Statistical Analysis of Wireless Communication Systems Using Hidden Markov Models

Ishtiaq Rouf

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

Master of Science
In
Electrical and Computer Engineering

Dr. William H. Tranter
Dr. Jeffrey H. Reed
Dr. Tim Pratt

July 2, 2009
Blacksburg, Virginia

Keywords: Hidden Markov Model, Position Location, Stochastic Games, Network Security Situation Awareness

Statistical Analysis of Wireless Communication Systems Using Hidden Markov Models

Ishtiaq Rouf

ABSTRACT

This thesis analyzes the use of hidden Markov models (HMM) in wireless communication systems. HMMs are a probabilistic method which is useful for discrete channel modeling. The simulations done in the thesis verified a previously formulated methodology (1). Power delay profiles (PDP) of twelve wireless receivers were used for the experiment. To reduce the computational burden, binary HMMs were used. The PDP measurements were sampled to identify static receivers and grid-based analysis. This work is significant as it has been performed in a new environment.

Stochastic game theory is analyzed to gain insight into the decision-making process of HMMs. Study of game theory is significant because it analyzes rational decisions in detail by attaching risk and reward to every possibility.

Network security situation awareness has emerged as a novel application of HMMs in wireless networking. The dually stochastic nature of HMMs is applied in this process for behavioral analysis of network intrusion. The similarity of HMMs to artificial neural networks makes it useful for such applications. This application was performed using simulations similar to the original works.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

This thesis is dedicated to my parents, family, and teachers for blessing me with their love and guidance. I am eternally grateful to them for helping me come this far. I am especially thankful to my mother for doing her best to get me through graduate school.

I extend special thanks to Dr. William Tranter for providing me with exceptional opportunities in graduate school and advising me through my graduate studies. I would also like to thank Dr. Jeffrey Reed for looking after me during the good times and bad, and Dr. Tim Pratt for his support as a committee member and excellent motivations as a teacher.

I am grateful to Mr. Haris Volos for providing me with the most important part of my thesis – the measured data that I have used for my work. I am indebted to Dr. Ihsan Akbar for his supervision of my work and specific technical advice. I am also thankful to Dr. Maruf Mohammad for his advice and guidance. I owe a lot to Dr. S. M. Hasan for his advice.

Finally, I would like to thank my brothers in arms – Mahmud Harun, Ashrafee Azmir Hossain, Arafat Islam Khan, and Wasef Hossain Khaled – for being with me during perhaps the most challenging time of my life. I owe them my happiness, my optimism, and my spirit.

CHAPTER 1: INTRODUCTION

Hidden Markov Models (HMM) are one of the most important statistical modeling concepts developed in the last few decades. It is particularly useful because the use of hidden or unobservable states makes the model generic enough to handle a variety of complex real-world time series, while the relatively simple prior dependence structure still allows for the use of efficient computational procedures (2).

HMMs offer an efficient method of analyzing wireless communication systems. The use of state-based expression enables symbol-by-symbol analysis as opposed to wavelength-based analysis. In this method, error in communication channels becomes a characteristic property. This information is analyzed to obtain insight about the channel.

In this thesis, HMMs have been used to train profiles of specific wireless receivers. Profiling was done using power delay profiles of the received signals. The profiles took into account the burst error in the channels. The complete set of profiles was later used to test and identify receivers. Since varying signal strength at specific receivers is similar to the varieties of channel conditions, HMMs can be used to locate and identify channel conditions as well.

This thesis uses a previously formulated methodology to position-location application. The method was applied to a new environment to successfully verify the methodology. Binary HMMs and Baum-Welch Algorithm-based parameter estimation were used to determine the effectiveness of the method to the worst case scenarios.

The scope of the thesis is the effect of fading only. Aspects of wireless communication such as radio-frequency communication, modulation, etc are not considered. The presence and absence of fading in a system as a result of channel conditions is taken into account to locate wireless receivers.

Position location or classification applications of HMMs require decision-making. While HMMs simplify the analysis by evaluating the stochastic nature of the systems, the decision-making on the basis of these analyses is usually dependent on stochastic distance measures. One method to improve the performance is by associating the decision-making process with stochastic game theory. Game theory is a branch of mathematics that evaluates the decision-making process by attaching risk and reward to every possibility. Thus, the use of stochastic games can provide the best of both worlds. Stochastic game theory is explained in this thesis, and some illustrative examples are shown.

The ability of HMMs to dynamically model the characteristics of a system has been used in behavior analysis as well. One example of it is the use of HMMs in Network Security Situation

Awareness (NSSA). This thesis simulates the process of NSSA analysis using the Viterbi algorithm for parameter estimation. The successful result validates the process used in previous research.

A quick summary of each chapter and the findings therein is provided below.

1.1 FUNDAMENTALS OF MARKOV THEORY

The first chapter presents the fundamentals of hidden Markov models. Practical examples were provided to illustrate how a process becomes Markovian and what benefits such processes offer. Parameters of HMMs as well as the methods of analyzing them have been discussed.

Variations to the basic HMMs are often used to extract better results. Gilbert and Fritchman models are two such structures that are used in wireless communications. In order to reduce the computational burden, semi-hidden or parallel Markov models are used for certain applications. These variations have been discussed in this chapter.

Finally, the Baum-Welch algorithm was described in this chapter. This is an effective algorithm for estimating parameters of an HMM. Once the parameters have been accurately estimated, they can be further analyzed for dissimilarities that enable classification. The direct distance measure and KLD were discussed as possible methods to achieve this.

1.2 STOCHASTIC GAMES USING HIDDEN MARKOV MODELS

The second chapter focuses on stochastic games and hidden Markov models. These theories were independently developed, and both were very important in wireless communications. Game theory was used in wireless communications when mobile ad hoc networks are required to make decentralized decisions. It is used to manage resources efficiently in a logically defined method. Each decision in a game is associated with a reward or cost of the controllers.

Hidden Markov models offer the opportunity to stochastically analyze a system. This method enables the use of higher levels of abstraction. Merging these two theories ensures that the benefits of both the theories can be applied. Practical examples have been provided in this chapter which demonstrate the simultaneous use of Markov processing and reward or cost of decisions in a stochastic game.

1.3 POSITION LOCATION USING HIDDEN MARKOV MODELS

As briefly mentioned in 4.4 , this work adds specific value to the practical aspect of a previous work in (1). Following a previously developed methodology, HMM-based profiles were used for receiver positions to identify them. The application of stochastic methods to wireless communications is a significant aspect of this method.

There can be different ways of utilizing this profile-and-locate method. One possible use is to consider receivers as transponders and, given the location of any receiver and its known response to signal- and position-variation, a radio system can be designed that dynamically adjusts with its environment. As such, it can be a possible plug-in to cognitive radio network.

Another possible application can be to use receiver location profiles as a way to identify a receiver. For example, signal sent by a trapped miner can be received and analyzed using HMM profiling to estimate his or her exact location.

This work applies the previously formulated methodology in a new environment with new data. The successful completion validates the methodology of using HMMs for position location.

1.4 NETWORK SECURITY SITUATION AWARENESS USING HIDDEN MARKOV MODELS

NSSA is a comparatively new aspect of network security. If implemented properly, it can be a significant improvement to network administration as we know it today. The research is in line with inserting limited cognition into communication systems. The notion of growing a sense about the security status includes many holistic and cognitive aspects. It is about not only knowing what an element has done to the system, but also predicting what it might do next. This introduces the need to statistically analyze the system.

The HMM-based model mentioned in this paper can add some of these qualities to the current solutions. The similarity of HMMs to artificial neural networks enables the system to learn and adjust to its situation (3). Once that is done, an administrator can develop appropriate security metrics according to specific networks. This work can be improved by using actual attacker scenarios and network setup. The attack model described in the paper was chosen based on holistic estimation. A detailed model regarding the inter-relation between layers is necessary for optimum functioning.

Other approximation algorithms can be applied to this method. Also, advanced matrix analysis techniques can be applied to improve the comparison metrics between different HMMs

CHAPTER 2: FUNDAMENTALS OF MARKOV THEORY

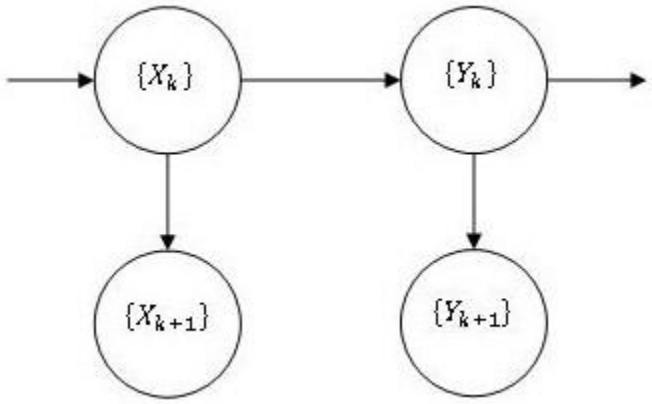
2.1 MARKOV PROCESSES

The basis of an HMM is a Markov chain denoted by $\{X_k\}$, where k is an integer. In most cases, the Markov chain is assigned to take positive numbers from a finite set only. The Markov chain, $\{X_k\}$, remains hidden or unobservable in the HMM. However, the inner workings of this chain are reflected in an observable sequence, $\{Y_k\}$, where k is an integer. These two sequences are both stochastic and are inter-linked. The characteristics of the observable sequence, Y_k, are a direct result of the operations in the hidden Markov chain, $\{X_k\}$.

Since the observable sequence is governed by the hidden sequence, analytical methods are often used to obtain insight about the hidden sequence from the nature of the observable sequence. For instance, if $\{Y_k\}$ is a normal or Poisson distribution, its mean and variance will be determined by $\{X_k\}$.

The underlying hidden sequence, $\{X_k\}$, is called a Markov chain when k is an integer. It is sometimes referred to as the $regime$ or $state$. All stochastic inferences about the system are done from the observable sequence only. It is also assumed that the hidden sequence, $\{X_k\}$ is the only factor that affects the outcomes of the observable sequence, $\{Y_k\}$. This assumption makes HMMs a bivariate discrete time process denoted by $\{X_k, Y_k\}$, where k is a positive integer. The dependence structure defined here can be illustrated by the following figure.

Figure 1: Graphical representation of the dependence structure of an HMM. X and Y denote the hidden and observable sequences in an HMM, respectively.



Many variations of HMMs are used within the basic structure shown in **Error! Reference source not found.**. A classical example of the use of HMMs is the analysis of human actions depending on the state of the weather. We assume that two friends, Alice and Bob, are in regular contact with each other. Alice has no way of knowing the weather conditions of the city Bob is in. However, she knows that Bob does one of three things during the day – $walk$, $shop$, or $clean$. We assume there to be two possible weather conditions on a given day – $rainy$ and $sunny$. These two weather conditions constitute of the two $states$ in the unobservable or hidden sequence, $\{X_k\}$. The set of Bob's actions is the observable set, $\{Y_k\}$.

Since Alice does not know about the weather in Bob's city, she can only assume that weather conditions follow a Markov process, and apply probabilistic values to them. The initial probabilities for rainy and sunny weather can be assumed to be $\{Rainy: 0.6, Sunny\ 0.4\}$.

Alice knows that Bob's actions are directly dependent on the weather conditions, and has some idea about what Bob likes to do in certain weather conditions. Bob's preferences for a given weather can be defined by the following probabilities.

Table 1: Emission probabilities for HMM example. This table shows the probability of walking, shopping, and cleaning depending on the weather condition.

Weather	Walk	Shop	Clean
Rainy	0.1	0.4	0.5
Sunny	0.6	0.3	0.1

These values denote the emission probabilities in an HMM. The emission probabilities depend not only on how the weather is on a particular day but also on how the following day's weather is. Since one cannot surely know the weather of a future time, the only way is to develop a probabilistic model. The following table shows the probabilities of weather transition.

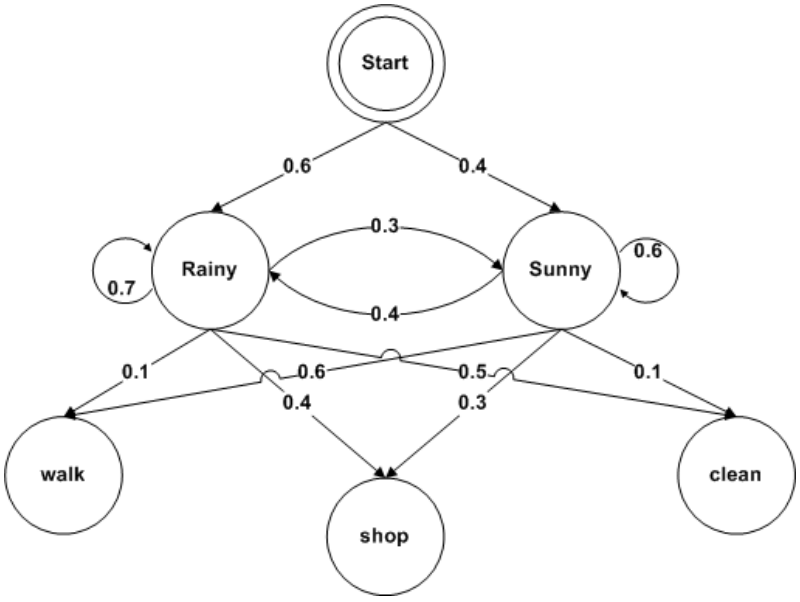
Table 2: Transition probabilities for HMM example. This table shows the probability of the next day's weather being sunny or rainy based on today's weather.

Weather	Rainy	Sunny
Rainy	0.7	0.3
Sunny	0.4	0.6

The transition matrix and emission matrix are integral in modeling using HMMs. These, as well as Markov chains and other technical concepts, will be explained in detail shortly.

To Alice, the only observable value is Bob's action on a particular day. Based on that, she can estimate the weather in Bob's city on a given day. The method that Alice should use is an HMM. The complete probabilistic model can be expressed by the following illustration (4).

Figure 2: Complete probabilistic model of HMM example. This example shows the probability of walking, shopping, or cleaning on the basis of the weather condition. Here, the choice of activity is an observable sequence that can be used to estimate the weather condition based on the list of probabilities.



2.2 DISCRETE-TIME MARKOV CHAINS

Discrete-time Markov processes are stochastic processes in which the current state of the system depends on the k previous state(s). Such systems can be either finite or infinite. A random process, $\{X(t)\}$, is a Markov process if the future of the process given the present is independent of the past (5). For arbitrary times $t_1 < t_2 < \cdots < t_k < t_{k+1}$ and discrete-valued $X(t)$, the probability of X at time $t + 1$ is as below (5).

$$P[X(t_{k+1}) = x_{k+1} | X(t_k) = x_k, \dots, X(t_1) = x_1] = P[X(t_{k+1}) = x_{k+1} | X(t_k) = x_k] \tag{1.1}$$

The equation above is a condition for Markov processes. For continuous-valued $X(t)$, the condition is as below.

$$P[a < X(t_{k+1}) \leq b \,|X(t_k) = x_k, \dots, X(t_1) = x_1] = P[a < X(t_{k+1}) \leq b \mid X(t_k) = x_k] \tag{1.2}$$

If the samples of $X(t)$ are jointly continuous, the equation is equivalent to the following.

$$f_{X(t_{k+1})}(x_{k+1} |X(t_k) = x_k, \dots, X(t_1) = x_1) = f_{X(t_{k+1})}(x_{k+1} |X(t_k) = x_k) \tag{1.3}$$

These equations are referred to as the Markov property. Here, t_k denotes the $present$ time, t_{k+1} denotes the $future$, and $t_1, t_2, \dots, t_{k-1}$denote the $past$. In Markov processes, probability mass functions (pmf) and probability density functions (pdf) are conditioned on several time instants to reduce them to only the recent-most time instance.

The assumption that the future state depends only the last k states greatly reduces the computational load in a system, simplifies the behavior of the system, and makes it analytically tractable. Although it is difficult to prove that any system is completely Markovian or non-Markovian, the assumption, none-the-less, is very useful when working with complex systems.

Another advantage of Markov chains is that they can always be reduced to the first order. This can be done at the expense of using more $states$. Typically, an N-state k-th order Markov chain requires N^k states in order to be properly represented in the first order.

2.2.1 States and Transitions

Use of the concept of $states$ and $transition$ between states is a particular strength of HMMs. The system under consideration is divided up among a number of quantized $states$ or $regimes$ to aid the modeling. Physical systems can often be completely described by a specific number of variables and parameters. Each state represents a particular variable or parameter of the system. Depending on the realm of simulation, the states can denote a variety of things. For example, an electric circuit can be described by states such as voltage, current, and impedance. Knowledge about the values of these states helps us describe the system in full. In digital logic systems, $states$ are defined by a certain combination of binary digits that represent various aspects of the system. In speech analysis, each phoneme can be considered a $state$.

The concept of system states is useful in wireless channel modeling as it introduces a new method of analysis. It is different from the traditional methods that analyze noise, interference, and other disturbances. These methods are applied in the waveform level and can be very cumbersome to implement. The introduction of discrete states enables the use of symbol-by-symbol analysis.

A system is said to be fully defined when it is expressed in terms of all of its states. As such, a system is described as a function of its states. Static systems can be completely defined by a set of values for the states only. Dynamic systems, however, change states with time. Therefore, they require information regarding the $transition$ between states over a period of time. The following example illustrates the concept of states and transitions.

Suppose stochastic analysis is applied to a five-letter word, 'Space'. Here, the placeholder for each letter is a part of the sequence. The entire word is expressed by the set $\{X_k\}$, where k=1,2,3,4,5. The first state in the sequence, X_1, is an initial value, 'S'. For the next letter, the state transitions to the state 'p'. Consequently, the states 'a', 'c', and 'e' are transitioned to. Each letter in the word is a $state$ and the change from one state to another is a $transition$.

Although there are 26 letters in the English alphabet, and therefore 26 possible states, the probability of each of these to be the following state is not the same. The probability of a 'p' following

an 'S' is much higher than the probability of a 'z' or an 'x' following the 'S'. Therefore, we know *a priori* that a word starting with 'S' is more likely to have a 'p' as its second letter than a 'z'.

The adaptive and uneven probability of state transitions can be utilized by HMMs since the Markovian assumption requires that the future state will depend only on the last k states of the system. The example mentioned here is a case used for speech and handwriting recognition using HMMs. The use of this algorithm enables intuitive learning and adaptive stochastic prediction of a system. Further examples of HMMs will be discussed in this thesis.

2.2.2 Transition Matrix

One of the building blocks of an HMM is the transition matrix. This matrix represents the probability of transition between different states. This matrix is extensively used during both the training and testing phase of hidden Markov modeling.

2.2.2.1 Definition of the Transition Matrix

Let X_k be a discrete-time integer-valued Markov chain that starts at $n = 0$ with pmf as below.

$$p_j(0) = P[X_0 = j], j = 0,1,2, \dots \tag{1.4}$$

The joint pmf for the first $k + 1$ values of the process is as below.

$$\begin{aligned} P[X_k = i_k, \dots, X_0 = i_0] = P[X_k = i_k \,|X_{k-1} = i_{k-1}] \dots P[X_1 = i_1 \mid X_0 \\ = i_0] P[X_0 = i_0] \end{aligned} \tag{1.5}$$

Therefore, the joint pmf for a particular sequence is the product of the probability for the initial state and the probabilities for the subsequent one-step state transitions. The one-step state transition probabilities are fixed and do not change with time. An expression of one-step state transition is as below.

$$P[X_{k+1} = j \,|X_n = i] = p_{ij}, for\ all\ k \tag{1.6}$$

X_k is said to have homogeneous transition probabilities. The joint pmf for $X_k, \dots, X_0$ is given by the following.

$$P[X_k = i_k, \dots, X_0 = i_0] = p_{i_{k-1},i_k} \dots p_{i_0,i_1} p_{i_0}(0) \tag{1.7}$$

Thus, X_k is completely specified by the initial pmf, $p_i(0)$ and the matrix of one-step transition probabilities, P.

$$P = \begin{bmatrix} p_{00} & p_{01} & p_{02} & \cdots & p_{0k} \\ p_{10} & p_{11} & p_{12} & \cdots & p_{1k} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ p_{i0} & p_{i1} & p_{i2} & \cdots & p_{1k} \end{bmatrix} \tag{1.8}$$

This matrix is called the $transition\ matrix$ of an HMM. It is sometimes denoted by A and called the A-matrix. Since a state must either transition to another state or remain in that state itself, the sum of each row in P should be unity.

2.2.2.2 Inferences into the Transition Matrix

The transition matrix defines the state transition probabilities within the Markov model. It should be noted that HMMs do a stochastic analysis within reasonable error limits. The transition probabilities in this matrix determine the sequences of states and, in turn, the nature of the system. In the example of Alice and Bob, the probability chart mentioned in Table 2 is the transition. It can be inferred from the table that it is more like than not that two consecutive days will have the same weather (0.7 for rainy and 0.6 for sunny). Also, the probability of a rainy weather continuing is higher than that of a sunny weather continuing (0.7 to 0.6). Given the initial condition, a Markov model can be prepared using this transition matrix.

