Detecting a Sequential Signal Signature in the Radio Frequency Spectrum

A Thesis

Submitted to the Faculty

of

Drexel University

by

Christopher S. Lester

in partial fulfillment of the

requirements for the degree

of

Master of Science in Electrical Engineering

January 2012

Dedications

To Ryan, and all who have served.

Eternal Father, strong to save,
Whose arm hath bound the restless wave,
Who bidd'st the mighty ocean deep
Its own appointed limits keep;
O hear us when we cry to Thee
For those in peril on the sea!

Eternal Father, Faithful Friend,
Be swift to answer those we send
In brotherhood and urgent trust
On hidden missions dangerous;
O hear us when we cry to Thee
For SEALs in air, on land, and sea!

Eternal Father, grant, we pray
To all Marines, both night and day
The courage, honor, strength, and skill
Their land to serve, thy law fulfill;
Be Thou their shield forevermore
From every peril to the Corps.

Eternal Father, Lord of Hosts,
Watch o'er the men who guard our coasts;
Protect them from the raging seas
And give them light and life and peace;
Grant them from Thy great throne above
The shield and shelter of Thy love.

Lord, guard and guide the men who fly
Through the great spaces in the sky;
Be with them always in the air,
In dark'ning storms or sunlight fair;
O hear us when we lift our prayer
For those in peril in the air!

Acknowledgements

The work contained in this thesis is a compilation of several related lines of research that were performed by members of the Data Fusion Laboratory at Drexel University from 2008 to 2010. Each subject covered in this document was part of a larger effort, undertaken as part of a research project conducted by Drexel University.

Individuals who contributed to the content compiled in this document include (in alphabetical order): Bradford Boyle, Raymond Canzanese, David Dorsey, Gabriel Ford, Christopher Lester, Ryan Measel, and Richard Primerano. Supervising faculty included: Moshe Kam, Kapil Dandekar, and John Walsh.

Finally, I must also thank those around me who provided the encouragement I needed in order to see this thesis through to completion. Without the loving support of my wife, Ashley, and that of the rest of my family, this thesis would still remain uncompleted.

Table of Contents

List of Tables vi

List of Figures vii

List of Algorithms x

List of Acronyms xi

Abstract xiii

1. Introduction 1

    1.1 Defining the Problem 1

    1.2 Development of the Solution 3

    1.3 Pseudo-code Algorithms in this Text 4

    1.4 Terminology 5

    1.5 Organization 5

2. Signal Detection and Classification System Architecture 6

    2.1 Basic classification problem 6

    2.2 Features 6

        2.2.1 Low level features 7

        2.2.2 Higher level features 7

    2.3 Feature selection 8

    2.4 General System Architecture 9

    2.5 Physical Tuners 10

    2.6 Feature Extraction 12

    2.7 Target Signal Detection 14

    2.8 Simple Example 16

    2.9 Implementation 19

        2.9.1 Implementation of the Physical Tuners Subsystem 20

        2.9.2 Implementation of the Feature Extraction Subsystem 21

        2.9.3 Implementation of the Target Signal Detection Subsystem 22

        2.9.4 Implementation of the Controller and GUI 24

3. Feature Extraction Exemplified Through DTMF Detection 27
    3.1 Overview of DTMF and DTMF Signaling 27
    3.2 Fast and Automatic DTMF Detection 28
    3.3 The Optimal Parameter Vector 32
    3.4 Optimal Amplitude Parameters 33
    3.5 Numerical Solution for Frequency and Phase Parameters 35
        3.5.1 Gauss-Newton Method 35
        3.5.2 Step-size Selection 36
        3.5.3 Full vs. Reduced Parameter Vector 37
    3.6 Determining DTMF vs. non-DTMF 38
    3.7 Empirical Distributions 40
    3.8 Detector Performance 41
        3.8.1 DTMF Key Identification 44
        3.8.2 DTMF Detection Algorithm 45
    3.9 Non-Ideal DTMF: Frequency Deviation 46
    3.10 DTMF Feature Extraction Conclusions 47
4. Classification 49
    4.1 Linear Classification 49
    4.2 Linear Classifiers in Radio Frequency Communications 50
        4.2.1 Computing the Weight Vector 52
        4.2.2 Example 56
    4.3 Extension to Multiple Layers 59
    4.4 Sequential Nature of Target RF Communication Events 59
    4.5 Profile Hidden Markov Model 61
    4.6 Example with PHMMs 63
5. Conclusion 67
Bibliography 69
A. Full List of Features 72

List of Tables

3.1 DTMF frequency table 28

3.2 DTMF parameters for empirical distributions 40

3.3 Maximum recommended frequency variation 46

4.1 Sampling of the more than 36, 000 signals that were observed during system testing and used for training and testing the linear classifier 57

List of Figures

1.1 Fundamental blocks of the signal sequence detector . . . 3

1.2 Partial timeline depicting activity between two or more hijackers in the 9/11 terrorist attacks. Increased activity in the whole network is indicative of the impending attack [30]. . . . 4

2.1 General architecture of an RF signal detection system . . . 9

2.2 General decomposition of the PHYSICAL TUNERS subsystem . . . 11

2.3 General decomposition of the FEATURE EXTRACTION block . . . 13

2.4 General decomposition of the TARGET SIGNAL DETECTION subsystem . . . 14

2.5 Operational flow diagram of the RF signal detection system . . . 16

2.6 Block diagram of one implementation of the target signal detection system . . . 20

2.7 Operational flowchart through the FEATURE EXTRACTION block . . . 21

2.8 Operational flowchart through the TARGET SIGNAL DETECTION block . . . 23

3.1 Definition of observation window. Top graph shows original signal, continuously observable. Bottom graph shows the same signal as observed through the time window (shaded regions) with length T_w. The duty cycle of the observation window is the ratio of the length of time when the signal can be seen to the total length of the window. . . . 29

3.2 A DTMF tone is depicted at various stages in the estimation process. The top plot shows the original DTMF signal as it would be received if there were no windowing effects. The second plot shows the effects of applying the window to the signal. The third plot shows the result of concatenating the look-through portions of the windowed signal. The last plot shows the result of overlaying the original signal with the minimum mean squared error best fit waveform. . . . 30

3.3 Representative empirical distributions of $\epsilon_n(\boldsymbol{\theta}^*)$ for the case where there is amplitude imbalance but the frequencies are fixed at the values specified for each subfigure. Increasing the value of every parameter increases the separation in the two distributions, making the decision between accepting H_0 or H_1 easier and thus resulting in better detector performance. . . . 41

3.4 ROC curves for the DTMF detector as a function of window length T_w . . . 42

3.5 ROC curves for the DTMF detector as a function of duty cycle τ 42

3.6 ROC curves for the DTMF detector as a function of SNR...................... 42

3.7 DTMF detector's key identification performance 44

3.8 ROC curves for the DTMF detector with the non-ideal effect of frequency deviation (sinusoid frequencies uniformly distributed ±1.8% about nominal) . 47

4.1 Black and White signals in 2-space with 3 discriminant functions. H_3 doesn't separate the 2 classes. H_1 does but with a small margin while H_2 separates Black from White with the maximum margin...................... 50

4.2 Diagram of the linear classifier. The classifier has d input units corresponding to the values of the features of a measured communication. Each input feature x_i is multiplied by its corresponding weight w_i. The single bias unit always emits a value of 1. The final output unit $g(\mathbf{x})$ emits a $+1$ if $\mathbf{w}^{\mathrm{T}}\mathbf{x} > 0$ and a -1 otherwise. 51

4.3 Two-dimensional projections of the 7-dimensional feature space. Non-target signals are shown as blue asterisks, while target signals are shown as red squares. Any linear discriminant function will result in misclassifying either some target signals, some non-target signals, or both, due to the overlap of the two classes of signals in every feature subspace. 60

4.4 Example of a multi-layer perceptron. All inputs x_i feed into individual perceptrons with distinct weights and thresholds for each perceptron (left half). The outputs of these perceptrons can then be used as inputs to a second layer of perceptrons (right half). 61

4.5 Profile Hidden Markov Model. Each square block represents a "match" state (M_j), where that state represents one stage of the target signature. The number of match states is equal to the number of steps in the target signature. Between each match state are (optional) diamond blocks (I_j), representing inserted signals between each match state. A path through circular blocks (D_j) is also present, representing deleted or skipped match states that were not observed. Each arrow represents the manner in which flow may move from one block to another; the model includes a probability associated with each arrow.. 62

4.6 Profile Hidden Markov Model classifier test results. The vertical axis shows the log-likelihood ratio of the likelihood of the target signature model having generated the observed data to the likelihood of the background noise model having generated the observed data; the higher the value, the more likely it is for the observed data to contain the target signature. The three dotted “Original Training Set” series show the results of running the classifier on data used to train the target signature model. The “Testing Set with Complete Signature” is a unique series collected in the same manner as the training sets but not used in training. The three “Incomplete Testing Set” series represent data that were collected with the attributes shown (portions of the target signature were intentionally omitted when collecting the data), while the “Complete Testing Set with Third Event Removed” represents all the same data as the first “Complete” testing set except for the signals comprising the third event, which have been manually removed from the data set. The “Background Testing Set” is a representation of data that was collected in the same environment as the other data but without any signals from the target signature present. 64

List of Algorithms

2.1 Programmatic description of the *Controller* functionality 25

3.1 Time domain Dual-Tone Multi-Frequency detection 45

4.1 Least-Squares Error linear classifier training..................................... 56

4.2 Linear classification process 56

List of Acronyms

		Page where first used:
AM	Amplitude Modulation	6
AoA	Angle of Arrival	17
ASK	Amplitude Shift Keying	48
AWGN	Additive White Gaussian Noise	29
DTMF	Dual-Tone Multi-Frequency	7
FCC	Federal Communications Commission	12
FFT	Fast Fourier Transform	21
FM	Frequency Modulation	6
FSK	Frequency Shift Keying	48
GPS	Global Positioning System	11
GUI	Graphical User Interface	19
HMM	Hidden Markov Model	23
IQ	In-phase and Quadrature	9
ITU	International Telecommunication Union	27
LMMSE	Linear Minimum Mean Squared Error	53
LSE	Least-Squares Error	54
MLP	Multi-Layer Perceptron	59
MSDD	Miniature Software Defined Digitizer	20
PSD	Power Spectral Density	46
PTT	Push To Talk	56
PHMM	Profile Hidden Markov Model	61
PSK	Phase Shift Keying	48

QAM	Quadrature Amplitude Modulation	73
RF	Radio Frequency	6
ROC	Receiver Operating Characteristic	40
SNR	Signal to Noise Ratio	43
UHF	Ultra High Frequency	56
ULS	Universal Licensing System	12
USB	Universal Serial Bus	20
VHF	Very High Frequency	56

Abstract

Detecting a Sequential Signal Signature in the Radio Frequency Spectrum

Christopher S. Lester
Advisor: Moshe Kam, Ph. D.

We outline algorithms and methods that are useful in identifying and detecting the presence of a particular signal of interest in a noisy Radio Frequency (RF) environment. We define a signal of interest as having particular RF features (*e.g.*, frequency or bandwidth) dictated by the equipment and the standard used to generate the signal. We formulate the concept of a *target RF communication sequence signature* which defines a specific sequence of communication signals, then use this concept to explore methods of detecting sequences of signals of interest. We present a review of features that would be useful in performing signal classification, and outline a methodology for extracting a particular feature (DTMF audio) from a signal. This methodology is useful in performing feature extraction for other similar features. Finally, we demonstrate that the Profile Hidden Markov Model method would be capable of handling the classification requirements of the studied scenario.

1. Introduction

As the popularity of wireless communication devices grows, with unprecedented numbers of individuals and businesses operating radios, cell phones and sensors, it becomes increasingly difficult to detect and demodulate a particular signal of interest in the Radio Frequency (RF) environment [29, 13]. Here, "signal of interest" is a generic term referring to any RF transmission that is desired to be separated from the background RF environment. The techniques outlined in this document have been developed in an effort to identify (detect) the presence of a particular signal of interest and to aid in its demodulation.

The main objective of this work is to outline algorithms and methods that have been developed to improve this detection process. The methods presented here may be used to implement an automated signal detection system. Recent developments in cognitive radio systems could benefit from this work by separating useful signals from unwanted signals in the environment [21].

1.1 Defining the Problem

Many scenarios exist where knowledge of specific patterns of communication represents actionable evidence. Law enforcement monitoring of illegal drug or arms dealings depend on the communications between dealers and buyers. Increased traffic on certain networks may be indicative of rapid changes in supply and demand of commodities, preceding economic, health or financial crises (*e.g.*, [9, 31]). When the communications enter the realm of wireless devices, the problem often requires separating those signals representing the "targeted communications" from communications that are present due to non-targeted users operating over similar or identical means as those users who are targeted. While it may be easy in some circumstances for a human being to identify signals that would naturally be considered part of the same "conversation" or communications scenario, we are seeking automated means that would detect "targeted communications" with little or no human interaction.

The solution set forth in this document seeks to outline a general method for detecting a particular sequence of signals amidst other signals and noise. In doing so, a *sequence*, or a series of transmissions, is identified by properties that are known about it *a priori*. These properties are assembled to form a *signature* that describes all sequences matching a particular pattern.

In this study, the signal of interest is not a single transmission in time; rather, it is comprised of transmissions from multiple sources that may not have identical characteristics. Here we consider the case where single transmissions are, by themselves, unremarkable; they become signals of interest when they appear along with other signals in some interrelated sequence or order.

In order to be compatible with various systems that might be able to take advantage of our detection schemes, it is desired that our solution be modular. It should be possible to insert the solution wholly into another signal processing system as the stand-alone signal detection module. Additionally, we seek real time operation. The module should indicate when it has detected particular communications of interest. We also seek a *confidence level* associated with each detection and, to the extent possible, the reasoning behind the detection indication. The solution should have high quality detection performance (as measured by receiver operating characteristics or confusion matrices), and be capable of operating over a wide range of frequencies and accepting input from a wide range of RF equipment.

A diagram of the fundamental parts of the system is depicted in Figure 1.1 on the following page. The initial input to the system comes from the RF environment; signals from the RF environment are picked up by a tuner, which provides low level inputs to the detector (*e.g.*, center frequency, bandwidth, In-phase and Quadrature (IQ) data). These low level inputs are processed in a feature extraction block in order to obtain higher level inputs to the detector (*e.g.*, modulation type, radio service to which signal

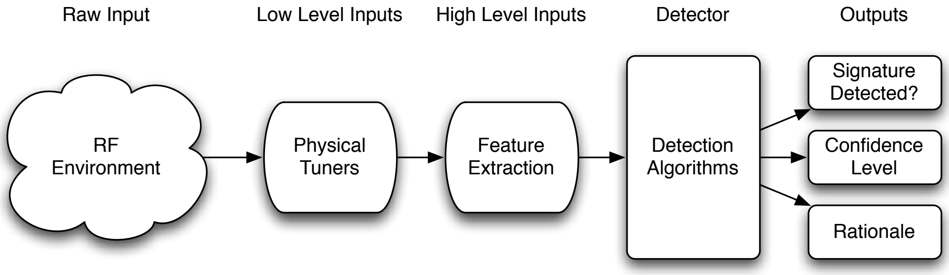


Figure 1.1: Fundamental blocks of the signal sequence detector

belongs). Both the low and high level inputs are passed to the detector, which provides as output whether a target signature was detected (along with its confidence of this decision and an indication of how the decision was reached).

Several examples of the importance of analyzing communication sequences are provided in [4] and its references. In these studies it is shown that simple frequency counts of communications between members of a team are not sufficient to understand the collaborative element of a team and that additional study of communication sequences can provide the missing data which characterize better versus worse performing teams.

An extreme example of the kind of patterns we are looking for is provided by the analysis of communications between perpetrators of the 9/11 terrorist attacks [30]. Here a communication event between any two members of the plot is of low significance, whereas the increased activity in the whole network is potentially an indication of impending illicit activity (see Figure 1.2 on the next page).

1.2 Development of the Solution

The detection scheme is made up of a series of discrete steps. First, the RF environment must be observed and analyzed. While decisions cannot be made from a single snapshot of the RF environment, individual snapshots make up the input to more complex detection schemes. Features must be estimated for each signal that is seen in each snapshot of the environment in order to distinguish it from other signals, and to identify

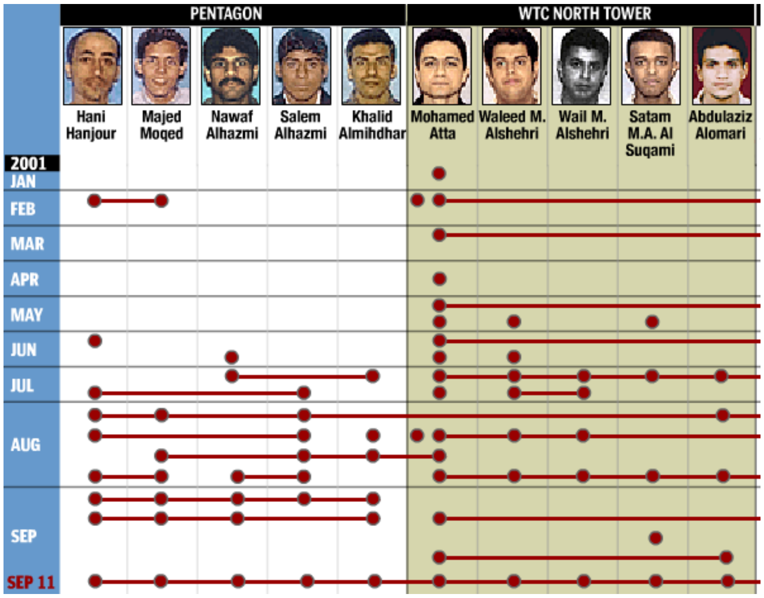


Figure 1.2: Partial timeline depicting activity between two or more hijackers in the 9/11 terrorist attacks. Increased activity in the whole network is indicative of the impending attack [30].

it in the context of a sequence of expected signals.

Second, features that have been estimated at each time step are processed to determine whether a target sequence exists in the observed environment. Through this processing, a *confidence indicator* is developed.

1.3 Pseudo-code Algorithms in this Text

Various algorithms and programmatic ideas are presented throughout this text in pseudo-code form. They are not intended to run or compile in any particular programming language, but are meant as an aid in developing code to reproduce the techniques described here.

1.4 Terminology

For the purposes of discussing the detection of certain signals, we will use the following terms as defined below.

Communication event An instance of a particular RF communication signal as observed in the RF environment.

Sequence A series of distinct **communication events** occurring in some (any) pattern over a period of time.

Signature A description of the set of all **sequences** conforming to a particular pattern.

Target sequence A particular **sequence** matching the **signature** being used for detection.

Target signal A constituent signal of a **target sequence**.

Non-target sequence/signal Any sequence/signal not matching the **signature** being used for detection.

