Statistical Analysis of Wireless Systems Using Markov Models

Ihsan A. Akbar

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

Doctor of Philosophy
in
Electrical Engineering

William H. Tranter, Chair
William T. Baumann
Christopher Beattie
R. Michael Buehrer
Jeffrey H. Reed

January 22, 2007
Blacksburg, Virginia

Keywords: Markov chains, hidden Markov models, fading channels, error source modeling, cognitive radio, cyclostationarity, dynamic spectrum allocation.

Copyright © 2007, Ihsan A. Akbar

Statistical Analysis of Wireless Systems Using Markov Models

Abstract

Ihsan A. Akbar

Being one of the fastest growing fields of engineering, wireless has gained the attention of researchers and commercial businesses all over the world. Extensive research is underway to improve the performance of existing systems and to introduce cutting edge wireless technologies that can make high speed wireless communications possible.

The first part of this dissertation deals with discrete channel models that are used for simulating error traces produced by wireless channels. Most of the time, wireless channels have memory and we rely on discrete time Markov models to simulate them. The primary advantage of using these models is rapid experimentation and prototyping. Efficient estimation of the parameters of a Markov model (including its number of states) is important to reproducing and/or forecasting channel statistics accurately. Although the parameter estimation of Markov processes has been studied extensively, its order estimation problem has been addressed only recently. In this report, we investigate the existing order estimation techniques for Markov chains and hidden Markov models. Performance comparison with semi-hidden Markov models is also discussed. Error source modeling in slow and fast fading conditions is also considered in great detail.

Cognitive Radio is an emerging technology in wireless communications that can improve the utilization of radio spectrum by incorporating some intelligence in its design. It can adapt with the environment and can change its particular transmission or reception

parameters to execute its tasks without interfering with the licensed users. One problem that CR network usually faces is the difficulty in detecting and classifying its low power signal that is present in the environment. Most of the time traditional energy detection techniques fail to detect these signals because of their low SNRs. In the second part of this thesis, we address this problem by using higher order statistics of incoming signals and classifying them by using the pattern recognition capabilities of HMMs combined with cased-based learning approach.

This dissertation also deals with dynamic spectrum allocation in cognitive radio using HMMs. CR networks that are capable of using frequency bands assigned to licensed users, apart from utilizing unlicensed bands such as UNII radio band or ISM band, are also called Licensed Band Cognitive Radios. In our novel work, the dynamic spectrum management or dynamic frequency allocation is performed by the help of HMM predictions. This work is based on the idea that if Markov models can accurately model spectrum usage patterns of different licensed users, then it should also correctly predict the *spectrum holes* and use these frequencies for its data transmission. Simulations have shown that HMMs prediction results are quite accurate and can help in avoiding CR interference with the primary licensed users and vice versa. At the same time, this helps in sending its data over these channels more reliably.

Acknowledgement

All praise and thanks to Allah, the One, the only Creator and the Ruler of this universe. I wish I could thank Him enough for all the blessings and favors He has bestowed upon me and granted me with unlimited bounties to enjoy. He is indeed the source of all knowledge, the all knowing.

I would like to express my deepest gratitude to my advisor Dr. William H. Tranter, without whom I would not have been able to accomplish my Ph.D. research work at Virginia Tech. He has been a source of guidance, support and encouragement for me during the entire course of my graduate studies at Tech. I also thank him for providing a friendly and supportive environment which facilitated my research.

I want to show my sincere appreciation to the members of my doctoral committee whose guidance and advice from time to time made my task more achievable. Their dedication and expertise in their respective fields will always be a source of inspiration for me.

I would also like to thank my co-workers and staff at the Mobile and Portable Radio Research Group (MPRG) of Virginia Tech. I am also grateful to all my friends here in Blacksburg who gave me encouragement and love and just about enough entertainment to rejuvenate my spirit every now and then when I needed to.

My wife, Fatima, has been along my side throughout my Ph.D. Her relentless understanding, support, and love has helped immensely in getting me where I am today and I pray to Allah that we will always live a very pleasant and joyous life. I would like

to express my deepest gratitude to her parents, Masood A. Zuberi and Shaukat A. Zuberi, for their continuous prayers, encouragement and support.

To my daughter, Raviha, who gave me moments of tranquility and refuge no matter the level of neglect she faced during my long stays at the office and my late night studies during the last two years.

I cannot find the words to express my profound feelings of gratitude for my parents, my father Inamuddin Akbar, my mother Tahira Darakhshan, and other family members who helped me shape my dreams and focus on what I really wanted in my life. I thank all of my family, my sisters, and brother for their prayers, persistent support, patience and encouragement.

Table of Contents

1. Introduction ---------- 1

2. Markov Theory ---------- 12

2.1 Markov Processes ---------- 12

2.2 Markov Chain ---------- 15

2.3 Hidden Markov Models ---------- 16

2.4 Semi-Hidden Markov Models ---------- 19

2.5 EM Algorithms for HMM and Semi-HMM Parameter Estimation ---------- 21

2.5.1 The Forward-backward BWA for HMM ---------- 21

2.5.2 Scaling of the Forward and Backward Variables ---------- 26

2.5.3 The Forward-only BWA for HMMs ---------- 27

2.5.4 The Forward-backward BWA for Semi-HMMs ---------- 29

2.5.5 The Forward-only BWA for Semi-HMMs ---------- 32

2.5.6 Phase-Type (PH) Distribution Approximation ---------- 34

2.6 Other Direct Maximization Techniques ---------- 37

3. Order Estimation of Discrete Markov Channel Models ---------- 38

3.1 Hypothesis Testing in Markov Order Estimation ---------- 39

3.2 Binary Markov Chain Order Estimation ---------- 40

3.3 Likelihood of Binary Markov Processes ---------- 43

3.4 MC and HMM Order Estimation Using AIC and BIC ---------- 47

3.5 Identifiability---- 49
3.5.1 Identifiability of a Binary Hidden Markov Process---- 50
3.6 Autocorrelation Functions of Markov Processes ---- 51
3.7 Information Theoretic Measures for Markov Processes---- 55
3.7.1 Kullback-Leibler and Divergence-Rate Distance ---- 56
3.7.2 Shannon-McMillan-Breiman Theorem ---- 56
3.8 Modeling Quantized Fading Statistics ---- 58
Case 1 - PH Approximation of Rayleigh Fading with f_d = 10Hz and f_s = 0.5Kbps---- 58
Case 2 - PH Approximation of Rayleigh Fading with f_d = 10Hz and f_s = 10Kbps---- 63
3.9 Order Estimation of Binary Markov Chain Using Goodness of Fit---- 65
Case 1- Zero-Order Markov Chain ---- 65
Case 2 - First-Order Markov Chain---- 66
Case 3 - Second-Order Markov Chain ---- 67
Case 4 - Third-Order Markov Chain---- 69
3.10 Forward-backward and Forward-only BWA Simulation---- 71
3.11 MC and HMM Order Estimation Using AIC and BIC---- 74
Case 1 - First-Order Markov Chain ---- 74
Case 2 - Second-Order Markov Chain---- 76
Case 3 - Three-State HMM---- 79
Case 4 - Five-state HMM---- 81
Case 5 - Six-State HMM---- 84
Case 6 - Seven-State HMM ---- 86
3.12 Order Estimation of Binary Hidden Markov Rayleigh Fading Channel Models --------- 88
Case 1 - Rayleigh Fading with 7Hz Doppler Frequency, 0.5Kbps Sampling Frequency and 10dB SNR.---- 89
Case 2 - BPSK Rayleigh Fading with 7Hz Doppler Frequency, 1Kbps Sampling Frequency and 10dB SNR.---- 93
Case 3 - BPSK Rayleigh Fading with 7Hz Doppler Frequency, 1Mbps Sampling Frequency and 10dB SNR. ---- 95
3.13 Neural Network and its Connection With the HMM----108
3.14 Inference from the Simulation Results----110
3.15 Conclusion ----111

4. Signal Classification and Detection Using HMMs ---- 112

4.1 Introduction ---- 113

4.2 Cyclostationary Spectral Analysis ---- 114

4.3 Cyclic Spectral Analysis of Modulated Signals ---- 120

4.3.1 SCF and Cycle Frequency Domain Profile of DSB-SC AM ---- 120

4.3.2 SCF and Cycle Frequency Domain Profile of BPSK ---- 121

4.3.3 SCF and Cycle Frequency Domain Profile of B-FSK ---- 121

4.3.4 SCF and Cycle Frequency Domain Profile of MSK ---- 122

4.3.5 SCF and Cycle Frequency Domain Profile of QPSK ---- 122

4.4 Signal Detection ---- 127

4.4.1 The Crest Factor as a Detection Threshold ---- 128

4.4.2 Threshold Test for Signal Detection ---- 130

4.5 HMM As Signal Classifier ---- 131

4.6 Simulation Result ---- 133

4.6.1 Signal Detection ---- 133

4.6.2 Signal Classification ---- 136

4.7 Conclusion ---- 139

5. Dynamic Spectrum Sharing in Cognitive Radio Using HMMs ---- 140

5.1 Theory ---- 142

5.1.1 Markov-based Channel Prediction Algorithm (MCPA) ---- 143

5.2 Simulation Results ---- 145

Case 1 - MCPA Performance when the Spectral Usage Patterns of the Licensed Users are Exponentially Distributed ---- 146

Case 2 - MCPA Performance When the Spectral Usage Patterns of the Licensed Users are Poisson Distributed ---- 151