2.2.3 Emission Matrix

Another very important building block of HMMs is the emission matrix. This matrix is essential to produce a set of simulation data that can be verified later.

2.2.3.1 Definition of the Emission Matrix

The $emission\ matrix$ in a Markov chain refers to the set of all possible outputs. The probabilities of all possible emissions comprise the emission matrix. The resulting emissions are the actual observable values of an HMM. These observations are used to estimate the hidden state transitions and analysis the nature of the system.

The emission matrix can often be empirically defined based on sample data. The initial matrix is later used to estimate the transition matrix. Together, the transmission matrix, the emission matrix, and the state probability distribution form the HMM profile of a system.

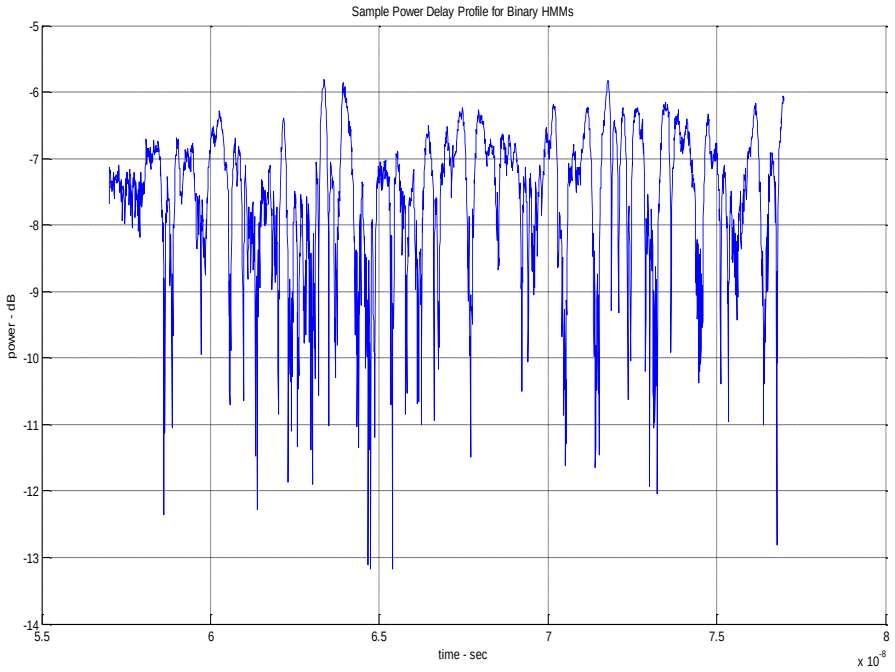
2.2.3.2 Inferences into the Emission Matrix

The emission matrix of the example of Alice and Bob in Table 1 contains the probability of Bob choosing to do a certain activity, given the weather condition. Alice came to know about these values after due observation. These probabilities, analyzed against an actual sequence of activities, enable Alice to estimate the weather in Bob's city.

The significance of the transmission and the emission matrices can be illustrated by the following example of the use of HMMs.

A two-state model can be assumed for receiver input signal level in a wireless communication system as in (6). A possible plot of the receiver input signal level can be as below.

Figure 3: Sample power delay profile to illustrate Binary HMMs. Here, the noticeable drops indicate burst errors. These are considered to be samples with no fading in the signal.



Once an arbitrary threshold is set (say, $-9dB$ in the figure above), it can be assumed that states that have are equal to or higher than the threshold are $good$ states while states below the threshold are bad states. A possible transition matrix, A, can be assumed as follows.

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \tag{1.9}$$

It can be assumed that a_1 denotes a good state and a_2 denotes a bad state. Therefore, the significances of the four elements of the transition matrix are as below.

a_{11} = Both the initial and the future states are good

a_{12} = The received signal transition from a good state to a bad state

a_{21} = The received signal transition from a bad state to a good state

a_{22} = Both the initial and the future states are bad

There is, however, another factor to consider. The transitions observed in the system are subject to error. Therefore, a transition from $good$ to bad, for example, can be correct or erratic. There needs to be a provision to consider this possibility. The emission matrix, called the $error\ generation\ matrix$ in this case, serves the purpose. The emission matrix can be defined as below.

$$B = \begin{bmatrix} b_{C|g} & b_{C|b} \\ b_{E|g} & b_{E|b} \end{bmatrix} \tag{1.10}$$

The significances of the four elements in the emission matrix are as below.

$b_{C|g}$ = Probability of getting a transition to $good$ state correctly

$b_{C|b}$ = Probability of getting a transition to bad state correctly

$b_{E|g}$ = Probability of getting a transition to $good$ state by error

$b_{E|b}$ = Probability of getting a transition to bad state by error

2.3 HIDDEN MARKOV MODELS

A Markov chain is called 'hidden' when one of the two processes in the doubly stochastic system is unobservable. In HMMs, one can only observe if the values are correct or erratic. Which of the numerous states caused the error is unknown to the observable. There is neither a way to analytically derive the exact source of the error.

2.3.1 Parameters of Hidden Markov Models

Since the underlying unobservable states and the observable sequence are both stochastic in nature, HMM-based profiles cannot be created based on an exact order of sequences or outputs. Instead, HMM profiles are generally created using a set of probability matrices.

Each HMM profile is characterized by two probability matrices – the state transition matrix and the emission matrix. The transition matrix contains probability based on the first order Markov chain. The emission matrix contains discrete output probabilities that define the conditional

probability of emitting an output symbol. In case of discrete-time HMMs, the emission matrix contains the probability of choosing an output symbol from a finite set of outputs.

Besides the transition matrix and the emission matrix, the following elements also determine HMM profiles (1). Combinations of these values are used to create profiles during the training phase of HMM modeling. Later, models of actual situations are compared to these profiles to classify them.

i. Number of States

The number of states in an HMM is the first parameter of a Markov model. The number of states, N, can often determine the accuracy of the model. A large number of states can increase accuracy at the cost of workload. The state sequence can be defined as $\boldsymbol{S} = [s_1, s_2, \dots, s_N]$.

ii. Number of Observation Symbols

The size of the observation sequence refers to the number of elements in the discrete alphabet. This number, M, corresponds to the number of physical outputs required to express the system in full. As such, the observation sequence can be defined as $\boldsymbol{V} = [v_1, v_2, \dots, v_N]$. The number of observation symbols to be used may vary due the nature of the system.

iii. State Transition Probability

The state transition probability matrix refers to the transition matrix defined previously. The elements of this matrix are defined as $\boldsymbol{A} = \{a_{ij}\}$, where $1 \leq i, j \leq N$. Also, a_{ij} denote elements of the transition matrix. The previously defined properties of the state transition matrix can be summarized as below.

$$a_{ij} = P(q_{t+1} = s_j \mid q_t = s_i) \tag{1.11}$$

$$0 \leq a_{ij} \leq 1 \tag{1.12}$$

$$\sum_{j=1}^{N} a_{ij} = 1 \tag{1.13}$$

Each state in an HMM follows these definitions. The nature of the transition probabilities is a characteristic property of different Markov profiles.

iv. Observation Symbol Probability

The observation symbol probability matrix, i.e., the emission matrix, is denoted as $B = \{b_j(l)\}$, where $1 \leq j \leq N$, $1 \leq l \leq M$. As mentioned before, the emission matrix is also a stochastic matrix. The previously defined properties of the emission can be summarized as below.

$$b_j(l) = P(v_l at\ t | q_t = s_j) \tag{1.14}$$

$$0 \leq b_j(l) \leq 1 \tag{1.15}$$

$$\sum_{l=1}^{M} b_j(l) = 1 \tag{1.16}$$

v. Initial State Distribution

As shown before, the initial state distribution is the only aspect of an HMM that is not stochastic in nature. How a system is initiated has an effect on the final profiles created. However, all modeling applied in this thesis have been done with an evenly defined initial state distribution matrix. This matrix is mathematically defined as $\boldsymbol{\pi} = \{\pi_i\}$, where $1 \leq i \leq N$. The properties of the initial state distribution matrix can be summarized as below.

$$\pi_i = P(q_l = s_i) \tag{1.17}$$

$$0 \leq \pi_i \leq 1 \tag{1.18}$$

$$\sum_{i=1}^{N} \pi_i = 1 \tag{1.19}$$

When combined, these parameters can be used to model a system as $\lambda = (\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\pi})$, where $\boldsymbol{A}$, $\boldsymbol{B}$, and $\boldsymbol{\pi}$ represent the state transition matrix, the emission matrix, and the initial state distribution, respectively.

2.3.2 Output Sequence Generation in Hidden Markov Models

Once a Markov model has been defined as $\lambda = (\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\pi})$, it can be used to generate an observation sequence. The generated model has the same stochastic nature as defined by the Markov

profile, λ. For an observation sequence of length M and an observation symbol o_i from alphabet $\boldsymbol{V}$, the observation sequence at time instant t is generated as below (1).

i. An initial state $q_1 = s_i$ is chosen by generating a uniformly distributed random number in the range $(0,1]$ and compared against the initial state distribution, $\boldsymbol{\pi}$
ii. The time index is set to $t = 1$.
iii. The observation symbol $o_i = v_1$ is chosen by generating a uniformly distributed random number in the range $(0,1]$, and compared against symbol probability distribution in state $s_i, b_i(l)$.
iv. The next state, $q_{t+1} = s_j$ is chosen by generating a uniformly distributed random number in the range $(0,1]$, and compared against state transition probability distribution in state s_i, a_{ij} .
v. Set $t = t + 1$.
vi. Terminate the process if $t > M$, repeat from step 3 otherwise.

Repeated operation of this algorithm is used to generate an output sequence from a given HMM. The resulting model can be of various structures depending on the nature of the system under study.

2.3.3 Structures of Hidden Markov Models

HMMs can be structured according the requirements presented by a particular system. However, there are two major variations that are widely used for modeling. These two structures are:

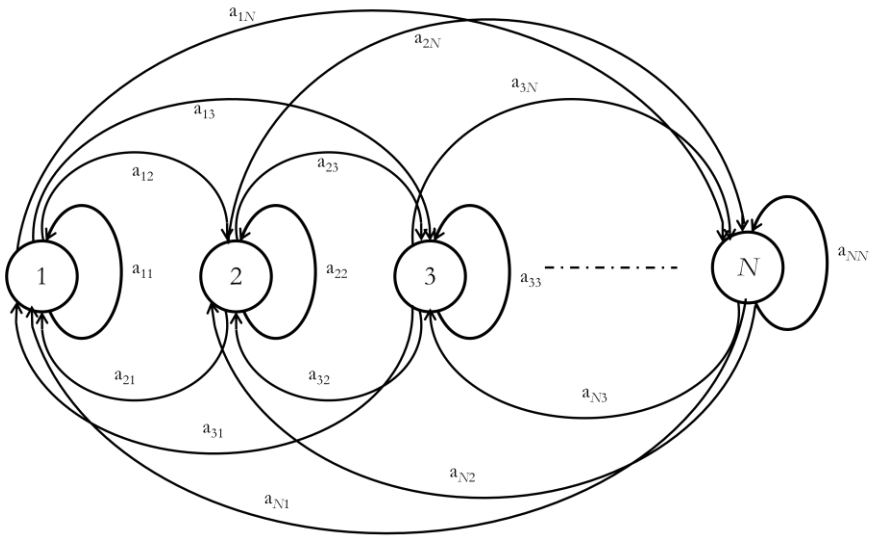
i. The ergodic model or fully-connected HMM, and
ii. The Bakis model or left-right HMM.

These two structures are briefly explained below.

2.3.3.1 Ergodic Hidden Markov Models

HMMs that are fully connected are referred to as Ergodic structures. In such structures, transitions are allowed between all the states in the HMM. As shown in the following figure, ergodic structures allow transition to any other state through a finite number of steps.

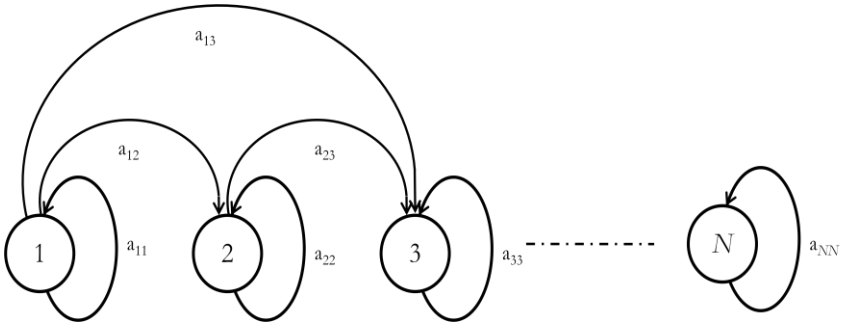
Figure 4: Ergodic Structure of Hidden Markov Models.



2.3.3.2 Left-Right Hidden Markov Models

Applications such as speech processing experience change in properties over time. In such cases, a structure is required that factors in the inherent discontinuities of the system. This model is also known as the Bakis model.

Figure 5: Left-Right Structure of Hidden Markov Models.



In a left-right model, no transition is allowed to the states whose indices are lower than that of the current state. This condition is mathematically expresses as the following.

$$a_{ij} = 0, j < 1 \tag{1.20}$$

Usually, the initial state distribution, π, is set to 1 for $i = 1$ and 0 for all other indices. To ensure that the system is manageable, the number of states that can be jumped are often limited in these models.

2.3.4 Gilbert and Fritchman Models

The Gilbert model (6) is two-state hidden Markov model with a good error-free state and a bad state with error probability. This method is used to calculate the capacity of a channel with burst errors.

The Fritchman model is a popular option in modeling burst errors in mobile radio channels. They are also easier to analyze and estimate. In Fritchman models, the state space is divided into good and bad states only. The good and bad states represent error-free transmissions and transmission error, respectively. As a result, the emission matrix is made up of zeros and ones only, and does not need to be estimated (6). A general state transition matrix for the Fritchman model can be as below.

$$A = \begin{bmatrix} A_{gg} & A_{gb} \\ A_{bg} & A_{bb} \end{bmatrix} \tag{1.21}$$

Here, the sub-matrices represent the transition probability between various good and bad states. This model can be further modified to incorporate error run vectors and diagonal matrices.

2.3.5 Variations of Hidden Markov Models

Despite their utility, HMMs can be very cumbersome for certain applications. The high computation complexity as well as the size of the matrices often renders HMMs unsuitable for applications. The fundamental theories of HMMs are, therefore, sometimes modified to fit wider use. The two most widely used variations are the semi-hidden Markov models and the parallel Markov models. Besides these, there are methods of optimizing HMMs depending on their applications.

2.3.5.1 Semi-Hidden Markov Models

HMMs are very computation intensive and, thus, often require extensive processing. To reduce the computational complexity, semi-hidden Markov models (SHMM) are often used. This variation is also referred to as the Markov Renewal Process.

In an SHMM, successive state occupancies are described by the transition probabilities of a Markov chain, while the state duration is determined by an integer-valued number that depends only on the current and future states. Both the structures maintain similar properties during transition. A major difference between HMM and SHMM is the concept of a $holding\ time$.

Say, an SHMM is in state a_i. Based on the transition probability, a_{ij}, the next state, j, is chosen. Unlike HMMs, SHMMs wait for a time interval, τ_{ij}, before the transition is made. The holding time is usually expressed in terms of the holding time mass function for the corresponding transition. This probability distribution is given by the following (1).

$$P(\tau_{ij} = m) = h_{ij}(m), 1 \leq i, j \leq N, m = 1,2, \ldots \quad (1.22)$$

All holding time mass functions are expressed compactly in the state duration matrix, $\boldsymbol{H}$, where $\boldsymbol{H} = \{h_{ij}\}_{N \times N}, 1 \leq i, j \leq N$. This matrix is an addition to the conventional HMMs and is the most significant different between HMM and SHMM.

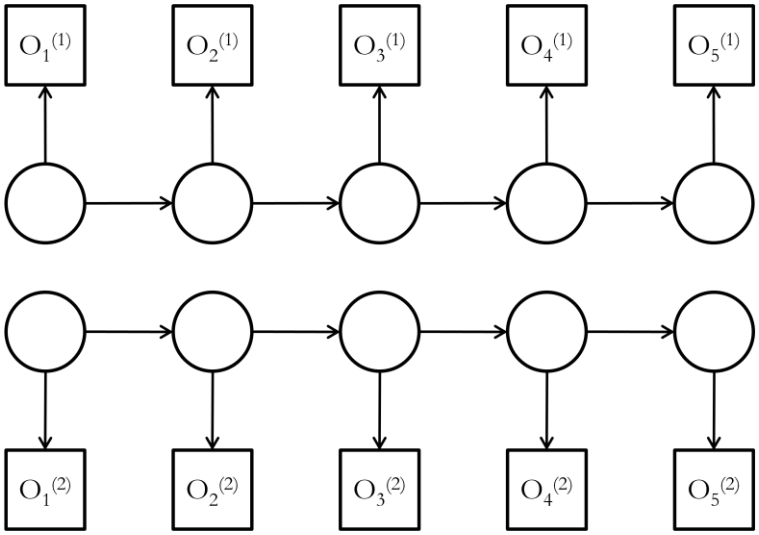
An advantage of using the holding time mass function is the ability to use non-geometric distributions such as the normal distribution, gamma distribution, etc. In HMMs, only geometric distributions can be used. If the holding time times are made of exactly one time unit, then SHMM becomes HMM.

2.3.5.2 Parallel Hidden Markov Models

Parallel hidden Markov models (PHMM) are another important variation of the Markov process. PHMMs have been successfully used for applications such as sign language recognition where there are two processes in parallel (7). In sign languages, both the left and right hand make different moves depending on the word or phrase being interpreted. PHMMs, in this case, offer scalability to the system.

As seen in Figure 6 below, the output observations of PHMMs are separate and the internal states do not affect each other.

Figure 6: Parallel Hidden Markov Models.



PHMMs greatly increase the size of the set of observation possibilities. Therefore, such systems need to be very well-defined. The issue of scalability needs to be addressed in order to apply PHMMs widely. It is quicker than other similar extensions of the basic theory of HMMs because the two processes operate simultaneously.

In the application on American Sign Language, it has been suggested in (7) that this method reduces the order of the HMM to $(30 + 8 + 20 + 40) \times 2 = 196$ instead of an order of 10^8 HMMs.

The drastic reduction of computational complexity makes this an intriguing option for training and testing using HMMs in complex situations.

2.3.5.3 Block Equivalent Hidden Markov Models

Wireless communication channels often observe burst errors. Markov models need to be able to accommodate long observation periods in order to accurately model such systems. As such, the transition matrix elements have to have large dynamic range. While large matrices increase the accuracy, they also add more computational burden to the system.

To mitigate this problem, block equivalent HMMs are often used. This was used proposed as a method to reduce the overall computational burden. In this method, the N states of the system can be

partitioned into numerous subsets. As a result, the transition matrix can be expressed as a matrix of matrices as shown below.

$$A = \begin{bmatrix} A_{00} & A_{01} \\ A_{10} & A_{11} \end{bmatrix} \tag{1.23}$$

Here, 0 and 1 represent good and bad states, respectively. Splitting the states makes the errors deterministic as opposed to a probabilistic function of states. The computational complexity is reduced by pre-computing and reusing exponents of the sub-matrices. With this model, the channel remains in the same state during an error burst and changes state only at the end of the burst (1).

2.3.6 Examples of Hidden Markov Models

As mentioned previously, HMMs have an observable sequence and an unobservable sequence of states. Models that are comprised of such unobservable random variables called latent variable models, missing data models, or models with incomplete data (2). The following examples reflect on some of the applications of HMMs and the methods used to solve them.

2.3.6.1 Use of HMMs in Biology

HMMs are used in Biology to stochastically model biological sequences. They are used to find genes in the DNA (deoxyribonucleic acid) and to functionally annotate sequenced genomes (2). All living organisms carry a blueprint of the molecules they need for living. DNA is a chain-like molecule that can be specified uniquely by listing the sequence of amine bases from which it is composed. This is called the sequencing process. The DNA is abstracted by the letters A, C, G, and T (for adenine, cytosine, guanine, and thymine) to denote the sequences.

The role of DNA is as a storage medium for information about the individual molecules needed in the biochemical processes of the organism. A region of the DNA that encodes a single functional molecule is referred to as a gene. There is no easy method to discriminate coding regions from non-coding regions. The dimension of the problem is enormous as typical genomes can be millions of bases long with the number of genes to be located ranging from few hundreds to few thousands (2).

One method of solving this problem was offered in (8). This method consisted of modeling the observed sequences of bases, $\{Y_k\}_{k\geq 0} \in \{A, C, G, T\}$, by a two-state HMM such that the non-

observable state is binary-valued with one state corresponding to non-coding regions and the other one to coding regions. The simplest form of the model consisted of a conditional distribution, Y_k, given an independent distribution, X_k. The distributions are parameterized by the vector of probabilities of observing A, C, G, or T.

This method was improved upon by applying more complicated methods that incorporate behavioral analysis or actual DNA sequences (9) (10). In some applications, higher order HMMs have been proposed in which the distribution of Y_k does not only depend on the current state, X_k, but also on the previous two observations, Y_{k-1} and Y_{k-2}.