1.5 Organization

The remainder of this thesis is organized as follows. In Chapter 2 (*Signal Detection and Classification System Architecture*), we describe the feature vectors that are used to characterize RF signals, and the architecture of the classification system that we propose to identify target sequences in the RF environment. Chapter 3 (*Feature Extraction Exemplified Through DTMF Detection*) continues with an in-depth review of extracting a particular feature (Dual-Tone Multi-Frequency (DTMF)) from a signal in order to use that feature for signal classification. Finally, we examine several forms of classifiers in Chapter 4 (*Classification*), and propose the use of a Profile Hidden Markov Model (PHMM) for use in our system.

2. Signal Detection and Classification System Architecture

In order to look for patterns or *signatures* in Radio Frequency (RF) emissions that would match the signature of a known communication sequence, it is necessary to capture a description of the state of the RF environment. The method we have developed uses a *feature vector* to describe an observed emission; this vector is passed as an input to the appropriate classification algorithm. The classifier makes a decision as to the presence of signals that match the target signature based on the sequence of observed feature vectors.

2.1 Basic classification problem

The RF classification problem consists of identifying particular RF signals and determining to which *class* they belong, based on features of the signal. Here, a class might be the basic modulation type of the signal (that is, whether the modulation is analog or digital), it might identify the particular modulation of the signal (whether the signal is Amplitude Modulation (AM) or Frequency Modulation (FM)), or indicate whether the signal belongs to a user in a known group of users (indicating membership in the class of commercial radio broadcasters). The decision regarding to which class (or classes) a signal belongs is the output of a classification model. Based on this output, one might take a particular action, *e.g.*, switch a receiver to a mode more suitable to processing the signal.

2.2 Features

A variety of features can be used to describe the RF signals that we wish to classify. The performance of our classification algorithms is determined by the specific features that are chosen. Some features will be inherently better at discriminating between different signal classes, depending on the performance index.

Listed here are some features that might be useful in an automatic classification system for RF signals. Some of these are easy to measure and there exists hardware and/or techniques for measuring them. Others are significantly more difficult to capture and may even entail a protocol-specific exploitation. For a more complete list, see Appendix A.

2.2.1 Low level features

We define low level features to be those features that require very little processing. These features require little to no inference about the signal and are obtainable without demodulation of the signal. Examples of low level features include:

Center Frequency – The measured center frequency of an observed signal.

Bandwidth – The measured bandwidth of an observed signal.

Received Power – The total received power within the specified bandwidth.

Knowledge of basic, low level features gives the fundamental means by which one might determine the uniqueness of a particular RF signal. Furthermore, the number of transmitters and sometimes even their locations can be estimated from this information. It is usually possible to estimate a low level feature without knowledge of any other features.

2.2.2 Higher level features

Higher level features are more specific to the user behind the RF emission. Examples of higher level features include:

Device Type – One-way versus two-way radio, commercial or home-built.

Radio Service – Radio service to which the communication event belongs.

Dual-Tone Multi-Frequency (DTMF) Key-press – Specific DTMF tone present in communication event, if any.

Higher level features usually require more processing power to estimate than low level features. Additionally, it is usually necessary to already have an estimate of other features in order to obtain a higher level feature.

2.3 Feature selection

While many features can be used to describe RF emissions, the computational effort to obtain some of these features is too great for them be of practical use. Other features, while relatively easy to obtain, are less discriminatory. The challenge is to identify a subset of all possible features that is both measurable and provides an accurate representation of the signal for the task at hand. For example, certain features (such as center frequency) are relatively easy to measure in a short amount of time. Other features (such as modulation type) require the receiver to stare at a particular emission for an extended period time, the result of which is that other emissions may be left unobserved.

While many of these higher level features are more costly to determine, both in terms of computational power and time, they have the potential to be more discriminating features than some of the lower level features. Consider that, while it is beneficial to know that there is an RF signal operating at 500 MHz, it is much more meaningful to say that the device operating at 500 MHz is probably a home-built FM transceiver, versus a (perhaps commonly available) commercial radio. The more detail that one can discern from the RF signal, the more precise the target signature that one can construct for performing RF classification.

The rest of this chapter describes the general architecture of a signal classification system that takes as its input a subset of the features just described and makes a determination as to whether a detected RF **communication event** is benign or is part of a **target sequence**.

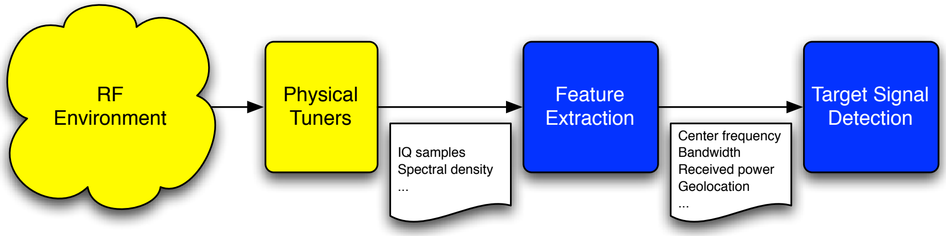


Figure 2.1: General architecture of an RF signal detection system

2.4 General System Architecture

The signal detection system consists of three main components as outlined below. A diagram of the system architecture as it relates to data flow among the blocks is shown in Figure 2.1. An alternative depiction of the system from an operational flow perspective appears in Figure 2.5 on page 16 along with a simple example of the operation of the system.

1. PHYSICAL TUNERS block

   Function: Monitor the RF environment, provide initial raw data

   Input: RF signals from the environment

   Output: (1) In-phase and Quadrature (IQ) data, (2) environmental information, and (3) pre-supplied expert knowledge

2. FEATURE EXTRACTION block

   Function: Process raw data and provide a vector of features that describe each observed RF emission

   Input: Raw data from the PHYSICAL TUNERS block

   Output: For each RF signal present in the raw data, a vector containing features that describe that signal

3. TARGET SIGNAL DETECTION block

    Function: Determine, based on a statistical model, whether an observed RF emission is benign or part of a target sequence

    Input: (1) Vectors of features generated by FEATURE EXTRACTION block, from a database, and (2) **target signature** model definitions

    Output: (1) Decision of whether any target sequence is present, (2) confidence levels for decision, and (3) mapping from target sequence components to target signature components

The first two blocks process data successively in a continuous loop, with the FEATURE EXTRACTION block writing vectors of features to a database for each signal that is detected. Meanwhile, the TARGET SIGNAL DETECTION block operates independently, utilizing all of the data that has been collected. In order to control the computational complexity of the detection problem and to operate with finite storage capacities, in practice we discard data older than five minutes, using only recent data in the detection problem.

The following sections provide a detailed look at the makeup of each of these components.

2.5 Physical Tuners

The PHYSICAL TUNERS block provides information about the physical environment necessary for the FEATURE EXTRACTION block to extract the features described in Section 2.2. The components that comprise the PHYSICAL TUNERS block are shown in Figure 2.2 on the following page.

The input to this subsystem comes from the environment and returns real-time IQ samples and/or spectral density information. The PHYSICAL TUNERS block may comprise either a single physical tuner or an array of physical tuners. Each tuner is connected either to a single antenna or to an array of antennas that monitor the RF

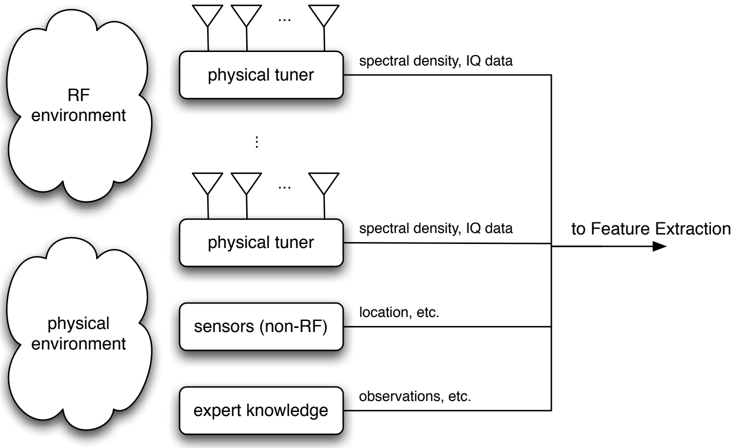


Figure 2.2: General decomposition of the PHYSICAL TUNERS subsystem

environment. Since the system must monitor all frequencies of interest, the tuners either

- are configured to monitor multiple smaller bands sequentially,
- are used in large quantities with each tuner monitoring a different band, or
- are capable of receiving sufficiently wide-band signals so as to cover the entire range of interest.

Sometimes, a combination of these three features will be used to monitor the bands of interest.

In addition to RF tuners, the PHYSICAL TUNERS block may also include other sensors that provide non-RF information about the physical environment. These sensors include Global Positioning System (GPS) receivers that can be used to provide location information for each physical tuner or antenna we are using. Other physical sensors might include optical sensors, ultrasonic sensors, or metal or radiological detectors.

A signal detection system can also use expert knowledge as an input to the system. For instance, a human operator may make the determination based on his or her own experience and observations that a particular situation or signal should be classified as a target signal, even though it does not otherwise match a preidentified target signature. Furthermore, in certain areas, agencies like the Federal Communications Commission (FCC) of the United States collect and make available to the public information regarding licensed use of the radio spectrum,[1] which could be used as a different form of expert knowledge.

The PHYSICAL TUNERS subsystem provides as output (1) information about the RF environment provided by the tuners, (2) information about the physical environment provided by additional sensors, and (3) expert knowledge provided by human operators and automated databases.

2.6 Feature Extraction

The FEATURE EXTRACTION block processes the data provided by the PHYSICAL TUNERS block and delivers as output vectors of features that describe observed RF emissions and communications. Figure 2.3 on the next page shows the structure of the FEATURE EXTRACTION block. The operation of this block is split into the following two stages.

Stage 1 of the block first detects RF emissions present in the data. This is accomplished through energy peak detection, where consecutive peaks of approximately the same amplitude are clustered together and denoted as a single signal.[2] Once a set of emissions is detected, the system continues to estimate low-level parameters such as center frequency, bandwidth, and received power for each one of them. Next, the system might perform direction finding or localization. These parameters are then combined

[1] The FCC provides this service in the United States through several online resources. The FCC Spectrum Dashboard provides a web browser interface to search and browse graphically through the various spectrum bands or via a map [10]. Similar information is also provided through the Universal Licensing System (ULS), which provides for large-scale database exports of licensee information based on a variety of parameters [11].

[2] This represents a limitation of the system, as it is conceivable that we could detect two signals close enough together in frequency that we would identify it as a single signal. For our purposes, however, we assume that such occurrences are rare, as we would expect the interference that such a situation would result in to prompt a change in operating behavior for one or both transmitters. Moreover, it would be exceedingly rare for two such signals to cycle “on” and “off” with sufficient synchronization for us to be unable to detect the presence of two separate signals.

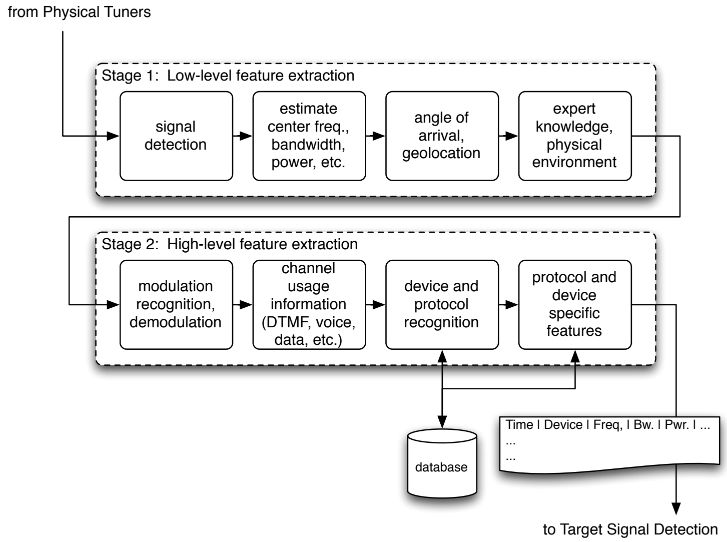

from Physical Tuners
Stage 1: Low-level feature extraction
signal detection
estimate center freq., bandwidth, power, etc.
angle of arrival, geolocation
expert knowledge, physical environment
Stage 2: High-level feature extraction
modulation recognition, demodulation
channel usage information (DTMF, voice, data, etc.)
device and protocol recognition
protocol and device specific features
database
Time | Device | Freq, | Bw. | Pwr. | ...
...
...
to Target Signal Detection

Figure 2.3: General decomposition of the FEATURE EXTRACTION block

with the estimated receiver location, time of emission, and additional information from the physical environment sensors or from expert knowledge.

Stage 2 begins after the low-level features of a signal have been determined. This stage involves extracting higher-level features that are required for the task at hand, such as identifying modulation type or channel usage information (*e.g.*, whether the demodulated signal contains voice/phone transmissions, data, or DTMF). Computations performed in this stage are typically more computationally expensive than those performed in *Stage 1* that gathered the low-level features.

As indicated in Figure 2.3, elements of *Stage 2* may use a database containing descriptions of both target and non-target devices and signals. For example, the database can contain information that describes the radios and devices used by local law enforcement for communication, or designated geographic areas that do not require attention. The database may also contain licensing information, such as the information from the FCC Universal Licensing System (ULS) database [11]. This information allows the system to search for the licensing information for detected signals based on the observed

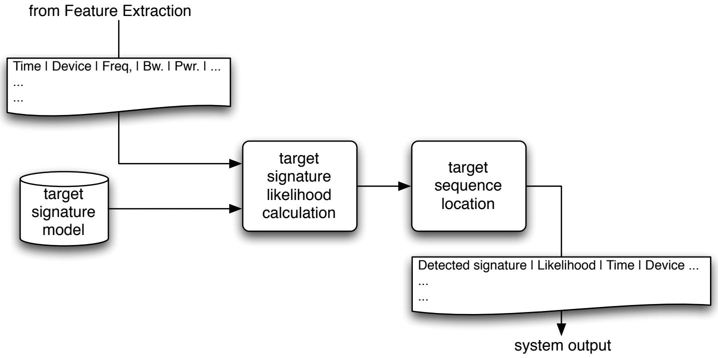

from Feature Extraction
Time | Device | Freq, | Bw. | Pwr. | ...
...
...
target signature model
target signature likelihood calculation
target sequence location
Detected signature | Likelihood | Time | Device ...
...
...
system output

Figure 2.4: General decomposition of the TARGET SIGNAL DETECTION subsystem

features, and verify whether the measured spectral characteristics of a detected signal conforms to a published license or standard. For example, if a detected signal does not conform to its published standards (*e.g.*, the received power is too high or the bandwidth is incorrect), the signal could be labeled as suspicious, or flagged for manual review. The output of the FEATURE EXTRACTION block is a vector of the extracted features for each detected RF emission.

2.7 Target Signal Detection

The final stage of the detection process is to perform target signal detection based on the features that have been extracted.

Figure 2.4 shows the structure of the TARGET SIGNAL DETECTION subsystem. This subsystem has as its input the database of all feature vectors generated by the FEATURE EXTRACTION module and provides as output a decision as to whether any target sequence is present, along with a confidence of the decision and a mapping to identify the signals that make up the sequence.

The objective of the TARGET SIGNAL DETECTION block is to use the feature vectors and temporal relationships of detected RF signals to determine the absence or presence of a target sequence based on a statistical model. This **target signature model** can describe either scenarios that should result in a positive detection or those that should preclude a positive detection. The model defines a sequence of signals in terms of the extracted features and the temporal sequences in which they occur. For example, the model would describe signals A, B, and C in terms of their frequency, bandwidth and modulation type (or any other subset of *extracted features*), and then describe a target sequence in terms of a temporal pattern. This pattern could be "signal A appearing for at least 15 seconds, followed by signals B and C appearing at some point simultaneously within 2 minutes of signal A's appearance." The form of this model is further clarified in the example of Section 2.8 on the following page.

The target signature model parameters can be obtained in one of two ways. The first method is to perform model training. This method begins with arbitrary model parameters that are adjusted to best fit a collection of data that has been identified as the desired target sequence. As an alternative to model training, the parameters may be manually set by an expert based on previous experience and knowledge of the target sequence. For a detailed discussion of the model development and design process, see Chapter 4.

Several components make up the TARGET SIGNAL DETECTION block (as shown in Figure 2.4 on the previous page). The first element of the TARGET SIGNAL DETECTION block is a likelihood calculation, where the presence or absence of the target sequence is assessed based on the target signature model. The second element of the block identifies the specific sequence of signals within the input data that cause a positive detection.

In addition to the actual indication of a detection, the output of the TARGET SIGNAL DETECTION block also includes the likelihood value that indicates the level of confidence in having made a correct detection. Furthermore, the TARGET SIGNAL DETECTION block also provides a mapping from the target signals to the position in the target sequence that was detected. In the event of detection being performed for multiple target signatures simultaneously, an identifier for the matching target signature would also be supplied. The output of target signal detection is intended to be suitable for

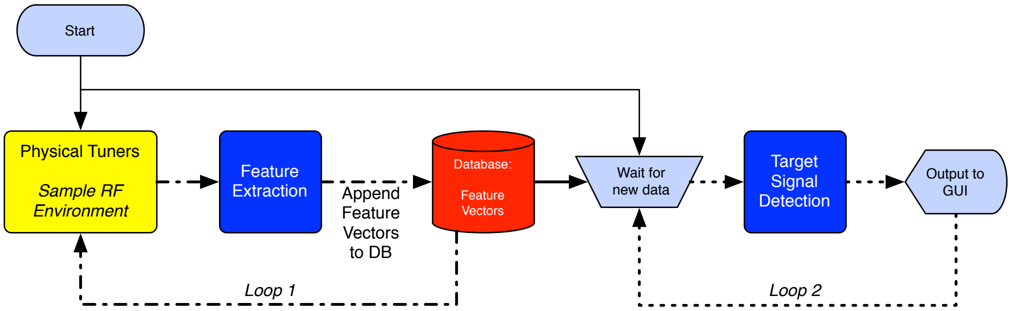


Figure 2.5: Operational flow diagram of the RF signal detection system

display to a human user for verification or initiation of an appropriate response.

2.8 Simple Example

The following paragraphs exemplify the operational flow of the system (depicted in Figure 2.5) for a simple scenario. The specific numbers given in this section are all hypothetical; in particular, the confidence numbers that are provided here are an abstraction of the log-likelihood ratio output from the classifier as discussed in Section 4.6 on page 63.

When the system starts, the database is empty of feature vectors, but has been seeded with the definitions for several target signature models, including one called Signature, which has been defined as

$$\textsc{Signature} \quad := \quad \left\{ \begin{array}{c} \textit{Signal-A} \text{ for at least 5 seconds, followed by} \\ \textit{Signal-B} \text{ for at least 5 seconds, followed by} \\ \textit{Signal-C} \text{ for at least 5 seconds,} \end{array} \right\} \text{ occurring in 45 secs.}$$

The signals *Signal-A*, *Signal-B*, and *Signal-C* are also then defined in terms of their features, including Angle of Arrival (AoA), as

$$\begin{aligned}
\textit{Signal-A} &:= \langle \text{Freq=100 MHz, Bandwidth=10 kHz, AoA } = * \rangle \\
\textit{Signal-B} &:= \langle \text{Freq=300 MHz, BW=1 MHz, AoA } = [\textit{Signal-A}\text{'s AoA}] \rangle \\
\textit{Signal-C} &:= \langle \text{Freq=100 MHz, BW=10 kHz, AoA } \neq [\textit{Signal-A}\text{'s AoA}] \rangle \,.
\end{aligned}$$

The definition of Signature indicates that an alert should be triggered if a target sequence is detected that matches the sequence of events laid out. A small subset of features has been used to define the particular signals that would comprise this sequence.