5.3 Conclusion ---- 156

6. Conclusion and Future Work ---- 157

References ---- 160

Vita ---- 171

List of Figures

Figure 1.1: The pond abstracted..... 2
Figure 1.2: Illustration of different states of a Markov process and their transitions..... 3
Figure 1.3: A two-state MC realization of a wireless fading envelope. The observation sequence (a binary error sequence) is a deterministic function of states..... 6
Figure 1.4: A three-state HMM realization of a wireless fading envelope. The observation sequence (a binary sequence) is a probabilistic function of states..... 7
Figure 1.5: Continuous-Phase Binary FSK spectral coherence and α-domain profile at 9dB SNR..... 9
Figure 1.6: Illustration of spectrum holes in cognitive radio network. The cognitive network is utilizing licensed frequency bands by performing CSMA based spectrum allocation..... 10
Figure 2.4.1: Semi-hidden Markov process state transitions..... 20
Figure 2.5.1.1: Computation of forward variable $\boldsymbol{\alpha}$ in the forward-backward BWA..... 23
Figure 2.5.1.2: Computation of backward variable $\boldsymbol{\beta}$ in the forward-backward BWA..... 24
Figure 2.5.1.3: Parameter estimation using forward and backward variables in BWA..... 25
Figure 3.2.1: An example of pessimist test result of MC order estimation using goodness of fit approach. Here the CDF of I is lower than that of 0.5 χ^2 plot..... 41
Figure 3.2.2: An example of optimist MC order estimation test result using goodness of fit approach. Here the theoretical CDF is lower than the empirical CDF..... 42
Figure 3.2.3: An example of true test MC order estimation result using goodness of fit. Here the two CDFs overlap one another..... 43

Figure 3.3.1: A conceptual profile of the log-likelihood surface during HMM parameter estimation using the BWA. 44
Figure 3.3.2: Log-likelihood values obtained by running the BWA 100 times. Significant improvement can be achieved by running the BWA multiple times with different initial conditions..... 46
Figure 3.3.3: Autocorrelation of the original input sequence with the autocorrelations of the two HMMs. 46
Figure 3.8.1: Probability density function of Rayleigh fading envelope. Rayleigh fading parameters are $f_d = 10\text{Hz}$ and $f_s = 0.5\text{Kbps}$. 59
Figure 3.8.2: Fading envelope quantized to different fading levels. Rayleigh fading parameters are $f_d = 10\text{Hz}$ and $f_s = 0.5\text{Kbps}$. 60
Figure 3.8.3: Fade duration distribution and its approximation for the sixth quantization level. Rayleigh fading parameters are $f_d = 10\text{Hz}$ and $f_s = 0.5\text{Kbps}$ 61
Figure 3.8.4: Fade duration distribution and its approximation for the seventh quantization level. Rayleigh fading parameters are $f_d = 10\text{Hz}$ and $f_s = 0.5\text{Kbps}$ 62
Figure 3.8.5: Fade duration distribution and its approximation for the eighth quantization level. Rayleigh fading parameters are $f_d = 10\text{Hz}$ and $f_s = 0.5\text{Kbps}$ 62
Figure 3.8.6: Fade duration distribution and its approximation for the seventh quantization level. Rayleigh fading parameters are $f_d = 10\text{Hz}$ and $f_s = 10\text{Kbps}$ 63
Figure 3.8.7: Fade duration distribution and its approximation for the eighth quantization level. Rayleigh fading parameters are $f_d = 10\text{Hz}$ and $f_s = 10\text{Kbps}$ 64
Figure 3.8.8: Fade duration distribution and its approximation for the ninth quantization level. Rayleigh fading parameters are $f_d = 10\text{Hz}$ and $f_s = 10\text{Kbps}$ 64
Figure 3.9.1: Markov chain order estimation using goodness of fit. The original sequence is a memoryless chain. 65
Figure 3.9.2: Markov chain order estimation using goodness of fit. The true order of Markov chain is 1..... 66
Figure 3.9.3: Markov chain order estimation using goodness of fit. The true order of the Markov Chain is 2..... 68

Figure 3.9.4: Markov chain order estimation using goodness of fit. The true order true is 3..... 70
Figure 3.10.1: Convergence of log-likelihood values using the FB-BWA..... 72
Figure 3.10.2: Comparison of error-free intervals of original data and HMM generated data using the FB-BWA..... 72
Figure 3.10.3: Comparison of error-free intervals of original data and HMM generated data using FO-BWA. 73
Figure 3.11.1: First 20 lags of sample ACF and order-1 MC ACF. Sample ACF is obtained from the data generated by a first-order MC..... 75
Figure 3.11.2: First 20 lags sample ACF and 2-state HMM ACF. Sample ACF is obtained from the data generated by a first-order MC. 75
Figure 3.11.3: First 20 lags of sample ACF and 3-state HMM ACF. Sample ACF is obtained from the data generated by a first-order MC..... 76
Figure 3.11.4: First 20 lags of sample ACF and second-order MC ACF. Sample ACF is obtained from the data generated by a second-order MC. 77
Figure 3.11.5: First 20 lags of sample ACF and first-order MC ACF. Sample ACF is obtained from the data generated by a second-order MC. 78
Figure 3.11.6: First 20 lags of sample ACF and 4-state HMM ACF. Sample ACF is obtained from the data generated by a second-order MC. 78
Figure 3.11.7: First 20 lags of sample ACF and first-order MC ACF. Sample ACF is obtained from the data generated by a three-state HMM. 80
Figure 3.11.8: First 20 lags of sample ACF and 2-state HMM ACF. Sample ACF is obtained from the data generated by a three-state HMM. 80
Figure 3.11.9: First 20 lags of sample ACF and 4-state HMM ACF. Sample ACF is obtained from the data generated by a three-state HMM. 81
Figure 3.11.10: First 20 lags of sample ACF and 2-state HMM ACF. Sample ACF is obtained from the data generated by a five-state HMM. 82
Figure 3.11.11: First 20 lags of sample ACF and 3-state HMM ACF. Sample ACF is obtained from the data generated by a five-state HMM. 83
Figure 3.11.12: First 20 lags of sample ACF and 4-state HMM ACF. Sample ACF is obtained from the data generated by a five-state HMM. 83

Figure 3.11.13: First 20 lags of sample ACF and 5-state HMM ACF. Sample ACF is obtained from the data generated by a 5-state HMM. 84
Figure 3.11.14: First 20 lags of sample ACF and 2-state HMM ACF. Sample ACF is obtained from the data generated by a six-state HMM. 85
Figure 3.11.15: First 20 lags of sample ACF and 6-state HMM ACF. Sample ACF is obtained from the data generated by a six-state HMM. 86
Figure 3.11.16: First 20 lags of sample ACF and 2-state HMM ACF. Sample ACF is obtained from the data generated by a seven-state HMM. 87
Figure 3.11.17: First 20 lags of sample ACF and 7-state HMM ACF. Sample ACF is obtained from the data generated by a six-state HMM. 88
Figure 3.12.1: Rayleigh fading envelope, $f_d = 7\text{Hz}$, $f_s = 0.5\text{Kbps}$, and $\text{SNR} = 10\text{dB}$ 90
Figure 3.12.2: A segment of Rayleigh fading envelope from 17.1sec to 17.2sec, $f_d = 7\text{Hz}$, $f_s = 0.5\text{Kbps}$, and $\text{SNR} = 10\text{dB}$. 90
Figure 3.12.3: The Runs Test applied to the error trace obtained using BPSK with $f_d = 7\text{Hz}$, $f_s = 0.5\text{Kbps}$, and $\text{SNR} = 10\text{dB}$. 91
Figure 3.12.4: Comparison of autocorrelations of the original data with a 2-state and a 3-state HMM. Original error trace is obtained using BPSK simulation with $f_d = 7\text{Hz}$, $f_s = 0.5\text{Kbps}$, and $\text{SNR} = 10\text{dB}$. 92
Figure 3.12.5: A segment of Rayleigh fading envelope from 18.85sec to 18.95sec, $f_d = 7\text{Hz}$, $f_s = 1\text{Kbps}$, and $\text{SNR} = 10\text{dB}$ 93
Figure 3.12.6: The Runs Test applied to the error trace obtained using BPSK with $f_d = 7\text{Hz}$, $f_s = 1\text{Kbps}$, and $\text{SNR} = 10\text{dB}$ 94
Figure 3.12.7: ACF comparison of the original data with a 2-state HMM and a 3-state HMM. Original trace is obtained using BPSK simulation with $f_d = 7\text{Hz}$, $f_s = 1\text{Kbps}$, and $\text{SNR} = 10\text{dB}$. 94
Figure 3.12.8: A segment of Rayleigh fading envelope from 2.1sec to 2.2sec, $f_d = 7\text{Hz}$, $f_s = 1\text{Mbps}$, and $\text{SNR} = 10\text{dB}$ 96
Figure 3.12.9: A segment of Rayleigh fading envelope from 7.5sec to 7.6sec, $f_d = 7\text{Hz}$, $f_s = 1\text{Mbps}$, and $\text{SNR} = 10\text{dB}$ 96

Figure 3.12.10: The Runs Test applied to the error trace obtained using BPSK with $f_d = 7\text{Hz}$, $f_s = 1\text{Mbps}$, and $\text{SNR} = 10\text{dB}$. 97
Figure 3.12.11: Comparison of autocorrelations of the original data with 2-state, 3-state, 4-state, and 8-state HMMs. Original error trace is obtained using BPSK simulation with $f_d = 7\text{Hz}$, $f_s = 1\text{Mbps}$, and $\text{SNR} = 10\text{dB}$. 98
Figure 3.12.12: Distribution of the error-free intervals of the original and HMM generated error sequences. 99
Figure 3.12.13: Comparison of autocorrelations of the original data with a 4-state semi-HMM and a 8-state semi-HMM. Original error trace is obtained using BPSK simulation with $f_d = 7\text{Hz}$, $f_s = 1\text{Mbps}$, and $\text{SNR} = 10\text{dB}$. 100
Figure 3.12.14: Illustration of good and bad traces obtained from the original error trace. 101
Figure 3.12.15: The Runs Test applied to the bad trace that is obtained from spitting the original error trace into good and bad traces. 101
Figure 3.12.16: Comparison of autocorrelations of the bad data trace with a 2-state HMM..... 102
Figure 3.12.17: Comparison of autocorrelations of the bad data trace with a 3-state HMM 103
Figure 3.12.18: Comparison of autocorrelations of the bad data trace with a 5-state HMM..... 103
Figure 3.12.19: Comparison of autocorrelations of the original data with a 10-state HMM..... 104
Figure 3.12.20: Comparison of autocorrelations of the original data with a 15-state HMM..... 104
Figure 3.12.21: Comparison of runlength distributions of the good trace with the distribution of the inverse Gaussian process..... 105
Figure 3.12.22: Comparison of autocorrelations of the original data with the data generated using an inverse Gaussian process and a 15-state HMM. 106
Figure 3.12.23: Distribution of the error-free intervals of the original trace and the trace produced by the inverse Gaussian process and the 15-state HMM. 107