2.3.6.2 Speech Recognition

A very significant application of HMMs is speech recognition. This was the first area where HMMs were applied extensively to automatically determine what a person said from a recording of his or her voice.

In this application, a recorded and sampled speech signal is slotted into short sections or frames. The typical representation contains about 20 milliseconds of the original signal. Each section is analyzed separately to produce a set of coefficients that represent the estimated power spectral density of the signal in the frame. This results in a discrete-time multivariate time series of spectral coefficients (2). For a given word to be recognized, the length of the series of vectors depends on the time taken for the speaker to utter the word. Therefore, the primary requirement of this model is to adjust the time alignment so that the multivariate sequences of unequal lengths can be compared.

Each word in continuous speech can be viewed as a sequence of phonemes. The state of the Markov chain is a hypothetical phoneme that is currently being uttered at a given time slot. As each letter of a word is spoken, the probability of the next letter varies. The observation vectors associated with the states are assumed to be independent and are assigned a multivariate distribution, which is most often a mixture of Gaussian distributions.

The variability induced by the distribution is used to model spectral variability within and between speakers (2). The actual speech recognition is realized by running the recorded word as input to several different HMMs, and selecting the one that assigns the largest likelihood to the observed sequence. The recognition using HMM-based method works better for phonemes compared to segmented words.

2.4 PARAMETER ESTIMATION OF MARKOV MODELS

Parameter estimation is one the most important aspects of hidden Markov models. Various algorithms are used to estimate the parameters of an HMM. Of these, the Baum-Welch algorithm and the Viterbi algorithm are very popular. It is a variation of the Expectation Maximization (EM) algorithm.

2.4.1 Expectation-Maximization (EM) Algorithm

The EM algorithm iteratively estimates the parameters of an HMM. As depicted in the name, this algorithm sets an expectation and attempts to maximize it (11). The expectation is with respect to the unknown underlying variables. The maximization step provides a new estimate of the parameters based on the assumption that the expectation step is correct (1).

The likelihood function can be described by the following equation.

$$L(\tau) = \sum_{x} f(x) \log\big(g(x,\tau)\big) = E\left[\log\big(g(x,\tau)\big|f(x)\big)\right] \tag{1.24}$$

Here, $f(x)$ and $g(x,\tau)$ are the probability distributions of the original process. $E[\cdot]$ denotes the expectation equation. Next, another function, $K(z;x,\tau) > 0$, is assumed whose sums over z is constant. It is viewed as a probability distribution of z by normalizing the function. It is called the auxiliary probability distribution. The normalized function is as below.

$$\sum K(z;x,\tau) = 1 \tag{1.25}$$

A new function is defined at this point as below.

$$\Psi(z;x,\tau) = g(x,\tau)K(z;x,\tau) \tag{1.26}$$

$$\log g(x,\tau) = \Psi(\mathrm{z};\mathrm{x},\tau) - \log\mathrm{K}(\mathrm{z};\mathrm{x},\tau) \tag{1.27}$$

As shown in (12), both sides of the equation above can be multiplied by $r(z;x,\tau) = f(x)K(z;x,\tau)$ to come up with the following equation.

$$L(\tau) = Q(\tau, \tau_p) + H(\tau, \tau_p) \tag{1.28}$$

Here, $Q(\tau, \tau_p)$ and $H(\tau, \tau_p)$ are defined as below.

$$\begin{aligned} Q(\tau, \tau_p) &= \sum_x \sum_z \log \Psi(z; x, \tau) r(z; x, \tau_p) \\ &= E[\log \Psi(z; x, z) | r(z; x, \tau_p)\,] \end{aligned} \tag{1.29}$$

$$\begin{aligned} H(\tau, \tau_p) &= \sum_x \sum_z \log \mathrm{K}(z; x, \tau) r(z; x, \tau_p) \\ &= -E[\log\ \mathrm{K}(z; x, z) | r(z; x, \tau_p)\,] \end{aligned} \tag{1.30}$$

Following these steps, the expectation step is executed by calculating the likelihood function $E[\log \Psi(z; x, \tau) | r(z; x, \tau_p)]$. For the maximization step, the maximum value of this function is computed.

The EM algorithm is used in computer vision, natural language processing, machine learning, and other applications.

2.4.2 Forward-Backward Algorithm (FB)

The forward-backward algorithm is used to determine the probability of a particular observation sequence. This algorithm has three steps – computation of forward probabilities, computation of backward probabilities, and computing smoothing values.

During the first step, the algorithm moves forward from the first observation to the last. It then returns to the first observation. At each single observation in the sequence, probabilities to be used for calculations at the next observation are computed. The smoothing step takes place during the backward pass. This allows the algorithm to consider past observations of output for computing more accurate results.

The Baum-Welch algorithm is a variation of this algorithm.

2.4.3 Baum-Welch Algorithm (BWA)

The Baum-Welch algorithm is used to optimize the parameters of HMMs using empirical observations. It can compute maximum likelihood estimates and posterior mode estimates for the transition and emission probabilities of an HMM by using emissions as training data.

Upon computing the forward and backward probabilities, this algorithm determines the frequency of the transition-emission pair values and divides it by the probability of the entire string. This is done by calculating the expected count of the specific pair of transition-emission values. Each time a particular transition is found, the value of the previously mentioned quotient increases. This value is then made the new value of the transition (13).

Say, the observation sequence for an HMM is $O = o_1, o_2, o_3, \dots, o_l$. The sequence takes values from a set of observable symbols, $o_k \in \boldsymbol{V} = \{v_1, v_2, v_3, \dots, v_M\}$. In order to calculate the probability of the observation sequence, $\boldsymbol{O}$, each possible state sequence, $\boldsymbol{Q} = \{q_1, q_2, q_3, \dots, q_T\}$, is enumerated over. Here, the state sequence is given by $\boldsymbol{S} = \{s_1, s_2, s_3, \dots, s_N\}$. Given these variables, the probability of the observation sequence, $\boldsymbol{O}$, for the state sequence, $\boldsymbol{Q}$, is given by the following.

$$P(\boldsymbol{O}|\boldsymbol{Q}, \lambda) = \prod_{t=1}^{T} P(o_t|q_t, \lambda) \tag{1.31}$$

Here, $\lambda(\boldsymbol{A}, \boldsymbol{B}, \pi)$ denotes the parameters of the HMM. The above equation can be rewritten as following if statistical independence is assumed.

$$P(O|Q, \lambda) = b_{q1}(o_1) \cdot b_{q2}(o_2) \cdot \dots \cdot b_{qT}(o_T) \tag{1.32}$$

The probability of the state sequence, $\boldsymbol{Q}$, is expressed as below.

$$P(\boldsymbol{Q}|\lambda) = \pi_{q_1} a_{q_1 q_2} a_{q_2 q_3} \dots a_{q_{T-1} q_T} \tag{1.33}$$

Multiplying the previous two equations, the joint probability can be expressed as below.

$$P(O, Q|\lambda) = \pi_{p_1} b_{q_1}(o_1) \prod_{k=2}^{T} a_{q_{k-1} q_k} \, b_{q_k}(o_k) \tag{1.34}$$

Next, the Forward-Backward algorithm is applied. The forward variable, α, is defined as below (1).

$$\alpha_t = P(o_1, o_2, \dots, o_t, q_t = s_i|\lambda) \tag{1.35}$$

The algorithm, then, is run iteratively to compute the forward variables. The initialization, induction, and termination steps are done as shown in (13) (1) (3). Finally, the backward variables, β_t, are similarly defined and iterated.

Although not the fastest, the BWA adapts the parameters of the HMMs to maximize the probability of the observed sequence by summing the probability over all possible state sequences (1). It is also superior to the K-means algorithm as it takes a full-likelihood approach and is more computationally efficient compared to the genetic algorithm. The genetic algorithm, however, is better than BWA when a global approach needs to be taken to achieve more than one local maximum.

2.5 DISTANCE MEASURE OF HIDDEN MARKOV MODELS

Classification problems require the comparison between different profiles. Methods need to be employed that analyze and differentiate between the dissimilarity measures. As such, parameter estimation is a major part of HMM-based analysis.

Dissimilarity measures in a set of numerical profiles are usually analyzed in terms of distance of divergence. Distance measures use Euclidean distance between profiles to determine which profile the test value fits to. Divergence measures are used to analyze the statistical distance using complex methods.

The distance between two HMM profiles, $\lambda_1(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\pi})$ and $\lambda_2(\boldsymbol{A'}, \boldsymbol{B'}, \pi')$, has been used to successfully differentiate HMM profiles (14). The simple distance metric used in this thesis is the Euclidean distance between pairs of transition and emission matrices. The following equation was used for this distance measure (1).

$$D_{Eu}(\lambda_1, \lambda_2) = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{N}\left|a_{ij} - a'_{ij}\right|^2} + \sqrt{\frac{1}{N}\sum_{j=1}^{N}\sum_{l=1}^{M}\left|b_j(l) - b'_j(l)\right|^2} \tag{1.36}$$

Here, N and M denote the number of states and the number of observation symbols per state, respectively. This method is very simple and requires less computational burden compared to more complicated methods. As a result, this is a fast method at the price of improved accuracy. This method has been applied in this thesis since the applications considered were simulated for near-real time use.

The short-coming of this is method is the inability to analyze the probabilistic nature of the transmission and emission matrices. As a result, much of the characteristic information within the HMM profiles may be unused. Direct distance measure may result in conflicting situations where a test profile is equally distanced from two HMM training profiles. This leads to the analysis of HMMs in probability space as well as Cartesian space.

A popular method of divergence analysis is the Kullback-Leibler Divergence (KLD). This method measures the relative entropy by analyzing the average discrimination information between two random models (1). If the probability density functions of two random variables are denoted by $F_1(z)$ and $F_2(z)$, their KLD is defined as below.

$$D_{KL}(F_1||F_2) = \int F_1(z) \log \frac{F_1(z)}{F_2(z)} dz \tag{1.37}$$

Properties of KLD such as additivity, non-negativity, convexity, and asymmetry make it very useful for use in wireless communications.

CHAPTER 3: STOCHASTIC GAMES USING HMMS

3.1 GAME THEORY

Game theory is an area that studies how rational decision-makers would act and react to different situations. Games are characterized by a number of players or decision-makers who interact between each other and form coalitions, take actions under uncertain conditions, and receive some reward or punishment or monetary loss as a result of their decisions (15).

A classical example of Game Theory is the prisoner's dilemma. In this game, there exist two prisoners who have been apprehended for a crime and are held in separate rooms for interrogation. A clever prosecutor presents both the suspects with a number of options. Although both have the option to either confess or remain silent, their counterpart's decision would play a role in their prosecution. If one suspect confesses and the other remains silent, the one who co-operates would walk free while the other will be put away in jail for maximum punishment. If both confess and, thus, produce two convicting statements, they face reduced sentence. Lastly, if both remain silent, they get only a minimum punishment.

For rational players or controllers, such situations produce $games$. The solutions each have a certain $payoff$ associated with it. A grid representation of the solutions as well as their respective payoffs can be formulated for each game. The equilibrium point of these solutions is referred to as the Nash Equilibrium after the famous mathematician John Nash.

Despite conceptual similarities, the theories of HMMs and Game Theory have been independently studied. This chapter reviews some of the recent works in stochastic HMMs and their applications. The development of stochastic games and Markov decision processes parallels the development of differential games and optimal control theory, respectively, in the sense that they are both initially derived from single-controller, non-competitive cases. Upon more complex theoretical development, stochastic games and Markov decision process can be unified into a combined study of competitive Markov decision processes.

3.2 STOCHASTIC GAMES

Stochastic games are a generalization of the competitive Markov decision processes to the case of two or more $controllers$ or $players$. The classical case of the prisoner's dilemma mentioned earlier is a two-controller game since two controllers determine the outcome of the game. Likewise, a three-way dual can be an example of three-controller game, and so on. As the number of controllers increase, the difficulty of the analysis increases, too. Some of the definitions and properties related to stochastic games are discussed below.

3.2.1 Summable Markov Decision Processes

A summable Markov process, Γ, can be defined that is observed at time instances, $t = 0,1,2,3, \dots$. These time instances can be considered as Markovian $stages$ as shown before. The state at time instant t can be defined by S_t. If $\boldsymbol{S}$ is a finite set containing all the values, it can be defined as $\boldsymbol{S} = \{1,2,3, \dots, N\}$.

The process is controlled by one or more $controllers$ or decision-makers who choose an action, $a \in \boldsymbol{A}(s) = \{1,2,3, \dots, m(s)\}$ at time t if the process is in state s at that time. The choice can be regarded as a realization of a random variable, A_t, denoting the controller's choice at time t. Furthermore, the choice can be associated with a reward of payoff, $r(s,a)$, and a probabilistic transition to a new state, $s' \in \boldsymbol{S}$. As can be inferred from simple observation, this is a Markovian process (16).

3.2.2 Stationary Markov Transition Property

The stationary Markov transition property states that, for every $(s,s') \in \boldsymbol{S}$ and $a \in \boldsymbol{A}(s)$, the probability that $S_{t+1} = s'$ given that $S_t = s$ and the controller chooses an action, a, is independent of time and any previous states and actions. It can be expressed as below.

$$p(s'|s,a) = \Pr\{S_{t+1} = s'|S_t = s, A_t = a\} \tag{2.1}$$

This definition is consistent with the definition of a Markovian process explained in section 2.2 previously.

In order to understand the $rewards$ associated with each decision taken through competitive Markov processes, an example can be considered. Say, a Markov process has a finite state space, $\boldsymbol{S} = \{1,2,3\}$. The choice of possible actions can be defined as $\boldsymbol{A}(1) = \boldsymbol{A}(2) = \{1,2\}$ and $\boldsymbol{A}(3) = 1$. The defined system can be graphically expressed using the following structure (16).

Table 3: Rewards or payoffs associated with Stochastic Games. The table illustrates the probability of each decision in the right hand side of each cell. The left hand side indicates the risk or reward attached with each such decision.

-5	(0.9,0,0.1)
10	(0,1,0)

State 1

5	(0,1,0)
0	(0.8,0.2,0)

State 2

20	(0.9,0.1,0)

State 3

The values in boldface represent the reward or payoff associated with each choice of action. The values in parentheses represent the transition probabilities for each state. The significance of this model can be inferred from the table of values mentioned above. For instance, a choice of action 1 in state 1 will result in a cost of 5. Here, the reward, $r(1,1) = -5$. The transition probabilities are $p(1|1,1) = 0.9$, $p(2|1,1) = 0$, and $p(3|1,1) = 0.1$. The transition probabilities shown here are similar to those in a fully-connected Markov model, as explained in section 2.3.3.2 previously. The only difference is the rewards associated with each decision or transition.

In (16), the strategic analysis of stochastic games is demonstrated through a simple example. A strategy, $\boldsymbol{f}$, can be assumed such that $\boldsymbol{f} = (\boldsymbol{f}(1), \boldsymbol{f}(2), \dots, \boldsymbol{f}(s), \dots, \boldsymbol{f}(N))$ is a block row vector whose s-th block is a non-negative row vector $\boldsymbol{f}(s) = (f(s,1), f(s,2), \dots, f\big(s, m(s)\big))$ with entries that satisfy the Markov requirement $\sum_{a=1}^{m(s)} f(s,a) = 1$. Here, $f(s,a)$ signifies the probability that the controller chooses action $a \in \boldsymbol{A}(s)$ in state $s \in \boldsymbol{S}$ whenever s is visited.

In this case, a deterministic strategy would be one where $f(s,a) \in \{0,1\}$ for all $a \in \boldsymbol{A}(s)$, $s \in \boldsymbol{S}$. For each $s \in \boldsymbol{S}$, a pure control selects some action, a_s, with probability of 1 in state s whenever this state is visited.

The property that the controller's decisions in state s are invariant with respect to the time of visit to s is referred to as the $stationarity$ of the strategy (16).

3.2.3 $\boldsymbol{\beta}$-Discounted Markov Decision Model, $\boldsymbol{\Gamma_\beta}$

The rewards resulting from the choice of a particular competitive Markov strategy can have numerous inferences. One such method is the discounted Markov decision model. Let $\{R_t\}_{t=0}^{\infty}$ denote the sequence of random rewards with R_t being the reward for the period $[t, t+1)$. Once an initial state, $\boldsymbol{s}$, and a strategy, $\boldsymbol{f}$, are specified, then so is the distribution R_t for every $t = 0,1,2, \ldots$.

This happens because a probability tree can be created whose branches represent all possible state-action realizations up to time t and then assign each branch both a reward and a probability of observing that branch. Therefore, the expectation of R_t can be written as below.

$$E_{s\boldsymbol{f}}[R_t] := E_{\boldsymbol{f}}[R_t | S_0 = s] \tag{2.2}$$

Next, the overall discount value of the chosen strategy, $\boldsymbol{f}$, from the initial state, s, can be defined by the following.

$$v_\beta(s, f) := \sum_{t=0}^{\infty} \beta^t E_{s\boldsymbol{f}}[R_t] \tag{2.3}$$

Here, $\beta \in [0,1)$ is called the discount factor. The model uses the above equation as its performance criterion. It is sometimes referred to as the discounted Markov decision process (DMD) (16). As can be inferred from the equation above, the reward or output of one unit at a time $t+1$ is worth only $\beta < 1$of what t was worth at time t.

3.2.4 Termination Markov Decision Model, $\boldsymbol{\Gamma_\tau}$

Termination Markov decisions are derived from the stationary Markov transition property mentioned in 3.2.2 . In this model, the following relaxed model is assumed.

$$\sum_{s'=1}^{N} p(s'|s, a) < 1 \tag{2.4}$$

Here, the equation holds for all $a \in \boldsymbol{A}(s)$, $s \in \boldsymbol{S}$. This assumption has a the interpretation that, with every action $a \in \boldsymbol{A}(s)$ selected in every state s, there is a positive stopping probability as below (16).

$$p(0|s,a) := 1 - \sum_{s'=1}^{N} p(s'|s,a) > 0 \tag{2.5}$$

This signifies the termination of the process the absorption in an artificial absorbing state 0. According to this model, a transition matrix of $P(\boldsymbol{f})$ has the following property.

$$\sum_{s'=1}^{N} p(s'|s,\boldsymbol{f}) < 1 \tag{2.6}$$

This equation holds for all $s \in \boldsymbol{S}$. It is, then, analogous to the DMD mentioned before and has a terminating value vector, $\boldsymbol{f}$, as below.

$$v_\tau(\boldsymbol{f}) := \sum_{t=0}^{\infty} P^t(\boldsymbol{f})\boldsymbol{r}(\boldsymbol{f}) = [I - P(\boldsymbol{f})]^{-1}\boldsymbol{r}(\boldsymbol{f}) \tag{2.7}$$

The corresponding terminating optimal Markov control problem is the optimization problem $\max v_\tau(\boldsymbol{f})$ subject to $\boldsymbol{f} \in \boldsymbol{F}_S$.

3.3 BEHAVIOR AND MARKOV STRATEGIES

Stochastic games are a generalization of the Markov decision processes to the case of two or more controllers. This enables behavioral analysis using Markov strategies. The discounted stochastic games defined in 0can be generalized for two controllers to demonstrate some of the strategic decision-making.

If the process is in state $s \in \boldsymbol{S} = \{1,2,\dots,N\}$ at time t, players 1 and 2 can independently choose actions $a^1 \in \boldsymbol{A}^1(s)$ and $a^2 \in \boldsymbol{A}^2(s)$. The respective rewards are $r^1(s,a^1,a^2)$ and $r^2(s,a^1,a^2)$. The stationary transition probabilities can be generalized to the following.

$$p(s^{'}|s,a^1,a^2) := P\{S_{t+1} = s^{'}|S_t = s, A_t^1 = a^1, A_t^2 = a^2\} \quad (2.8)$$

This holds for all $t = 0,1,2,....$ Here, S_t is the state at time t, and A_t^1, A_t^2 denote the actions chosen by players 1 and 2. The fact that the rewards and transition probabilities depend on the actions of both players as well as the current state, implies that the fate of the two players is coupled in this process, even though their choices of actions are independent of one another (16).

Say, the sets of stationary strategies for two players are denoted by $\boldsymbol{F_S}$ and $\boldsymbol{G_S}$ and the set of choices by the two players is denotes by $(\boldsymbol{f}, \boldsymbol{g})$. The respective rewards for two players can be denoted by $r^1(\boldsymbol{f}, \boldsymbol{g})$ and $r^2(\boldsymbol{f}, \boldsymbol{g})$. Therefore, the set of all possible strategy pairs is $(\boldsymbol{f}, \boldsymbol{g}) \in \boldsymbol{F_S} \times \boldsymbol{G_S}$. In general, the discounted value of a strategy pair to player k can be given by the following equation.

$$v_\beta^k(s, \boldsymbol{f}, \boldsymbol{g}) := \sum_{t=0}^{\infty} \beta^t E_{s\boldsymbol{fg}}(R_t) \quad (2.9)$$

This is a Markov problem for the players if they attempt to maximize their rewards, as expected.