After the start of the system, *Loop 2* will begin in the "Wait for new data" sleep process until the database has new data for the detector to begin processing. *Loop 1* begins the data collection process with the physical tuners block gathering IQ data and information from other sensors.

The first data collected from the tuners is passed along to the feature extraction block. The data is processed in this block and peak detection discovers two signals:

$$\begin{aligned}
\textit{Signal-R} &= \langle \text{ID=}R\text{, Begin=0, Freq=150 MHz, BW=6 kHz, AoA } = 180^\circ \rangle \\
\textit{Signal-S} &= \langle \text{ID=}S\text{, Begin=0, Freq=100 MHz, BW=10 kHz, AoA } = 300^\circ \rangle
\end{aligned}$$

These two feature vectors (the output from the feature extraction block) are appended to the database and then *Loop 1* cycles back to the tuners to acquire another set of data. Each time through the loop, the feature extraction block checks to see that these signals remain; when they cease to be present, an ending time attribute will be added to the entry in the database.

This process continues, building up the following database of feature vectors:

⟨ ID=R, Begin=0, End=9 Freq=150 MHz, BW=6 kHz, AoA =180° ⟩
⟨ ID=S, Begin=0, End=8 Freq=100 MHz, BW=10 kHz, AoA =300° ⟩
⟨ ID=T, Begin=3, End=6 Freq=160 MHz, BW=15 kHz, AoA =345° ⟩
⟨ ID=U, Begin=12, End=20 Freq=300 MHz, BW=1 MHz, AoA =300° ⟩
⟨ ID=V, Begin=19, End=21 Freq=120 MHz, BW=10 kHz, AoA =225° ⟩
⟨ ID=W, Begin=21, End=58 Freq=250 MHz, BW=1 MHz, AoA =180° ⟩
⟨ ID=X, Begin=26, End=35 Freq=330 MHz, BW=200 kHz, AoA =45° ⟩
⟨ ID=Y, Begin=33, End=35 Freq=100 MHz, BW=10 kHz, AoA =135° ⟩
⟨ ID=Z, Begin=38, End=44 Freq=100 MHz, BW=10 kHz, AoA =90° ⟩

Each time one of the feature vectors is appended or modified in the database, *Loop 2* breaks out of its sleep process and runs the target signal detection subsystem on the most recent set of data that has been collected. The detection subsystem first determines a likelihood value for whether the Signature pattern is contained in the data. A visual inspection of the vectors above would indicate that there are 3 signals (S, U, and Z) that exactly correspond to *Signal-A*, *Signal-B* and *Signal-C* from the Signature pattern. Consider the output of the subsequent runs of the target signal detection subsystem.

Initially, when the detector runs, it sees just two signals, R and S. It can identify that one of the three components of Signature is present in the data: S matches *Signal-A* perfectly. The computed confidence is 10%. By time 6, the detector is able to produce the following output:

T=6, Sig=Signature, Conf=10%, *Signal-A* → S, *Signal-B* → nul, *Signal-C* → nul

At time 20, the detector sees signals R through V (though V has not yet ended). The confidence rises to 35%, as the detector now can see two signals (S and U) that match the first two components of the SIGNATURE. The output at this time step is:

T=20, Sig=SIGNATURE, Conf=35%, *Signal-A* → S, *Signal-B* → U, *Signal-C* → NUL

At time 35, the detector can see all but the last signal. The confidence has risen to 90%. While there is a match for all three signals (note that Y has all the parameters needed to be *Signal-C*), the duration of Y has not met the criteria to fully match the SIGNATURE. The detector recognizes that the match is not perfect (this is reflected in the confidence level), while at the same time detecting that a sequence nearly similar to the one defined by the SIGNATURE is present.

T=35, Sig=SIGNATURE, Conf=90%, *Signal-A* → S, *Signal-B* → U, *Signal-C* → Y

At time 44, the detector finally sees Z, and raises its confidence to 100%. The SIGNATURE definition has been perfectly matched.

T=44, Sig=SIGNATURE, Conf=100%, *Signal-A* → S, *Signal-B* → U, *Signal-C* → Z

Both operational loops continue in a similar manner until the entire system process is terminated.

2.9 Implementation

Figure 2.6 on the following page shows a block diagram of one possible implementation of the target signal detection system. It consists of the PHYSICAL TUNERS, FEATURE EXTRACTION and TARGET SIGNAL DETECTION blocks as shown in Figure 2.1 on page 9, along with a controller for coordinating the execution of the system and a Graphical User Interface (GUI) where the output of the system is presented.

For the purposes of testing the algorithms presented in this work, code was developed in MATLAB and was designed to run on commercial PC hardware. The rest of the

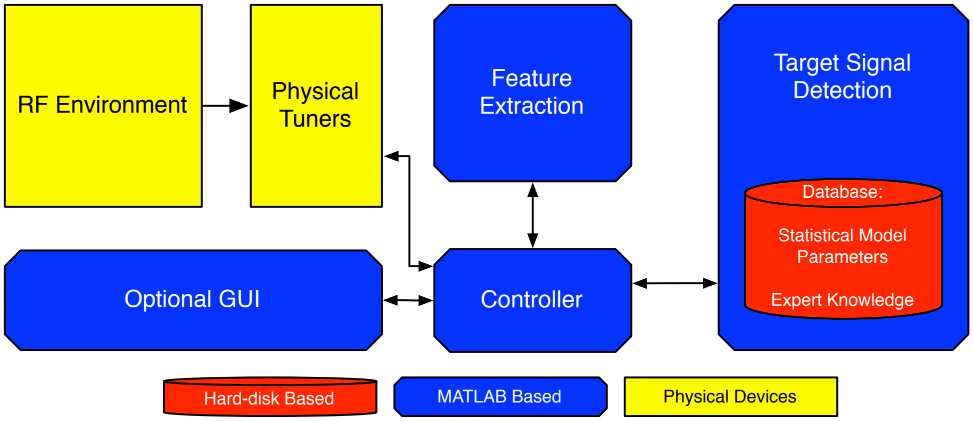


Figure 2.6: Block diagram of one implementation of the target signal detection system

equipment is likewise commercially available hardware, including:

- Softronics Miniature Software Defined Digitizer (MSDD)-3000 software defined radio tuner, capable of tuning to frequencies in the 30-3000 MHz range [23],
- BU-353 Universal Serial Bus (USB) GPS receiver [14], and
- Dell Mobile Precision M4500 laptop running Microsoft Windows 7 [7].

The MSDD-3000 tuner and the GPS receiver comprise the PHYSICAL TUNERS block, with the FEATURE EXTRACTION and TARGET SIGNAL DETECTION blocks running in MATLAB.

2.9.1 Implementation of the Physical Tuners Subsystem

The system uses a BU-353 GPS receiver and a Softronics MSDD-3000 software defined radio tuner as its PHYSICAL TUNERS. The BU-353 receiver provides the GPS location of the receiver antenna while the MSDD-3000 tuner provides both power spectral density data and real-time IQ samples. The MSDD-3000 tuner is capable of tuning to frequencies from 30 MHz to 3000 MHz. The system uses the tuner to sequentially

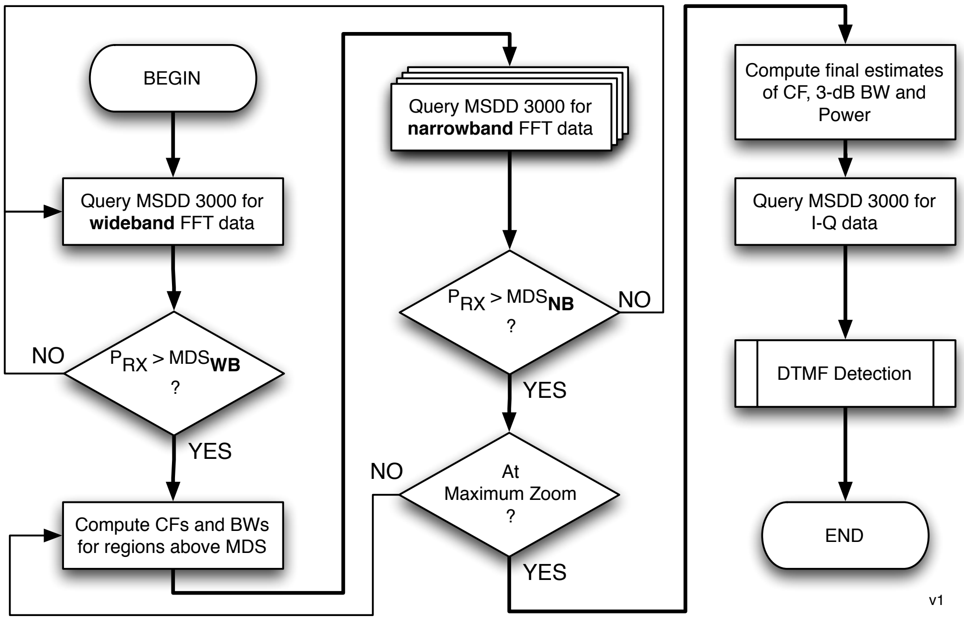

BEGIN
Query MSDD 3000 for **wideband** FFT data
P_{RX} > MDS_{WB} ?
NO
YES
Compute CFs and BWs for regions above MDS
Query MSDD 3000 for **narrowband** FFT data
P_{RX} > MDS_{NB} ?
NO
YES
At Maximum Zoom ?
NO
YES
Compute final estimates of CF, 3-dB BW and Power
Query MSDD 3000 for I-Q data
DTMF Detection
END
v1

Figure 2.7: Operational flowchart through the FEATURE EXTRACTION block

monitor user-specified frequency ranges and pass the data to the FEATURE EXTRACTION block.

The output of the PHYSICAL TUNERS is comprised of:

- power spectral density data,
- real time IQ samples, and
- the GPS coordinates of the receiver.

2.9.2 Implementation of the Feature Extraction Subsystem

The FEATURE EXTRACTION subsystem consists of algorithms used for extracting the features of each detected signal. Figure 2.7 shows a flowchart of the execution of this subsystem.

First, the subsystem requests wide-band Fast Fourier Transform (FFT) data from the physical tuner. The FFT envelope is then compared to a configurable threshold,

MDS_{WB}, which represents the **minimum detectable signal** able to be received by the tuner in its wide-band configuration. Portions of the FFT that exceed MDS_{WB} are designated as RF emissions, and the system estimates the bandwidth and center frequency for these signals.

The estimates of center frequency and bandwidth are refined by requesting narrow-band FFT data from the tuner (obtained through decimated sampling). The resulting FFT envelopes are analyzed in the same manner as the wide-band envelopes, this time comparing with MDS_{NB}, a threshold adjusted for the minimum detectable signals at the decimated sampling rate. The center frequency and bandwidth are estimated again from these "zoomed in" data.

Once the final estimates of center frequency, bandwidth and power for each detected signal have been made, the FEATURE EXTRACTION subsystem requests additional IQ data from the tuner in order to perform additional feature extraction, in particular DTMF detection.

The output of this block is a vector of features that characterizes each detected signal. The list of features and a description of each feature follows. These are:

Time − the time the signal was observed

Center Frequency − the estimated center frequency

Bandwidth − the estimated half-power bandwidth

Power − the received power

DTMF − whether a DTMF tone was detected and the detected DTMF key-press (if applicable)

Location − the GPS location of the receiver at the time that the RF emission was detected.

2.9.3 Implementation of the Target Signal Detection Subsystem

The TARGET SIGNAL DETECTION subsystem uses the feature vectors output by the FEATURE EXTRACTION subsystem to determine whether a sequence of detected signals

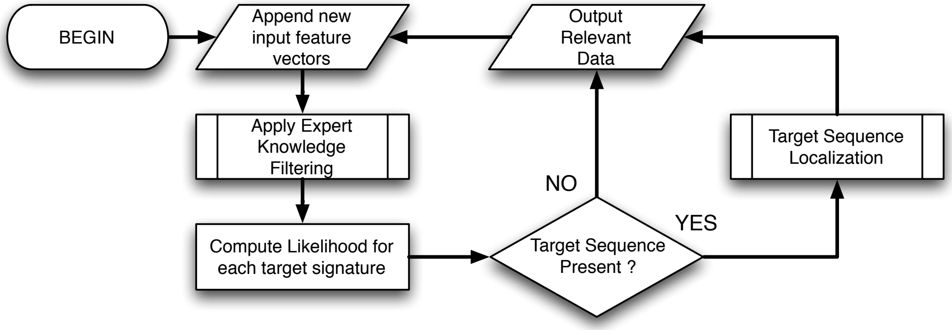

BEGIN
Append new input feature vectors
Output Relevant Data
Apply Expert Knowledge Filtering
Target Sequence Localization
NO
YES
Compute Likelihood for each target signature
Target Sequence Present ?

Figure 2.8: Operational flowchart through the TARGET SIGNAL DETECTION block

contains a known target signature. The basic flow within this subsystem is depicted in Figure 2.8. The TARGET SIGNAL DETECTION subsystem includes three main processes, *Feature Filtering*, *Classification* and *Localization*. Once the system is started, the data input vectors are loaded from the database. These data are then passed through the *Feature Filtering* process, where known non-target signals are filtered out based on expert knowledge that has been input into the system.

After filtering the data, the reduced dataset is passed to the *Classification* subsystem. This block determines whether the signals in the observed data form any of the patterns described by the target signatures that we are matching against. A log-likelihood ratio is calculated for each target signature, which is a measure of the likelihood that the observed data were generated by a model that contains the signature.[3] The current implementation of the *Classification* subsystem is a profile Hidden Markov Model (HMM). More detailed information about classification algorithms is included in Chapter 4.

The *Classification* process operates with a degree of flexibility when alerting to target sequences matching a target signature. It is possible that a sequence loosely matching a target signature (perhaps differing by one feature from the signature definition) would

[3] A log-likelihood ratio is a quantity used in comparing the relative probabilities of two hypotheses given a particular observed result. See the more detailed discussion of this quantity in [8].

be identified as a target sequence. As a result, even though the exact sequence defined by the target signature may not be present, the system can still alert to a target sequence if the confidence level meets the configurable threshold.

The *Classification* process is only capable of indicating whether or not a target sequence is present in the data (a binary output). If a target sequence is determined to be present, the system proceeds to identify which signals in the data comprise the sequence to which we are alerting. The *Localization* process goes back through the reduced dataset and identifies the particular signals that are the best match to the target signature. The identity of these signals is provided among the output of the subsystem.

If it is determined that no target sequence is represented in the current data (or that we are not confident enough that a sequence is present), the subsystem outputs informational data to that effect.

Regardless of whether or not a target sequence was detected, the subsystem then loops back to repeat the entire process with taking in new input from FEATURE EXTRACTION.

Since the system may be configured to search for more than one target signature, this detection process may occur multiple times in parallel over the same data set. The output of the TARGET SIGNAL DETECTION block indicates for each signature whether a target sequence was detected, along with the signals that were identified as part of that sequence. This output includes a mapping from each of the individual signals to the portions of the target signature that each signal corresponds to.

2.9.4 Implementation of the Controller and GUI

The final blocks of Figure 2.6 on page 20 are the *Controller* and the Graphical User Interface (GUI).

The *Controller* is implemented in MATLAB as the main program loop through which each of the other blocks are called. It directs the flow of data between the subsystems

Algorithm 2.1 Programmatic description of the *Controller* functionality

```
Main ()
    Gui.Initialize()
    Thread1()
    Thread2()
Thread1 ()
    while !exit do
        rawData ← PhysicalTuners.GetRawData()
        reducedData ← FeatureExtraction.Process(rawData)
        Database.Insert(reducedData)
    endwhile
Thread2 ()
    while !exit do
        repeat
            sleep
        until Database.NewDataAvailable()
        detectorResults ← TargetDetection.RunDetector()
        Alert(detectorResults)
        Gui.Update(detectorResults)
    endwhile
```

and regulates the communication among them. A pseudo-programmatic view of how the *Controller* functions appears in Algorithm 2.1.

The *Controller* initializes matlab objects for all hardware interfaces used by the system; it is also capable of interfacing with an RF simulator for use in place of physical tuners. The Main loop initializes all of the interfaces, including the GUI. It then starts two threads that run in parallel.

Thread1 operates the loop that runs the physical tuners and feature extraction subsystems. Data received from the first subsystem is passed to the second. The resulting reduced dataset is then appended to the Database.

Thread2 meanwhile operates a separate loop that waits until new data is available in the Database. Upon the arrival of new data, the target signal detection subsystem is called, with the resulting data (alerts to target signals, etc.) being displayed to the user (via the GUI or another suitable interface).

The *Controller* is designed to interface with a GUI when one is present, presenting information to a human operator for operational or debugging purposes.

An option not depicted in the algorithm is the ability for the *Controller* to be configured to log the output data (*rawData*, *reducedData*, and *detectorResults*) to a file. This capability is useful for both debugging and training purposes.

For use in training, once the data has been collected and logged, the resulting feature vectors may then be manually labeled by an expert, with occurrences of target sequences in the data identified. The resulting data set can then be used as training data and input to the training algorithms of the classifier in order to build statistical models for use by the TARGET SIGNAL DETECTION subsystem (see Section 4.2.1 on page 52).

3. Feature Extraction Exemplified Through DTMF Detection[4]

Many features mentioned in Chapter 2 are straightforward to calculate, while others require multiple steps of estimation and analysis to determine. One feature that, depending on the context, has the potential to be very discriminating is the presence and specific pattern of DTMF tones that appear in the audio channel.

3.1 Overview of DTMF and DTMF Signaling

DTMF is a method of multiple-frequency signaling over analog channels in the voice-frequency range [26]. The most common everyday usage for DTMF is in navigating automated telephone menu systems [16, 22]. In this usage, a user is presented with a list of options and then presses the numbered dialpad button corresponding to the appropriate choice. This button press generates a tone comprised of two pure sine waves, the combination of which is detected at the other end of the communication channel. Examination of the frequency components of the received audio tone reveals the specific button that was pressed. Table 3.1 on the next page shows the frequency pairing to button mapping for DTMF as defined in International Telecommunication Union (ITU)-T recommendation Q.23 [17].

Another application of DTMF signaling appears in the control of amateur radio repeaters. These radio transceivers (most commonly found in VHF/UHF range) “listen” on a particular input frequency and retransmit received signals on a different output frequency. Repeaters are used to retransmit weak or low-powered signals in order to allow users to communicate over a wider area of coverage; they are usually located atop a tall building or high up on a hill or mountain. These remote locations often benefit from some form of remote administration and this is typically accomplished via DTMF

[4] Portions of this chapter were developed in collaboration with Bradford D. Boyle.