Figure 3.12.24: Comparison of autocorrelations of the bad data trace with the 4-state and 10-state semi-HMMs. 108
Figure 3.13.1: A four inputs, one output and a single hidden layer ANN. 109
Figure 4.3.1: Spectral Coherence and cycle frequency domain profile of DSB-SC AM. 123
Figure 4.3.2: Spectral Coherence and cycle frequency domain profile of BPSK..... 124
Figure 4.3.3: Spectral Coherence and cycle frequency domain profile of binary FSK.. 125
Figure 4.3.4: Spectral Coherence and cycle frequency domain profile of MSK. 126
Figure 4.3.5: Spectral Coherence and cycle frequency domain profile of QPSK. 127
Figure 4.4.1.1: Flowchart of finding dominant peaks using crest factor. 129
Figure 4.4.1.2: Determination of crest factors of BPSK signal. 130
Figure 4.4.2.1: Distribution of crest factor when no signal is present. 131
Figure 4.5.1: Block diagram of HMM based signal classification system. 132
Figure 4.6.1.1: Probability of detection of AM signals with 10% false alarm. 133
Figure 4.6.1.2: Probability of detection for BPSK signals with 10% false alarm. 134
Figure 4.6.1.3: Probability of detection of FSK signals with 10% false alarm. 134
Figure 4.6.1.4: Probability of detection of MSK signals with 10% false alarm. 135
Figure 4.6.1.5: Probability of detection of QPSK signals with 10% false alarm. 135
Figure 4.6.2.1: Percentage of successful classification of AM when incoming signals have SNRs of -3dB and -5dB. Training signal has -3dB SNR and 300 observation length..... 136
Figure 4.6.2.2: Percentage of successful classification of BPSK when the incoming signals have -3dB and -5dB SNRs. Training signal has -3dB SNR and 300 observation length. 137
Figure 4.6.2.3: Percentage of successful classification of FSK when the incoming signals are coming at -3dB and -5dB. Training signal has SNR of -3dB and observation length of 300. 137
Figure 4.6.2.4: Percentage of successful classification of MSK when the incoming signals have SNRs of -3dB and -5dB. Training signal has -3dB SNR and 300 observation length..... 138

Figure 4.6.2.5: Percentage of successful classification of QPSK when the incoming signals have -3dB and -5dB SNRs. Training signal has SNR of -3dB and observation length of 300. 138
Figure 5.1: Illustration of spectrum holes in a cognitive radio network. Spectrum allocation is performed using the conventional CSMA based technique. 141
Figure 5.1.1: Flowchart of the MCPA for a single primary user case. Multiple channel case is an extension of this algorithm. 144
Figure 5.1.2: Illustration of dynamic spectrum allocation using MCPA. 145
Figure 5.2.1: Simulation scenario with four licensed users and one cognitive radio transceiver labeled as 'TX' and 'RX'. The primary users are labeled using upper case alphabets..... 146
Figure 5.2.2: Spectrum usage patterns of (a) - first licensed users, (b) - second licensed user, (c) - third licensed user and (d) - fourth licensed user. The spectral patterns are assumed exponentially distributed. 147
Figure 5.2.3: Comparisons of the primary user's SIR using CSMA based and MCPA based schemes. 148
Figure 5.2.4: BER performance of the MCPA with four primary users. 150
Figure 5.2.5: Percentage of time the cognitive radio uses four radio bands using (a) - CSMA based scheme and (b) - MCPA based scheme. 151
Figure 5.2.6: Spectrum usage patterns of (a) - first licensed users, (b) - second licensed user, (c) - third licensed user and (d) - fourth licensed user. The usage patterns are assumed to be Poisson distributed. 152
Figure 5.2.7: Comparisons of the primary user's SIR using CSMA based and MCPA based schemes. 153
Figure 5.2.8: Performance of the MCPA with four primary users. Throughput is also shown with the corresponding values of the threshold. 154
Figure 5.2.9: Percentage of time the cognitive radio uses each of the four radio bands using (a) - CSMA based scheme and (b) - MCPA based scheme. 155

List of Tables

Table 3.10.1: Probability of error intervals in original sample and HMM generated data obtained using the FB-BWA and the FO-BWA. ---------- 73

Table 3.11.1: AIC and BIC results for different Markov models (original sequence is a first-order binary MC). ---------- 74

Table 3.11.2: AIC and BIC values of different Markov models (original sequence is a second-order binary Markov Chain). ---------- 77

Table 3.11.3: AIC and BIC values of different Markov models (original sequence is a 3-state binary hidden Markov sequence). ---------- 79

Table 3.11.4: AIC and BIC values of different Markov models (original sequence is a five-state binary HMP). ---------- 81

Table 3.11.5: AIC and BIC values of different Markov models (original sequence is a six-state binary HMP). ---------- 84

Table 3.11.6: AIC and BIC values of different Markov models (original sequence is a seven-state binary HMP). ---------- 87

Table 3.12.1: AIC and BIC values of different HMMs (original sequence is an error trace obtained from BPSK with Rayleigh fading simulation with 7Hz doppler frequency, 0.5Kbps sampling frequency, and 10dB SNR). ---------- 92

Table 3.12.2: AIC and BIC values of different HMMs (original error trace is obtained using BPSK simulation with $f_d = 7\text{Hz}$, $f_s = 1\text{Kbps}$, $\text{SNR} = 10\text{dB}$). ---------- 95

Table 3.12.3: AIC and BIC values of different Markov models (original error trace is obtained using BPSK simulation with $f_d = 7\text{Hz}$, $f_s = 1\text{Mbps}$, and $\text{SNR} = 10\text{dB}$).--- 98

Table 3.12.4: Comparison of error bursts of different lengths in the original and HMM-generated trace.---------- 99
Table 3.12.5: AIC and BIC values of different Markov models (original sequence is binary bad trace). ----------105
Table 3.12.6: Comparison of error bursts of different lengths in original and the trace generated from the inverse Gaussian process and the 15-state HMM.----------107
Table 5.2.1: Performance of the MCPA in terms of bit error rate and throughput. ------149
Table 5.2.2: Performance of the MCPA in terms of bit error rate and throughput. ------154

Chapter 1

Introduction

We can specify a physical system by giving the values of a number of variables that describe the system. For example, a chemical system can often be specified by the values of temperature, pressure, and volume, whereas the instantaneous description of a spacecraft would include its position in spatial coordinates, its mass, and its velocity. These critical variables of a system are called state variables. When the values of all state variables of a system are known, we can say that its state has been completely specified. Thus we can say that the state of a system is all that is needed in order to describe the system at any instant. In the course of time a system passes from state to state and thus exhibits dynamic behavior. Such changes are called state-transitions or simply transitions.

To explain a Markov process, let us start with an example of a frog and a lily pond as described in [1]. Assume that there are a large number of lily pads in a pond, and a frog is sitting on one of these pads. Let us also assume that the frog must always sit on a pad and never swim in the water. It occasionally jumps in the air and lands on a lily pad; sometimes the one it had been sitting on and sometimes a different one. For the moment we are interested not in the time pattern of its jumping, but in the location of the frog after successive jumps.

To make our discussion specific, lets consider that the pond has a finite number of lily pads, N, numbered from 1 to N. If we use n to denote the number of jumps made

by the frog, and S_n to denote the number of the pad occupied after the nth jump, then the sequence

$$S_0, S_1, S_2, ..., S_n, S_{n+1}$$

specifies the frog's itinerary. Such a sequence is known as the trajectory of the process.

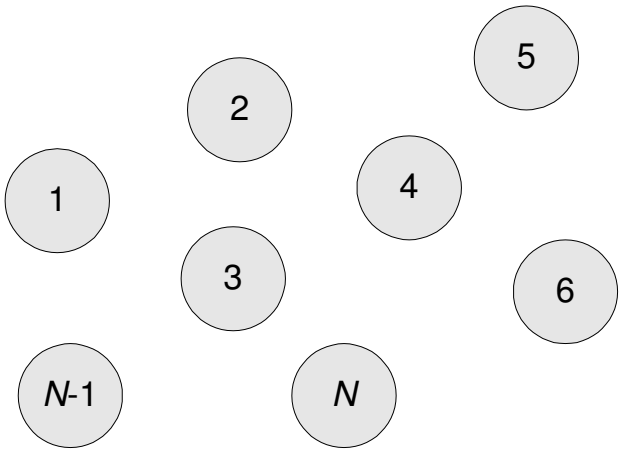


Figure 1.1: The pond abstracted.

We now have a physical realization of a *state-transition* process. The state of the process is the number of the pad occupied by the frog. The quantity S_n represents the state of the process after its nth transition, which is called its state at time index n. In an N-state process, S_n may take on any integer value $1, 2, ..., N$ at any time index $n = 0, 1, 2, ...$ that shows the number of transitions made by the process. Figure 1.1 shows the abstraction of the pond.

As we observe the sequence of the states occupied by the frog, i.e., the process trajectory, it can be easily observed that it is random. The statistical behavior of this process could be specified if the probability

$$\Pr(S_{n+1} = j \mid S_n = i, S_{n-1} = k, ..., S_0 = m) \tag{1.1}$$

was known for all values of its arguments. This is the probability that each state will be occupied after the $n+1$th transition, given the entire trajectory or history of state occupancies through time n.