The classical non-cooperative assumption of game theory states that players choose their decisions independently, and attempt to maximize personal reward functions only. When the added assumption that each player knows about the other's reward function is included, the concept of Nash Equilibrium is introduced. A strategic pair, $(\boldsymbol{f}^0, \boldsymbol{g}^0)$, can be a Nash equilibrium in a two-player game if it fulfills the following two conditions.

$$v_\beta^1(\boldsymbol{f}, \boldsymbol{g}^0) \leq v_\beta^1(\boldsymbol{f}^0, \boldsymbol{g}^0), \boldsymbol{f} \in \boldsymbol{F_S} \quad (2.10)$$

$$v_\beta^2(\boldsymbol{f}^0, \boldsymbol{g}) \leq v_\beta^2(\boldsymbol{f}^0, \boldsymbol{g}^0), \boldsymbol{g} \in \boldsymbol{G_S} \quad (2.11)$$

Numerous Nash equilibriums can exist in a game with various payoffs associated with them. Some examples of its application are mentioned in the following sections.

3.3.1 Travelling Inspector Model

One example of an application of stochastic game theory and Markov decision process is the travelling inspector model (16). This game consists of two players – an inspector who is tasked with finding and fining plants that dump toxic waste illegally, and plant controllers who intend to dump pollutants while avoiding the inspector. Each result of the two players' choices results in a reward or cost for the two parties. Since a previous action affects the future actions by the two players, this is a Markovian process, too.

Since real-life reward functions are very difficult to determine for such policing actions, an assumption of aggregation is suggested that assumes that inspector and the plant controllers have antagonistic interests (16). This game is controlled by the inspector as he can independently choose where to inspect. However, the reward for the inspector is associated with what the plant controllers decide. This game also needs to be played for a long time such that the stochastic nature can play out, and a limiting average for the rewards is reached. The reward for the plants is to be able to dump the waste successfully, and the cost is the heavy fine charged by the government if caught. For the inspector, the cost function consists of the cost to move from one location, s, to another, s', denoted by $\sigma(s, s')$. The reward for the inspector is associated with the tonnage of waste he catches the plants dumping. Depending on the actions chosen by the inspector, $\boldsymbol{a} = \{a_1, a_2, \dots, a_N\}$, the tonnage of waste can be denoted by $\tau_s(\boldsymbol{a})$. Given these functions, the inspector's immediate loss can be determined by the following function.

$$C(s, \boldsymbol{a}, s') = \sum_{\bar{s} \in \boldsymbol{S}} C(s, a_{\bar{s}}, s') \tag{2.12}$$

Where,

$$C(s, a_{\bar{s}}, s') := \begin{cases} \tau_{\bar{s}}(\boldsymbol{a}) \\ \sigma(s, s')\tau_{\bar{s}}(\boldsymbol{a}) - 2[1 - \sigma(s, s')]\tau_{\bar{s}}(\boldsymbol{a}) \end{cases} \tag{2.13}$$

A sample violation cost in tons of waste is shown below (16).

Table 4: Sample violation cost in tons of waste for the Travelling Inspector Model. The tonnage of waste is denoted by τ. s denotes the next position. a denotes the action chosen by the inspector.

$\tau_s(\boldsymbol{a})$	$s = 1$	$s = 2$	$s = 3$
$a_s = 0$	0	0	0
$a_s = 1$	20	10	20
$a_s = 2$		15	25

Similarly, a table of the inspector's travel costs can be depicted as below.

Table 5: Sample travel cost for the Travelling Inspector Model. σ denotes the cost of moving from position s to position s'.

$\sigma(s, s')$	$s' = 1$	$s' = 2$	$s' = 3$
$s = 1$	0	0.3	0.6
s = 2	0.3	0	0.3
$s = 3$	0.6	0.3	0

Interpretation of the first table is simplistic in the sense that each choice of the inspector is associated with a definite amount in loss. The second table, however, is stochastic in the sense that it is made up of the probability of missing toxic dumping at the original location, s.

3.3.2 Duopoly Model

A duopoly situation can be considered where two competing firms make similar products. In a fixed market volume, both firms' target is to gain the highest market share possible. The action of each firm, i, can be denoted by (a_i, p_i, t_i) where a_i denotes the level of advertisement, p_i denotes the price that the respective firm sets, and t_i indicates the level of the development of new technologies.

Both firms update their decisions at time $t = 0,1,2, \ldots$. The market share of a firm depends on both the levels of advertisement and the prices. The function representing this is $m_i(a_1, a_2, p_1, p_2) \in [0,1]$. If N is the market volume, the production cost of $m_i N$ items can be expressed by the function $c_i(t_i)$. The profit of firm i can be expressed by the following equation (16).

$$r_i(a_1, p_1, t_1, a_2, p_2, t_2) := p_i m_i(a_1, a_2, p_1, p_2) N - m_i(a_1, a_2, p_1, p_2) N c_i(t_i) - a_i \tag{2.14}$$

When a certain play, (a_1, p_1, t_1) or (a_2, p_2, t_1), is decided by the firms, the new state becomes $(m_1(a_1, a_2, p_1, p_2), m_2(a_1, a_2, p_1, p_2))$. For any state, $m_1(.) + m_2(.) = 1$. This basic model can be extended to accommodate more complex situations. Since consumer behavior is less predictable than probability distributions, the model often requires modifications that incorporate real stochastic transitions.

3.3.3 Strike Negotiation Model

Another example of stochastic games and HMMs is a strike negotiation model (16). In this scenario, it is assumed that the management and the union are negotiating on salary. Each day, both parties in the negotiation re-adjust their strategy. In this respect, the game is Markovian.

Let it be assumed that, at a certain day $(t-1)$, the management had offered an increase of $x_1(t-1)$ while the union demands $x_2(t-1)$, where $x_2(t-1) > x_1(t-1)$. Therefore, the renewed choices on day t will be as below.

$$x_k(t) \in [x_1(t-1), x_2(t-1)], k = 1,2 \tag{2.15}$$

As expected, the strike will continue if $x_2(t) > x_1(t)$. In this scenario, the reward structure becomes significant because each party has a price to pay for each day that the strike continues. The cost for the management each day can be denoted by $l(t)$. Similarly, the cost for the union will be incurred from suspended salary. Upon deducting the financial support provided by a strike fund, the daily loss incurred by the union can be defined by $s(t)$. Say, the agreeable solution to this strike is $x_a = \frac{1}{2}\big(x_1(t) + x_2(t)\big)$.

The final agreements of such negotiations are usually based on future prospects. The utilities for each party are denoted by the function $f_i(x_a, t)$. Here, $f_2(.)$ denotes the positive for the union and $f_1(.)$ denotes the cost for the management. If the unknown decision moment is denoted by time t_a, the management will try to minimize the following function.

$$(1-\beta)\sum_{\tau=0}^{t_a-1}\beta^{\tau} l(\tau) + (1-\beta)\beta^{t_a} f_1(x_a, t_a) \tag{2.16}$$

Likewise, the union will try to maximize the following function.

$$(1-\beta)\beta^{t_a} f_2(x_a, t_a) - (1-\beta)\sum_{\tau=0}^{t_a-1} \beta^{\tau} s(\tau) \tag{2.17}$$

Upon execution of this game, a steady state will be reached.

CHAPTER 4: POSITION LOCATION USING HIDDEN MARKOV MODELS

4.1 INTRODUCTION

It is often imperative to accurately find the location of wireless receivers. Locating an access point can be of vital importance in many areas such as public safety, mining, law enforcement, traffic directions, and rescue operations. As such, various methods have been developed over time to identify wireless receivers.

Location sensing of wireless receivers can be of great importance in cognitive radios. With proper knowledge about the receivers, cognitive radios can potentially accommodate high data rate multi-media applications through improved radio resource management (1). Classical methods generally employ line of sight (LoS)-based procedures. Such methods present various limitations during inclement conditions. Therefore, it is important to formulate methods that do away with the LoS requirement and provide a level of awareness to the radio.

A Hidden Markov Model (HMM)-based solution to this problem was presented in (1). The suggested solution used power delay profiles (PDP) of wireless signals to train HMM profiles. This method was particularly useful in situations where multipath fading occurs and LoS is not available. In this thesis, the methods used in (1) are evaluated and simulated under different conditions.

4.2 RELATED WORK

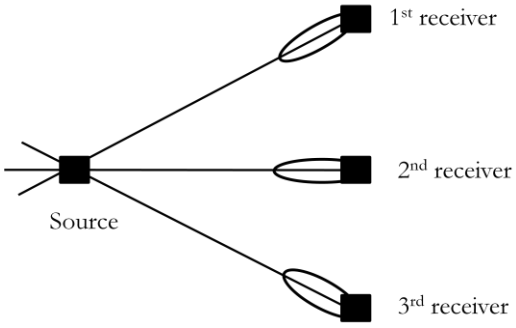
Classical position location algorithms are based on radio frequency (RF) wave propagation. Electromagnetic waves at RF are capable of penetrating most objects. Therefore, position location has been attempted by taking direct measure of the signals. Some of the most common techniques are as below.

4.2.1 Angle of Arrival Based Technique

Techniques based on angle of arrival are implemented by estimating the direction of the signal of interest. When a single measurement is taken, there is a high risk of getting erratic locations. To offset the limited accuracy of single measurements, multiple observations are used to accurately tri-

laterate the location of signal. The following figure shows an example of angle of arrival based position location.

Figure 7: Example of angle of arrival-based position location.



The method, although simple, has some limitations. It is required that there exists a LoS between the transmitter and the receiver. Furthermore, there needs to be a narrow angle of spread. This is especially difficult in wireless communication systems due to the presence of multipath fading. To ensure accuracy, a number of antennas are required. The arrangement would require regular calibration to ensure accuracy of measurement.

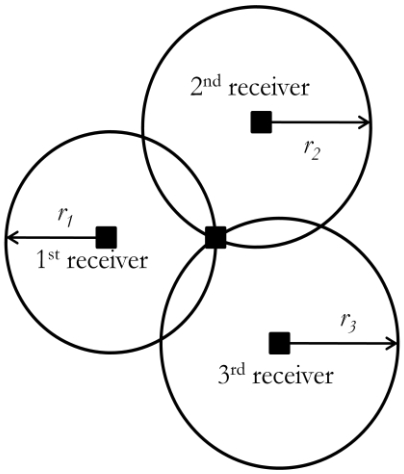
4.2.2 Time of Arrival Based Technique

In time of arrival based techniques, receivers determine the time taken by the signal to propagate from the source to the receiver. It is achieved by measuring the time in which the source responds to an inquiry or an instruction transmitted to the receiver. This measurement includes the roundtrip delay (RTD) as well as other delays (processing, queuing, and transmission). If the processing delays within the source are known, it is possible to estimate the RTD. Half of that quantity represents an estimate of the signal delay in one direction, which then provides an approximate distance of the source from the receiver (1).

Once the process is property tested and established, multiple measurements can be taken from various different locations. They can be later triangulated to estimate the location of the source.

If the source response can be detected at two additional receivers, the triangulation method can be employed to find the position of the source at the intersection of the circles determined by the time delay measurements.

Figure 8: Example of time of arrival-based position location.



This method is also used in other applications. One such case is a scheduling algorithm for radio frequency identification (RFID) readers to avoid collision. This anti-collision algorithm schedules readers depending on the amount of power left in them. Post-processing is applied to reduce possible duplicates.

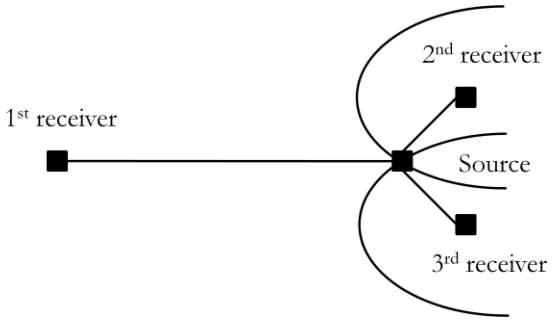
The disadvantage of using this method is that it is dependent on time reference. A single incidence of timing error can have severe ripple effect in the system. The absence of a line of sight component can cause errors due to multiple signal reflections (17).

4.2.3 Time Difference of Arrival Based Technique

The time difference of arrival (TDoA) technique is based on the difference in the arrival times of the signal from the source at multiple receivers. This is achieved by taking a snapshot of the signal at multiple receivers over a synchronized period of time. Next, the different versions of the signal are cross-correlated. The time difference is calculated by comparing the peaks of the cross-correlated output.

Once the time difference is estimated, it can be used to define a hyperbola between the two receivers assuming that the source and the receivers are co-planar (1). The method can be used for all of the numerous receivers. Finally, the resulting hyperbolas can be used to estimate the location of the source. The following image (1) demonstrates the hyperbolic position location scheme used in TDoA.

Figure 9: Example of hyperbolic position location scheme.



The TDoA method is also referred to as the hyperbolic position location method (1). The TDoA method has a specific advantage over the ToA method in the sense that TDoA does not require any time reference to estimate position. However, there needs to be accurate synchronization between receivers. Since any timing bias is cancelled in the differential time measurement, the TDoA methods outperforms the ToA methods in the absence of line of sight propagation between the source and the receivers (1). The disadvantage of TDoA is that additional timing delay due to non-line of sight propagation may result in errors in TDoA based location measurements.

4.2.4 Hybrid Techniques

The methods discussed previously can be combined to form hybrid techniques in order to achieve better performance. When the AoA and ToA methods are combined, it is possible to locate a source even when only one receiver receives the signal with sufficient strength. The hybrid scheme is the only applicable solution in such cases. Similarly, the AoA and the TDoA methods can be combined to form an AoA-TDoA hybrid scheme. With this scheme, multiple receivers receive signals from the source unit. The AoA estimates from each receiver and the TDoA estimates between multiple receivers are combined to determine the target location (1). Although this combination improves the accuracy, there remains a risk of the entire system performance being subjected to degradation due to one error. Such ripple effects need to be addressed in the hybrid systems.

Although these methods are simple in the sense that they use geometric principles only, they all require a line of sight reference between the transmitter and the receiver. Otherwise, the accuracy of the system gets severely degraded. Wireless communication systems can offer significant improvement because, in this field, there has been extensive research on multipath fading in case of non-line of sight

(NLoS) propagation. It is practically impossible to apply triangulation and tri-lateration in these situations. HMM based position location provides a significant advantage in this area.

4.3 HMM-BASED POSITION LOCATION

In (1), HMMs were used to formulate a position location application that is more accurate than the traditional methods. This method uses Power Delay Profiles (PDP) to obtain information about the channel. While traditional methods utilize angle and time of arrival only, PDP-based techniques utilize more technical information. Much of the extra information – such as the presence of a line of sight (LoS) component, multipath component strengths, and number of multipath components, delay spread – allows better modeling of the wireless channel. In fine, this method extracts location-dependent features from the PDP and uses HMM to exploit the signature statistics to identify locations (1).

As discussed in previous sections of this thesis, neither HMMs nor their application as classifiers are novel. The classification methods have been used previously in handwriting recognition, speech recognition, cryptanalysis, gene prediction, and machine learning. There has also been use of Artificial Neural Networks (ANN) for pattern matching algorithms (3).

The PDP-based techniques are especially useful in fading multipath channels.

4.3.1 Fading Multipath Channels

A fading multipath channel has $three$ unknown variables – number of paths, power attenuation, and delay profile. The relationship between attenuation and delay is expressed by the following equation (18).

$$c(\tau; t) = \alpha(\tau; t)e^{-j\,2\pi f_c \tau} \tag{3.1}$$

Here, $c(\tau; t)$ represents the response of the channel at time t due to an impulse applied at time $(t - \tau)$, and $\alpha(\tau; t)$ represents the attenuation of the signal components at delay τ and at time instant t.

The impulse response $c(\tau; t)$ from the multipath equation can be used to model the channel under various conditions. If $c(\tau; t)$ is a zero-mean, complex-Gaussian process, the envelope $|c(\tau; t)|$ at any instant t is said to be $Rayleigh$ distributed. The Rayleigh distribution contains a line of sight component.

Under fixed scatterers and signal reflectors, $c(\tau; t)$ can no longer be modeled as zero-mean. In such cases, the envelope $|c(\tau; t)|$ has a $Ricean$ distribution.

The Gaussian and Rayleigh distributions can be obtained from the Ricean distribution under special conditions. This work initially assumes Ricean distribution, and later varies the parameters to evaluate Gaussian and Rayleigh channels.

4.3.2 Ricean Power Delay Profiles

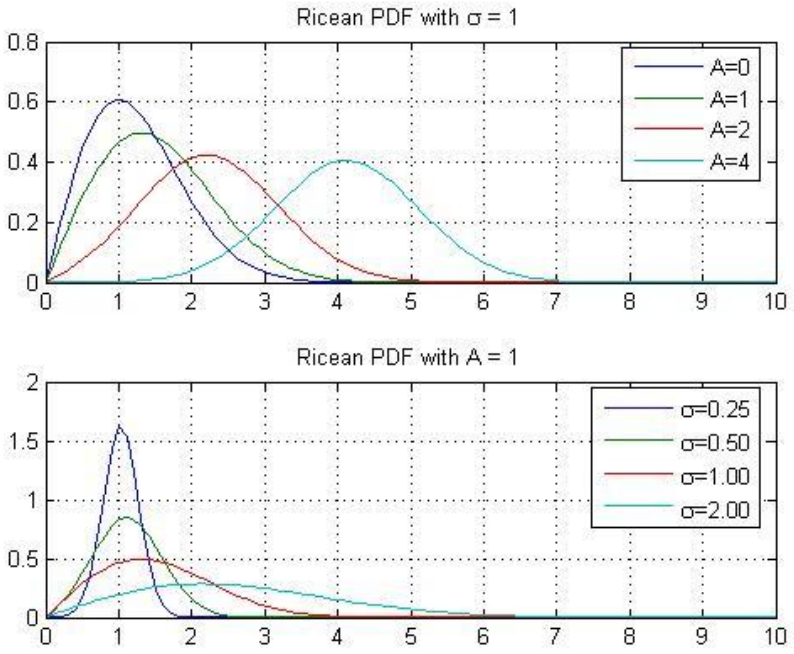
The input data is simulated using a Ricean approximation. A Ricean channel is defined as below.

$$f_R(r) = \frac{r}{\sigma^2} I_0\left(\frac{A_r}{\sigma^2}\right) \exp\left(-\frac{r^2 + A^2}{2\sigma^2}\right) \tag{3.2}$$

Here, A is the amplitude of the message signal sinusoid, r is the Jacobian of the transformation, $\sigma^2$is the common variance, and $I_0(.)$ is the modified Bessel function of first kind. The Ricean K-factor is defined as $\frac{A^2}{2\sigma^2}$. A Jacobian matrix is the matrix of all first-order partial derivatives of a vector-valued function. The Doppler effect is the change in frequency and wavelength of a wave as perceived by an observer moving relative to the source of the waves.

The following plot shows the simulation results for various values of K and A.

Figure 10: Ricean probability distribution functions. Here, A is the amplitude of the message signal sinusoid, σ is the standard deviation, K is the Ricean K-factor. The X-axis shows the K-factors. We observe that a Ricean distribution becomes Gaussian for high values of K.



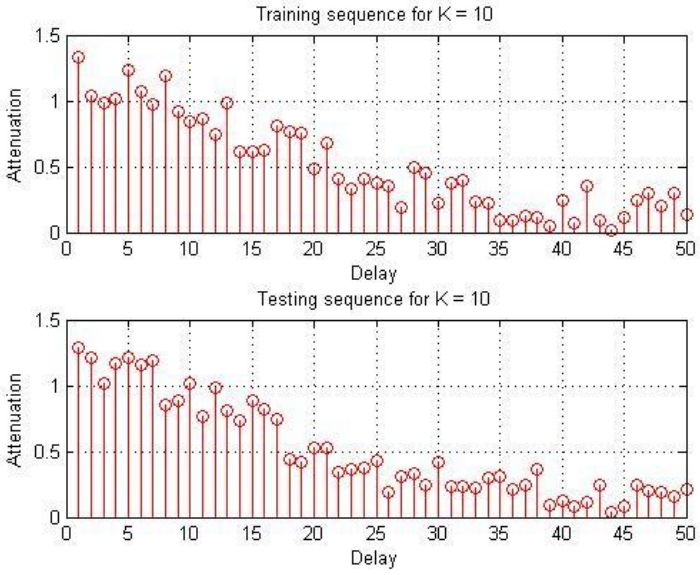
It can be observed from Figure 10 that, for a high value of K, the Ricean PDF becomes Gaussian. The Gaussian case was exclusively considered in (1).

4.3.3 Power Delay Profiles

Power Delay Profiles (PDP) are used in communication systems to observe the nature of multipath fading of a system. A typical PDP contains information about the attenuation, delay spread, and the Doppler Effect. However, the Doppler Effect is considered for moving objects only.

Not considering the Doppler Effect allows the use of 2-D plots for multipath fading analysis. Figure 11 shows a multipath fading channel in absence of a Doppler spread. Multipath fading affects the signal by dispersing it (time spreading and frequency selectivity) and adding a time-variant behavior.

Figure 11: Simulated power delay profiles for Ricean distributions. This figure demonstrates the HMM-based training and testing schemes using simulated data for a specific Ricean *K*-factor.