Table 3.1: DTMF frequency table

	1209 Hz	1336	1477	1633
697 Hz	1	2	3	A
770	4	5	6	B
852	7	8	9	C
941	$*$	0	#	D

signaling on the repeater's input channel [6].

In the appropriate contexts, the presence or absence of DTMF audio in the signal being observed might delineate a signal that is of interest from one that is ordinary. Similarly, while multiple signals might be expected to carry DTMF tones, the presence of DTMF arranged in a specific pattern may indicate a signal of interest.

3.2 Fast and Automatic DTMF Detection

The DTMF detection problem can be stated in general for any given signal as

> *"Is there DTMF audio in this signal?"*

. . . and the follow up question,

> *"Given that there* is *DTMF audio in this signal, which of the 16 signals (key presses) is present?"*

Traditionally, DTMF detection has been done with banks of narrow band-pass filters [5]. This technique is only effective for signals that are continuously observable, however. We are interested in the case where a signal is observable only during specific **observation time windows**. These time windows occur at fixed intervals with a known periodic frequency and duty cycle, as depicted in Figure 3.1 on the following page. This type of windowing is seen with RF detectors that see only small portions of the RF spectrum at a time due to the large width of spectrum under observation.

This observation window may be parameterized with window length (T_w) and duty cycle (τ). We let T_w denote the length (in seconds) of one window, while T_n denotes

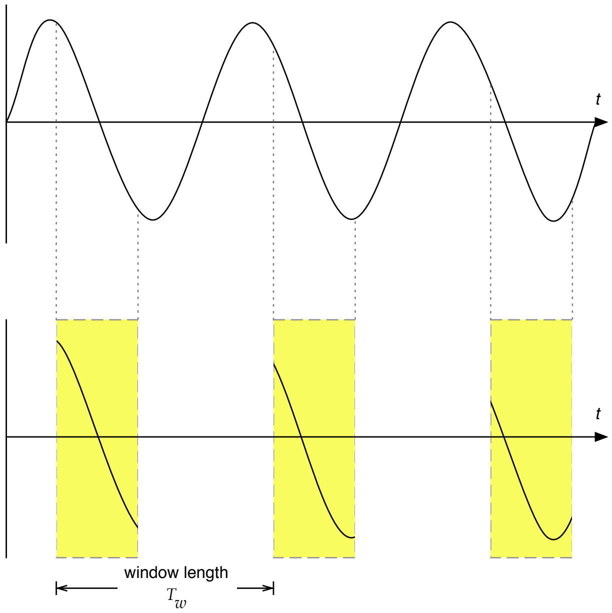


Figure 3.1: Definition of observation window. Top graph shows original signal, continuously observable. Bottom graph shows the same signal as observed through the time window (shaded regions) with length T_w. The duty cycle of the observation window is the ratio of the length of time when the signal can be seen to the total length of the window.

the same period in number of samples. As indicated in Figure 3.1, the window length is one full period, from the point when the signal appears, through the time when the signal is not observable, until the point when the signal reappears. The duty cycle of the observation window is τ. The duration of the look-through (period when the signal is observable) is given as τT_w seconds, or $\lfloor \tau T_n \rfloor$ samples.

In the context of the DTMF detection problem, we also add a third parameter, the number of consecutive windows (N) available for use in detection. The total number of samples used for detection becomes $n = \lfloor \tau T_n \rfloor N$.

Figure 3.2 on the following page shows pictorially the problem under consideration. A DTMF tone in Additive White Gaussian Noise (AWGN) is shown in the top plot, while the actual sampled signal is shown in the second with non-“visible” portions of the signal truncated to 0. Since we know the pattern of the look-through windows

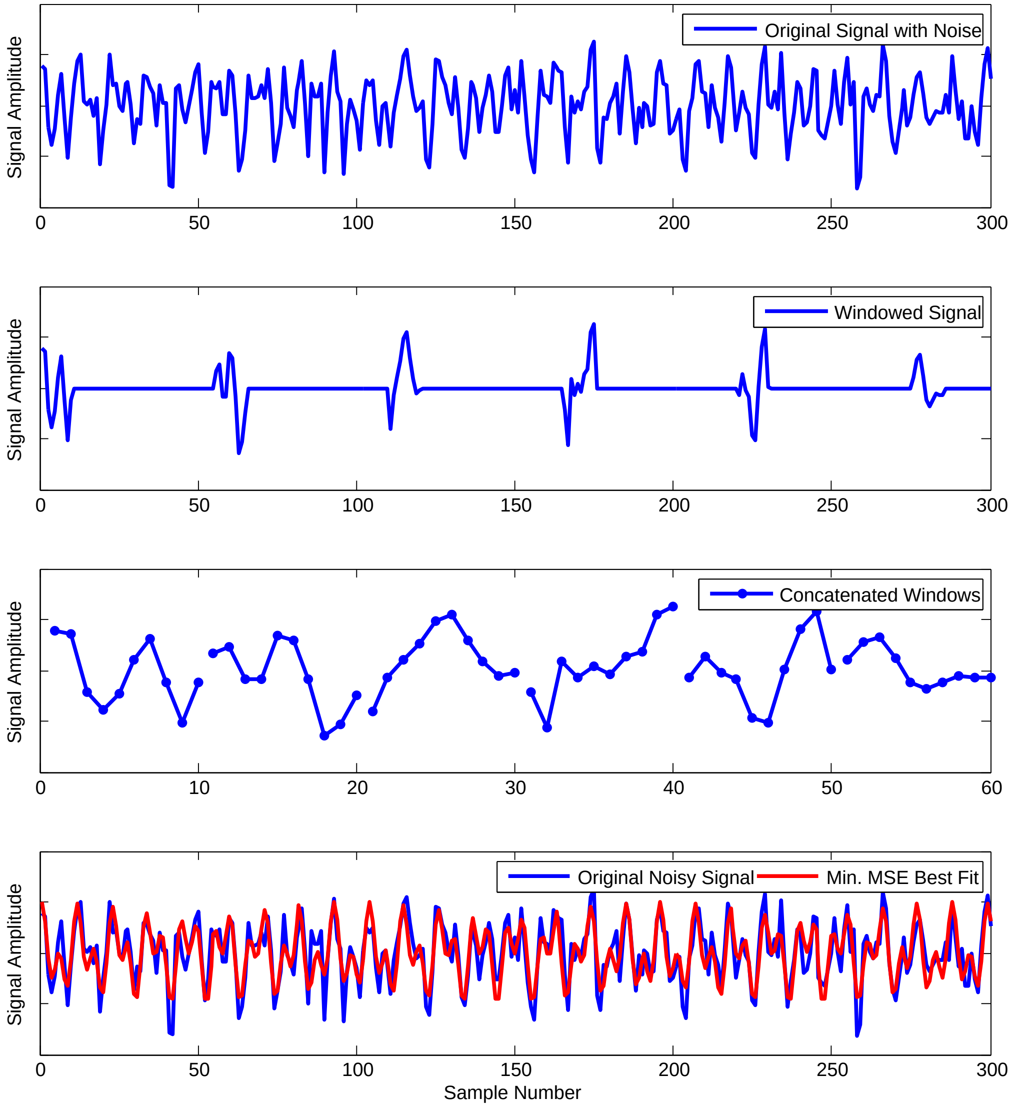


Figure 3.2: A DTMF tone is depicted at various stages in the estimation process. The top plot shows the original DTMF signal as it would be received if there were no windowing effects. The second plot shows the effects of applying the window to the signal. The third plot shows the result of concatenating the look-through portions of the windowed signal. The last plot shows the result of overlaying the original signal with the minimum mean squared error best fit waveform.

(which will result in valid samples), it is not necessary to sample during periods when it is known that the signal is not present. For faster processing time and reduced storage costs, we concatenate the blocks of samples together, as shown in the third plot. The final plot shows the original signal (DTMF corrupted by AWGN) overlaid with a generated DTMF tone corresponding to the best fitting (*i.e.*, minimum mean squared error) tone as determined by the process described in this chapter.

Each DTMF tone is characterized by its parameter vector, $\boldsymbol{\theta}$, defined in (3.1), where α is the amplitude, ω is the frequency and ϕ is the phase offset of the component tone, and the l or h subscript indicates that the parameter is for the low or high frequency tone in the DTMF. An ideal DTMF tone, $s(iT_s;\boldsymbol{\theta})$, is formed according to (3.2). This form allows for **amplitude imbalance**, where the amplitudes of the two component sinusoids are not the same.

$$\boldsymbol{\theta} = [\alpha_l \ \omega_l \ \phi_l \ \alpha_h \ \omega_h \ \phi_h] \tag{3.1}$$

$$s(iT_s;\boldsymbol{\theta}) = \alpha_l \cos(\omega_l iT_s + \phi_l) + \alpha_h \cos(\omega_h iT_s + \phi_h) \tag{3.2}$$

The approach that has been developed distinguishes between DTMF and non-DTMF signals in the time-domain. Furthermore, this approach is capable of distinguishing between different DTMF key presses. The technique that was developed is described by Equations (3.3)–(3.4) below. Initially, a signal of interest is sampled, down-converted, and FM demodulated giving the discrete waveform $r(iT_s)$. We denote the set of sample indices by $\mathcal{I}$ and let $n = |\mathcal{I}|$ be the total number of samples collected. The best fitting ideal DTMF tone $s(iT_s;\boldsymbol{\theta}^*)$ (which includes the time-domain effects of windowing, see Figure 3.2 on the previous page) is found and the resulting mean squared error $\varepsilon_n(\boldsymbol{\theta}^*)$ is recorded.

$$\varepsilon_n(\boldsymbol{\theta}) = \frac{1}{n}\sum_{i\in\mathcal{I}}[r(iT_s) - s(iT_s;\boldsymbol{\theta})]^2 \tag{3.3}$$

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \ \varepsilon_n(\boldsymbol{\theta}) \tag{3.4}$$

3.3 The Optimal Parameter Vector

The parameter vector that minimizes the mean squared error between the received signal and the two-tone model of (3.2) is the solution to an unconstrained optimization problem (that presented in (3.4)). Ideally, we would like to find a closed-form expression for $\boldsymbol{\theta}^*$ as a function of the received signal samples $r(iT_s)$. As a first step to finding a solution, we look at the first and second order necessary conditions [3].

First Order Necessary Condition In order for $\boldsymbol{\theta}^*$ to be a local minimum, it must satisfy $\nabla\varepsilon_n(\boldsymbol{\theta}^*) = \mathbf{0}$, that is, the partial derivative of the error with respect to each parameter being estimated must be zero. Using this relationship, we can develop the condition in terms of the received signal and the pure DTMF signal.

$$\frac{\partial\varepsilon_n(\boldsymbol{\theta})}{\partial\theta_j} = \frac{\partial}{\partial\theta_j}\left[\frac{1}{n}\sum_{i\in\mathcal{I}}\Big[r(iT_s) - s(iT_s;\boldsymbol{\theta})\Big]^2\right] = 0$$

$$\frac{1}{n}\sum_{i\in\mathcal{I}}\left[\frac{\partial}{\partial\theta_j}\Big[r(iT_s) - s(iT_s;\boldsymbol{\theta})\Big]^2\right] = 0$$

$$\frac{-2}{n}\sum_{i\in\mathcal{I}}\left[\Big[r(iT_s) - s(iT_s;\boldsymbol{\theta})\Big]\frac{\partial}{\partial\theta_j}s(iT_s;\boldsymbol{\theta})\right] = 0$$

$$\sum_{i\in\mathcal{I}} r(iT_s)\frac{\partial}{\partial\theta_j}s(iT_s;\boldsymbol{\theta}) = \sum_{i\in\mathcal{I}} s(iT_s;\boldsymbol{\theta})\frac{\partial}{\partial\theta_j}s(iT_s;\boldsymbol{\theta}) \tag{3.5}$$

Second Order Necessary Condition In order for $\boldsymbol{\theta}^*$ to be a local minimum (and not a maximum), it must also satisfy $\nabla^2\varepsilon_n(\boldsymbol{\theta}^*) \geq 0$, that is, the second partial derivative of the error with respect to each parameter must be non-negative.

$$\frac{\partial^2\varepsilon_n(\boldsymbol{\theta})}{\partial\theta_k\partial\theta_j} = \frac{\partial}{\partial\theta_k}\left(\frac{\partial\varepsilon_n(\boldsymbol{\theta})}{\partial\theta_j}\right) \geq 0$$

$$\frac{-2}{n}\sum_{i\in\mathcal{I}}\left[\frac{\partial}{\partial\theta_k}\left(\Big[r\left(iT_s\right) - s\left(iT_s;\boldsymbol{\theta}\right)\Big]\frac{\partial}{\partial\theta_j}s(iT_s;\boldsymbol{\theta})\right)\right] \geq 0$$

$$\frac{2}{n}\sum_{i\in\mathcal{I}}\left[\frac{\partial}{\partial\theta_k}s(iT_s;\boldsymbol{\theta})\frac{\partial}{\partial\theta_j}s(iT_s;\boldsymbol{\theta})+\Big[r(iT_s)-s(iT_s;\boldsymbol{\theta})\Big]\frac{\partial}{\partial\theta_k\partial\theta_j}s(iT_s;\boldsymbol{\theta})\right]\geq 0$$

$$\sum_{i\in\mathcal{I}}r(iT_s)\frac{\partial}{\partial\theta_k\partial\theta_j}s(iT_s;\boldsymbol{\theta})\geq\sum_{i\in\mathcal{I}}\left[s(iT_s;\boldsymbol{\theta})\frac{\partial}{\partial\theta_k\partial\theta_j}s(iT_s;\boldsymbol{\theta})-\frac{\partial}{\partial\theta_k}s(iT_s;\boldsymbol{\theta})\frac{\partial}{\partial\theta_j}s(iT_s;\boldsymbol{\theta})\right] \tag{3.6}$$

Finding a closed-form expression from (3.5) and (3.6) as a function of the received samples for candidate $\boldsymbol{\theta}^*$s is difficult, if not impossible. As a result, we use a numerical approach to finding the optimal parameter vector, $\boldsymbol{\theta}^*$.

3.4 Optimal Amplitude Parameters

Since it is not feasible to find a closed-form expression for the entire optimal parameter vector, we proceed by finding the expression for just the amplitude portion of this vector. The amplitude vector $\boldsymbol{\alpha} = [\alpha_l\ \alpha_h]^{\mathrm{T}}$ for a given set of frequencies and phases can be derived in the following manner. First, let

$$\begin{aligned}\boldsymbol{\omega} &= [\omega_l\ \omega_h]^{\mathrm{T}}\\ \boldsymbol{\phi} &= [\phi_l\ \phi_h]^{\mathrm{T}}\\ \boldsymbol{\alpha} &= [\alpha_l\ \alpha_h]^{\mathrm{T}} \text{ and}\\ \boldsymbol{\theta} &= [\alpha_l\ \omega_l\ \phi_l\ \alpha_h\ \omega_h\ \phi_h]^{\mathrm{T}}.\end{aligned}$$

We can then write the low and high frequency tones as

$$\mathbf{s}_l(\omega_l,\phi_l) = [\ldots\ \cos(\omega_l iT_s+\phi_l)\ \ldots]^{\mathrm{T}}$$

and

$$\mathbf{s}_h(\omega_h,\phi_h) = [\ldots\ \cos(\omega_h iT_s+\phi_h)\ \ldots]^{\mathrm{T}}$$

and the DTMF model can be written as

$$\mathbf{s}(\boldsymbol{\theta}) = \alpha_l\mathbf{s}_l+\alpha_h\mathbf{s}_h. \tag{3.7}$$

We can write the received signal's samples in a similar manner:

$$\mathbf{r} = [\ldots\ r(iT_s)\ \ldots]^{\mathrm{T}}.$$

With these definitions, it can be shown that the first derivatives of $\varepsilon_n(\boldsymbol{\theta})$ with respect to the amplitude parameters $(\alpha_l,\ \alpha_h)$ are, modulo multiplicative constants,

$$\frac{\partial \varepsilon_n(\boldsymbol{\theta})}{\partial \alpha_l} = -\langle \mathbf{s}_l, \mathbf{r}\rangle + \alpha_h \langle \mathbf{s}_h, \mathbf{s}_l\rangle + \alpha_l ||\mathbf{s}_l||_2^2 \ \text{ and} \tag{3.8}$$

$$\frac{\partial \varepsilon_n(\boldsymbol{\theta})}{\partial \alpha_h} = -\langle \mathbf{s}_h, \mathbf{r}\rangle + \alpha_l \langle \mathbf{s}_h, \mathbf{s}_l\rangle + \alpha_h ||\mathbf{s}_h||_2^2. \tag{3.9}$$

The matrix of mixed partial derivatives (Hessian) is given as (again, modulo multiplicative constants)

$$\mathbf{H} = \begin{bmatrix} ||\mathbf{s}_l||_2^2 & \langle \mathbf{s}_h, \mathbf{s}_l\rangle \\ \langle \mathbf{s}_h, \mathbf{s}_l\rangle & ||\mathbf{s}_h||_2^2 \end{bmatrix}.$$

The Hessian is always positive semi-definite. To see this, we note that the determinant is given by

$$\det(\mathbf{H}) = ||\mathbf{s}_l||_2^2\ ||\mathbf{s}_h||_2^2 - \langle \mathbf{s}_h, \mathbf{s}_l\rangle^2 \geq 0$$

where the last inequality holds from the *Cauchy–Schwarz* inequality. This tells us that any local minimum (with respect to the amplitude parameters) is also a global minimum and we can find the optimal amplitude parameters by solving the first-order necessary conditions. These are

$$\begin{bmatrix} \frac{\partial \varepsilon_n(\boldsymbol{\theta})}{\partial \alpha_l} \\ \frac{\partial \varepsilon_n(\boldsymbol{\theta})}{\partial \alpha_h} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

which lead via (3.8) and (3.9) to

$$\boldsymbol{\alpha}^* = \begin{bmatrix} ||\mathbf{s}_l||_2^2 & \langle \mathbf{s}_h, \mathbf{s}_l\rangle \\ \langle \mathbf{s}_h, \mathbf{s}_l\rangle & ||\mathbf{s}_h||_2^2 \end{bmatrix}^{-1} \begin{bmatrix} \langle \mathbf{s}_l, \mathbf{r}\rangle \\ \langle \mathbf{s}_h, \mathbf{r}\rangle \end{bmatrix}. \tag{3.10}$$

3.5 Numerical Solution for Frequency and Phase Parameters

Because of the difficulty in finding a closed-form expression for the rest of the optimal parameter vector (frequency and phase), we turn our attention to numerical techniques for solving the optimization problem. Since we are able to take both first and second derivatives of (3.3), we can make use of *gradient methods* [3, 25]. The general gradient method can be written as

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k - \beta_k \mathbf{D}_k \nabla \varepsilon_n(\boldsymbol{\theta}_k)$$

where $\mathbf{D}_k$ is some positive definite matrix and β_k is the size of the step taken at iteration k.