Fig. 1.2 shows a typical state transition diagram of an *N*-state Markov process. The states are labeled using numbers 1,2, …, *N*, and possible transitions are shown by arrows originating from one state and transitioning into a target state.

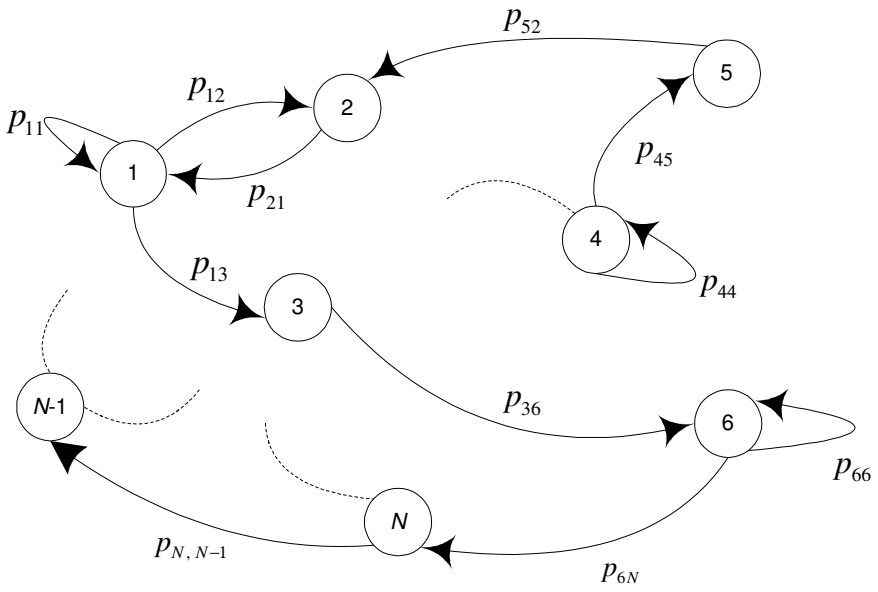


Figure 1.2: Illustration of different states of a Markov process and their transitions.

In this thesis, the terms Markov Chain (MC), Hidden Markov Model (HMM) and Semi-Hidden Markov Model (S-HMM) are used specifically for the mathematical models that generate Markov chain process (MCP), hidden Markov process (HMP) and semi-hidden Markov process (S-HMP), respectively. Although these categories of Markov models are related to one another, their mathematical interpretations vary widely and cannot be generalized.

To develop a certain Markov model based on some observed data, we need to estimate its parameters, i.e., the probability of transitioning from one state to another and

in case of HMMs, also the probabilities of producing different symbols while being in a particular state. For this purpose, maximum likelihood estimate is used for MCs and the Baum-Welch algorithm (BWA), a special case of Expectation Maximization (EM) algorithm, is widely used for discrete HMM parameter estimation [2]. A derived form of the BWA is available for the parameter estimation of S-HMMs [3], [4]. Note that in all these estimation problems, we first have to assume a fixed number of states of model under study.

Although Markov models are used extensively in wireless channel modeling, an important point about these models should be noted. Markov channel models are brittle in the sense that a slight change in system parameters, e.g., a change in received SNR, in transmitter-receiver separation, in channel impulse response, in the modulation format, or in the receive signal processing, requires that new system performance data be generated and the Markov parameters re-estimated.

The accurate modeling of error sequences that occur in wireless channels is necessary for a better understanding of network performance and for improving the design of communication protocols. For example, a thorough understanding of the packet loss and busty nature of errors is essential for the proper design and parameter tuning of error control protocols, etc. The term *discrete channel model* is usually used for technique that can simulate the error traces produced by wireless channels [5]. One of the primary advantages of these channel models is fast execution time. Using these models, one can dynamically generate artificial error patterns for the communication system under study and use these traces to simulate, and thus better understand, the performance of a given system. Due to their ease of use and repeatability, the system engineer can utilize these error patterns for characterizing and improving the performance of a wireless system.

Chapter 3 of this thesis deals with the modeling of error traces that occur in wireless channels. Real world wireless channels often contain memory and hence these error events cannot be treated as independent random processes. It is therefore not possible to model such data by using standard random variables available in literature. In such situations, Markov models have proved to be a valuable tool in analyzing and modeling/reproducing observed data. The natural candidates for this purpose are Markov

Chains (MCs) and hidden Markov models (HMMs) [6], [4]. Both of these models are flexible enough to fit a wide variety of experimental data with high accuracy. Although we usually use MC and HMMs for modeling experimental data, semi-hidden Markov models (S-HMMs) are also becoming popular especially in the field of telecommunications [7], [4]. This is primarily because of their reduced computational burden in parameter estimation as compared to traditional Markov models.

Although the theory of parameter estimation of Markov processes is well established and understood, the work on estimating the order of these models has started in early 1990s. Finesso, in 1990, used the Law of Iterated Logarithms (LIL) and information theoretic measures to estimate the order of MCs and HMMs [8]. Khundapur and Narayan have recently proposed an order estimation technique for S-HMMs that assumed that the model under study has only one bad state [9]. The order of a Markov process is defined as the minimum number of states required to model the data accurately. In MCs, this corresponds to the depth of memory of the process whereas in HMMs, the order corresponds to the number of quantized state levels. As the problem of Markov order estimation is yet to be solved satisfactorily, we present a simulation based approach to the order estimation of discrete wireless channel models. Specifically, we study the order estimation of hidden Markov models that are used to represent the error traces generated by the Rayleigh fading channels. We will discuss order estimation of Markov models in detail in Chapter 3.

Let's discuss error traces that are generated by wireless channels in more detail. Fig. 1.3 shows the realization of a fading envelope of wireless channel [10] using a Markov chain model. The envelope is divided into two states, i.e. a good state and a bad state. The channel generates no errors (zeros) when it is in good state and generates errors (ones) when it is in bad state. In this way, the channel produces error patterns by transitioning from one state to another state. Note that even when the transmitter is turned off, the maximum Bit Error Rate (BER) that a communication system can incur can not exceed 0.5. Since the process is deterministic, the state sequence can be generated just by looking at the observed error patterns. The increase in the number of states of a MCP corresponds to increase in the memory of the process.

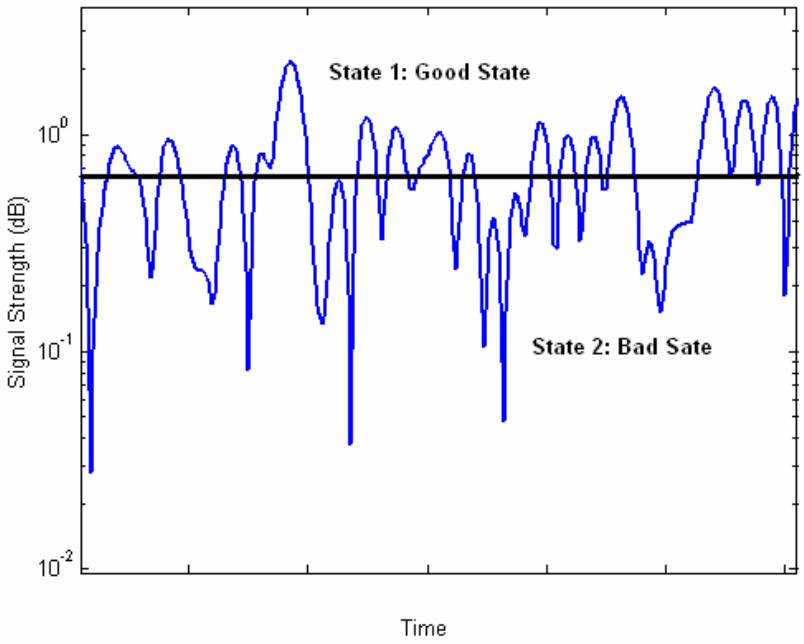


Figure 1.3: A two-state MC realization of a wireless fading envelope. The observation sequence (a binary error sequence) is a deterministic function of states.

Fig. 1.4 shows the realization of similar wireless channel fading envelope using a three-state hidden Markov process. The envelope is now divided into three quantized levels and the divided regions so formed are labeled as a good state, an intermediate state and a bad state. We assume that when this system is in the good state, its performance is quite satisfactory whereas when in the intermediate state, the probability of error generation is considerably high as compared to that in the good state. In the bad state, we experience significant degradation in system performance.

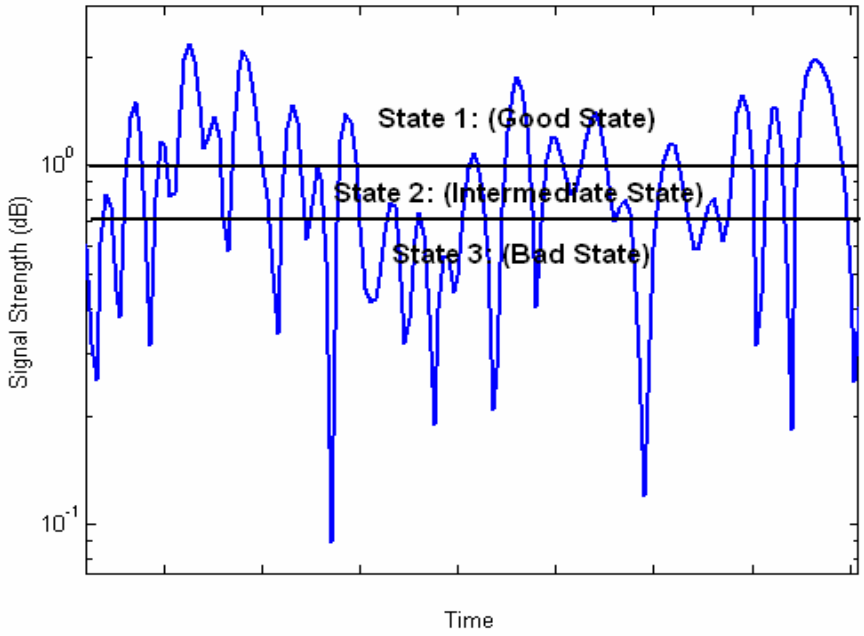


Figure 1.4: A three-state HMM realization of a wireless fading envelope. The observation sequence (a binary sequence) is a probabilistic function of states.