PDP is defined, as an extension from the delay and attenuation equation, as the following.

$$p(\tau) = E[|\hat{c}(\tau; t)|^2] \tag{3.3}$$

The position location algorithm in (1) extracted key 'features' from the signal by taking Fourier Transforms (FT). This process can be better understood by looking at a practical example. Say, the information retrieved from the PDP provides relative attenuation values of $[1.0\ 0.0\ 0.35\ 0.1]$ at multipath components $[1\ 0\ 3\ 4]$. If the message signal is defined by $x = 10\ sin(2\pi 1000t)$, then the values of the received signal at $t = [1\ 2\ 3\ 4]$ will be $r = [10.0\ 0.0\ 3.5\ 1.0]$. Performing an FT would allow the extraction of these values analytically.

4.3.4 Previously Used Method

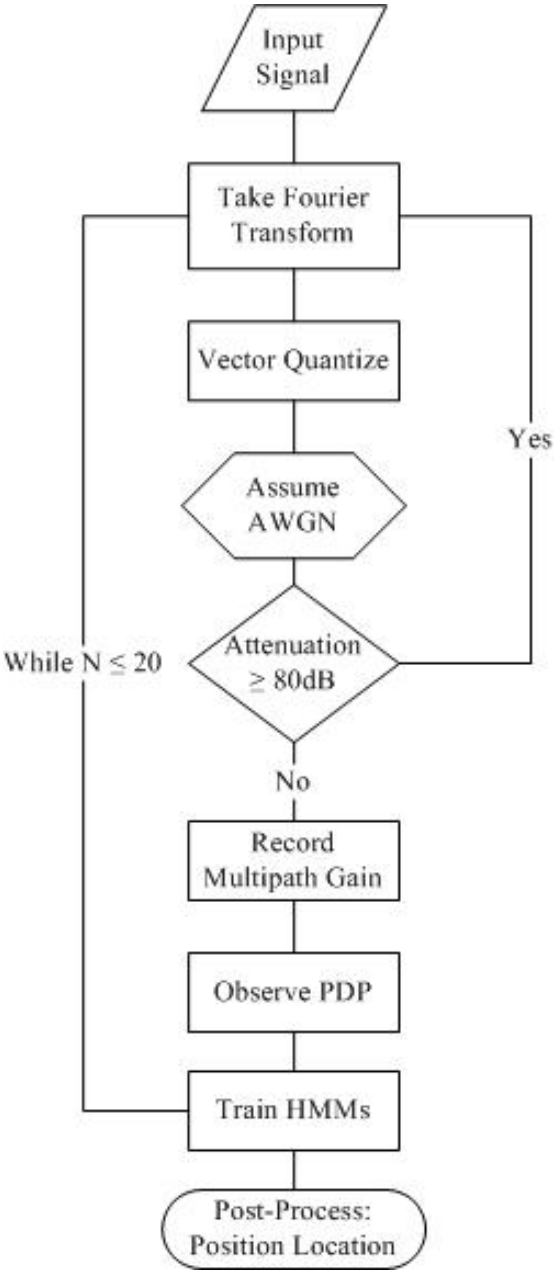
The previous work prepared a theoretical method that extracts signature characteristics from power delay profiles. These were used to discriminate between locations. Next, a suitable classification model was selected (1). During the first step, the required features were extracted from the PDP using Fourier transforms. Vector quantization was applied to convert the extracted features to an

observation sequence. Finally, HMM-based classification was done using training and testing procedures.

HMM-based profiles were created for each receiver location. Various assumptions were made during this step. Attenuation of less than $80\ dB$ was ignored assuming it to be thermal noise. HMM profiles $\lambda_k = (\boldsymbol{A}, \boldsymbol{B}, \pi)$ were created for the k receivers.

The algorithm used in (1) is illustrated below.

Figure 12: Position location algorithm used in prior work. This algorithm is used in a different environment to verify the methodology shown in the figure.

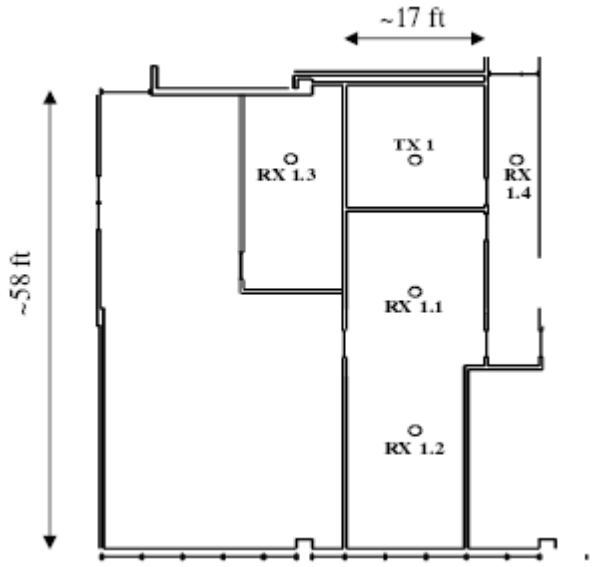


This method has two very significant advantages. Firstly, it exploits many of the relevant concepts in modern wireless system (multipath, fading, delay spread), thus enabling further research. Secondly, this method is not dependent on a specific LoS component and works seamlessly for both indoor and outdoor environments. However, a downside is the increased complexity of the system. Position location using a similar process was simulated in new experimental conditions.

4.3.5 Floor Map of Previous Test Scenario

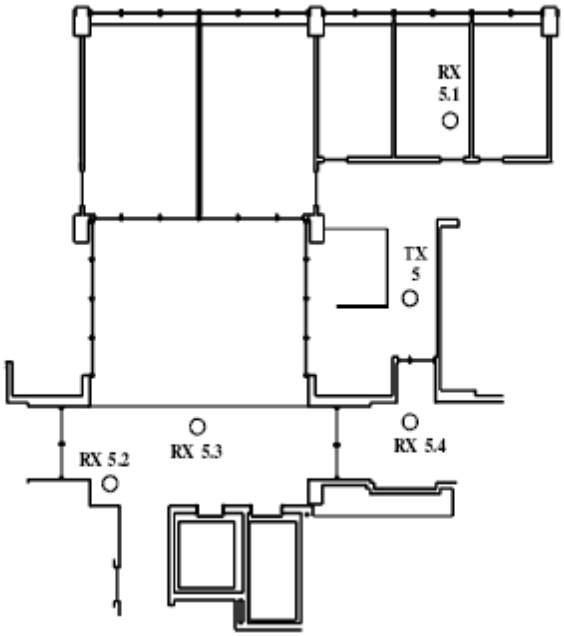
The previous work was conducted on the fourth floor of Durham hall (1). Two separate observations were done for the experiments. The following floor map shows the setup where the transmitter was placed in room no. 471. Four receivers were positioned as shown below.

Figure 13: Floor map of first experimental condition used in prior work.



The experiment was also done in another setup. The following figure shows the floor map for the second set of measurements. Here, the transmitter was placed in room no. 432 of Durham hall. The four receivers were positioned as shown in the figure below.

Figure 14: Floor map of second experimental condition used in prior work.



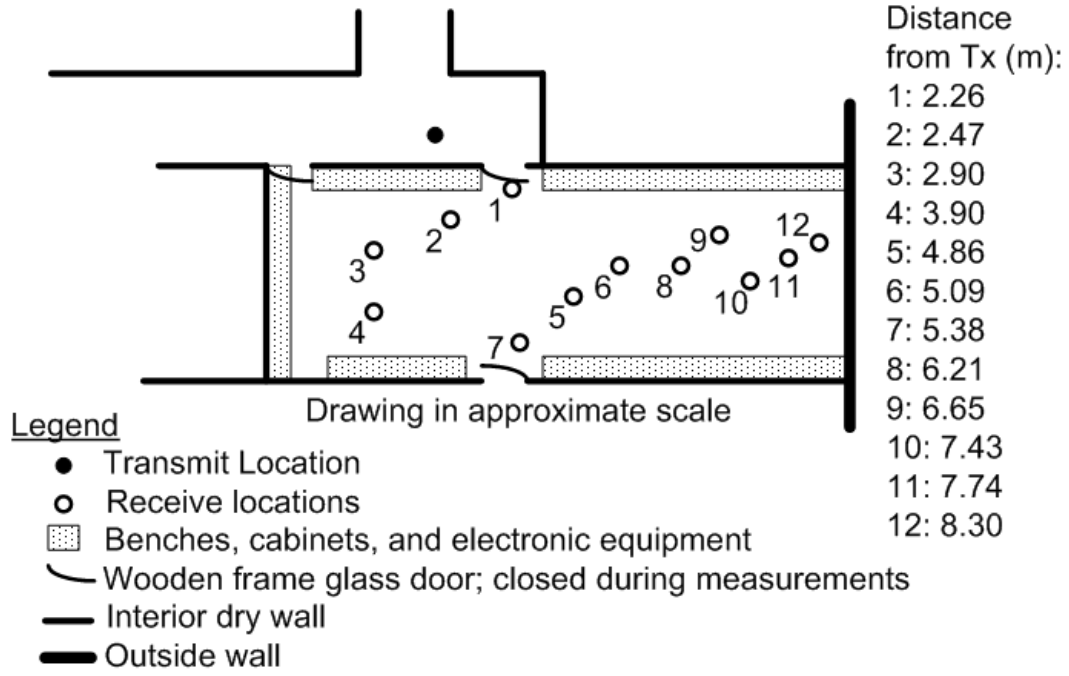
Measurements were taken in these conditions as part of an MS thesis (19). The observation sequence for the floor maps during testing were compared with original sequences observed during training the HMM profiles. The results were presented in a confusion matrix in (1).

For each of the receivers, 40 PDP values were taken. The quarter-wave frequency was $30mm$. Of the 40 values, 20 values were used for training and 20 values were used for testing. 4-state Markov models were used for all three receivers. Each state denoted the probability of transitioning to the other three receivers.

4.4 SIMULATION METHODOLOGY

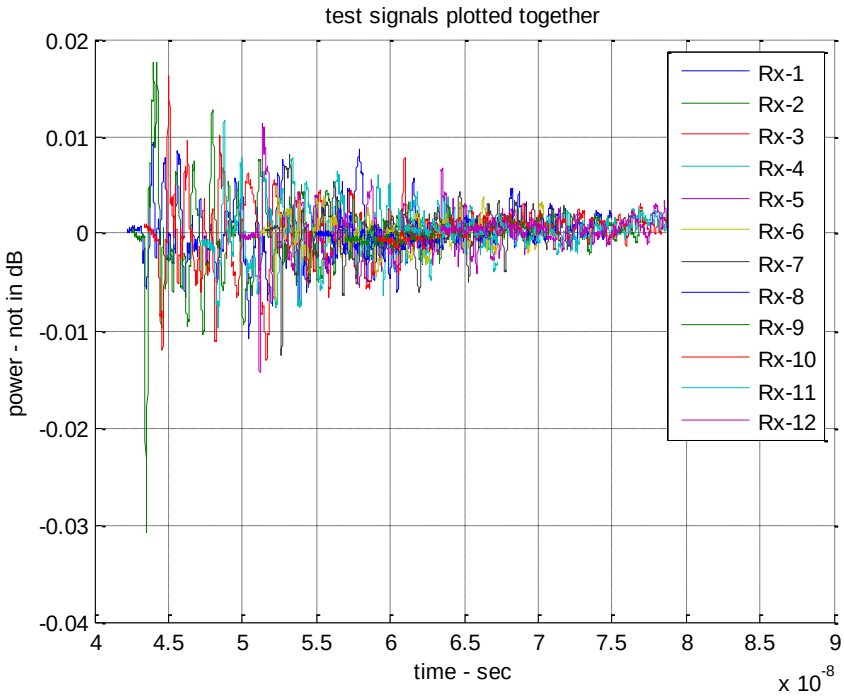
The work done for this thesis adds value to the previous work in two ways. Firstly, a fresh set of variables are used courtesy of Mr. Haris Volos of Wireless@VT. The following figure shows the floor map of the test setup.

Figure 15: Floor map of transmitter and receiver locations. Measurements for this setting have been used in the current work to verify the previously used methodology.



At each of the 12 receiving locations shown in Figure 15, PDP measurements for the transmitted signal were taken. Each set of data contained 2000 points. The measured signals are shown in a superposed image of all PDP values in Figure 16.

Figure 16: All measured signals plotted together (not in dB). The congested signal space illustrates the importance of proper position location and classification.



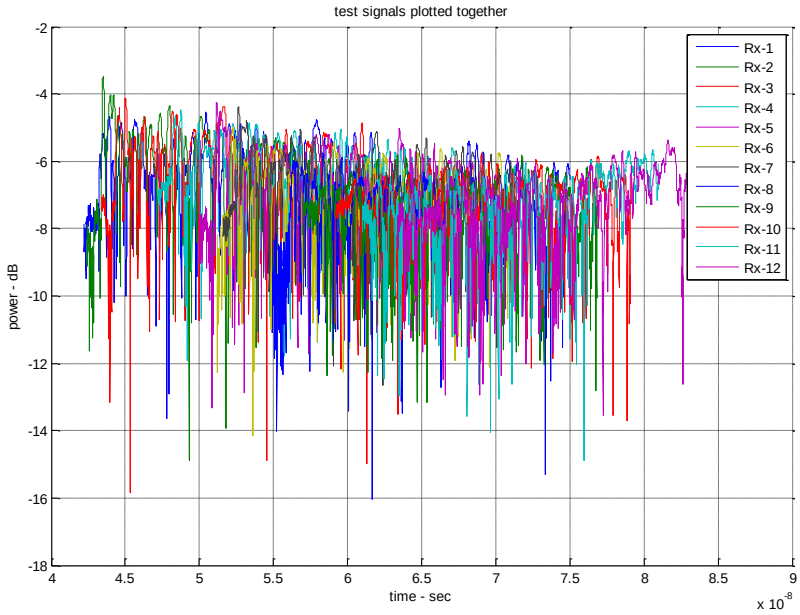
The figure demonstrates some important aspects of the simulation that require attention and processing. It can be observed that, due to the uneven distance of the receivers from the transmitter, there is a noticeable time delay between the various signals. Also, the presence of burst errors is observable, too.

The observed features are common to any wireless signal. They demonstrate practical variance from ideal Rayleigh or Ricean distributions, as expected theoretically. In order to use the available information properly, there needs to be a strong statistical tool that addresses these issues.

HMMs are useful in these situations as they allow symbol-by-symbol analysis as opposed to wavelength-level analysis. A first step towards using HMMs is to convert the measured data into the dB-scale. This makes it easier to infer the properties of the signal.

The following figure shows the PDP of receivers at all 12 receivers.

Figure 17: All test signals plotted together (in dB).This envelope of all signals illustrates the burse error in the signals. The fading multipath profile can also be observed from this figure.



The congested values show the importance of applying proper classification. Given the noticeable dip in signals during errors, it is possible to set an error threshold. Values below that level can be deemed in error. However, since the range of signal power varies depending on the location of the receiver, there needs to be heuristic adjustment to the threshold for receivers located further away from the transmitter. The correlation between distance and accuracy and the sensitivity to error threshold have been studied in this thesis.

The first 200 values of each set of 2000 values have been used for the simulation. In each case, alternate values were sampled to prepare $\boldsymbol{training}$ and $\boldsymbol{testing}$ sets. Training sets were used to create HMM profiles for each receiver location. The test sets were used to check the confidence and accuracy of the profiles created during the training phase.

Each HMM profile was denoted by $\lambda_k = (\boldsymbol{A}, \boldsymbol{B}, \pi)$, where k denotes the receiver number, and $\boldsymbol{A}$, $\boldsymbol{B}$, and π denote the transition matrix, emission matrix, and the initial probability, respectively. The testing profiles had a similar structure, i.e., $\lambda'_k = (\boldsymbol{A}', \boldsymbol{B}', \pi')$.

The method used to test the accuracy of the models was to compare the distance between the transition and emission matrices. This is the quickest method of classifying different HMM profiles.

Another possible method is to compare the symbols of each classified to signal to determine the accuracy.

The first phase of the simulation, as described thus far, determined the location of the receivers. The parameters of HMM profiling were relaxed during the second phase of the simulation to determine the usefulness of the methods used in case of signal fluctuation and receiver motion. For this, groups of receivers were clustered into grids according to their distance from the transmitted. These methods are discussed in further detail later.

The plot in Figure 17 shows all the signals, making it difficult to decipher the nature of the signals. Therefore, the training PDPs for all the receivers are shown in the following figures.

Figure 18: Training signal for Rx 1.

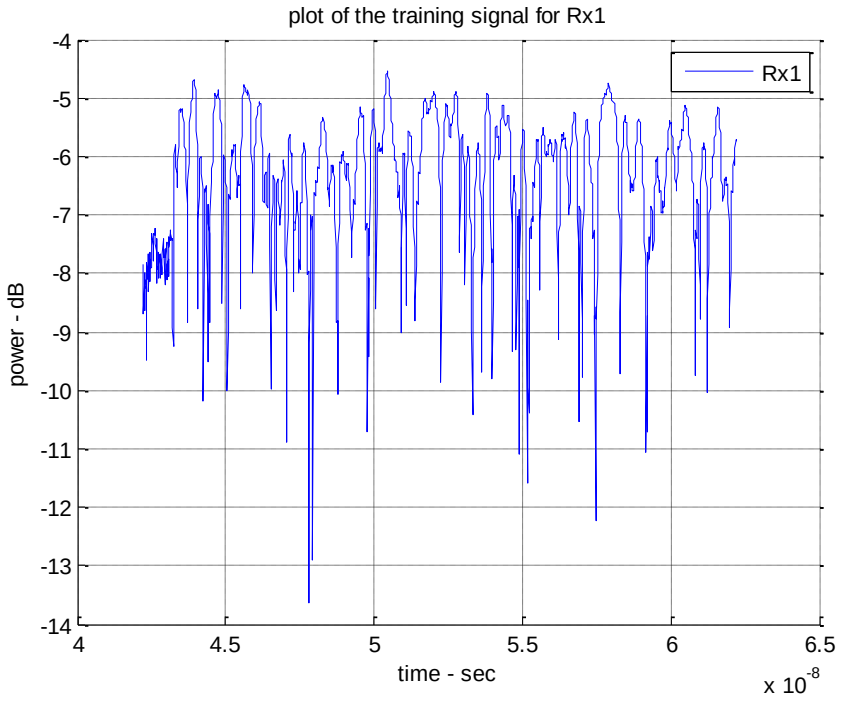


Figure 19: Training signal for Rx 2.

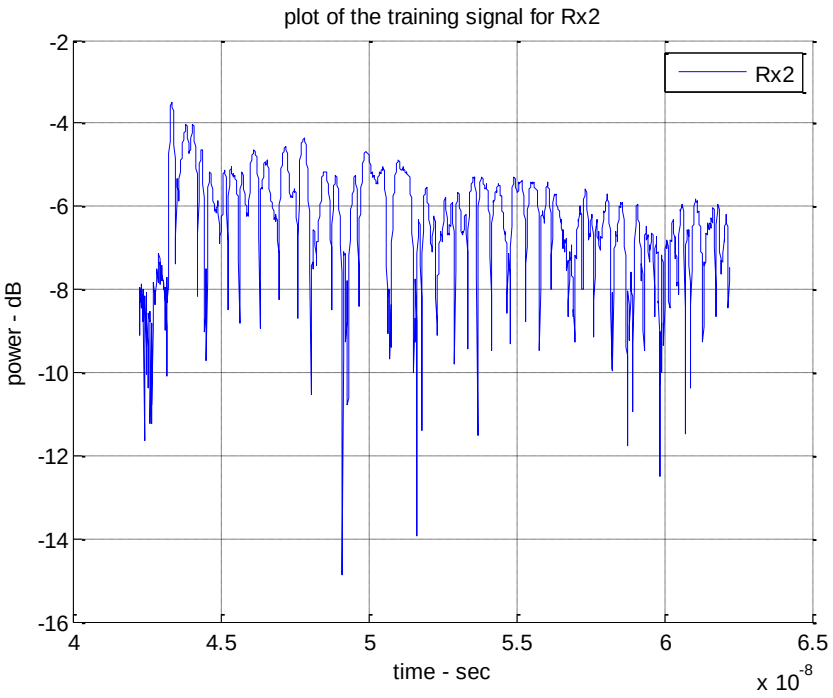


Figure 20: Training signal for Rx 3.