3.5.1 Gauss-Newton Method

We can rewrite (3.3) in the following manner

$$\varepsilon_n(\boldsymbol{\theta}) = \frac{1}{2} \left|\left| \sqrt{\frac{2}{n}} (\mathbf{r} - \mathbf{s}(\boldsymbol{\theta}^*)) \right|\right|_2^2 \tag{3.11}$$

where $\mathbf{r}$ is the row vector of received samples and $\mathbf{s}(\boldsymbol{\theta})$ is the row vector of model DTMF samples. By defining

$$\mathbf{g}(\boldsymbol{\theta}) = \sqrt{\frac{2}{n}} (\mathbf{r} - \mathbf{s}(\boldsymbol{\theta})),$$

the original optimization problem becomes

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \frac{1}{2} \left|\left| \mathbf{g}(\boldsymbol{\theta}) \right|\right|_2^2 . \tag{3.12}$$

We can solve (3.12) using a gradient method known as *Gauss-Newton method* [3] by letting

$$\mathbf{D}_k = \left[\nabla \mathbf{g}\left(\boldsymbol{\theta}\right) \nabla \mathbf{g}\left(\boldsymbol{\theta}\right)^{\mathrm{T}} \right]^{-1} .$$

If $\nabla \mathbf{g}(\boldsymbol{\theta})$ has rank equal to the number of parameters, then the matrix

$$\nabla \mathbf{g}(\boldsymbol{\theta}) \nabla \mathbf{g}(\boldsymbol{\theta})^{\mathrm{T}}$$

will be positive definite and therefore invertible. The gradient method can then be written as

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k - \beta_k \left[\nabla \mathbf{g}(\boldsymbol{\theta}) \nabla \mathbf{g}(\boldsymbol{\theta})^{\mathrm{T}}\right]^{-1} \nabla \mathbf{g}(\boldsymbol{\theta}) \mathbf{g}(\boldsymbol{\theta})^{\mathrm{T}}.$$

Substituting in for $\mathbf{g}(\boldsymbol{\theta})$, we have

$$\nabla \mathbf{g}(\boldsymbol{\theta}) = -\sqrt{\frac{2}{n}} \nabla \mathbf{s}(\boldsymbol{\theta})$$

$$\mathbf{D}_k = \frac{n}{2} \left[\nabla \mathbf{s}(\boldsymbol{\theta}) \nabla \mathbf{s}(\boldsymbol{\theta})^{\mathrm{T}}\right]^{-1}$$

which yields

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k + \beta_k \left[\nabla \mathbf{s}(\boldsymbol{\theta}) \nabla \mathbf{s}(\boldsymbol{\theta})^{\mathrm{T}}\right]^{-1} \nabla \mathbf{s}(\boldsymbol{\theta}) \left[\mathbf{r} - \mathbf{s}(\boldsymbol{\theta})\right]^{\mathrm{T}}. \tag{3.13}$$

3.5.2 Step-size Selection

For choosing the step-size (β_k), there are a variety of methods to choose from. These methods include:

constant step-size The simplest technique is to use a fixed constant $\beta_k = \beta \; \forall \; k$ for every iteration. This is attractive because it is simple; however, choosing the wrong constant can cause the method to diverge or perhaps to converge very slowly.

diminishing step-size In this method, the step-sizes are chosen so that

$$\lim_{k \to \infty} \beta_k = 0 \quad \text{while also} \quad \sum_{k=0}^{\infty} \beta_k = \infty.$$

This prevents the sequence $\{\boldsymbol{\theta}_k\}$ from converging to a non-stationary (and hence non-optimal) point. Unfortunately, this selection rule will generally result in slow convergence which is not ideal given the problem statement.

Armijo Rule This is a simple successive step-size reduction method that is attractive because of its ease of implementation and because at each iteration it guarantees a sufficiently large improvement in the objective function. A more thorough discussion is available in [3].

For simplicity, we have chosen to use the constant step-size for our implementation of the Gauss-Newton method.

3.5.3 Full vs. Reduced Parameter Vector

If we take $\boldsymbol{\theta}$ to be the full parameter vector as in (3.2) on page 31, we can use (3.13) to solve for $\boldsymbol{\theta}^*$. In this case,

$$\nabla \mathbf{s}(\boldsymbol{\theta}) = \begin{bmatrix} \frac{\partial}{\partial \omega_L}\mathbf{s}(\boldsymbol{\theta}) \\ \frac{\partial}{\partial \phi_L}\mathbf{s}(\boldsymbol{\theta}) \\ \frac{\partial}{\partial \omega_H}\mathbf{s}(\boldsymbol{\theta}) \\ \frac{\partial}{\partial \phi_H}\mathbf{s}(\boldsymbol{\theta}) \end{bmatrix} = \begin{bmatrix} \dots & -iT_s \sin(\omega_L i T_s + \phi_L) & \dots \\ \dots & -\sin(\omega_L i T_s + \phi_L) & \dots \\ \dots & -iT_s \sin(\omega_H i T_s + \phi_H) & \dots \\ \dots & -\sin(\omega_H i T_s + \phi_H) & \dots \end{bmatrix}.$$

Unfortunately, there are several stationary points that are not optimal and that do not correspond to a valid frequency pair for DTMF. We can exploit the fact that the frequency tuple (ω_H, ω_L) can only be one of 16 possible values and iterate over these fixed values. The parameter vector is then $\boldsymbol{\theta} = [\phi_L \, \phi_H]$ and we have

$$\nabla \mathbf{s}(\boldsymbol{\theta}) = \begin{bmatrix} \frac{\partial}{\partial \phi_L}\mathbf{s}(\boldsymbol{\theta}) \\ \frac{\partial}{\partial \phi_H}\mathbf{s}(\boldsymbol{\theta}) \end{bmatrix} = \begin{bmatrix} \dots & -\sin(\omega_L i T_s + \phi_L) & \dots \\ \dots & -\sin(\omega_H i T_s + \phi_H) & \dots \end{bmatrix}.$$

We then only need to use Gauss-Newton to find the *phase* that minimizes $\varepsilon_n(\boldsymbol{\theta})$. A drawback to this is approach is that we are now solving 16 optimizations (one for each frequency combination) instead of a single one increasing the time until we make a decision. In practice, even when running the optimizations in series (*i.e.*, one after the other) we observed convergence times less than one second, fast enough for the purposes of the target signal detection system. This time could be further reduced by running the optimizations in parallel and selectively pruning preemptively.

3.6 Determining DTMF vs. non-DTMF

In order to "detect" DTMF signals, we formulate the decision problem as a classical hypothesis testing with the minimum mean squared error acting as a performance index. More formally, let

$$\begin{aligned} H_0 &: \text{DTMF} \\ H_1 &: \text{Not DTMF.} \end{aligned}$$

The Bayes risk is then

$$\mathcal{R} = \sum_{i=0}^{1}\sum_{j=0}^{1} P_j C_{i,j} \int_{z_i} p_{\varepsilon_n(\boldsymbol{\theta}^*)|H_j}(\varepsilon_n(\boldsymbol{\theta}^*)|H_j)\, d\varepsilon_n(\boldsymbol{\theta}^*)$$

where P_j is the *a priori* probability associated with hypothesis H_j, $C_{i,j}$ is the cost associated with declaring hypothesis H_i to be true, when in fact hypothesis H_j is true, and z_i is the range of values for $\varepsilon_n(\boldsymbol{\theta}^*)$ where we declare H_i to be true. If we assume that the cost of errors are identical and that there is no cost associated with correct decisions, *i.e.*, $C_{0,1} = C_{1,0}$ and $C_{0,0} = C_{1,1} = 0$, then the decision rule that minimizes the Bayes risk is

$$\Lambda(\varepsilon_n(\boldsymbol{\theta}^*)) \underset{H_0}{\overset{H_1}{\gtrless}} \frac{P_0}{P_1}, \tag{3.14}$$

as shown in [28].

We decide in favor of H_0 if the left hand side is less than the quantity on the right hand side and decide against H_0 (in favor of H_1) if the left hand side is greater. The quantity $\Lambda(\varepsilon_n(\boldsymbol{\theta}^*))$ is defined as

$$\Lambda(\varepsilon_n(\boldsymbol{\theta}^*)) = \frac{p_{\varepsilon_n(\boldsymbol{\theta}^*)|H_1}(\varepsilon_n(\boldsymbol{\theta}^*)|H_1)}{p_{\varepsilon_n(\boldsymbol{\theta}^*)|H_0}(\varepsilon_n(\boldsymbol{\theta}^*)|H_0)}. \tag{3.15}$$

Looking at equations (3.14) and (3.15), we know that in order to make a decision using a Bayes' test we must know: 1) the *a priori* probabilities of a signal being DTMF or non-DTMF and 2) the probability distribution of $\varepsilon_n(\boldsymbol{\theta}^*)$ under the different hypotheses. Fixing the *a priori* probabilities is difficult and an incorrect guess can lead to poor detector performance.

We can remove the dependence on these values by considering a Neyman-Pearson style detector, wherein the probability of false alarm is fixed and the threshold level is set such that this false alarm rate is met. The probability of false alarm (P_F) and probability of detection (P_D) are given by

$$P_F = \int_{z_0} p_{\varepsilon_n(\boldsymbol{\theta}^*)|H_1}(\varepsilon_n(\boldsymbol{\theta}^*)|H_1)\, d\varepsilon_n(\boldsymbol{\theta}^*)$$

$$P_D = \int_{z_0} p_{\varepsilon_n(\boldsymbol{\theta}^*)|H_0}(\varepsilon_n(\boldsymbol{\theta}^*)|H_0)\, d\varepsilon_n(\boldsymbol{\theta}^*).$$

Specifying a maximum allowable false alarm rate (α), we solve

$$\alpha = \int_{z_0} p_{\varepsilon_n(\boldsymbol{\theta}^*)|H_1}(\varepsilon_n(\boldsymbol{\theta}^*)|H_1)\, d\varepsilon_n(\boldsymbol{\theta}^*)$$

for z_0 and find the corresponding P_D. While this removes the dependency on the *a priori* probabilities, we still need the probability distributions for $\varepsilon_n(\boldsymbol{\theta}^*)$. Finding closed-form expressions for these distributions is difficult. As an alternative, we approximate these distributions by observing the empirical distributions over a variety of settings.

Table 3.2: DTMF parameters for empirical distributions

T_w (T_n)	$\{8, 12, 16, 20\}$ ms ($\{200, 300, 400, 500\}$ samples)
τ	$\{8, 12, 16, 20\}\%$
N	$\{1, 2, 5\}$
SNR	$\{0, 3, 6\}$ dB

3.7 Empirical Distributions

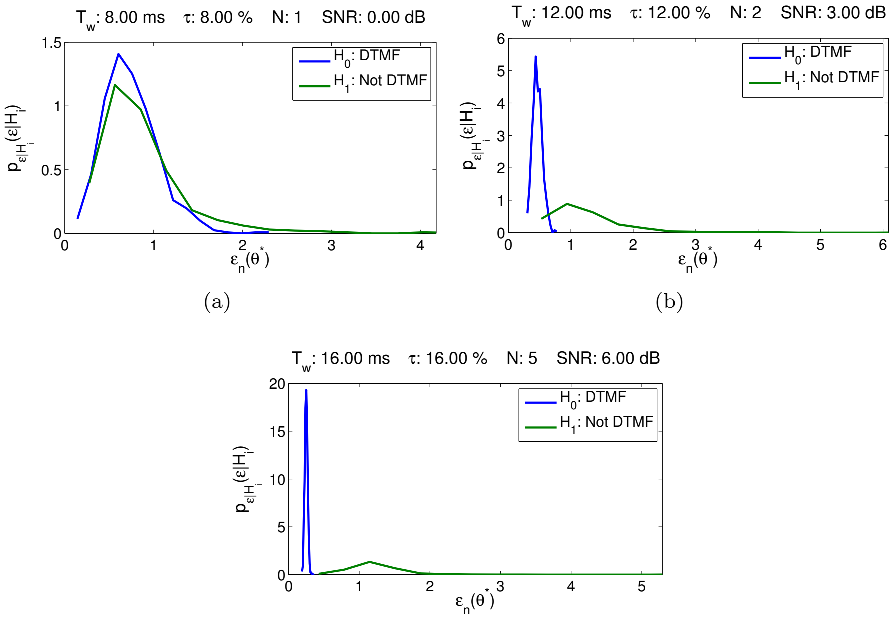
To build up the empirical distributions (*i.e.*, histograms) of $\varepsilon_n(\boldsymbol{\theta}^*)$ under the different hypotheses, an arbitrarily large number of DTMF and non-DTMF samples were generated randomly and windowed. Additive white Gaussian noise was added to the samples and the optimal $\boldsymbol{\theta}^*$ values were computed. The resulting value of $\varepsilon_n(\boldsymbol{\theta}^*)$ was utilized to create histograms of $p_{\varepsilon_n(\boldsymbol{\theta}^*)}(\varepsilon_n(\boldsymbol{\theta}^*))$ under both hypotheses based on the window parameters.

The DTMF samples were generated according to (3.16), which explicitly includes the non-ideal effects of amplitude imbalance, *i.e.*, when the amplitudes of the two sinusoids comprising the DTMF signal are not equal.[5]

$$\begin{aligned} r &= a_l \cos(\omega_L i T_s + \phi_L) + a_h \cos(\omega_H i T_s + \phi_H) \qquad (3.16) \\ a_l, a_h &\sim \text{Uniform}(0, 1) \end{aligned}$$

Distributions for $\varepsilon_n(\boldsymbol{\theta}^*)$ were generated for a variety of windowing parameters, the full set of which are shown in Table 3.2. Three representative distributions are shown in Figure 3.3 on the following page. With these distributions, we can plot Receiver Operating Characteristic (ROC) curves describing the expected performance of the detector.

[5] The ITU-T Recommendation Q.23 [17] does not specify the relative power levels of each frequency component. While it is common that a Recognized Private Operating Administration (*e.g.*, AT&T) would specify the maximum power level difference between each frequency [18], it is not guaranteed.



(a)

(b)

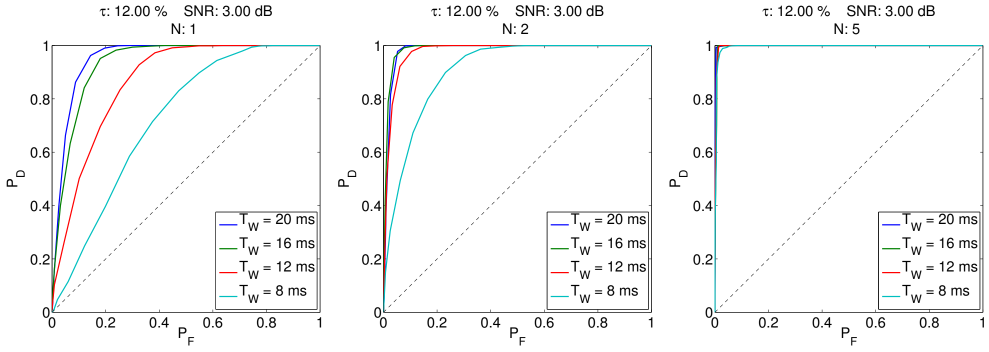
(c)

Figure 3.3: Representative empirical distributions of $\varepsilon_n(\boldsymbol{\theta}^*)$ for the case where there is amplitude imbalance but the frequencies are fixed at the values specified for each subfigure. Increasing the value of every parameter increases the separation in the two distributions, making the decision between accepting H_0 or H_1 easier and thus resulting in better detector performance.

3.8 Detector Performance

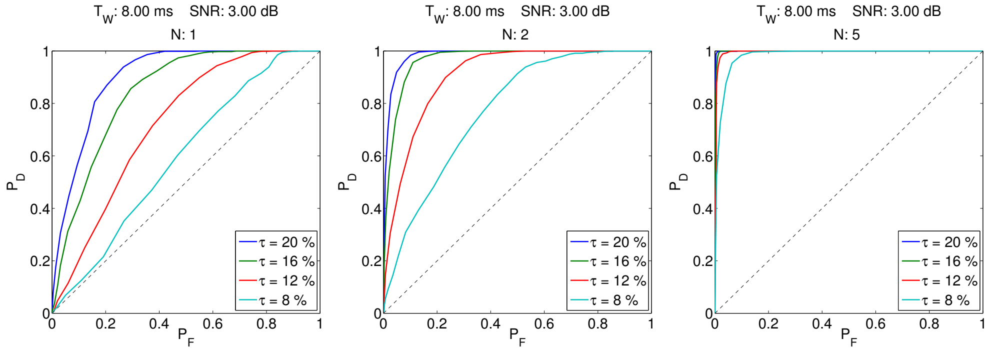
Plotted in Figures 3.4 – 3.6 on the next page are ROC curves which show the probability of correct detection, P_D (*i.e.*, that a signal is DTMF when it is in fact DTMF) versus the probability of false alarm P_F (*i.e.*, deciding that a signal is DTMF when it is in fact **not** DTMF). An ideal detector would have $P_D = 1$ while $P_F = 0$; however, realistic detectors will have make a trade off between detection and false alarm. This trade off is captured in the ROC curve; the ROC curve also allows us to make comparisons between the performance of the detector under different window parameters.

Figure 3.4 on the following page shows how the performance of the DTMF detector is affected by increasing the length of the window (T_w) while all other window parameters



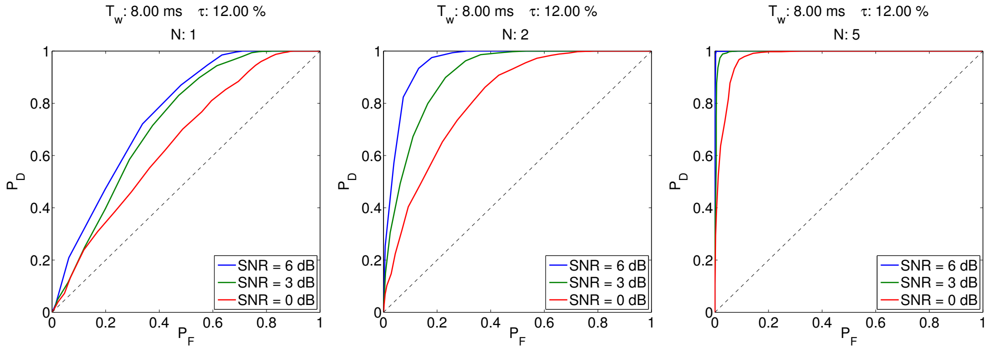
(a) $N = 1$ (b) $N = 2$ (c) $N = 5$

Figure 3.4: ROC curves for the DTMF detector as a function of window length T_w



(a) $N = 1$ (b) $N = 2$ (c) $N = 5$

Figure 3.5: ROC curves for the DTMF detector as a function of duty cycle τ



(a) $N = 1$ (b) $N = 2$ (c) $N = 5$

Figure 3.6: ROC curves for the DTMF detector as a function of SNR

and Signal to Noise Ratio (SNR) are held constant. The number of samples available from each look-through window is $\lfloor \tau f_s T_w \rfloor$, so increasing the window length while holding all other window parameters constant results in a larger number of received signal samples. With a larger number of samples to work with, we would intuitively expect the detector to have better discriminatory performance when deciding between DTMF and non-DTMF. Inspection of Figure 3.4 reveals that as the window length increases, the detector's probability of correct detection (for a fixed probability of false alarm) increases, confirming the intuition that detector performance would be increasing in window length. The effect of varying window length is most clearly seen for the case where only one window ($N = 1$) is used for reaching a decision.

