal Scene Analyzer", July 2005. Whitepaper.

20. Statistical Signal Processing, Inc. "CuHBC Automatic Signal Classifier". Product Presentation, January 2005.

21. Waters, Angela M. *Modification of a Modulation Recognition Algorithm to Enable Multi-Carrier Recognition*. Master's thesis, AFIT, Wright-Patterson Air Force Base, Ohio, March 2005.

22. Wong, M.L.D. and A.K. Nandi. "Automatic Digital Modulation Recognition using Spectral and Statistical Features with Multi-Layer Perceptrons". International Synposium on Signal Processing and its Applications. August 2001.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704–0188

1. REPORT DATE *(DD–MM–YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From — To)*
22–03–2007	Master's Thesis	Sept 2005 — Mar 2007

4. TITLE AND SUBTITLE

MULTI-DIMENSIONAL CLASSIFICATION ALGORITHM FOR AUTOMATIC MODULATION RECOGNITION

5a. CONTRACT NUMBER

DACA99–99–C–9999

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Ouail Albairat, First Lieutenant, USAF

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Air Force Institute of Technology
Graduate School of Engineering and Management(AFIT/EN)
2950 Hobson Way, Bldg. 640
WPAFB OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT/GE/ENG/07-01

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Vasu D. Chakravarthy
AFRL/SNR
2241 Avionics Circle, Bldg 620
Wright-Patterson AFB, OH 45433
DSN 785-6127×4245, vasu.chakravarthy@wpafb.af.mil

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

Approval for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

This thesis proposes an approach for modulation classification using existing features in a more efficient way. The Multi-Dimensional Classification Algorithm (MDCA) treats features extracted from signals of interest as elements with irrelevant identities, hence eliminating any dependence of the classifier on any particular feature. This design enables the use of any number of features, and the MDCA algorithm provides the capability to classify modulations in higher dimensions. The use of multiple features requires an equal number of data dimensions, and thus classification in as high a dimensional space as possible can improve final classification results. Finally, the MDCA algorithm uses a relatively small number of simple operations, which leads to a fast processing time. Simulation results for the MDCA algorithm demonstrate good potential. In particular, the MDCA consistently performed well (at SNR levels down to -10dB in some cases) and in identifying more modulation types.

15. SUBJECT TERMS

automatic modulation recognition, multi-dimensional classification, feature independent classification

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Dr. Richard K. Martin
U	U	U	UU	138	19b. TELEPHONE NUMBER *(include area code)* 937–255–3636×4625, richard.martin@afit.edu