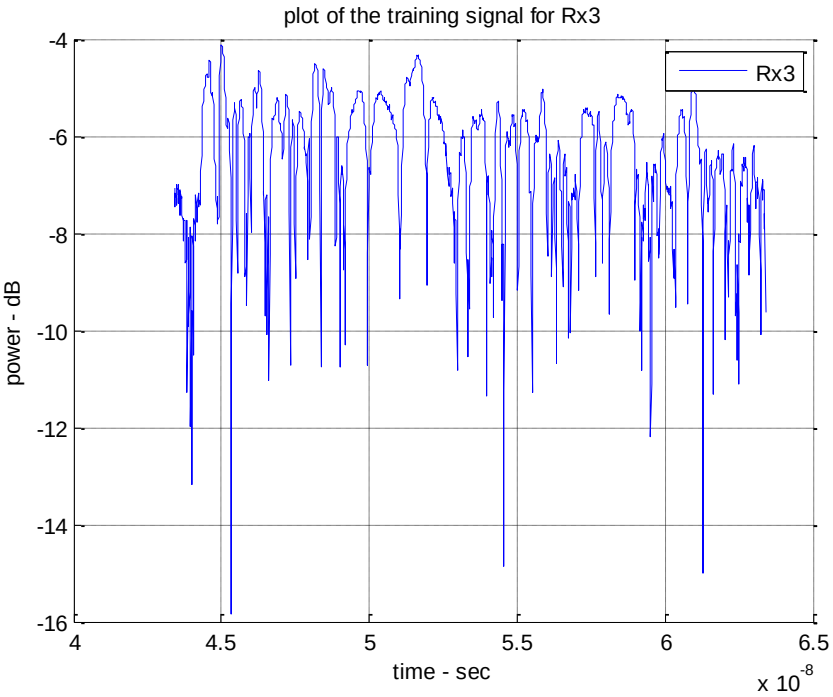


Figure 21: Training signal for Rx 4.

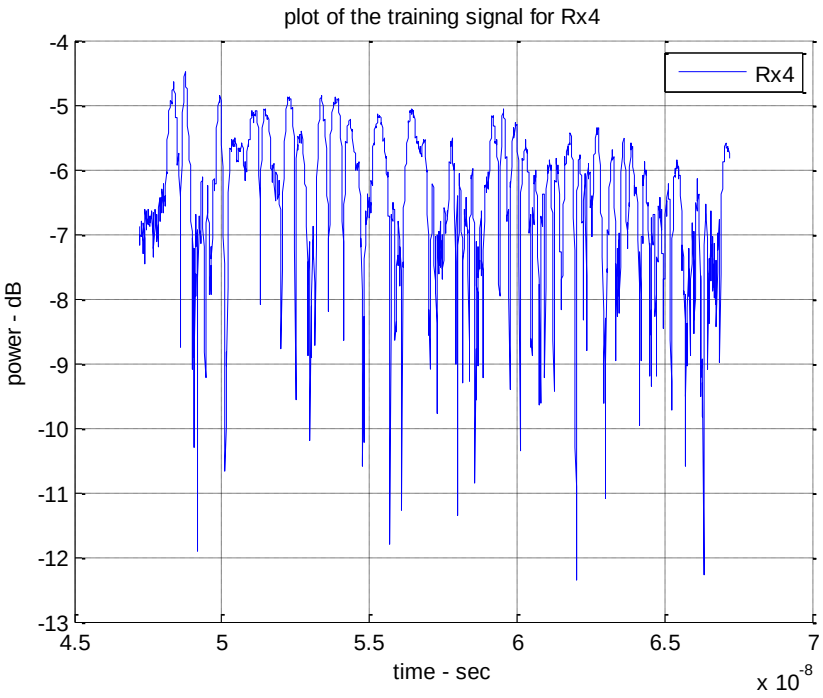


Figure 22: Training signal for Rx 5.

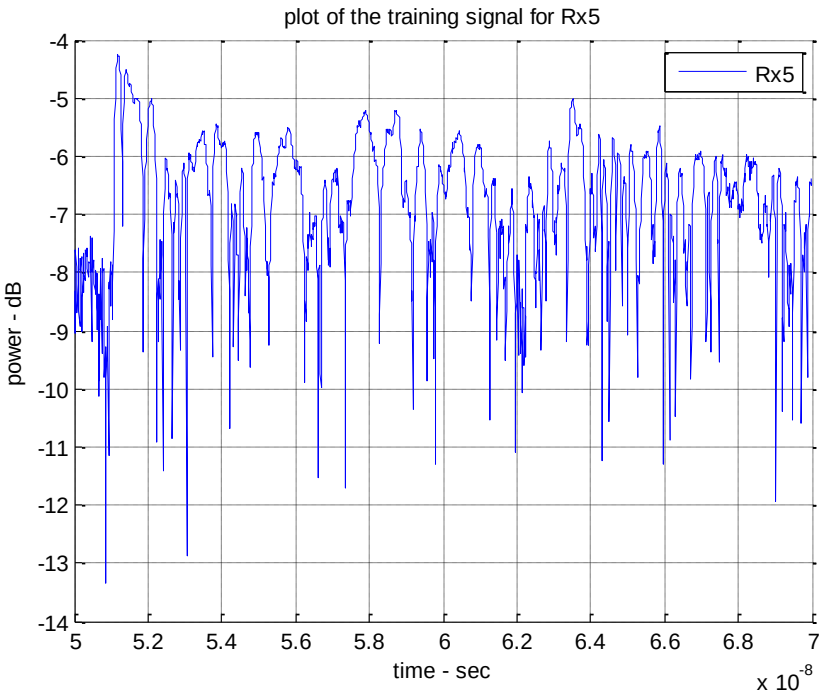


Figure 23: Training signal for Rx 6.

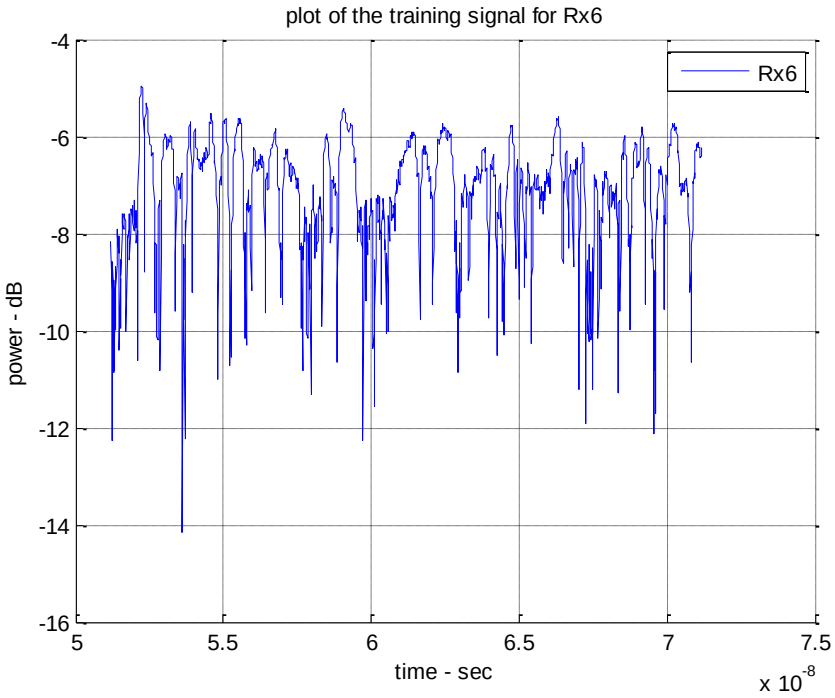


Figure 24: Training signal for Rx 7.

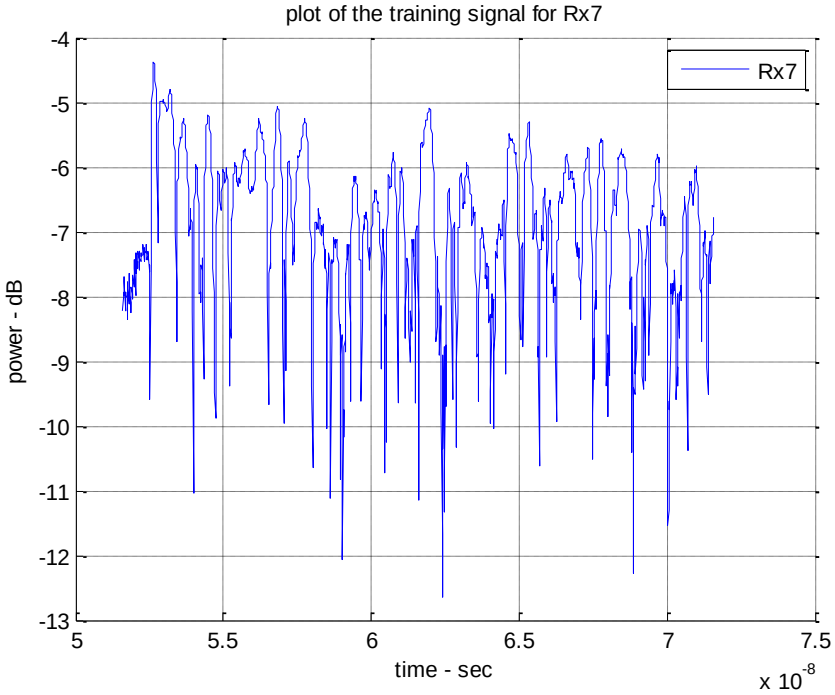


Figure 25: Training signal for Rx 8.

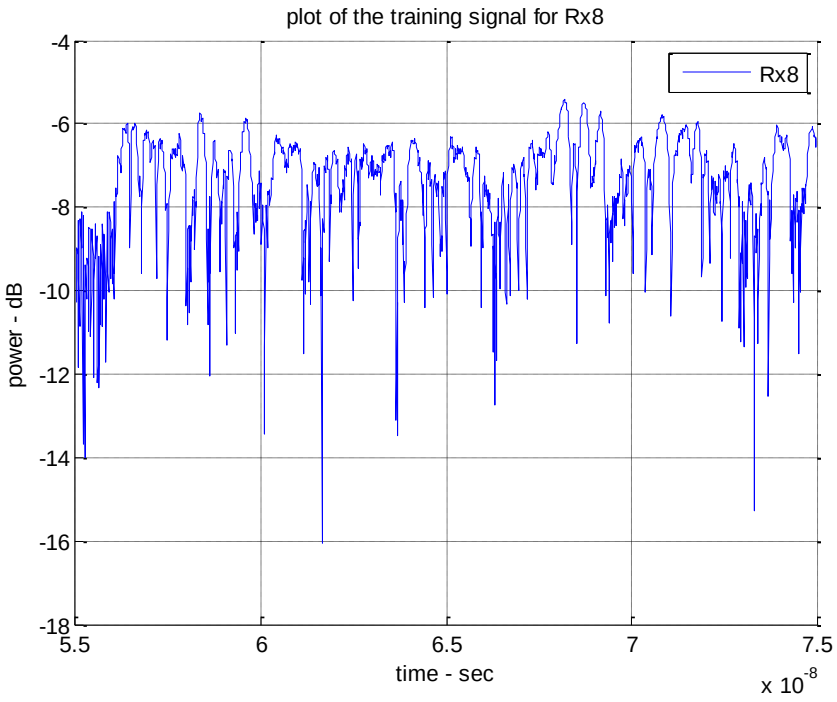


Figure 26: Training signal for Rx 9.

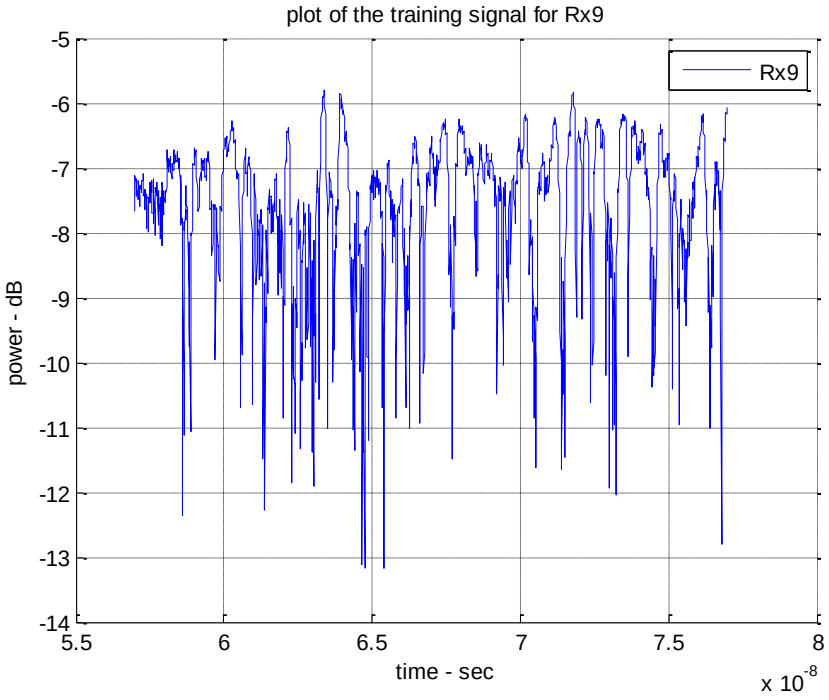


Figure 27: Training signal for Rx 10.

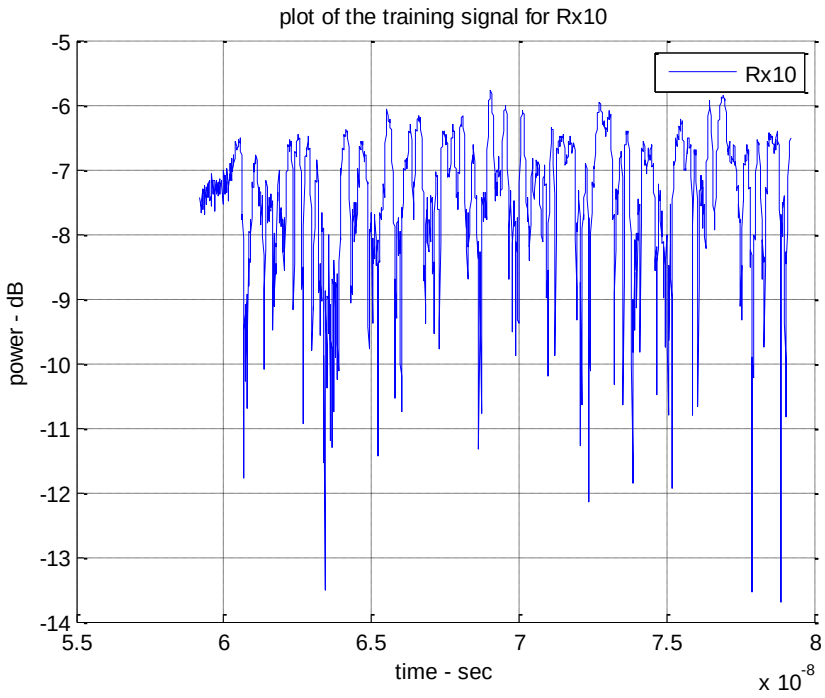


Figure 28: Training signal for Rx 11.

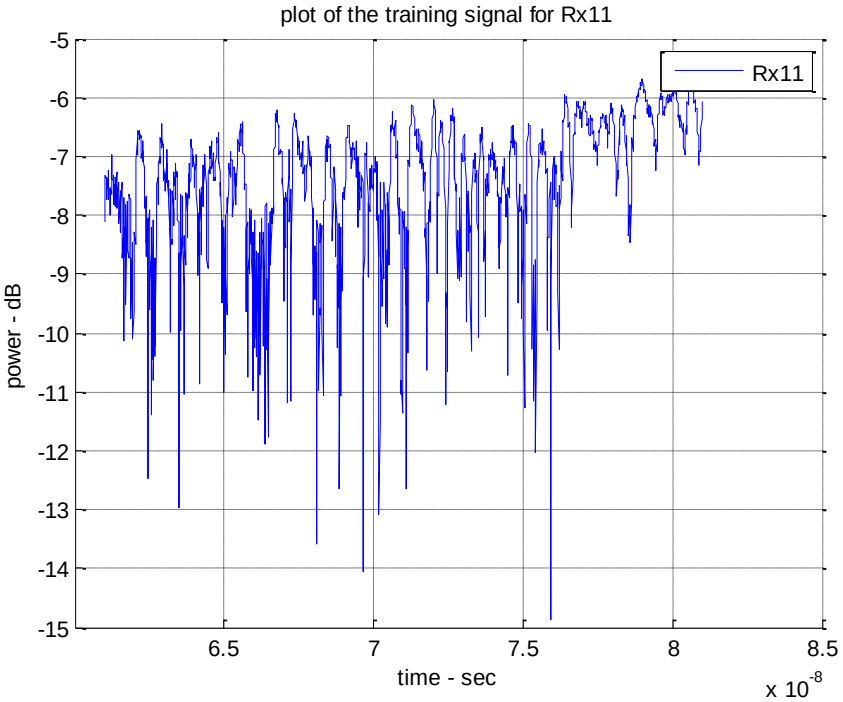
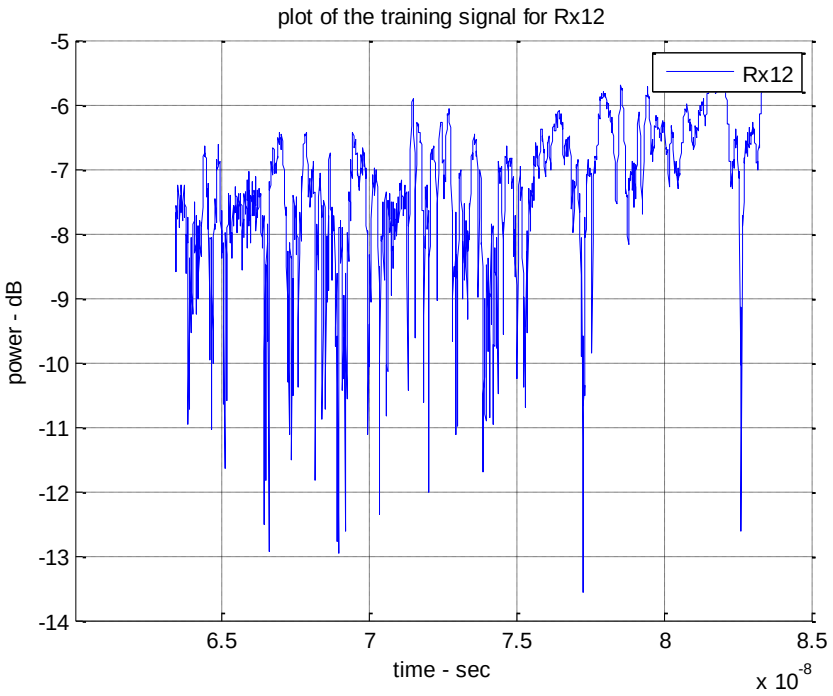


Figure 29: Training signal for Rx 12.



4.4.1 HMM-Based Location Training

In wireless communication systems, error usually occurs in bursts. Some error can be corrected by error-correction code such as Hamming Code. However, not all errors are equally likely, affecting the design of error-correcting codes used in practice. Performance degradations that cause errors may occur due to various reasons. For example, solar flare may cause static drowning out the transmission from a satellite for a short period of time (20) or a scratch on a CD may prevent the CD player from reading part of the disk. Such burst errors will wipe out a series of bits adjacent to each other. Practical error correcting codes are usually designed to correct several random (scattered) errors but much longer burst errors.

Since errors are a reality of wireless communication systems, it is more appropriate to factor them into any analysis. Burst errors can be analyzed to gain insight regarding the nature of the system. HMM profiling of receivers follows this idea through by analyzing the error patterns. Here, it is assumed that physical deformities and limitations are not present. Therefore, the errors in the signal pertain to the channel condition only.

Since the amplitude and accuracy of signals is a function of, among other things, the distance from the transmitter, some degradation in the system is expected with increased distances. The

accuracy of the profiling improves with the size of the code book. Increased number of states and knowledge about transmission probabilities between intermediate states enables a compact analysis of the system. However, the Baum-Welch algorithm (BWA) being highly computation-intensive, the increase of code book size would render the system to real time or near-real time use. Also, burst errors can be very easily identified by setting heuristic thresholds.

In this simulation, binary HMMs were used to optimize the difficulties. This choice allowed the use of fast HMM-based profiling using nominal resources. An alternative to using the BWA is the Viterbi algorithm, which is popular because of its ability to estimate most likely paths. Viterbi estimation is used in the second application considered in this thesis. The position location application is simulated using the BWA only.

Once the errors in the training signal are identified, HMM profiles were created for all 12 receivers. These training profiles were later used for all analyses.

4.4.2 Validation of Simulation Results

Successful identification of receivers was the primary focus of the simulation. The simulation started with testing methodology under the strictest restrictions. Consequently, the requirements were relaxed to understand the limitations of the system. The three major scenarios for this simulation are as below.

i. When training and test signals are very closely matched.
ii. When the signal quality fluctuates.
iii. When the receiver locations vary.

The first phase of position location pertained to the ability to identify the receivers in the best-case scenario when where training and test HMM profiles were a near-identical match. This check would help identify the usability and consistency of the data.

The 12 training profiles were tested with 100% accuracy, as expected.

The significance of this simulation was to verify that the data being used is suitable for differentiation between receiver profiles.

The next step was to consider variations that incorporated different situations. Among there were the consideration of static receivers with fluctuating signals and moving receivers with fluctuating signals.

The first setting was achieved by sampling every other sample of the available data to prepare the training and testing values. The HMM-based profiles created for all of these were of the form $\lambda_k = (\boldsymbol{A}, \boldsymbol{B}, \pi)$. The Euclidean distance-based was classification was used to identify receivers. The result showed 100% accuracy each time.