Figure 3.5 shows how the performance of the DTMF detector is affected by increasing the duty cycle (τ) (*i.e.*, decreasing the amount of time in a single window when the signal is not able to be observed) while all other window parameters and SNR are held constant. For a fixed window length, a larger duty cycle equates to a larger look-through window. This means that we have more contiguous data samples for processing and making a decision and we would expect better performance from the detector under these circumstances. Inspection of Figure 3.5 reveals that as the duty cycle increases, the detector's probability of correct detection (for a fixed probability of false alarm) increases, confirming the intuition about the effect of varying duty cycle on the detectors performance.

Finally, Figure 3.6 shows how the performance of the DTMF detector is affected by SNR while all window parameters are held constant. With a higher SNR, the received signal is less corrupted by noise allowing for better discriminating power with fewer received samples. This can be seen by noting that the best performing curve (6 dB) of $N = 2$ (in Figure 3.6(b)) is matched by the performance of the detector at an SNR of 0 dB with $N = 5$ (Figure 3.6(c)); the additional 6 dB of SNR allows for comparable performance with three fewer windows. We also observe near ideal detector performance for only $N = 5$ windows in the majority of cases.

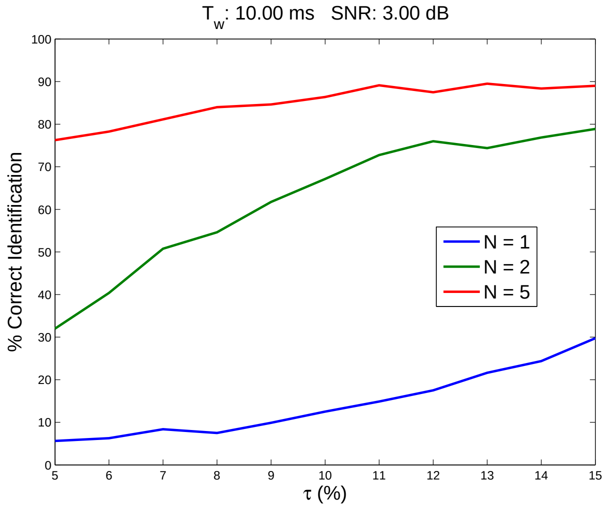


Figure 3.7: DTMF detector's key identification performance

3.8.1 DTMF Key Identification

In addition to determining whether a signal of interest uses DTMF, it is also useful to determine, for signals that use DTMF, which particular key from Table 3.1 (*i.e.*, which pair of frequencies) is being transmitted. To make this determination, after a signal is declared to be DTMF, the values of ω_l and ω_h that solve (3.4) are used as indices in Table 3.1 to find the corresponding DTMF key. Now, in addition to specifying the performance of the detector in determining DTMF versus non-DTMF signals, we must also consider the performance of the detection method at distinguishing individual keys.

Figure 3.7 depicts the conditional probabilities for correctly identifying a particular DTMF tone; this is the probability that, given we have declared the signal of interest to contain DTMF, the detector will also correctly identify the corresponding key. We see that for a fixed duty cycle (*e.g.*, $\tau = 10\%$), moving from one window ($N = 1$) to two windows ($N = 2$) results in a significant improvement in the detector's key accuracy ($12.5\% \rightarrow 67.1\%$); the performance increases even further when moving from two to five windows ($67.1\% \rightarrow 86.4\%$).

3.8.2 DTMF Detection Algorithm

Presented as Algorithm 3.1 is a pseudo-code implementation of the DTMF detector that we have developed. It is a collection and summarization of the key equations and design decisions described above.

Algorithm 3.1 Time domain Dual-Tone Multi-Frequency detection

```
Require: $\mathbf{r}$ the vector of received signal samples
         $\mathcal{I}$ the set of sample indices
         $F_s$ the rate at which the signal was sampled
         $\delta \ll 1$ stopping criteria for Gauss-Newton iteration
         $\eta$ threshold for deciding DTMF
  for all $f_l \in \{697, 770, 852, 241\}$ do
    $\omega_l \leftarrow 2\pi f_l$
    for all $f_h \in \{1209, 1336, 1477, 1633\}$ do
      $\omega_h \leftarrow 2\pi f_h$
      $k \leftarrow 0$
      $\boldsymbol{\phi}^{(0)} \leftarrow [0\ 0]^{\mathrm{T}}$
      $\boldsymbol{\alpha}^{(0)} \leftarrow [1\ 1]^{\mathrm{T}}$
      $\boldsymbol{\theta}^* \leftarrow \boldsymbol{\theta}^{(0)}$
      repeat
        $\mathbf{s}_l^{(k)} \leftarrow \mathbf{s}_l\left(\omega_l, \phi_l^{(k)}\right)$, $\mathbf{s}_h^{(k)} \leftarrow \mathbf{s}_h\left(\omega_h, \phi_h^{(k)}\right)$
        $\boldsymbol{\phi}^{(k+1)} \leftarrow \boldsymbol{\phi}^{(k)} + \left(\nabla_{\boldsymbol{\phi}}\mathbf{s}(\boldsymbol{\theta}^{(k)})\nabla_{\boldsymbol{\phi}}\mathbf{s}(\boldsymbol{\theta}^{(k)})^{\mathrm{T}}\right)^{-1}\nabla_{\boldsymbol{\phi}}\mathbf{s}(\boldsymbol{\theta}^{(k)})\left(\mathbf{r} - \mathbf{s}(\boldsymbol{\theta}^{(k)})\right)^{\mathrm{T}}$
        $\boldsymbol{\alpha}^{(k+1)} \leftarrow \begin{bmatrix} \left\|\mathbf{s}_l^{(k)}\right\|_2^2 & \left\langle \mathbf{s}_h^{(k)}, \mathbf{s}_l^{(k)} \right\rangle \\ \left\langle \mathbf{s}_h^{(k)}, \mathbf{s}_l^{(k)} \right\rangle & \left\|\mathbf{s}_h^{(k)}\right\|_2^2 \end{bmatrix}^{-1} \begin{bmatrix} \left\langle \mathbf{s}_l^{(k)}, \mathbf{r} \right\rangle \\ \left\langle \mathbf{s}_h^{(k)}, \mathbf{r} \right\rangle \end{bmatrix}$
        $k \leftarrow k + 1$
      until $\left|\varepsilon_n(\boldsymbol{\theta}^{(k-1)}) - \varepsilon_n(\boldsymbol{\theta}^{(k)})\right| < \delta$
      if $\varepsilon_n(\boldsymbol{\theta}^{(k)}) < \varepsilon_n(\boldsymbol{\theta}^*)$ then
        $\varepsilon_n(\boldsymbol{\theta}^*) \leftarrow \varepsilon_n(\boldsymbol{\theta}^{(k)})$
        $\boldsymbol{\theta}^* \leftarrow \boldsymbol{\theta}^{(k)}$
      end if
    end for
  end for
  if $\varepsilon_n(\boldsymbol{\theta}^*) < \eta$ then
    Declare $\mathbf{r}$ to be DTMF and use $\boldsymbol{\omega}^*$ to determine key
  end if
```

Table 3.3: Maximum recommended frequency variation

Low Frequencies		High Frequencies	
697 Hz	684.454 – 709.546 Hz	1209 Hz	1187.238 – 1230.762 Hz
770 Hz	756.140 – 783.860 Hz	1336 Hz	1311.952 – 1360.048 Hz
852 Hz	836.664 – 867.336 Hz	1477 Hz	1450.414 – 1503.586 Hz
941 Hz	924.062 – 957.938 Hz	1633 Hz	1603.606 – 1662.394 Hz

3.9 Non-Ideal DTMF: Frequency Deviation

In the previous sections, we made the assumption that the pair of frequencies in a DTMF signal could only be one of 16 possible pairs and in fact this assumption was exploited in the design of our detection algorithm. However, the ITU recommendation that standardizes DTMF specifies that the transmitted frequencies must be within ±1.8% of the nominal frequency [17]. Table 3.3 lists the ranges of allowed variance for each of the DTMF frequencies. In some circumstances, it is possible that observed frequency deviations might be even higher than ±1.8% (and the ITU recommendation for building DTMF receivers suggests that this possibility be taken into account [18]). An approximate maximum likelihood estimate for frequency involves the computation of the Power Spectral Density (PSD) of the received signal [19]; because of the windowing, though, this approach yields unsatisfactory performance, even with a significantly larger number of windows. Techniques based on moment matching and expectation maximization [19] were also attempted with even worse performance.

Naïve application of the developed technique to non-ideal DTMF results in the ROC curve shown in Figure 3.8 on the next page, where each DTMF tone was generated using sinusoid frequencies that were uniformly distributed ±1.8% from the nominal values (uniformly distributed in the ranges specified in Table 3.3). At first glance, this graph might appear to have rather counterintuitive behavior, as there is a decrease in performance as we increase from two to five to ten windows.

The explanation is that even with 10 windows, we are only observing a fraction of the total period of the DTMF tone. With only one window, there are so few samples

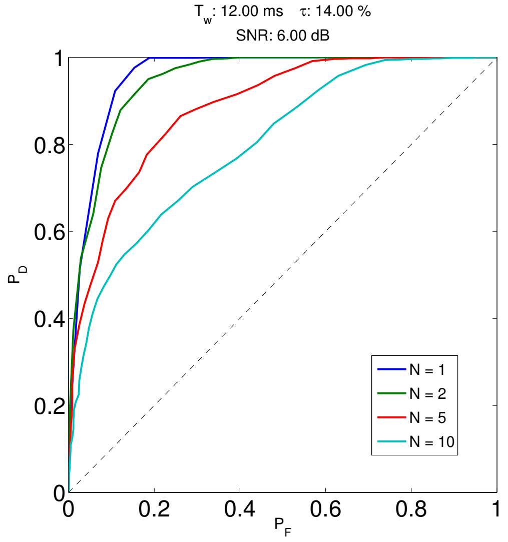


Figure 3.8: ROC curves for the DTMF detector with the non-ideal effect of frequency deviation (sinusoid frequencies uniformly distributed ±1.8% about nominal)

that it is easy to find a good fitting DTMF model. As the number of windows increases, though, the impact of the frequency deviation becomes larger, resulting in a poorer fit. Nevertheless, despite the performance hit when considering the non-ideal effects of frequency deviation, the detector's performance is significantly better than random coin flipping (which has the ROC curve given by the dashed line in the figure, $P_D = P_F$).

3.10 DTMF Feature Extraction Conclusions

While methods exist for DTMF detection and decoding, they have poor performance in scenarios where there is limited data availability due to periodic windowing of a signal that occurs under certain circumstances. We have developed a technique for detecting and decoding DTMF signaling based on minimizing the mean squared error between an idealized model for DTMF tones and the received signal. Using this technique, detection with a high probability of correct detection and a low probability of false alarm is possible in very low SNR environments and only requires a few look-through windows worth of data. The developed technique can be extended for other non-DTMF multiple-frequency signaling schemes through the development of an appropriate parametric model. It is also possible to extend this technique to digital modulation schemes such as Amplitude Shift Keying (ASK), Frequency Shift Keying (FSK), and Phase Shift Keying (PSK). Development of these additional detectors would provide additional features for the classification of RF signals and provide higher levels of discriminatory power for the detection and classification of target sequences.

4. Classification

The basic problem of classification involves determining to which *class* an *object* belongs, given that all objects are known to be members of one distinct class. This problem is exemplified in Chapter 3, where each observed RF signal is an object that is to be identified as being a member of one of the two mutually exclusive classes, CONTAINS-DTMF and DOES-NOT-CONTAIN-DTMF. A separate classification problem is solved when we place objects of class CONTAINS-DTMF into one of 16 mutually exclusive classes representing each of the 16 DTMF keys.

In the context of the larger scope of this paper, the *objects* that we seek to classify are the sequences of signals that are observed in the RF environment. We wish to classify every sequence as being either a TARGET-SEQUENCE or a NON-TARGET-SEQUENCE based on the target sequence signatures being employed.

4.1 Linear Classification

The classification problem can be seen in the example of BLACK signals and WHITE signals. Here, BLACK and WHITE are classes of signals.

The classification begins with the identification of suitable features for BLACK and WHITE signals. In this case, we measure two quantities for every signal, X_1 and X_2 (representative of any of the features that we previously discussed in Chapter 2). With a group of sixteen signals (eight BLACK and eight WHITE) plotted on the feature axis as in Figure 4.1 on the following page, one can see that a straight line is easily capable of separating the two classes of signals. The goal of linear classification is to identify the straight line, called the **discriminant function**, that separates one class from another.

This is a simple example of a linear classifier, which makes a decision based on a linear combination of the characteristics it uses. Unfortunately many decision problems do not lend themselves to this kind of simple process.

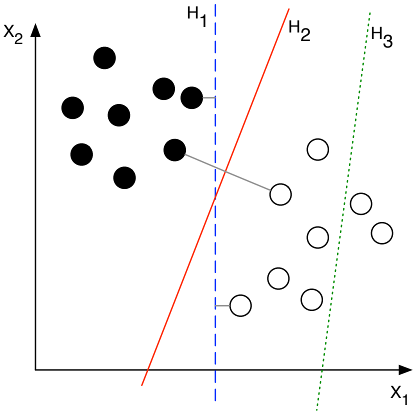


Figure 4.1: Black and White signals in 2-space with 3 discriminant functions. H_3 doesn't separate the 2 classes. H_1 does but with a small margin while H_2 separates Black from White with the maximum margin.

4.2 Linear Classifiers in Radio Frequency Communications

As described in Chapter 2, our classifier relies on a series of snapshots of the RF environment. The various signals, or communications events, that are detected will be identified and the features associated with each event will be identified. Each communication event is described by a feature vector whose elements are descriptive characteristics of a signal (*i.e.*, power, bandwidth, the presence of a DTMF sequence, direction of arrival, etc.). Each communication event detected during a sample period can thus be thought of as a point in d-dimensional space (one dimension for each feature in the vector). The use of a discriminative model implies that we do not know the underlying probability densities from which the communications and their features are generated, but we assume that the form of the function that separates a target-class communication from a non-target-class communication (the discriminant function) is a linear combination of the features. In the limited sense that the prior densities are assumed unknown, discriminative linear classifiers are considered non-parametric methods. The attractiveness of linear classifiers lies in their simplicity and tractability.

In its simplest form, the linear discriminant function $g(\mathbf{x})$ is a linear combination

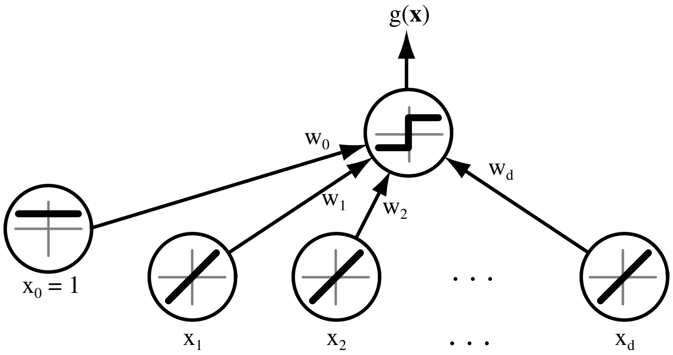


Figure 4.2: Diagram of the linear classifier. The classifier has d input units corresponding to the values of the features of a measured communication. Each input feature x_i is multiplied by its corresponding weight w_i. The single bias unit always emits a value of 1. The final output unit $g(\mathbf{x})$ emits a $+1$ if $\mathbf{w}^{\mathrm{T}}\mathbf{x} > 0$ and a -1 otherwise.

of weighted features, where the features of a communication are denoted $x_1, x_2, \ldots, x_d$ and the associated weights are denoted $w_1, w_2, \ldots, w_d$ as in (4.1).

$$g\left(\mathbf{x}\right) = w_0 + \sum_{i=1}^{d} w_i x_i \tag{4.1}$$

Here, w_0 is called the *bias* and the negative of the bias is called the *threshold*. We can rewrite the discriminant function using vector notation as

$$g\left(\mathbf{x}\right) = \mathbf{w}^{\mathrm{T}}\mathbf{x} + w_0. \tag{4.2}$$

The form of (4.2) suggests that we are seeking a hyper-plane in $d - 1$ dimensions that will separate the data points so that on one side there are target communications and on the other side are non-target communications. This structure is indicated in Figure 4.2.

Since there are two classes, $\mathcal{C}_1$ (target) and $\mathcal{C}_2$ (non-target), a communication event described by the vector $\mathbf{x}$ is assigned to class $\mathcal{C}_1$ if $g(\mathbf{x}) \geq 0$ and to class $\mathcal{C}_2$ otherwise, as depicted in Figure 4.2. Therefore, the *decision surface* is determined by the relation

$g(\mathbf{x}) = 0$. The decision surface divides the feature space by a hyper-plane. The orientation of the decision surface is determined by the weight vector $\mathbf{w}$ and the location of the boundary is determined by the threshold w_0. To see this, consider two signals described by $\mathbf{x}_A$ and $\mathbf{x}_B$ that are both on the decision boundary, then

$$\mathbf{w}^{\mathrm{T}}\mathbf{x}_A + w_0 = \mathbf{w}^{\mathrm{T}}\mathbf{x}_B + w_0,$$

or

$$\mathbf{w}^{\mathrm{T}}\left(\mathbf{x}_A - \mathbf{x}_B\right) = 0,$$

which shows that $\mathbf{w}$ is perpendicular to any vector in the hyper-plane.

4.2.1 Computing the Weight Vector

Estimates of the classifier weights are obtained through the use of *labeled* sample data obtained through simulation, experiments, or descriptive reports. The data should be labeled correctly. The training data are provided as pairs $\{\mathbf{X}, \mathbf{s}\}$, where $\mathbf{X}$ contains descriptions of communication signals and $\mathbf{s}$ is a vector of labels, one for each communication. Thus, $\mathbf{X}$ is an $N \times d$ matrix where each row represents a single communication event, $\mathbf{x}_i$, described by d features listed across the columns. The label vector is a $N \times 1$ vector with components equal to either $+1$ or -1 (a value of $+1$ in the ith position indicates that the communication described by $\mathbf{x}_i$ is labeled "target").

The methods used to compute the classifier weights are based on some assumptions about the data. The first is that the parameters of the underlying distribution are stationary and the values are independent (*i.e.*, the occurrence of a particular communications event does not affect the likelihood of any other communications event also occurring). In order to simplify the calculations, we also assume that the data are zero-mean. To meet this requirement, we simply subtract the mean of each column in the $\mathbf{X}$ matrix from the entries in the respective column. We also require that the number of labeled samples N be large enough to estimate the cross-correlation and the correlation of the true underlying distributions.

Lastly, an additional modification of the data is performed in order to compute the bias or threshold. A column consisting of all ones is appended to $\mathbf{X}$; this is a dummy variable, $\mathbf{x}_0$ corresponding to the weight w_0.

With this structure in place, we proceed with determining the weight vector by one of two methods.