Contrary to the Markov Chain process, in HMM the observation sequence is a probabilistic function of its states. Therefore in our case of bit error patterns, a probability is assigned to each state that corresponds to the likelihood of generating zeros or ones. In other words, every state behaves as a Bernoulli source [11] with usually different probability of error generation. In a good state, the probability of producing 0s (no errors) is significantly higher than producing 1s (errors) and vice versa. Note that even if we further divide a state into two sub-states (or in other words if we increase the number of states of the HMM), the depth of the memory still remains the same. State splitting is generally assumed, in practical applications of HMMs, with an assumption that the system will perform better by increasing the number of hidden states. In our work of wireless channel error trace modeling, we have found that this is not necessarily true and a two-state HMM is usually sufficient for modeling the error behavior of wireless channel. PH distributions approximations [4] are used to model the fade durations in

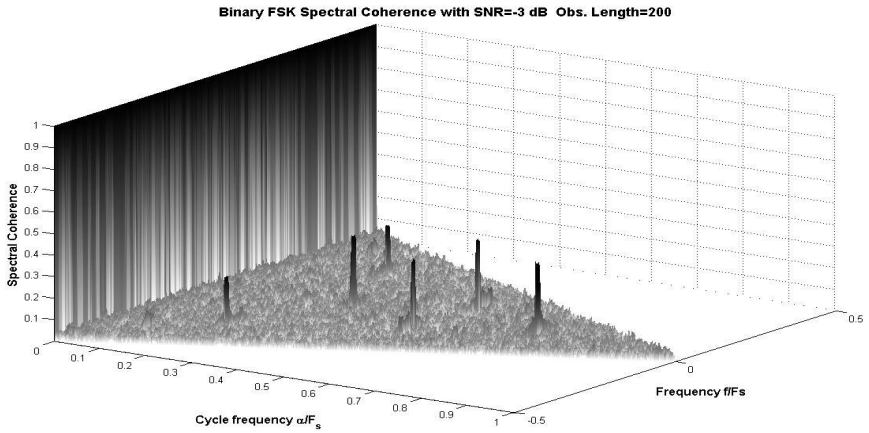
wireless channels. Results of modeling both slow and fast fading channel error sources are presented in this report.

Chapter 4 and 5 of this thesis deals with cognitive radio (CR). CR is a *brain-empowered* wireless communication paradigm that is aware of its environment and uses the methodology of understanding-by-building to learn from the environment and to adapt to statistical variations in the input stimuli. The two primary objectives for the design of cognitive radio are highly reliable communications whenever and wherever needed and a more efficient utilization of the radio spectrum. The three fundamental cognitive tasks are radio-sense analysis, channel-state estimation, transmit power control and dynamic spectrum management. The six (ideal) characteristics of a cognitive radio are as follows:

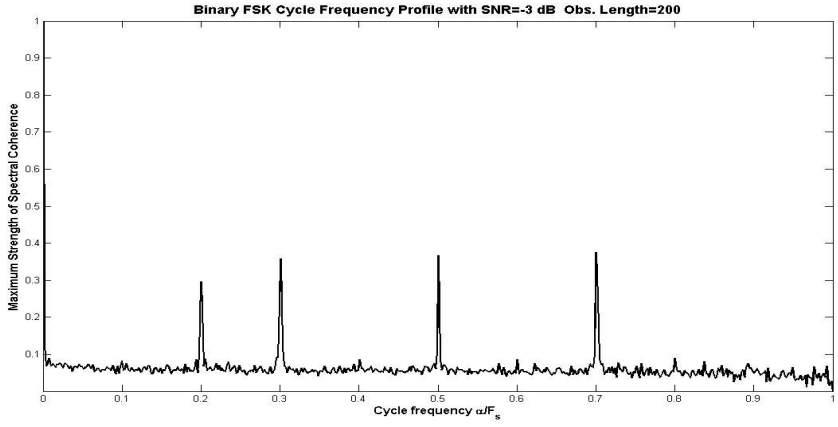
Intelligence - allows the radio to have the ability to perform autonomous decisions.
Awareness - lets the radio be familiar with itself, its capabilities and its surroundings.
Learning - to be capable of dealing with situations that were not anticipated during the network's initial deployment.
Adaptability - to be able to adapt to new regulations.
Reliability - should be able to establish reliable communications whenever and wherever possible.
Efficiency - should be efficient in performing its tasks.

A cognitive engine can be defined as the *intelligent* agent that manages the cognition tasks in a cognitive radio. This agent can be thought of as an independent entity that oversees the cognitive operations of the radio. Given the input from its environment or users, the cognitive engine analyzes the situation, performs necessary calculations and proceeds by responding or reacting to the stimulus. As an example, this response can be adapting some of radio's parameters such as modulation scheme, frequency of transmission, frequency of reception, etc. given the user requirement and the current radio environment.

Detection of CR signal is difficult by using conventional energy detection techniques because of very low Signal-to-Noise Ratio (SNR). Second order cyclostationarity [12] allows signal sensors or receivers to classify the incoming signal in very low SNR situations. It also reveals distinctive features in the cycle frequency domain according to the modulation parameters of transmitted signal. We use second order statistics of incoming signal in the frequency domain and utilize its two dimensional statistics, called the α-domain profile, as the training sequence for the HMM. Maximum likelihood method is used for decision making along with case-based reasoning to reduce the number of HMMs. A typical spectral correlation function and its corresponding α-domain profile of an FSK signal at 9dB SNR is shown in Fig. 1.5 below.



(a) Spectral Coherence



(b) Cycle Frequency Domain Profile

Figure 1.5: Continuous-Phase Binary FSK spectral coherence and α-domain profile at 9dB SNR.

In November 2002, the Federal Communications Commission (FCC) published a report prepared by the Spectrum-Policy Task Force aimed at improving the way in which this precious resource is managed in the United States. In their report they state that "In many bands, spectrum access is a more significant problem than physical scarcity of spectrum, in large part due to legacy command-and-control regulation that limits the ability of potential spectrum users to obtain such access."

Measurements have shown that there are frequency bands even in urban areas that are largely unoccupied, some of them are only partially occupied and some are heavily used. This spectrum usage statistics gives us a notion of *spectrum holes* which is defined in [13] as a band of frequencies assigned to the primary user, but at a particular time and specific geographical location, the band is not being utilized by that user. The concept of spectrum holes can be shown in Fig. 1.6.

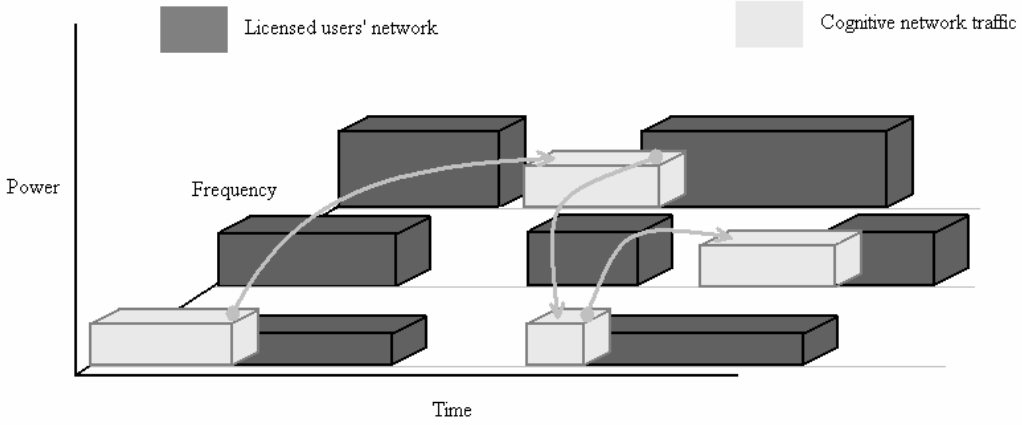


Figure 1.6: Illustration of spectrum holes in cognitive radio network. The cognitive network is utilizing licensed frequency bands by performing CSMA based spectrum allocation.

Spectrum utilization can be improved significantly by making it possible for a secondary user (who is not being serviced) to access the spectrum hole unoccupied by the primary user at the right location and at the right time. Cognitive radio has recently been

proposed as the method to promote the efficient use of the spectrum by exploiting the existence of spectrum holes. It observes the current spectrum conditions in the given band, orients itself, identifies the available spectrum, plans the possible user for it, decides on the best plan, and acts by allocating resources, negotiating protocols, etc. While engaged in the transmission, the radio may observe the signal of the incumbent user. If such signal is detected, the radio's cognition cycle should direct itself to jump out of the spectrum to minimize interference. It should then look for other available spectrums to continue its data transmission.

We propose to predict the spectrum usage patterns of different licensed users using HMMs. Different HMMs would first train themselves from to the spectrum usage patterns of the incumbent users and once reliable models are developed, the spectrum manager would decide which frequency to use based on the likelihood of spectrum holes of these bands. In our preliminary work, usage patterns are assigned binary bits with zeros indicating no traffic and ones indicating that the spectrum is being used by the incumbent user at that particular time. Hence we obtain binary vectors for different frequency bands and use these vectors as training sequences for HMMs assigned for different radio bands. These HMMs will then predict the occurrence of spectrum holes for their respective users and the cognitive engine will decide which frequency to use based on these predictions.