The next setting was implemented by selecting three receivers – namely Rx 1, 4, and 9 – and attempting to identify them from all 12 receivers. As can be seen in Figure 15, some of the receivers are very close to the transmitter and, therefore, enjoy LoS-like signal strength. Consistent with this observation, Rx 1 and 4 were accurately identified 100% of times. However, Rx 9 is placed at a much distant location compared to the transmitter. Consequently, the observation was not as accurate for this receiver. Under similar conditions, this receiver was accurately identified 70% of times. This phenomenon is consistent with the expected performance degradation.

Finally, a distance-based grid was prepared. This was used to simulate the effect of roaming receivers. Receivers 1, 2, 3, 4 were in the first group. The second group consisted of receivers 5, 6, 7, 8, and 9. Finally, receivers 10, 11, and 12 comprised the third group. Receivers 2, 6, and 12 were used to train for these receiver grids.

The error signals for the three training receivers were as below.

Figure 30: Error in training signal for Rx2.

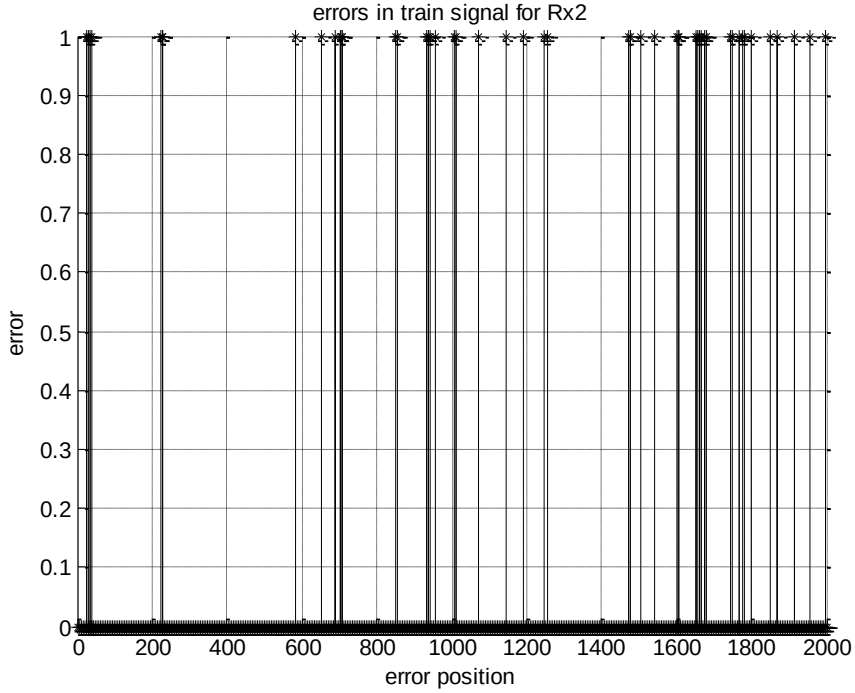


Figure 31: Error in training signal for Rx6.

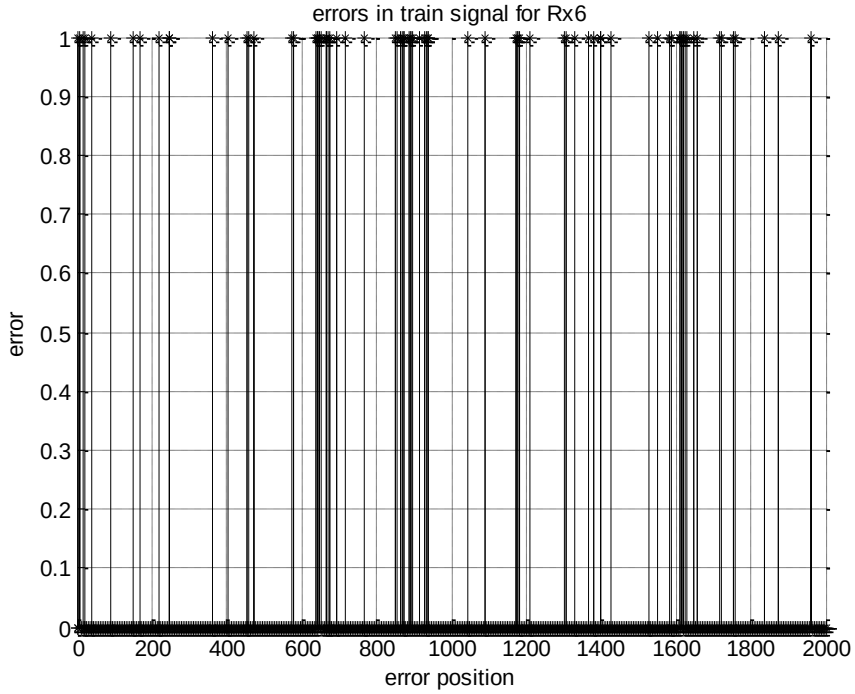
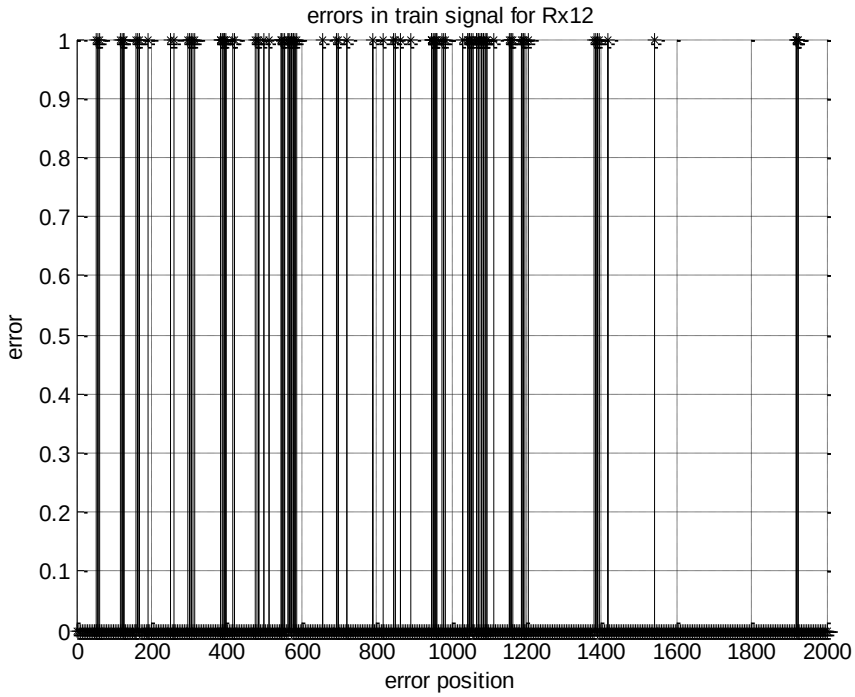
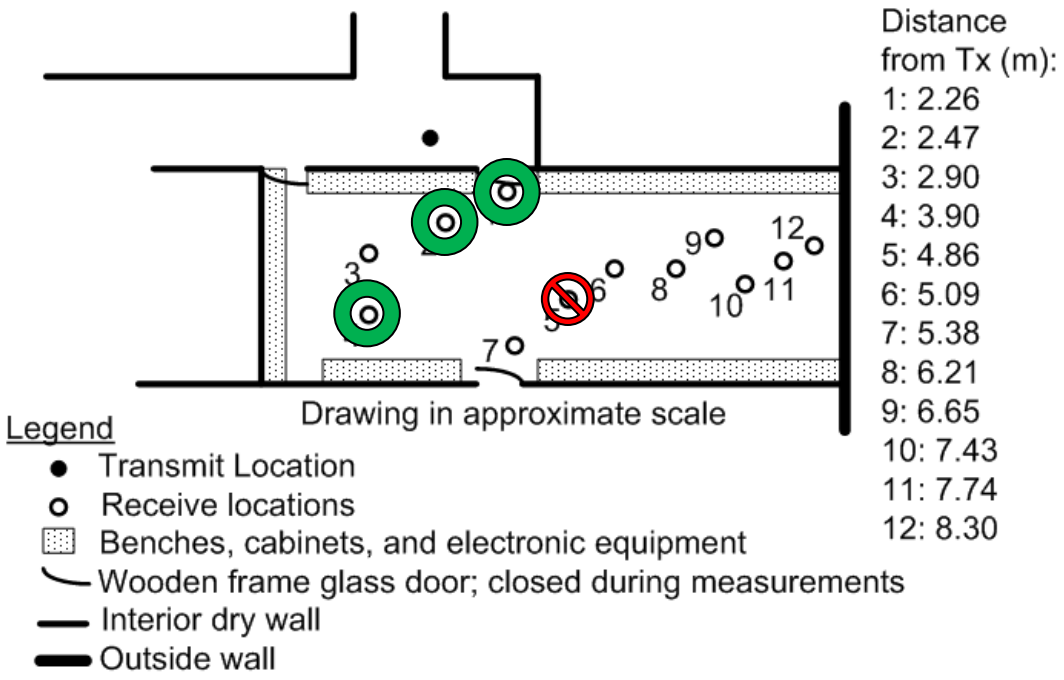


Figure 32: Error in training signal for Rx12.



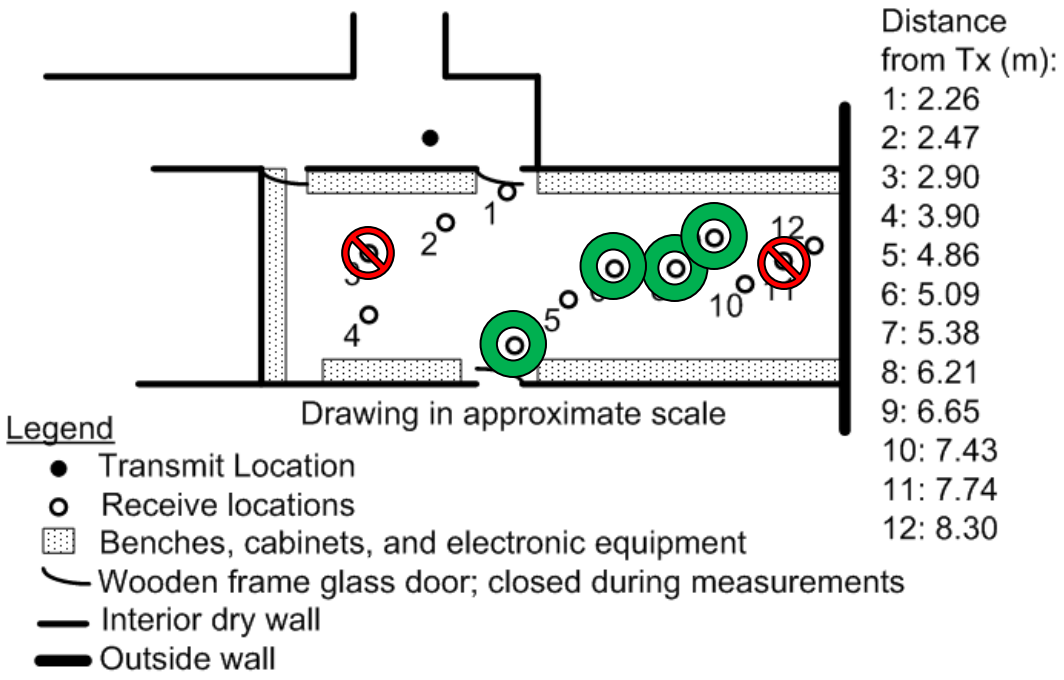
Receivers that were identified as members of the first group were Rx 1, 2, 4, 5. The actual members of this group were Rx 1, 2, 3, 4. The results are shown in the figure below.

Figure 33: Position location result of grid-based analysis for grid 1. The grid consisted of Rx 1, 2, 3, 4.



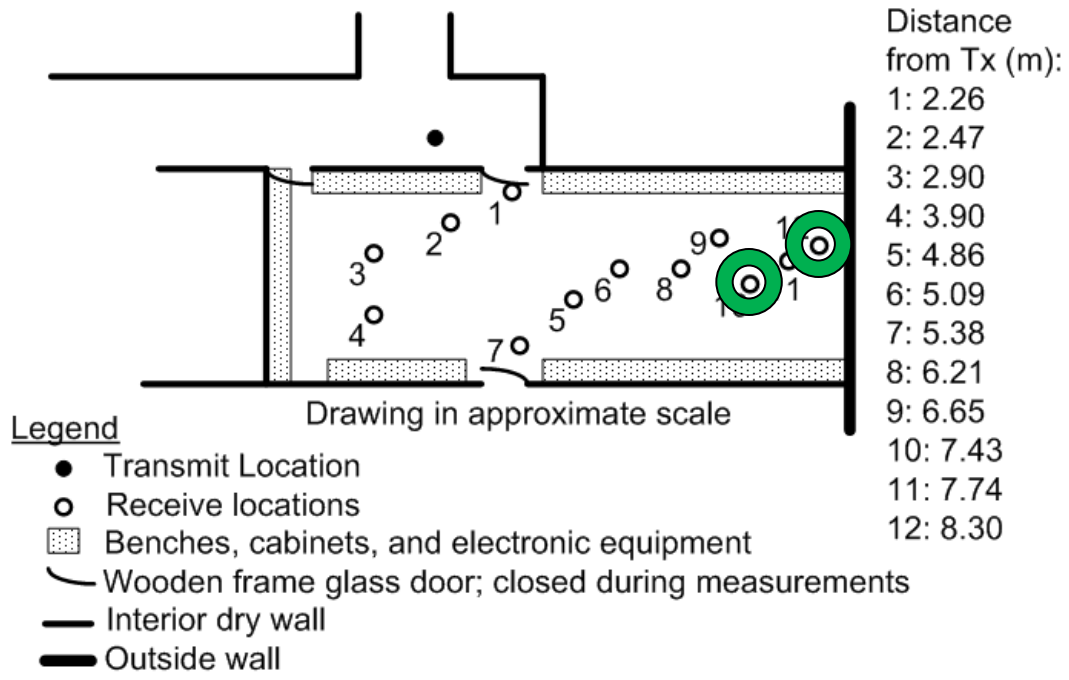
Receivers that were identified as members of the second group were Rx 3, 6, 7, 8, 9, 10. The actual members of this group were Rx 5, 6, 7, 8, 9, 11.

Figure 34: Position location result of grid-based analysis for grid 2. The grid consisted of Rx 5, 6, 7, 8, 9.



Receivers that were identified as members of the third group were Rx 10, 12. The actual members of this group were Rx 10, 11, 12.

Figure 35: Position location result of grid-based analysis for grid 3. The grid consisted of Rx 10, 11, 12.



As can be seen, some false alarms as well as false detections were observed during the simulations. However, only one erratic observation was present in each group. These results, obtained for binary HMMs and for very closely placed receivers, reinforce the methodology of using HMMs for position location. Repeated simulations provided the same outcome for the given set of data.

As the distance between the transmitter and the receiver increases, the effect of noise becomes more prominent. Also, burst errors become more frequent during this time. Use of higher order HMMs as well as a larger code book to incorporate the significance of the severity of the burst errors can improve the performance of this methodology.

The use of stochastic methods for grid-based identification, although at a price, can provide for fast and improved service.

CHAPTER 5: NETWORK SECURITY SITUATION AWARENESS USING HIDDEN MARKOV MODELS

5.1 INTRODUCTION

These days, a network is only as useful as its security level. Given the enormity of the modern wired and wireless networks, it is essential that they provide robust security. The parameters that determine the security of a system usually include counter-measures to already-known threats. While this is a very important method of providing post-attack care, it is vital to know how much harm a malicious agent can cause, how long it would take for the network to be compromise, and how long it might take to remove the threat. Such knowledge regarding the attack scenario and its possible effects can help the network administrator greatly. As a result, the need to be able to predict anomalies in a system is on the rise.

Prediction of the security threat level at any point in time requires extensive knowledge regarding the behavior of the network. Implementation of such features requires a blend of cognition, prediction, and intuitive learning on the part of the network. Cognition would help the network to identify that something is not right about the way things are behaving, prediction would help the network to get an estimate of how long it might be in the attacked mode, and intuitive learning would enable the network to better itself by evolving with time.

The current search-and-destroy methods do not offer such advanced intelligence. As a result, the network, in many cases, is already compromised before it is rescued. It should, therefore, be significant to be able to employ a sense of security (or the lack of it) in the network. This feature is referred to as network security situation awareness (NSSA).

This work evaluates previous research on NSSA and attempts to further the results.

5.2 RELATED WORK

The concept of cyberspace situational awareness was first proposed in 1999. In (21), Tim Bass proposed some of the fundamental functional frameworks required for subsequent development in

NSSA. It has been referred to as one of the two key technologies to assure information domination (22).

In (22), a novel approach of using Hidden Markov Models (HMM) to provide situational awareness in the network was suggested. HMMs are a dually stochastic method that is very similar to artificial neural networks (3). HMMs have the property of training according to changing situations. This paper extends the idea presented in (22) by applying the Viterbi algorithm for parameter estimation.

The significance of the NSSA approach is that it attempts to go beyond traditional qualitative explanations, and quantify the threats. It models both the behavior and impact of the attacker on the network. This helps to prepare metrics that include conceptual elements of security such as confidentiality, availability, reliability, etc. The concept of situation awareness (SA) originated from research on human factors in aviation (22). It was later suggested that SA can be treated as a state.

Two models are frequently used in network security evaluation – behavior modeling and impact modeling. Behavior modeling focuses on events that have already taken place, or are in the process of occurring. It is usually based on system connections, integration technique, and system state analysis. System connection analysis extracts and analyzes key information regarding connections, and informs the network administrator about the security status through visualization (23), (24). Integration analysis technique attempts to offer in-depth care by integrating various existing techniques. This was demonstrated in (25), (26). Lastly, system state analysis evaluates the system at its initial, intermediate, and final expected states.

This method is run recursively to detect intrusion (27), (28). The methods mentioned above focus on a blend of behavior modeling and statistical analysis. They evaluate only the anomaly behaviors of the system. However, the interactions between the anomalies are not included in the analysis. This leaves the necessity to analyze the anomalies deeper. Impact modeling offers this opportunity.

NSAA is a global concept that helps to grow a high-level knowledge about the network's security situation. It takes into account both the anomaly behaviors and their impact. Stochastic modeling methods based on hidden Markov models have been used in (22) to implement NSAA for various network services. In (22), NSSA was used for various network services such as dns, www, ftp, nfs, and mail. The Baum-Welch algorithm was used to estimate the parameters of the HMMs.

5.3 CONTRIBUTIONS TO PREVIOUS WORK

This paper attempts to extend the previous work in (22) by implementing the idea of using HMMs for threat detection in the OSI model. For the ease of operation, five of the seven layers have been discussed. These are the physical, link, transport, network, and application layers. These five layers have been chosen because they are most vulnerable to network attacks. This work is unique in the sense that it evaluates the inter-relation between the selected layers of the OSI model while attempting to implement NSSA.

Also, the parameter estimation algorithm used in this paper is different from the one used in (22). The Viterbi algorithm is used in this paper to take advantage of its probabilistic estimation model. It is also a very fast algorithm that estimates the maximum likelihood path of states (29).

Estimating state transitions within an HMM is similar to decoding the transition in paths in a communication network. Therefore, the characteristic of the problem in this paper is such that the Viterbi algorithm is a good fit to solve it.

Finally, a simulation-based approach has been adopted so that all possible attack scenarios can be modeled using this method. The parameters of the HMM can be used to represent various scenarios. The simulation-based approach can be used as a technique to follow in future applications of probabilistic modeling and estimation.

5.4 SIMULATION METHODOLOGY

This thesis employs a computer simulation-based approach to demonstrate NSSA. The simulation parameters are selected to represent an actual computer network. These parameters were heuristically selected after careful study of the relative security of different layers (30).

First, a sequence of Markov observations was created. The observation sequence was prepared with a known set of transition and emission matrices. This was done with a view to effectively verify the process. The model was prepared for the physical, link, network, transport, and application layers in the OSI model. For each layer, the status of the network was defined by four states. These states were defined as hacked, compromised, attacked, and normal. The definition is similar to that used in (22).

The transition matrix, A, was heuristically defined as below to represent an attack scenario. This is the model of an attack pattern which causes most damage to the physical and link layers, leaving the other layers relatively unharmed. The rows of this matrix denote the five layers, and the columns denote the probability of the layer being hacked, compromised, attacked, or normal.

$$A = \begin{bmatrix} 0.6 & 0.2 & 0.1 & 0.1 & 0.0 \\ 0.5 & 0.3 & 0.1 & 0.1 & 0.0 \\ 0.5 & 0.1 & 0.2 & 0.1 & 0.1 \\ 0.4 & 0.2 & 0.2 & 0.1 & 0.1 \\ 0.3 & 0.2 & 0.2 & 0.2 & 0.1 \end{bmatrix} \quad (4.1)$$

In this model, if the physical layer (row 1) is attacked, the probability that it will be hacked is very high. Besides the physical layer, the attacker will also harm the link and network layers (rows 2 and 3). It can be seen that, although the link and network layers have equal probability of being hacked, the probability of the link layer being compromised is higher than the network layer.