Linear Minimum Mean Squared Error Estimator The first approach to determining the linear classifier weights is based on a linear estimation technique taken from a class of estimators generally called *Wiener filters* [19]. Given labeled target data, where $s_i \in \{\pm 1\}$ is the label for the i^{th} communication event ($+1$ denotes a target signal and -1 denotes a non-target signal), we can find appropriate weights, $\mathbf{w}$, for the d features that describe our communication signals in order to get an estimator ($\hat{\mathbf{s}}$) for $\mathbf{s}$:

$$\begin{aligned} \hat{s}_i &= \mathbf{w}^{\mathrm{T}} \mathbf{x}_i \\ &= w_0 + w_1 x_{1,i} + w_2 x_{2,i} + \ldots w_d x_{d,i}. \end{aligned} \tag{4.3}$$

We desire that the weights be chosen so as to minimize the expected mean squared error $\mathbb{E}\left[(\mathbf{s}-\hat{\mathbf{s}})^2\right]$ for new unlabeled data. Equation (4.3) describes the classifier consisting of a weighted sum of the features of each new communication; the result of the calculation is compared to a threshold.

$$\hat{s}_i \underset{\text{non-target}}{\overset{\text{target}}{\gtrless}} \tau \tag{4.4}$$

If s_i exceeds the threshold, then the i^{th} communication event (s_i) is designated a target signal. Thus, the coefficients in $\mathbf{w}$ determine whether a signal is part of a target sequence or not. This estimator is the Linear Minimum Mean Squared Error (LMMSE) estimator, and it is optimal in the mean squared sense, that is, it is the best *linear* estimator assuming stationarity of the parameters of the underlying probability distributions ([28], Section 6.1).

In order to train the estimator, we label each row of the training data with the value $s_i \in \{\pm 1\}$; here $\tau = 0$. Then, we estimate the correlation $\mathbf{R}_{sx} \in \mathbb{R}^{1\times d}$ between $\mathbf{s}$ and $\mathbf{x}$,

$$\mathbf{R}_{sx} = \frac{1}{N}\sum_{i=1}^{N} s_i \mathbf{x}_i, \tag{4.5}$$

and the autocorrelation $\mathbf{R}_{xx} \in \mathbb{R}^{d\times d}$ of $\mathbf{x}$,

$$\mathbf{R}_{xx} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{x}_i \mathbf{x}_i^{\mathrm{T}}. \tag{4.6}$$

The desired weights for the features we seek are then given by

$$\mathbf{w} = \mathbf{R}_{xx}^{-1}\mathbf{R}_{sx}. \tag{4.7}$$

By using this approach we are able to obtain appropriate weights for each feature based on *labeled data*, which makes the feature reduction simple: we simply throw out features with negligible weights to eliminate non-discriminatory features. Furthermore, we have an estimator in (4.3) that can be used to decide whether an unlabeled signal x_i is likely a target signal or not. This linear estimator is considered optimal in the *mean squared error* sense if the parameter s is assumed to be stationary and the values s_i are assumed to be independent. The mean squared error can be used to rate the performance of the estimator in classifying target signals.

Least-Squares Error (LSE) and the Pseudo-inverse A second approach, one that does not make any assumptions about the statistical structure of the data, is the *Least-Squares Error* (LSE) criterion [19]. We seek a vector $\mathbf{w}$ such that $\mathbf{X}\mathbf{w} = \mathbf{s}$. If $\mathbf{X}$ were non-singular, then we could simply compute $\mathbf{w} = \mathbf{X}^{-1}\mathbf{s}$ and obtain a solution. However, the data matrix will typically have more rows than columns, and so $\mathbf{w}$ will be over-determined. Instead, we can obtain a vector $\mathbf{w}$ that minimizes a function of the error between $\mathbf{Xw}$ and $\mathbf{s}$. The classical approach is to minimize the squared length of the error vector, $\mathbf{Xw}-\mathbf{s}$, which leads to the LSE criterion function:

$$\begin{aligned} J(\mathbf{w}) &= \|\mathbf{Xw}-\mathbf{s}\|^2 \\ &= \sum_{i=1}^{N}\left(\mathbf{w}^{\mathrm{T}}\mathbf{x}_i - s_i\right)^2. \end{aligned}$$

A simple closed-form solution can be obtained by forming the gradient with respect to $\mathbf{w}$

$$\begin{aligned} \nabla_{\mathbf{w}} J &= \sum_{i=1}^{N} 2\left(\mathbf{w}^{\mathrm{T}}\mathbf{x}_i - s_i\right)\mathbf{x}_i \\ &= 2\mathbf{X}^{\mathrm{T}}\left(\mathbf{Xw}-\mathbf{s}\right) \end{aligned}$$

and setting it equal to zero, yielding the necessary condition

$$\mathbf{X}^{\mathrm{T}}\mathbf{X}\mathbf{w} = \mathbf{X}^{\mathrm{T}}\mathbf{s}. \tag{4.8}$$

The $d \times d$ matrix $\mathbf{X}^{\mathrm{T}}\mathbf{X}$, known as the *sample correlation matrix*, is square and typically non-singular, so we can solve for $\mathbf{w}$ uniquely as

$$\mathbf{w} = \left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{s}. \tag{4.9}$$

The expression $\left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}\mathbf{X}^{\mathrm{T}}$ is known as the *Moore-Penrose pseudo-inverse* [15]; it is a generalization of the notion of a matrix inverse to non-square matrices. The `pinv` command in MATLAB can be used to compute the pseudo-inverse of a matrix, so a simple procedure for computing the weights and training estimates is given by Algorithm 4.1.

Using either method to determine $\mathbf{w}$ allows for the linear classification to be implemented by the procedure given in Algorithm 4.2.

Algorithm 4.1 Least-Squares Error linear classifier training

```
Require: $\mathbf{X}$ the $N \times d$ feature matrix
  $\mathbf{s}$ the class label vector
  $\mathbf{X} \leftarrow [\mathbf{X}\ \mathbf{1}_N]$
  $\mathbf{X}^\dagger \leftarrow \left(\mathbf{X}^\mathrm{T}\mathbf{X}\right)^{-1}\mathbf{X}^\mathrm{T}$
  $\mathbf{w} \leftarrow \mathbf{X}^\dagger\mathbf{s}$
  return $\mathbf{w}$ the LSE linear classifier weights
```

Algorithm 4.2 Linear classification process

```
Require: $\mathbf{w}$ the $d+1$-dimensional linear classifier weights
  $\mathbf{x}$ the $d$-dimensional feature vector for an observed signal
  $\mathbf{x} \leftarrow [\mathbf{x}\ 1]$
  $\hat{s} \leftarrow \langle \mathbf{w}, \mathbf{x} \rangle$
  if $\hat{s} > 0$ then
    Observed signal described by $\mathbf{x}$ is a target signal
  else
    Observed signal described by $\mathbf{x}$ is a non-target signal
  end if
```

4.2.2 Example

To clarify how the classifier works, we consider an example. Each communication event is represented as a point in m-dimensional space (a dimension for each feature). The form of (4.3) suggests that we are seeking a hyper-plane in $m-1$ dimensions that will separate the data points so that on one side there are target communications and on the other side are non-target communications. To collect training and testing data for the linear classifier, contrived scenarios were captured from an urban environment in Camden, New Jersey. The noisy background RF environment consisted of several Very High Frequency (VHF) and Ultra High Frequency (UHF) phone paging systems, public safety communications, commercial and private Push To Talk (PTT) walkie-talkies, and cellular telephone transmissions. A simple three transmitter sequence was developed for use as an injected "target" sequence.

Table 4.1: Sampling of the more than $36,000$ signals that were observed during system testing and used for training and testing the linear classifier

Time (s)	CF (MHz)	BW (kHz)	Power (dBm)	DTMF key	Latitude	Longitude
290.2836	461.0404	3.9063	-95.0132	–	39.9512	-75.1257
290.2836	461.1250	4.6875	-86.5957	–	39.9512	-75.1257
290.3206	462.4902	1.9531	-82.2488	–	39.9512	-75.1257
290.3206	462.6880	1.5625	-66.8078	1	39.9512	-75.1257
290.7489	463.2815	8.5938	-78.6614	–	39.9512	-75.1257
290.7489	463.4033	5.8594	-78.6820	–	39.9512	-75.1257
290.7905	464.6566	1.5625	-85.1937	–	39.9512	-75.1257
291.0104	851.1701	12.5000	-65.2993	–	39.9512	-75.1257
291.0556	853.1688	4.6875	-85.9458	–	39.9512	-75.1257
291.0903	853.6689	8.2031	-78.0500	–	39.9512	-75.1257

In the following example, the observed signals are described by the following seven features:

(1) the time the RF signal was observed,

(2) the center frequency (CF) of the RF signal,

(3) the bandwidth (BW) of the RF signal,

(4) the received power of the RF signal,

(5) whether the RF signal contains a DTMF tone,

(6) the latitude of the receiver when the RF signal was observed, and

(7) the longitude of the receiver when the RF signal was observed.

Note that these features were chosen based on perceived usefulness in distinguishing target communication events along with their relative ease of calculation. The target sequence consisted of communications utilizing differing center frequencies, transmit powers, and channel usage (*i.e.*, voice versus DTMF).

In order to train the classifier, over 36,000 communication events were observed, of which fewer than 1% were a part of the target sequence. Target signals and non-target signals were labeled accordingly in the incoming data. A subset of the observed signals

(10 communication signals out of over $36,000$) that were used to test the trained classifier is shown in Table 4.1. After collecting the training data, we trained the classifier and computed the coefficients, $\mathbf{w}$. For this experiment, we used both LMMSE and LSE methods and obtained identical weight vectors

$$\mathbf{w} = \{0.0000,\ 0.0000,\ 0.0001,\ 0.0001,\ 0.2186,\ 0.0000,\ -0.1278,\ -0.9975\}$$

where the value -0.9975 is the bias term w_0. Thus the classification test was formulated as

$$0x_1+0x_2+0.0001x_3+0.0001x_4+0.2186x_5+0x_6-0.1278x_7-0.9975 \underset{\text{non-target}}{\overset{\text{target}}{\gtrless}} 0. \quad (4.10)$$

Looking at the values of the linear classifier weights, the two features with the largest coefficients are the presence of DTMF on the channel and the longitude of the receiver when the signal was observed; since these features have the largest coefficients in the classifier, they would provide the most discriminative representation of the data in this example. Also, notice that the coefficients for the first feature (time signal was observed) and the sixth feature (latitude of the receiver when the signal was observed) are zero, indicating that they provide no useful information to separate signal classes in this example.

Running the unlabeled training data through this linear classifier, we obtain a probability of correct detection (i.e. the probability that we correctly label a target signal as being a target signal) of only 0.3125 with a probability of false alarm (i.e. the probability of incorrectly identifying a non-target signal as being a target signal) of 0. Applying the linear classifier to an unlabeled data set that was not used in training, the probability of correct detection decreases to 0.2857. The linear classifier has unacceptably low performance even when being run on the same data sets that were used to train it and this is the case where we would expect the best performance. It can be concluded that a detection and classification system based on linear classification does not correctly identify enough target signals to be useful.

The reason for such poor performance from the linear classifier is that with the selection of features listed above, the classification problem is not *linearly separable*; no hyper-plane through the feature space will sufficiently separate target signals from non-target signals. To see this, consider Figure 4.3 on the next page, which shows a subset of all of the 2-dimensional feature subspaces of the 7-dimensional feature space in this example. Non-target signals are plotted as a blue asterisks, while target signals are plotted as red squares. Looking at the plot of DTMF key index versus longitude (which should have the most discriminating representation of the data in this example), it is clear that any linear discriminant function will result in misclassifying all non-target signals as being target signals or in declaring several target signals to be non-target signals. This is true regardless of which feature subspace is considered.

4.3 Extension to Multiple Layers

A single layer linear classifier as described above requires labeled data sets that are known to be linearly separable in order to function properly. When the data are not linearly separable, it is sometimes possible to combine several layers of linear classifiers to build a successful classifier. A Multi-Layer Perceptron (MLP) is a multi-layer realization of the test given by (4.3) and (4.4). It can be viewed as a *weighted sum of features* detector, with optimal weights that are learned from labeled training data. A simple MLP is illustrated in Figure 4.4 on page 61.

MLPs have the advantage that they are capable of realizing more complicated decision surfaces than a simple linear classifier alone. However, they are somewhat harder to design than a single-layer realization of (4.3) and (4.4).

4.4 Sequential Nature of Target RF Communication Events

The classification techniques described in this chapter, including both the simple linear classifier and the Multi-Layer Perceptron, make classification decisions based on a

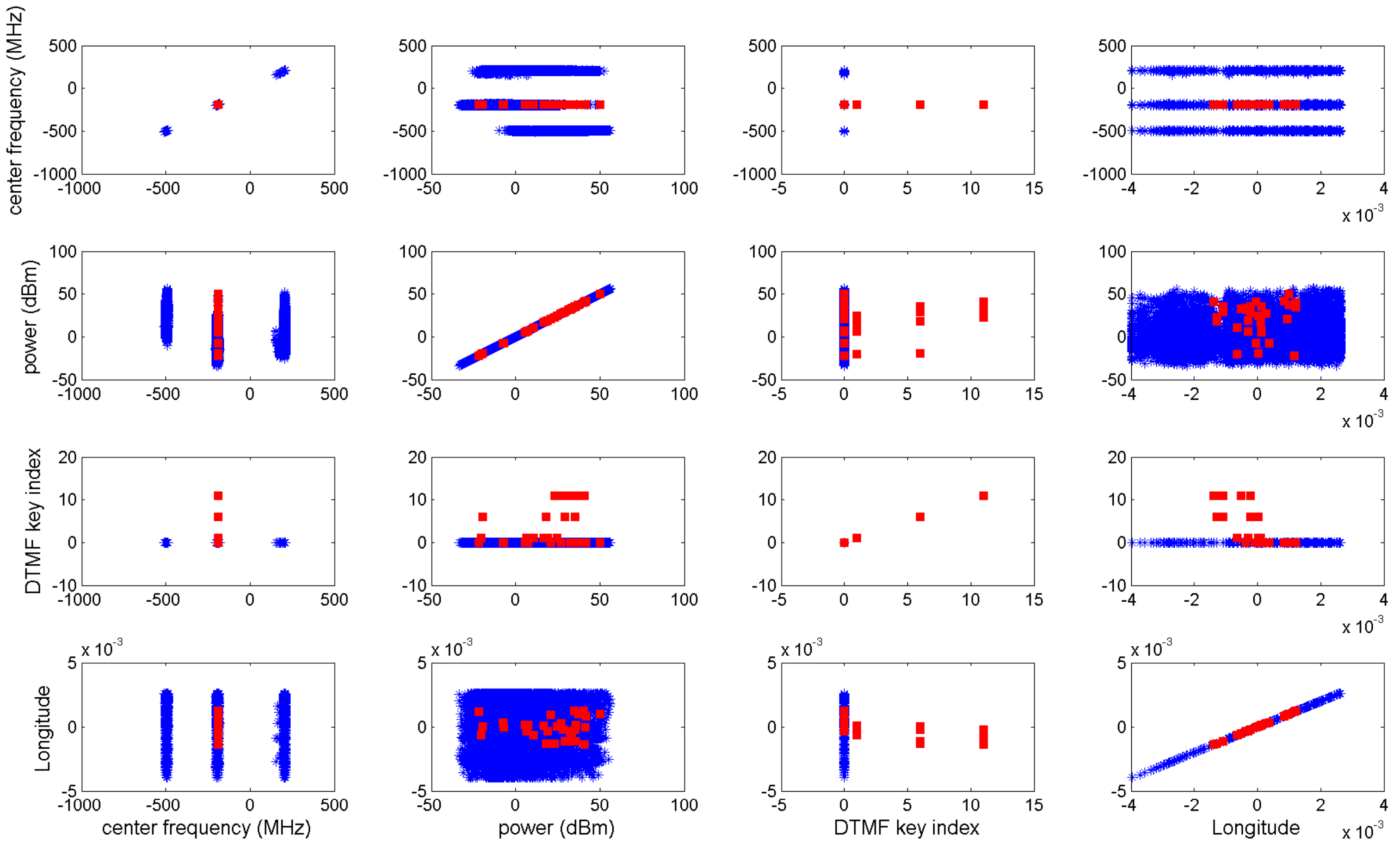


Figure 4.3: Two-dimensional projections of the 7-dimensional feature space. Non-target signals are shown as blue asterisks, while target signals are shown as red squares. Any linear discriminant function will result in misclassifying either some target signals, some non-target signals, or both, due to the overlap of the two classes of signals in every feature subspace.

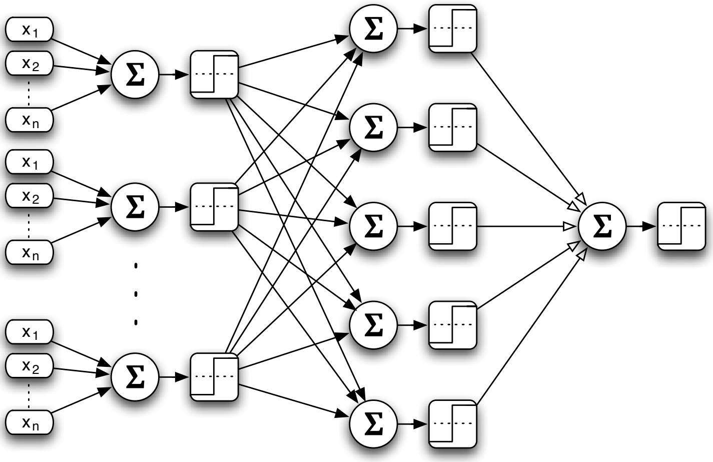


Figure 4.4: Example of a multi-layer perceptron. All inputs x_i feed into individual perceptrons with distinct weights and thresholds for each perceptron (left half). The outputs of these perceptrons can then be used as inputs to a second layer of perceptrons (right half).

snapshot of the decision surface at a single point in time. In the case of classifying target RF signals, these techniques make decisions based on the state of the RF environment at a single point in time. While the algorithms may be run multiple times at regular intervals (as in our final implementation), subsequent decisions will have no inherent knowledge of, or dependence on, previous decisions.

The sequential nature of target RF communication events is a result of some signals being benign when observed in isolation but target when observed as a part of a temporal sequence with other signals. In order to distinguish between these two cases, a classification sequence with some element of memory must be used.

4.5 Profile Hidden Markov Model

One advanced classification technique to capture the element of time needed for this problem is the Profile Hidden Markov Model (PHMM) [20]. The PHMM, setup in the proper manner, would be able to detect the target signature patterns (sequences) within

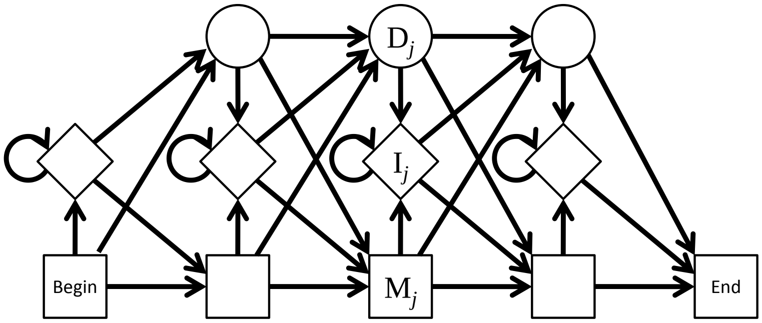


Figure 4.5: Profile Hidden Markov Model. Each square block represents a "match" state (M_j), where that state represents one stage of the target signature. The number of match states is equal to the number of steps in the target signature. Between each match state are (optional) diamond blocks (I_j), representing inserted signals between each match state. A path through circular blocks (D_j) is also present, representing deleted or skipped match states that were not observed. Each arrow represents the manner in which flow may move from one block to another; the model includes a probability associated with each arrow.

a series of feature vectors that are provided as input.