Some notable contributions are as follows:

- A simulation based study on the order estimation of binary hidden Markov wireless channel models.
- A simulation based study on the behavior of the Bayesian Information Criterion (BIC) and Akaike's Information Criterion (AIC) for binary Markov Chains and hidden Markov models.
- A novel HMM based classification of different types of signals having very low SNRs using cyclostationarity.
- A novel technique of dynamic spectrum allocation in cognitive radio networks using HMMs.

Chapter 2

Markov Theory

This chapter is organized as follows. We first present the basic Markov theory and discuss is mathematical formulation of Markov processes. We then discuss Markov chains, hidden Markov models and semi-hidden Markov models, the three major categories of Markov models. Parameter estimation of discrete Markov models is studied next. We will focus our study on different forms of the Baum-Welch algorithms that are used to estimate the parameters of HMMs and semi-HMMs. Phase-type distribution approximation is discussed next that is used for modeling quantized fading statistics.

2.1 Markov Processes

Markov processes are those processes in which the current state of system depends on the previous state (or states). A system can have finite number of states or it can have infinite states. However, a finite state assumption is obviously more practical. We define the discrete state space $\mathbb{S}$ with elements $s_1, s_2, ..., s_N$ or simply with numbers $1, 2, ..., N$ when there is no ambiguity.

A *homogeneous* finite state Markov process can be defined as a process in which the current state of the system depends on the previous state (or states) in such a way that the state dependency at a particular time instant is not dependent on the time at which the system is observed. Mathematically, if $x_1^t = x_1, x_2, ..., x_t$ is the state sequence and a_{ij} is the

probability of transitioning from state *i* to state *j*, then for a first-order Markov process we can write

$$a_{ij} = \Pr(x_t = i \mid x_{t-1} = j) \tag{2.1.1}$$

Consider a transition from a particular state s_i to some other state s_j in exactly n steps. This transition from state s_i to s_j can occur via different paths and the conditional probability that the system passes through a particular path, say $s_i \to s_{i_1} \to s_{i_2} \to \ldots \to s_{i_{n-1}} \to s_j$ is denoted by $a_{ii_1}a_{i_1i_2}\ldots a_{i_{n-1}j}$. The sum of the corresponding expressions for all possible paths is the probability of finding the system at time $r+n$ in state s_j, given that at time r it was in state s_i and is denoted by $a_{ij}^{(n)}$. Hence,

$$a_{ij}^{(1)} = a_{ij} \tag{2.1.2}$$

$$a_{ij}^{(2)} = \sum_u a_{iu}a_{uv} \tag{2.1.3}$$

and in general

$$a_{ij}^{(n+1)} = \sum_u a_{iu}a_{uj}^{(n)} \tag{2.1.4}$$

and

$$a_{ij}^{(m+n)} = \sum_u a_{iu}^{(m)}a_{uj}^{(n)} \tag{2.1.5}$$

The last equation reflects the fact that the first m steps lead the system from s_i to some intermediate state s_u, and the last n steps from s_u to final state s_j.

A Markov process is *irreducible* (or *indecomposable*) if every state is accessible from every other state. In other words, s_j can be reached from s_i with some t such that $a_{ij}^{(t)} > 0$. With the homogeneity condition, we can replace $a_{ij}^{(t)}$ by a_{ij}. A Markov process is *aperiodic* (or *acyclic*) if there exists no state such that the process will continue to return to that state with fixed periods $T, 2T, 3T, \ldots,$ where $T > 1$ is the greatest integer with this property. Positive *recurrent* (or *persistent*) Markov chains are those Markov processes in which the expected return time is finite for every state. A transition matrix $\mathbf{A} = [a_{ij};\ i, j = 1, 2, \ldots, N]$ which is positive (that is, every element of the matrix is

positive) is irreducible and aperiodic. A matrix is a stochastic matrix if and only if the rows sum to 1. Hence the state transition probability matrix is always a stochastic matrix. For a discrete state space, the integration in the *k*-step transition probability are summations that can be computed as $\mathbf{A}^k$ (the *k*-th power of the state transition matrix). In other words, if $\mathbf{A}$ is one-step transition matrix, then $\mathbf{A}^k$ is the transition matrix for the *k*-step transition. The stationary distribution of states is a vector $\boldsymbol{\pi}$ that satisfies $\boldsymbol{\pi}\mathbf{A} = \boldsymbol{\pi}$. We can also say that the stationary vector is a left eigenvector of the transition matrix, associated with the eigenvalue 1.

If $\mathbf{A}$ is irreducible and aperiodic, then there always exist a unique stationary distribution. It is also noted that $\mathbf{A}$ converges to a rank-one matrix in which each row denotes the stationary distribution of the states. Mathematically, $\lim_{k\to\infty} \mathbf{A}^k = \boldsymbol{\pi}$. This means that if we observe a random walk based on an irreducible and aperiodic transition matrix $\mathbf{A}$, then the long term probability of being in a given state is independent of the state from where the chain was started and is dictated by the stationary distribution. This is sometimes known as the *exponential forgetting* of Markov chains.

After defining these terms we are in a position to discuss ergodicity in Markov chains. An *ergodic* finite state Markov process is defined as a process that is aperiodic and irreducible. Most of the systems of our interest are modeled as ergodic Markov chains, as this corresponds to a well-defined steady-state behavior. For additional material on these topics, refer to [6], [7], and [14].

Broadly speaking, Markov processes can be divided into three sub-categories; Markov chains, hidden Markov processes and semi-hidden Markov processes. A Markov chain process (MCP) is the process in which the states are directly observable. In other words, the observation sequence is a deterministic function of the states. A hidden Markov process (HMP) is a doubly stochastic process in which the generation of observation symbols depends on the emission properties of the states. Therefore, a state can usually generate more than one observation symbol and we cannot directly observe the state sequence from the observation sequence. Semi-hidden Markov process (S-HMP) is a process in which each state can emit a sequence of observations before transitioning to another state. The main purpose of this report is to investigate the order estimation of binary Markov processes, show our preliminary results in this area and explain proposed

future research directions. Recall that by the order of Markov models we mean the minimum number of states that is required to correctly model an observed sequence. For the case of Markov chains, order corresponds directly to the memory of the chain whereas for the case of hidden Markov models and semi-hidden Markov models, this generally corresponds to the number of hidden states. In this report, we will be mainly focused on the order estimation of Markov chains and hidden Markov processes. A performance comparison with the semi-HMMs is also presented at the end of chapter 3.

2.2 Markov Chain

Markov chains (also known as *Random Walks on Graphs*) are the simplest forms of a Markov process. Markov chains are probably used in all fields of engineering, physics, biology, economy, etc. In wireless communications, they are mainly used to model complex channels that contain memory, especially in those places where we don't have closed form expressions to approximate or reproduce the behavior of such channels. Extremely complex error statistics can be successfully modeled using these simple mathematical models. In this section we formally define Markov chains and highlight some of their important characteristics.

A Markov chain of order k is defined as a process $X_1, X_2, ..., X_t$ such that

$$\Pr(X_{t+1} \mid X_1, X_2, ..., X_t) = \Pr(X_{t+1} \mid X_{t-k+1}, X_{t-k+2}, ..., X_t) \qquad (2.2.1)$$

For a first-order Markov chain, the state transition matrix $\mathbf{A}$ is written as

$$\mathbf{A} = (a_{ij}) = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1N} \\ a_{21} & a_{22} & \dots & a_{2N} \\ : & & & : \\ : & & & : \\ a_{N1} & a_{N2} & \dots & a_{NN} \end{bmatrix} \qquad (2.2.2)$$

with $a_{ij} = \Pr(X_{n+1} = i \mid X_n = j)$ and $\sum_{j=1}^{N} a_{ij} = 1$.

As an example of a first-order Markov chain, let us show how to predict the probability of a particular symbol pattern. Suppose there is a three state Markov chain process in which state 1 denotes *Rainy* (R), state 2 denotes *Cloudy* (C) and state 3 denotes *Sunny* (S) weather. The state transition probability matrix is assumed to be

$$\mathbf{A} = \begin{bmatrix} 0.4 & 0.3 & 0.3 \\ 0.2 & 0.6 & 0.2 \\ 0.1 & 0.1 & 0.8 \end{bmatrix} \tag{2.2.3}$$

We wish to compute the probability of observing S S R S C S given that today is S. Using the chain rule, we obtain

$$\begin{aligned} \Pr(\mathrm{S,S,R,S,C,S}) &= \Pr(\mathrm{S})\Pr(\mathrm{S}\,|\,\mathrm{S})\Pr(\mathrm{R}\,|\,\mathrm{S})\Pr(\mathrm{S}\,|\,\mathrm{R})\Pr(\mathrm{C}\,|\,\mathrm{S})\Pr(\mathrm{S}\,|\,\mathrm{C}) \qquad (2.2.4) \\ &= \pi_3 a_{33} a_{31} a_{13} a_{32} a_{23} \qquad (\pi_3 = 1, \text{ by assumption}) \\ &= 1(0.8)(0.1)(0.3)(0.1)(0.2) = 0.00048 \end{aligned}$$

For the case of discrete state space, the *k*-step transition probability can be computed as the *k*-th power of $\mathbf{A}$. If $\boldsymbol{\pi}$ is the stationary distribution, then the stationary Markov chain satisfies the relation $\boldsymbol{\pi}\mathbf{A} = \boldsymbol{\pi}$. If $\mathbf{A}$ is stationary and aperiodic, for *k* sufficiently large

$$\lim_{k\to\infty} \boldsymbol{\pi}_1 \mathbf{A}^k = \boldsymbol{\pi} \tag{2.2.5}$$

and hence the process is independent of the initial distribution $\boldsymbol{\pi}_1$. The mathematical form of multiple order Markov chains is more complex. We will discuss some cases of second-order and third-order binary Markov chains in chapter 3.