The emission matrix, B, was defined as below. The basis of this choice was the comparative strengths of different layers in the OSI model discussed in (30). In reality, this matrix corresponds to the dynamic between the various aspects of a network. This matrix can be produced by an a priori analysis of the system under hostile conditions.

$$B = \begin{bmatrix} 0.7 & 0.2 & 0.1 & 0.0 \\ 0.6 & 0.3 & 0.1 & 0.0 \\ 0.5 & 0.3 & 0.1 & 0.1 \\ 0.4 & 0.3 & 0.2 & 0.1 \\ 0.3 & 0.5 & 0.2 & 0.0 \end{bmatrix} \quad (4.2)$$

Using these values, a Markov model was generated. The observable sequence is shown in Figure 36. In an HMM, the state transitions are hidden. Since this work is simulation-based, a known set of state sequence was generated to aid the verification process. The corresponding state sequence is shown in Figure 37.

Figure 36: Actual observation sequence simulated for the experiment. This figure shows the internal state observations.

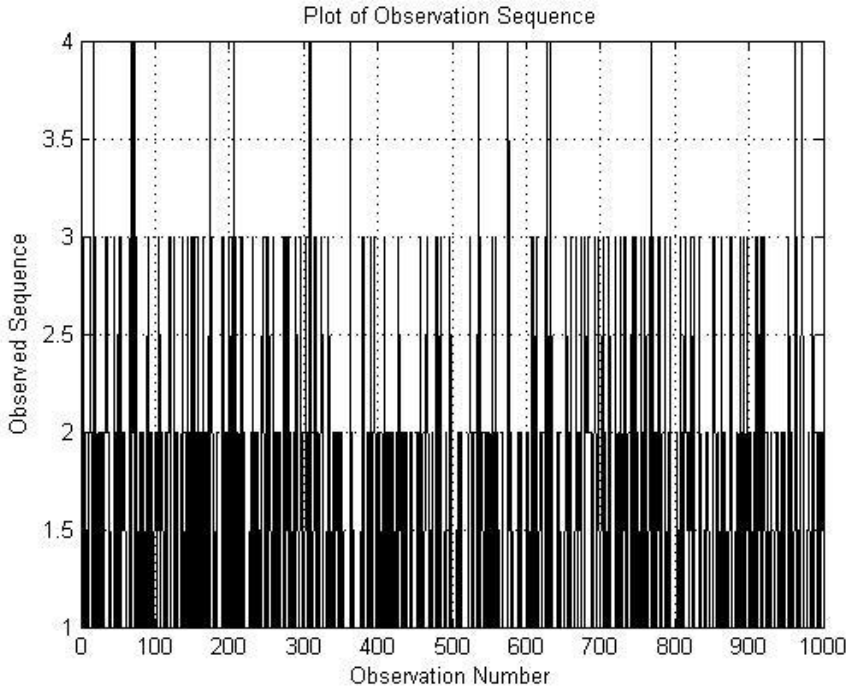
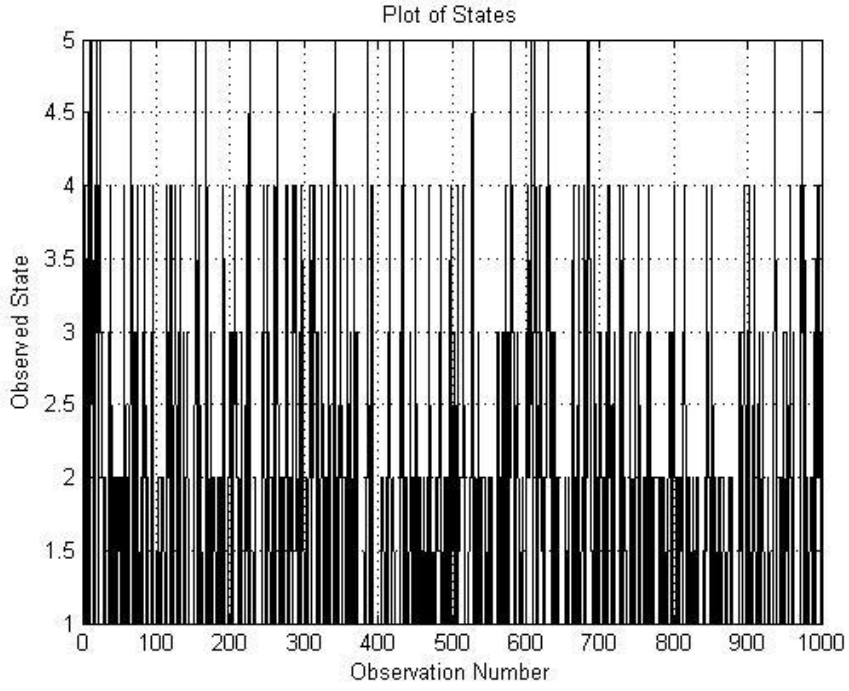


Figure 37: Actual state transitions that take place in the system. These values define the system, but are unobservable.



The objective of the simulation was to use the observation sequence to obtain an estimate of the A matrix. Consistent with the usual procedures of Markov modeling, a training model is used first. This was later used to contrast the situations and create a sense of security.

The initial estimate of the A matrix, $\hat{A}$, was uniformly defined as below. Upon HMM estimation, this matrix will be modified to reflect the current state of the system. Since this is a statistics-based model which shows transition probabilities to every other state for each layer, this can provide a wealth of information for the network administrator.

$$\hat{A} = \begin{bmatrix} 0.2 & 0.2 & 0.2 & 0.2 & 0.2 \\ 0.2 & 0.2 & 0.2 & 0.2 & 0.2 \\ 0.2 & 0.2 & 0.2 & 0.2 & 0.2 \\ 0.2 & 0.2 & 0.2 & 0.2 & 0.2 \\ 0.2 & 0.2 & 0.2 & 0.2 & 0.2 \end{bmatrix} \tag{4.3}$$

The uniform definition of $\hat{A}$ should result in an estimated value of $\hat{A}$ that shows trends similar to those shown in the original A, i.e., the estimated matrix, $\hat{A}$, should be able to determine that physical and link layers are most vulnerable to this attack scenario. To get improved results, the simulation can be run iteratively, using previous results.

The results obtained with the simulation parameters are validated in the following section.

5.5 VALIDATION OF SIMULATION RESULTS

The state sequence was estimated using the observation sequence. The Viterbi algorithm was used to perform this estimation. The resulting state sequence, shown in Figure 36, was compared with the actual state sequence to calculate the effectiveness of the system.

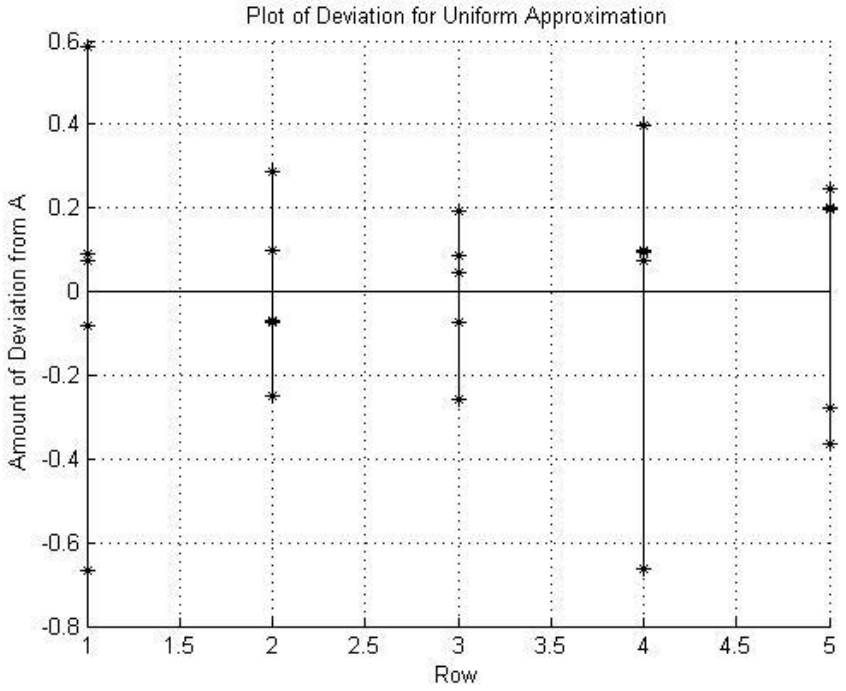
Markov models can be compared using various methods. Among these, the simplest method is to compare the Euclidian distance between two models $\lambda_1(\boldsymbol{A}, \boldsymbol{B}, \pi)$ and $\lambda_2(\boldsymbol{A}, \boldsymbol{B}, \pi)$. For better analysis, it is often useful to check the divergence between the matrices using Kullback-Leibler Divergence (KLD).

For the uniform $\hat{A}$ matrix defined above, the following results were obtained.

$$\hat{A} = \begin{bmatrix} 0.5370 & 0.0749 & 0.0190 & 0.2087 & 0.1604 \\ 0.0216 & 0.4693 & 0.1382 & 0.0002 & 0.3705 \\ 0.5323 & 0.0515 & 0.3625 & 0.0039 & 0.0498 \\ 0.0428 & 0.7582 & 0.0000 & 0.1987 & 0.0003 \\ 0.0147 & 0.0878 & 0.4077 & 0.4897 & 0.0001 \end{bmatrix} \quad (4.4)$$

The amount of deviation observed in this estimation is shown in Figure 38.

Figure 38: Deviation of simulated transition matrix from uniform estimation. This figure illustrates the deviation between the actual elements of the transition matrix and the estimated transition matrix.



As seen in Figure 38, the results showed some deviation from the actual transition matrix. Therefore, the following modified version of the initial estimate, $\hat{A}$, was applied in order to improve simulation performance. With the adjustment, the simulation results gained accuracy and were showing signs of growing an idea about the network's security status.

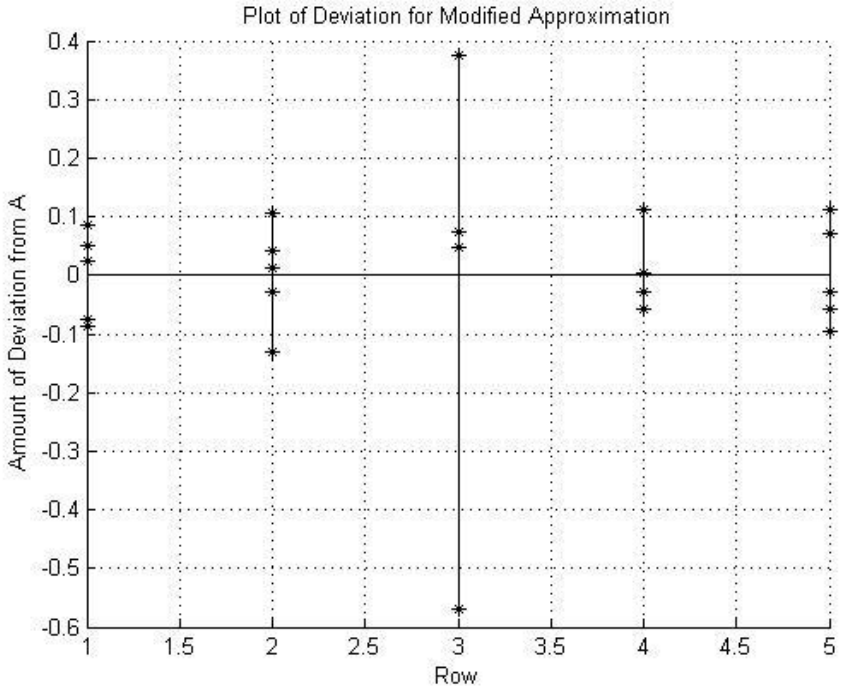
$$\hat{A} = \begin{bmatrix} 0.4 & 0.2 & 0.2 & 0.1 & 0.1 \\ 0.4 & 0.2 & 0.2 & 0.1 & 0.1 \\ 0.4 & 0.2 & 0.2 & 0.1 & 0.1 \\ 0.4 & 0.2 & 0.2 & 0.1 & 0.1 \\ 0.4 & 0.2 & 0.2 & 0.1 & 0.1 \end{bmatrix} \quad (4.5)$$

It was observed that the values changed in the proper direction. The results improved when the simulation was run with modified values of $\hat{A}$. The following shows a sample set of simulation results.

$$\hat{A} = \begin{bmatrix} 0.6073 & 0.1719 & 0.0490 & 0.0859 & 0.0859 \\ 0.3214 & 0.1792 & 0.3203 & 0.0896 & 0.0896 \\ 0.2168 & 0.2677 & 0.2477 & 0.1339 & 0.1339 \\ 0.3214 & 0.1792 & 0.3203 & 0.0896 & 0.0896 \\ 0.3214 & 0.1792 & 0.3203 & 0.0896 & 0.0896 \end{bmatrix} \tag{4.6}$$

The amount of deviation observed for the modified initial value is shown in Figure 39. As can be seen from the figure, with the exception of the network layer estimation, the values were very close to the initially define transition matrix, A. When the modified $\hat{A}$ was used, the accuracy of the parameter estimation increased by 27.25%. The accuracy improved with further iteration of the simulation.

Figure 39: Deviation of simulated transition matrix from modified estimation. This figure illustrates the improvement in performance as the simulation is run iteratively.



From the above simulation results, it can be inferred that the simulation methodology is valid and that the result can be improved by iteratively running the simulation. The results show that, the initially estimated $\hat{A}$ was successfully modified to reflect the trend of the attack. It successfully modeled that the physical layer was most vulnerable to being hacked. The vulnerability of the link and network layers was also reflected in the results. The probability of these layers being hacked or compromised was much higher than the rest.

Once a network is under attack, the simulation can quickly obtain a sense about the network's security situation. This is reflected in the estimated transition matrix. As seen above, the simulation can successfully offer an accurate understanding of the nature of the threat. By this, the network shows cognition. Over time, the network administrator can fathom the gravity of the attack varying the simulation parameters, and develop a security metric based on the effective predictions provided by the system. Finally, the system can intuitively learn and adapt to the threats through iterative simulations.

CHAPTER 6: CONCLUSION

6.1 SUMMARY

This thesis presented statistical methods to get insights into wireless communication systems. The first part of the thesis analyzed Hidden Markov Models (HMM) and illustrated various inferences into HMMs. Parameter estimation and classification of HMMs was explained in this part.

The second part of the thesis looked into stochastic games and HMMs. Stochastic games are a branch of game theory that, like the original theory, discusses the methods of rational decision-making. Stochastic games are Markovian processes that have reward or cost associated with each decision.

The third part of the thesis looked into the use of HMMs for position location. Position location is an important aspect of wireless communication networks. This chapter applied a previously developed algorithm in a completely new environment and successfully validated the previous work.

The fourth part of the thesis discussed Network security situation awareness (NSSA), an emerging field in computer network security. Researchers are working to include situation awareness in the system. HMMs' dual stochastic nature and similarity to neural networks make them a good method of modeling security sensing. This section of the thesis adopted a simulation-based approach to NSSA and performed general impact-analysis of network threats using the Viterbi algorithm to estimate Markov parameters.

6.2 FUTURE WORKS

The work presented in this thesis lays the foundation for a cognitive engine with improved position location. HMMs are useful for discrete channel modeling. They also provide the opportunity for dynamic adjustment with the environment. These are very important attributes required in a cognitive engine. The improvements offered by HMMs come at the cost of increased processing. For powerful engines, high-order HMMs can be used with great accuracy. The results in this thesis can be used in practical situations where binary HMMs can be applied to test the improvement in performance.

In this simulation, a code book size of 2, i.e., binary BWA, was used. It was observed that the accuracy of the system suffers for binary HMMs. Therefore, the increase of the size of the code book

can be used to improve accuracy. In such a case, a forward-only variation of the EM algorithm can be considered to reduce computational complexity.

Network security situation awareness applications can be readily used in an actual laboratory setting with simulated network intrusions. The experiment should produce results similar to the simulation. Such results would justify the possible use of HMMs for network security sensing.

CHAPTER 7: REFERENCES

1. **Mohammad, Maruf.** *Cellular Diagnostic Systems Using Hidden Markov Models.* s.l. : Virginia Tech, 2006.
2. **Cappe, Oliver, Moulines, Eric and Ryden, Tobias.** *Inference in Hidden Markov Models.* s.l. : Springer Series in Statistics, 2005.
3. **Akbar, Ihsan Ali.** *Statistical Analysis of WIreless Systems using Markov.* s.l. : Virginia Tech, 2007.
4. **Commons, WikiMedia.** [Online]
5. **Leon-Garcia, Alberto.** *Probability and Random Processes for Electrical Engineering.* 2007. p. 475.
6. **Tranter, William, et al.** *Principles of Communication Systems Simulation with Wireless Applications.* s.l. : Pearson Education, 2007.
7. *Parallel Hidden Markov Models for American Sign Language Recognition.* **Vogler, Christian and Metaxas, Dimitris.** Greece : Proceedings of the International Conference on Computer Vision., 1999.
8. **Churcill, G.** *Hidden Markov Chains and the Analysis of Genome Structure.* 1992. pp. 107-115.
9. *A Hidden Markov Model that Finds Genes in E. Coli.* **Krogh, A., Mian, I. S. and Haussler, D.** 1994.
10. *Sigma A Recognition Sites in the Bacilius Subtilis Genome.* **Jarner, H., et al.** 2001.
11. *Maximum Likelihood from Incomplete Data via the EM Algorithm.* **Dempster, A. P., Laird, N. M. and Rubin, D. B.** s.l. : Journal of the Royal Statistical Society., 1977.
12. **Turin, W.** *Digital Transmission Systems: Performance Analysis and Modeling.* s.l. : McGraw-Hill, 1999.
13. *A Maximization Technique Occuring in the Statistical Analysis of Probabilistic Functions of Markov Chains.* **Baum, Leonard E., et al.** 1, s.l. : The Annals of Mathematical Statistics., 1970, Vol. 41.
14. *A Probabilistic Distance Measure for Hidden Markov Models.* **Juang, B. H. and Rabiner, L.** s.l. : AT&T Technical Journal, 1985, Vols. 64, no.2.
15. **Ferguson, Thomas S.** Department of Mathematics, UCLA. [Online] http://www.math.ucla.edu/~tom/Game_Theory/Contents.html.
16. **Filar, Jerzy and Vrieze, Koos.** *Competitive Markov Decision Processes.* s.l. : Springer Publishers, 1997. 0-387-94805-8.
17. *A Non-Line-of-Sight Mitigation Technique Based on ML-Detection.* **Riba, Jaume and Urruela, Andreu.** s.l. : International Conference on Acoustics, Speech, and Signal Processing., 2004.
18. **Proakis, John G.** *Digital Communications.* 4th. s.l. : McGraw-Hill Higher Educations, 2000.
19. *Design and Implementation of an Ultrabroadband Millimeter-Wavelength Vector Sliding Correlator Channel Sounder and In-Building Multipath Measurements at 2.5 & 60 GHz.* **Anderson, Chris R.** 2002.
20. **University, Kansas State.** Department of Mathematics. [Online] Kansas State University.
21. *Intrusion Detection System and Multi-Sensor Data Fusion.* **Bass, Tim.** s.l. : Communications of the ACM, 2000, pp. 99-105.
22. *A Kind of Formal Modelling for Network Security Situational Awareness Based on HMM.* **Liang, Ying, Wang, Huiqiang and Pang, Yonggang.** s.l. : The Ninth International Conference on Web-Age Information Management, 2008. WAIM '08., 2008. pp. 598-605.
23. *Home-Centric Visualization of Network Traffic for Security Administration.* **Ball, Robert, Fink, Glenn A. and North, Chris.** Washington D.C. : Conference on Computer and Communications Security., 2004. Security Proceedings of the 2004 ACM Workshop on Visualization and Data Mining for Computer Security. pp. 55-64.
24. *Case study: Interactive visualization for Internet security.* **Teoh, Soon Tee, et al.** s.l. : IEEE Proceedings on Visualization 2002., 2002. pp. 505-508.
25. *A Technique Independent Fusion Model for Network Intrusion.* **Schifflet, Jason.** s.l. : Proceedings of the Midstates Conference on Undergraduate Research in Computer Science and Mathematics., 2004. Vol. 3, pp. 13-19.

26. **Batsell, Stephen G., Rao, Nageswara S. and Shankar, Mallikarjun.** Distributed Intrusion Detection and Attack Containment for Organizational Cyber Security. [Online] 2005. http://www.ioc.ornl.gov/projects/documents/containment.pdf.
27. *NVisionIP: NetFlow Visualizations of System State for Security Situational Awareness.* **Lakkaraju, Kiran, Yurcik, William and Lee, Adam J.** New York : Proceedings of the 2004 ACM Workshop on Visualization and Data Mining for Computer Security., 2004. pp. 65-72.
28. *State Transition Analysis: A Rule-Based Intrusion Detection Approach.* **Ilgun, Koral, Kemmerer, Richard A. and Porras, Phillip A.** 1995.
29. **Engineering, Viterbi School of.** About Viterbi Algorithm. *Viterbi School of Engineering.* [Online] http://viterbi.usc.edu/about/viterbi/.
30. **Rouf, Ishtiaq and Harun, Mahmud.** *A Survey of Authentication & Security Issues in Wireless Networks.* s.l. : Unpublished, 2008.
31. **Philosophy, Stanford Encyclopedia of.** Stanford University. [Online] http://plato.stanford.edu/entries/game-theory/.