The PHMM model (shown in Figure 4.5) describes a target signature (or "profile") for a multi-step target signature, similar to the one described in Section 4.2.2 on page 56. The model works by defining a match state for each step of the target signature, and then adding optional insert and deletion states to allow for non-target signals to be observed between each matched target signal, or for one or more steps in the target signature to be missing from the set of observed signals.

The probabilities associated with each transition shown in Figure 4.5 are determined through a training process (*e.g.*, Baum-Welch [2]). Then, with the model built, we can determine the probability of the collected observations being made given the model parameters. Because there are exponentially many possible state sequences, this problem is typically solved with a dynamic programming algorithm (*e.g.*, the *Forward Algorithm*) [27].

4.6 Example with PHMMs

To show that the PHMM is a feasible solution to our classification problem, the data that were collected from the example exercise of Section 4.2.2 on page 56 were processed using a PHMM as the classifier instead of a linear classifier.

A PHMM was constructed in MATLAB using the HMM Toolbox for MATLAB [24]. The form of this PHMM is the same as in Figure 4.5 on the previous page, with one match state for each step in the three stage example sequence.

The training process consisted of developing two sets of transition parameters, one to represent the model when the target signature is known to be present (the target or profile model), and the other for when the signature is known to be absent (a background model). For this example, the target model was trained with 8 datasets all collected from the same example scenario (*training sets*). The background model was similarly trained using data collected over an extended period of time.

For both models, a log-likelihood value is generated that indicates the probability density of the observed data, D, for each of the models. By taking the difference of the two log-likelihood values, we can form a log-likelihood ratio

$$\ln\left[L(H_P, H_B|D)\right] = \ln\left[L(H_P|D)\right] - \ln\left[L(H_B|D)\right]$$

where H_P represents the hypothesis that the data were generated by the target signature profile model and H_B represents the hypothesis that the data were generated by the background model [8]. This log-likelihood ratio is positive for the case where it is more likely that the data were generated by the target model and not by the background model. Negative values indicate the opposite, that it is more likely that the data were generated by the background model.

This log-likelihood ratio forms the basis for Figure 4.6 on the following page, which depicts the output of the PHMM classifier as this ratio versus time in relation to the three events that form the target sequence.

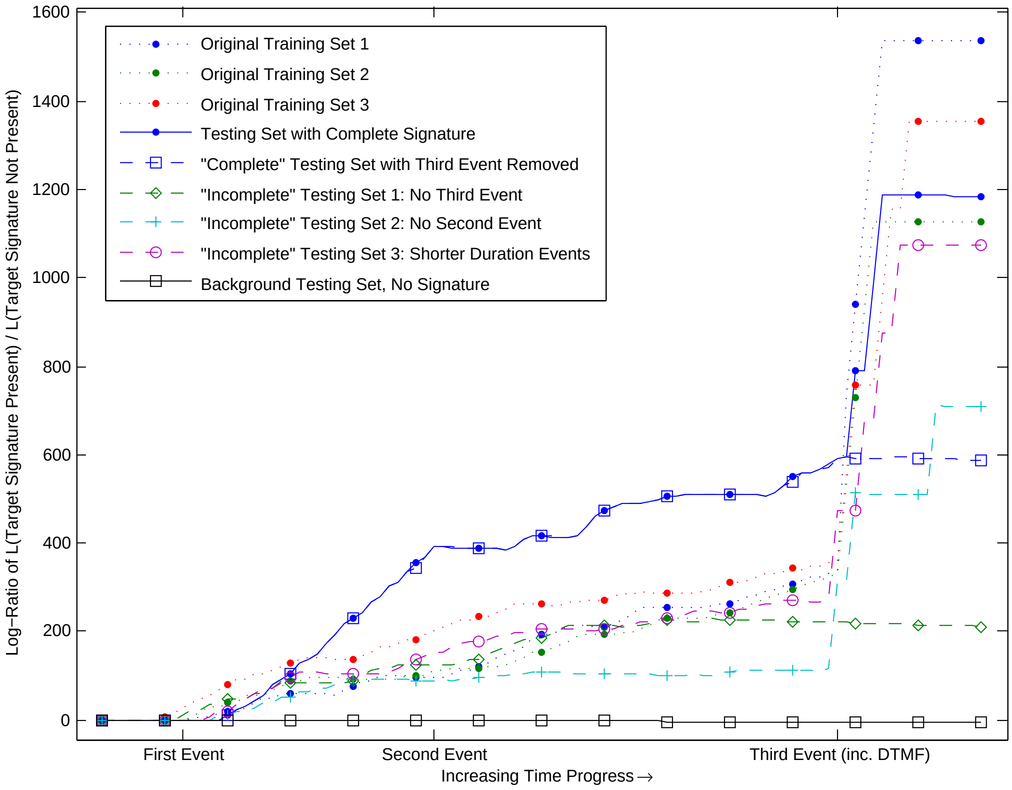


Figure 4.6: Profile Hidden Markov Model classifier test results. The vertical axis shows the log-likelihood ratio of the likelihood of the target signature model having generated the observed data to the likelihood of the background noise model having generated the observed data; the higher the value, the more likely it is for the observed data to contain the target signature. The three dotted "Original Training Set" series show the results of running the classifier on data used to train the target signature model. The "Testing Set with Complete Signature" is a unique series collected in the same manner as the training sets but not used in training. The three "Incomplete Testing Set" series represent data that were collected with the attributes shown (portions of the target signature were intentionally omitted when collecting the data), while the "Complete Testing Set with Third Event Removed" represents all the same data as the first "Complete" testing set except for the signals comprising the third event, which have been manually removed from the data set. The "Background Testing Set" is a representation of data that was collected in the same environment as the other data but without any signals from the target signature present.

To test the performance of the model, the PHMM classifier was used to analyze separate data sets, each with varying degrees of similarity to the original scenario (*testing sets*).

Three of the data sets that were used in training the target model are shown, each indicating relatively high ratios. While this is to be expected, when we use a testing set (a data set that was not used in training, but otherwise represents similar data), the classifier returns similar numbers.

One advantage of using the PHMM as a classifier is the ability to identify partial matches. This is seen in the testing sets where the observed target sequence is different than the sequence used when collecting training data. The classifier is able to identify testing sets that are missing steps in the target sequence or where the duration of the events in the sequence are altered.

To ensure that the classifier is not simply marking everything as a target sequence, we also show the result of passing in a unique background testing set. The log-likelihood ratio value in this case remains non-positive, indicating that it is more likely that the observed data came from the background model than from the target model.

As was mentioned, the log-likelihood ratio values that are graphed in Figure 4.6 are dependent on both the likelihood values for the target model and the background model. while the data for the signals that are a part of the target sequence remain mostly the same for each data set, the background signals in each are different (in some cases, they differ greatly). The difference in background signals drive the variation that is seen in the log-likelihood ratio level for each data set. There is no particular threshold that can guarantee the presence of the target sequence in the observed data; the threshold must be set based on the observed conditions in each particular operating area.

The classification process can be continued further by identifying the specific signals that comprise the target sequence. One algorithm, Viterbi Alignment, uses dynamic programming to determine the most probable sequence of states traversed by the PHMM [12]. Since each match state corresponds to a particular element of the target sequence signature, specific signals may be identified as being target signals.

The performance achieved by the PHMM classifier is excellent, particularly when contrasted to the 31% correct detection rate that was achieved by the linear classifier. The good performance, coupled with the versatility of being able to detect partial matches to signatures, shows the PHMM classifier to be ideal for use as the sequence classifier in the target signal sequence detector.

5. Conclusion

The primary objective of this work has been to automatically identify in real-time RF communications that match a target signature predetermined to have a high correlation to an anticipated action or event. We assume that target RF signals occur in temporal sequences of signals that can be characterized with a parameterized signature.

The goal in building an automatic detector is to develop a classifier that operates on observed characteristics (features) of RF signals and yields decisions as to whether the signals corresponding to these observations match the target signature. Furthermore, this classifier should be able to easily explain the rationale behind the decisions it yields. This rationale is useful for operational personnel who must use the output of the classifier to determine appropriate counteractions. Due to possible physical limitations that would prevent continuous monitoring of a particular frequency, the classifier must also be able to operate on sampled data that is limited to being collected only during windows when the frequency is “visible” to the RF detector.

In order to build such a classifier, we prescribed a list of features in Chapter 2 and Appendix A that can be used to describe a detected signal. A target signal detection system uses these features, provided from hardware sensors and/or signal processing as inputs. We also outlined an architecture for a target signal detection system that uses these inputs to make decisions as to whether a sequence matching a predetermined target signature appears in the RF environment.

The final step of the target signal detection process is to run algorithms that operate on the observed features and do inference on a statistical model to determine whether a target signature is present in the data. While a linear classifier provides a straightforward means of classifying data by defining a linear threshold to separate signals that match a target signature from those that do not, we showed in Chapter 4 that the data under consideration are not linearly separable. Accordingly, we considered a multi-layer perceptron, which is a weighted sum of features detector that combines multiple layers of linear classifiers. This approach, however, does not provide decisions for which a rationale can be easily deduced. Furthermore, neither the linear classifier nor the MLP provide a means of considering the sequential nature of a target signature in the time domain. The Profile Hidden Markov Model was finally identified as the best classification model due to its ability to detect temporal sequences of signals, and we show that it succeeds in distinguishing target signal sequences from the RF environment where the linear classifier fails.

In this document, we also described the process of extracting a feature relevant to signal classification and used in the implemented system, namely the presence of DTMF on an observed channel. In Chapter 3, we presented a detailed time-domain approach to detecting DTMF on an RF channel where physical limitations impose windowing constraints on the detected signal. This detection approach was implemented in MATLAB and shown to provide significant improvement over other classical methods of DTMF detection.

The detection methods presented in this thesis have been demonstrated to be effective for detecting signals conforming to signatures of target communication in real-time in an RF environment.

Bibliography

[1] Amoroso, F. The bandwidth of digital data signals. *IEEE Communications Magazine 18*, 6 (1980), 13–24. `doi:10.1109/MCOM.1980.1090291`.

[2] Baum, L. E., Petrie, T., Soules, G., and Weiss, N. A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *The Annals of Mathematical Statistics 41*, 1 (1970), 164–171. `doi:10.1214/aoms/1177697196`.

[3] Bertsekas, D. *Nonlinear Programming.* Athena Scientific, 2008.

[4] Bowers, C. A., Jentsch, F., Salas, E., and Braun, C. C. Analyzing communication sequences for team training needs assessment. *Human Factors: The Journal of the Human Factors and Ergonomics Society 40*, 4 (1998), 672–679. `doi:10.1518/001872098779649265`.

[5] Callahan, Jr., M. Integrated DTMF receiver. *Communications, IEEE Transactions on 27*, 2 (Feb. 1979), 343–348. `doi:10.1109/TCOM.1979.1094411`.

[6] Capitaine, T., et al. Signal processing algorithm on a low resource processor: DTMF strings identification on a DIL8 package microcontroller. In *Industrial Electronics, IEEE International Symposium on* (June 2008), pp. 1050–1054. `doi:10.1109/ISIE.2008.4676942`.

[7] Dell. Dell Precision m4500 mobile workstation details [online]. 2011. URL: `http://www.dell.com/us/business/p/precision-m4500/pd` [cited June, 2011].

[8] Edwards, A. W. F. *Likelihood.* Cambridge University Press, 1972.

[9] Eubank, S., et al. Modelling disease outbreaks in realistic urban social networks. *Nature 429*, 6988 (2004), 180–184. `doi:10.1038/nature02541`.

[10] Federal Communications Commission. FCC Spectrum Dashboard [online]. 2011. URL: `http://reboot.fcc.gov/reform/systems/spectrum-dashboard` [cited June, 2011].

[11] Federal Communications Commission. FCC Universal Licensing System [online]. 2011. URL: `http://wireless.fcc.gov/uls` [cited June, 2011].

[12] Forney, Jr., G. D. The Viterbi algorithm. In *Proceedings of the IEEE* (1973), vol. 61, IEEE, pp. 268–278. `doi:10.1109/PROC.1973.9030`.

[13] Ge, F., and Bostian, C. W. A parallel computing based spectrum sensing approach for signal detection under conditions of low SNR and Rayleigh multipath fading. In *New Frontiers in Dynamic Spectrum Access Networks, 3rd IEEE Symposium on* (2008), IEEE, pp. 1–10. `doi:10.1109/DYSPAN.2008.12`.

[14] GlobalSat Techonology Corporation. BU-353 USB GPS Receiver datasheet, 2009. URL: `http://www.usglobalsat.com/store/download/62/bu353_ds_ug.pdf` [cited June, 2011].

[15] Golub, G. H., and van Loan, C. F. *Matrix computations*, 3rd ed. Johns Hopkins University Press, 1996.

[16] Hayes, J. C., and Tunzi, B. R. DTMF Communication System. Patent, May 1978. US 4087638. URL: `http://www.patentlens.net/patentlens/patent/US_4087638/en/`.

[17] International Telecommunication Union. ITU-T Recommendation Q.23, Technical features of push-button telephone sets. In *Blue Book*, Fascicle VI.1. 1993. URL: `http://www.itu.int/rec/T-REC-Q.23-198811-I/en`.

[18] International Telecommunication Union. ITU-T Recommendation Q.24, Multifrequency push-button signal reception. In *Blue Book*, Fascicle VI.1. 1993. URL: `http://www.itu.int/rec/T-REC-Q.24-198811-I/en`.

[19] Kay, S. M. *Fundamentals of statistical signal processing: estimation theory*. Prentice-Hall, 1993.

[20] Krogh, A. An introduction to hidden Markov models for biological sequences. *New Comprehensive Biochemistry 32* (1998), 45–63. `doi:10.1016/S0167-7306(08)60461-5`.

[21] Lee, C.-h., and Wolf, W. Blind signal separation for cognitive radio. *Journal of Signal Processing Systems 63* (2011), 67–81. `doi:10.1007/s11265-009-0400-1`.

[22] Lin, Z.-M. A remote telephone-controlled home automation system. In *Southeastcon, Proceedings of IEEE* (Apr. 1998), IEEE, pp. 148–152. `doi:10.1109/SECON.1998.673314`.

[23] Midwest Microwave Solutions, Inc. VHF/UHF Digitizers product page [online]. 2010. URL: `http://www.mms-rf.com/rf-digitizers.html` [cited June, 2011].

[24] Murphy, K. Hidden Markov model (HMM) toolbox for Matlab, 1998. URL: `http://people.cs.ubc.ca/~murphyk/Software/HMM/hmm.html` [cited October, 2011].

[25] Ramsay, J. O. A Family of Gradient Methods for Optimization. *The Computer Journal 13*, 4 (1970), 413. `doi:10.1093/comjnl/13.4.413`.

[26] Sharma, R., Kumar, K., and Vig, S. DTMF based remote control system. In *Industrial Technology, IEEE International Conference on* (Dec. 2006), pp. 2380–2383. `doi:10.1109/ICIT.2006.372611`.

[27] Singh, M. Lecture 11: Profile hidden Markov models, Oct. 1999. URL: `http://www.cs.princeton.edu/~mona/Lecture/HMM1.pdf` [cited October, 2011].

[28] Van Trees, H. L. *Detection, Estimation, and Modulation Theory*, vol. I. John Wiley & Sons, 1968.

[29] Wang, X., and Poor, H. V. *Wireless communication systems: Advanced techniques for signal reception.* Prentice Hall, 2004.

[30] Webster, B., Calabro, G., et al. America at war: News graphics—The plot: A web of connections [online]. 2001. URL: `http://www.washingtonpost.com/wp-srv/nation/graphics/attack/investigation_24.html` [cited June, 2011].

[31] Wilson, K., and Brownstein, J. S. Early detection of disease outbreaks using the Internet. *Canadian Medical Association Journal 180*, 8 (2009), 829. `doi:10.1503/cmaj.1090215`.

Appendix A. Full List of Features

This section provides a list of features that might be useful in identifying target signatures in the RF environment with this target signal detection system. Inevitably, there are more features that could have been included on this list, as the options are limited only by the subjective level of usefulness in discriminating between communications that match target signature and those that do not. Therefore, this particular list should be considered non-exhaustive.

Center Frequency The measured center frequency of an observed signal

Bandwidth The measured bandwidth of an observed signal. There are several slightly different definitions of bandwidth, listed here are a few:

Fractional Power Containment Bandwidth The FCC defines the *occupied* bandwidth in rule 2.202 to be the range of frequencies such that the mean power radiated at frequencies within the range is 99% of the total radiated power [1]

Null-to-Null Bandwidth The width of the main spectral lobe [1]

Half-Power Bandwidth The width of the interval between the frequencies at which the spectral density has dropped by half [1]

Received Power The total received power within the specified bandwidth

Transmitted Power The total power at which the transmission was made

Distance The distance from the receiver to the transmitter

AoA The AoA as measured between the receiver and the transmitter, with respect to some cardinal direction

Modulation Class Whether the emission is analog or digital; for digital modulation, which family of modulations the emission corresponds to (i.e., PSK, FSK, ASK, Quadrature Amplitude Modulation (QAM), etc.)

Modulation Type For analog emissions, indicates whether it is an amplitude or frequency modulated signal. For digital modulations, specifies the type of modulation within the modulation class (i.e., QAM-16 vs QAM-64)

Symbol Rate For digitally modulated emissions, the rate symbols are being transmitted across the channel

Transmitter Location The location of the transmitter with respect to some global coordinate system

Receiver Location The location of the receiver with respect to some global coordinate system

Device Type One-way versus two-way radio, commercial or home-built

Device Manufacturer Who manufactured the device

Time Last Seen Time (with respect to a global clock) that the emission was last observed

Other Locations Seen Location (with respect to a global coordinate system) that the emission was last seen

Radio Service Which, if any, radio service does the emission belong to

Licensed or Unlicensed For emissions that are in a licensed radio service, whether there is a corresponding license that matches with the emission

Regulation Conformance Whether the emission corresponds to regulations associated with radio service/license or as specified by FCC or ITU regulations for unlicensed services

Traffic Type For analog modulation types, is the information being carried voice, DTMF (or other multi-frequency signaling scheme), or just noise

Mobility Whether the transmitter is stationary or mobile; if it is mobile, whether it is moving toward, away, or with the receiver

DTMF Key-Press If the transmission is DTMF, the sequence of keys that have been pressed

Keywords Whether a keyword of interest was transmitted as either voice or data

Speaker Recognition Whether the speaker on a voice channel is a person of interest