2.3 Hidden Markov Models

A simple HMP can be explained as follows. Suppose we have three urns and these three urns contain balls having different colors and with different proportions. There are four colored balls, i.e., green, blue, red and black. We randomly select an urn according to some probability distribution, and pick a ball from that urn at random. After noting the color of the ball, we place the ball in the same urn and then select another urn, based on a specific probability distribution. The urn might be the same urn selected

before or it can be a different urn. We now pick a ball from this newly selected urn and observe its color. Hence there are two random processes occurring one after another, i.e., the selection of an urn and the selection of a ball from that particular urn. If we repeat this process many times then we have a sequence identifying different balls that we picked from different urns. However note that looking at a particular ball does not identify the urn from which the ball was picked. We can observe that there is an underlying hidden process, i.e., the selection of urns (which is not directly observable when we look at the sequence of balls taken from these urns). The weather example discussed earlier can also be explained as a HMP if we include an underlying process such as the cloud conditions in a particular day. The wireless channel can be modeled in the same way where different quantized fading levels can define different states of the channel and where every state has a specific distribution of generating zeros and ones. The combined output of these states gives the overall bit error statistics of the communication system under study.

Let us formally define HMP based on stochastic realization theory. A hidden Markov process can be defined as the pair $\{X_t, Y_t;\ t \in \mathbb{N}\}$ of stochastic processes defined on the probability space (Ω, F, P). The finite set (X, Y) is said to be a stationary finite state system (SFSS) if the following conditions are met:

(i) (X_t, Y_t) are jointly stationary

(ii) $\Pr(Y_{t+1} = y_{t+1}, X_{t+1} = x_{t+1} \mid Y_1^t = y_1^t, X_1^t = x_1^t) = \Pr(Y_{t+1} = y_{t+1}, X_{t+1} = x_{t+1} \mid X_t = x_t)$

The processes X_t and Y_t are called the state and the output of the SFSS respectively. The cardinality of $\mathbb{S}$ will be called the size of SFSS. The probability distribution of discrete observations that can take *M* values is usually expressed in matrix form as

$$\mathbf{B} = \begin{bmatrix} b_{1Y_1} & b_{1Y_2} & \ldots & b_{1Y_M} \\ b_{2Y_1} & b_{2Y_2} & \ldots & b_{2Y_M} \\ : & & & \\ b_{NY_1} & b_{NY_2} & & b_{NY_M} \end{bmatrix} \tag{2.3.1}$$

with $b_{ik} = \Pr(Y_t = k \mid X_t = i) \quad i = 1, 2, ..., M;\ j = 1, 2, ..., N$. A stochastic process Y_t with values in the finite set $\mathbb{Y}$ is a hidden Markov process if it is equivalent to the output of the SFSS.

We now briefly discuss the estimation of these two probability distributions in HMMs, i.e. the probability of transitioning from one state to another or to itself and the probability of generating symbols from a particular state. The first contribution related to the parameter estimation of hidden Markov processes was made in the late sixties. Baum and Petrie [15] studied the Maximum Likelihood Estimation (MLE) of the parameters of an HMP. They also provided an algorithm for the numerical computation of the MLE, basically inventing the Expectation Maximization (EM) algorithm for HMM parameter estimation, that later became very popular. In 1975, the HMP was proposed by Baker as one of the models of choice for automatic speech recognition [16]. Computational aspects became very important and much work was done on the implementation of Baum's algorithm. Later a refined form of this algorithm was proposed that became popular as the Baum-Welch Algorithm (BWA). For details about the BWA and its different forms (for both HMMs and semi-HMMs), refer to [4].

We have previously used HMMs for modeling error sources in WCDMA systems [17] and implemented different BWA algorithms to fit real world bit error statistics. Our work was based on the assumption that the cardinality of state space can be either 2 or 3, i.e., we assumed that the HMMs used for modeling has 2 or 3 states. The idea of using a 2 state or a 3 state model was based on the assumption that the channel conditions are such that we have a state (that we call a good state) in which errors occur very rarely, also there is an intermediate state where errors occur with some higher rate and we have a third state (a bad state) in which errors occur quite frequently. For a 2-state model, we assume that the channel has two states, a good and a bad state. Clearly our assumption of using a 2-state or 3-state model could be incorrect. Hence we realize a need for an order estimation technique that can tell us the optimum number of binary HMM states that should be used for a given model.

At this point let us define the order of a HMM more formally. The order of an HMP Y_t is the minimum integer for which there exists a Markov chain X_t such that $Y_t = f(X_t)$ for some function f. The knowledge of the order of an observed HMP Y_t allows the construction of the *most economical* representation $f(X_t)$ in the sense that the number of parameters (the transition probabilities of X_t) is minimized. It can be shown

that the order cannot be estimated using the classical maximum likelihood technique because increasing the number of states automatically increases the likelihood [8]. This is a typical behavior of the likelihood function when the parameter is structural, i.e., when the parameter (usually integer valued) indexes the complexity of the model. Models with an equal number of states can each be described by the same number of parameters and are interpreted to have the same degree of complexity.

2.4 Semi-Hidden Markov Models

These are also known as *semi-Markov* processes or *Markov renewal* processes [7], [4]. The semi-hidden Markov approach often leads to a significant reduction of the model parameter set and hence computationally efficient models can be achieved as compared to the HMM in most cases. This is especially true when the observation symbol contains long stretches of identical symbols. We can view semi-hidden Markov process as the process whose successive state occupancies are governed by the transition probabilities of a Markov process, but whose stay in any state is described by an integer-valued random variable that depends on the state presently occupied, and on the state to which the next transition will be made. Thus at transition instants the semi-Markov process behaves just like a Markov process. However, the time at which transition occurs is governed by a different probabilistic mechanism. This process is also called the *embedded Markov process*. The semi-Markov process can be described mathematically in the following paragraphs.

Let p_{ij} be the probability that a semi-hidden Markov process that entered state i on its last transition will enter j on its next transition. The transition probabilities, p_{ij}, must satisfy the same equations as the transition probabilities for a Markov process discussed in section 2.2 and 2.3, i.e.,

$$p_{ij} \geq 0 \qquad i = 1, 2, ..., N; \quad j = 1, 2, ..., N \tag{2.4.1}$$

and

$$\sum_{j=1}^{N} p_{ij} = 1 \qquad i = 1, 2, ..., N \tag{2.4.2}$$

where the process has N states. Whenever the system is in state i, it determines the next state j according to the state i's transition probabilities $p_{i1}, p_{i2}, ..., p_{iN}$. However, after j has been selected, but before making this transition from state i to state j, the process *holds* for a time τ_{ij}, in state i. The holding time, τ_{ij}, is a positive, integer-valued random variable governed by a probability mass function $h_{ij}(.)$, known as the holding time mass function for a transition from state i to state j. Thus,

$$\Pr(\tau_{ij} = m) = h_{ij}(m) \qquad m = 1, 2, 3, ...; \; i, j = 1, 2, ..., N; \tag{2.4.3}$$

Assume that the means $\bar{\tau}_{ij}$ of all holding time distributions are finite and that all holding time mass distributions are at least one time unit in length, i.e.,

$$h_{ij}(0) = 0 \tag{2.4.4}$$

Fig. 2.4.1 shows the portion of a possible trajectory for a discrete-time semi-hidden Markov process. Large black dots show that the process enters a given state immediately after a transition. The figure shows that the process entered the state 1 at time 0, was held in state 1 for 3 time units, and then made the transition to state 2. It then held in state 2 for 1 time unit, and then jumped to state 3. It held in state 3 for 2 time units, and then jumped to state 2.

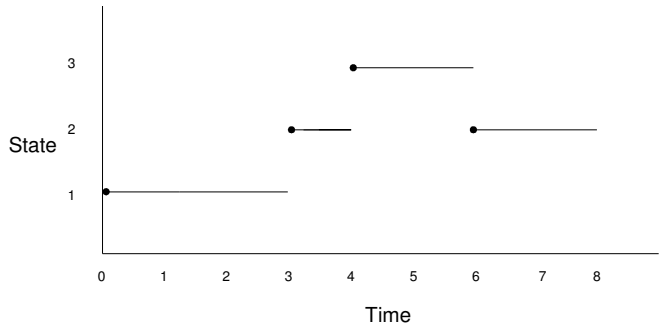


Figure 2.4.1: Semi-hidden Markov process state transitions

Algorithms similar to the BWA are available for the parameter estimation of S-HMMs [3], [4]. S-HMM parameter estimation algorithms for binary sources take

runlength vectors as the input sequence. A sequence of zeros and ones 00001100000000000000011100000001100000 can be compactly written as $0^4 1^2 0^{14} 1^3 0^6 1^2 0^5$. This method of compression is called *runlength coding*. Since error sequences produced by wireless channels usually contain long stretches of zeros (denoting no error in transmission), the equivalent runlength vector is significantly shorter in length than the original binary sequence. This results in considerable reduction in simulation runtime as compared to the traditional BWA [4], [17].

2.5 EM Algorithms for HMM and Semi-HMM Parameter Estimation

In this section we will discuss different forms of the Baum-Welch algorithm that we use to estimate the parameters of HMMs and semi-HMMs.

2.5.1 The Forward-backward BWA for HMM

In this section we will introduce the forward-backward BWA algorithm to estimate $\mathbf{A}$, $\mathbf{B}$, and $\boldsymbol{\pi}$ using the forward and backward variables given the observation sequence.

The Baum-Welch forward-backward algorithm [4], [18], and [19] is perhaps the most efficient and frequently used method for automatically estimating the parameters of discrete HMMs. Its biggest appeal lies in the way it effectively avoids the possible explosion in computational complexity resulting from the evaluation of all the states of a model of training data. The key to the elimination of redundant computation is to exploit the fact that all possible state sequences must merge into one of the complete states set at one time. The BWA is a special case of the EM algorithm.

The implementation of the forward-backward Baum-Welch algorithm involves computation of two different probability terms. First, the forward path probability is defined as the joint probability of having generated a partial observation sequence in the forward direction (i.e., from the start of the data), and having arrived at a certain state at a certain frame. Next, the backward path probability denotes the probability of generating a partial observation sequence in the reverse direction (from the final frame of data), given

that the state sequence starts from a certain state at a certain time. The parameters of the HMMs can be estimated by using these probabilities. The full details of the Baum-Welch procedure for parameter estimation, as well as the various implementation issues, is described in this section.

Let us define a sequence $y_1^T = y_1, y_2, ..., y_T$ and also the model $\theta = \{\mathbf{A}, \mathbf{B}, \boldsymbol{\pi}\}$. Define the forward variable

$$\alpha_t(i) \triangleq \Pr(y_1, y_2, ..., y_t, X_t = s_i \mid \theta) \tag{2.5.1.1}$$

as the probability of the partial observation sequence $y_1, y_2, ..., y_t$ in state s_i at times step t given the model θ. This variable can be computed inductively, and from these variables, $\Pr(y_1^t \mid \theta)$ is computed. The initialization is done as follows

$$\alpha_1(i, \theta) = \Pr(y_1, X_1 = s_i \mid \theta) = \Pr(y_1 \mid X_1 = s_i, \theta) \Pr(X_1 = s_i \mid \theta) \tag{2.5.1.2}$$

$$\alpha_1(i, \theta) = \pi_i b_i(y_1) \tag{2.5.1.3}$$

In the matrix form, we can write

$$\boldsymbol{\alpha}(y_1^1, \theta) = \boldsymbol{\pi} \tag{2.5.1.4}$$

and the induction is done as

$$\alpha_{t+1}(j) = \Pr(y_1, y_2, ..., y_{t+1}, X_{t+1} = s_j \mid \theta) \tag{2.5.1.5}$$

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \Pr(y_1, y_2, ..., y_t, X_t = s_i \mid \theta) \Pr(X_{t+1} = s_j \mid X_t = s_i, \theta) \right] \Pr(y_{t+1} \mid X_{t+1} = s_j, \theta) \tag{2.5.1.6}$$

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i)\, a_{ij} \right] b_j(y_{t+1}), \quad t \in \{1, 2, ..., T\text{-}1\}, \; j \in \{1, 2, ..., N\} \tag{2.5.1.7}$$

If we denote $\alpha_i(y_1^t, \theta)$ and as the elements of $\boldsymbol{\alpha}(y_1^t, \theta)$, then in matrix form it can be written as

$$\boldsymbol{\alpha}(y_1^t, \theta) = \boldsymbol{\alpha}(y_1^{t-1}, \theta)\mathbf{P}(y_t, \theta) \qquad t = 1, 2, ..., T \tag{2.5.1.8}$$

where $\mathbf{P}(y_t, \theta)$ is a function of the state transition matrix and the output symbol probability matrix, $\mathbf{B}$, for the current input sequence.

The computation of the forward variable $\alpha_{t+1}(j)$ in the induction step accounts for all possible transitions to state s_j from time step *t* to *t* +1, and the observable y_{t+1} at time *t* +1. Figure 2.5.1 shows the graphical computation for $\boldsymbol{\alpha}$. The forward variable $\boldsymbol{\alpha}$ is

usually stored as a row vector that needs to be normalized. The normalization can be done by dividing the elements of the forward variable by its sum for a particular time instant.

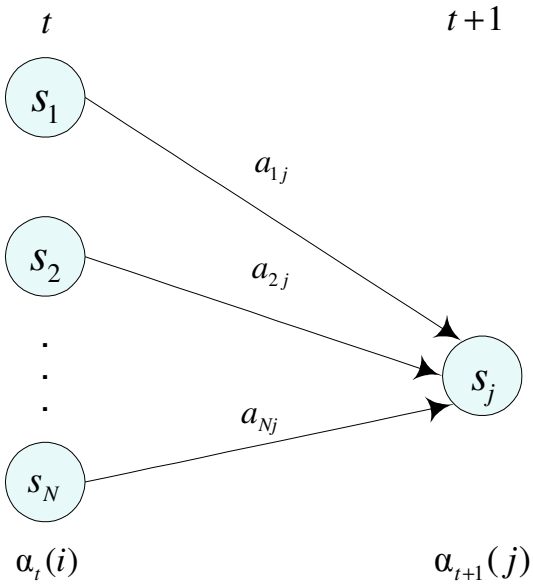


Figure 2.5.1.1: Computation of forward variable $\boldsymbol{\alpha}$ in the forward-backward BWA.

We now define a backward variable $\beta_t(i)$ as

$$\beta_t(i) \triangleq \Pr(y_{t+1}, y_{t+2}, ..., y_T \mid X_t = s_i, \theta) \tag{2.5.1.9}$$

which denotes the probability of the partial observation $y_{t+1}, y_{t+2}, ..., y_T$ given that at time step t the model is in state s_i and given the model θ. The initialization is done

$$\beta_T(i) \triangleq 1, \qquad i \in \{1, 2, ..., N\} \tag{2.5.1.10}$$

and the matrix form is written as

$$\boldsymbol{\beta}(y_T^t, \theta) \triangleq \mathbf{1} \tag{2.5.1.11}$$

The induction is done as

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(y_{t+1}) \beta_{t+1}(j), \qquad t = \{T-1, T-2, ..., 1\}, i = \{1, 2, ..., N\} \tag{2.5.1.12}$$

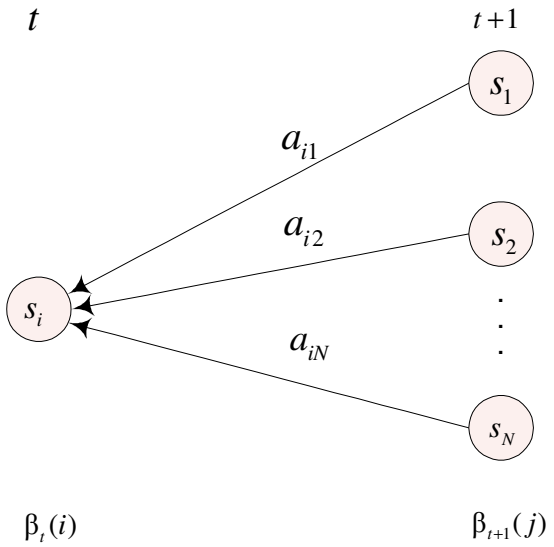


Figure 2.5.1.2: Computation of backward variable $\boldsymbol{\beta}$ in the forward-backward BWA.

If we denote $\beta_i(y_T^t,\theta)$ as the elements of $\boldsymbol{\beta}(y_T^t,\theta)$, then we can use the matrix form and write

$$\boldsymbol{\beta}(y_t^T,\theta)=\mathbf{P}(y_t,\theta)\,\boldsymbol{\beta}(y_{t+1}^T,\theta) \qquad t=T,T-1,...,1 \tag{2.5.1.13}$$

where $\mathbf{P}(y_t,\theta)$ is a function of the state transition matrix and the output symbol probability matrix $\mathbf{B}$ for the current input sequence. This backward variable is a column vector and it needs to be scaled to avoid computational underflow. The scaling can be performed by dividing the elements of the backward variable by the scaling factor that was obtained by adding the forward variables for the corresponding time index. Scaling is discussed in the next section. Note that

$$\Pr\left(y_1^T\mid\theta\right)=\sum_{i=1}^{N}\alpha_t(i)\beta_t(i) \qquad t\in\{1,2,...,T\} \tag{2.5.1.14}$$

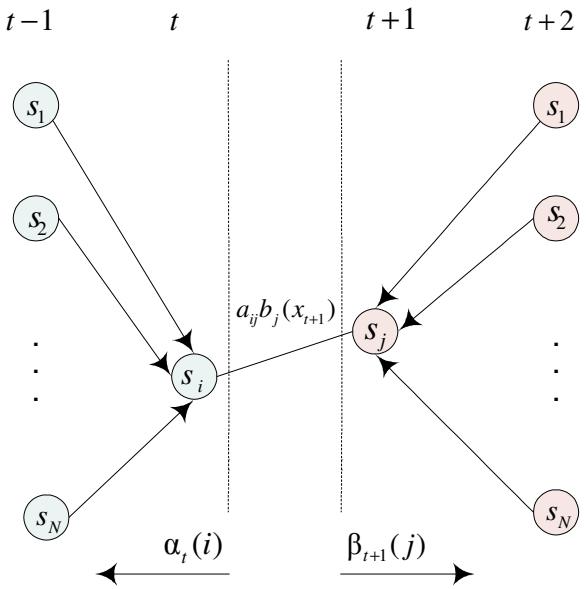


Figure 2.5.1.3: Parameter estimation using forward and backward variables in BWA.

The estimated parameters of the forward-backward BWA are given as

$$p_{ij} = \frac{\text{expected number of transitions from state } s_i \text{ to state } s_j}{\text{expected number of transitions from state } s_i} \tag{2.5.1.15}$$

$$b_j(k) = \frac{\text{expected number of times in state } s_j \text{ and observing symbol } k}{\text{expected number of times in state } s_j} \tag{2.5.1.16}$$

$$\pi_i = \text{expected number of times in state } s_i \text{ at time } t = 1 \tag{2.5.1.17}$$

The probability of obtaining a sequence given the model is

$$\Pr\left(y_1^T \mid \theta\right) = \sum_{i=1}^{N} \alpha_t(i)\beta_t(i) \qquad t \in \{1,2,...,T\} \tag{2.5.1.18}$$

The parameters are estimated as follows

$$p_{ij_{p+1}} = \frac{\sum_{t=1}^{T} \alpha_{t_p}(i)\, p_{ij_p} b_{j_p}(x_{t+1})\beta_{t+1_p}(j)}{\sum_{t=1}^{T} \alpha_{t_p}(i)\beta_{t_p}(i)} \tag{2.5.1.19}$$

$$b_{j_{p+1}}(k) = \frac{\sum_{\substack{t \\ \forall x_t = k}} \alpha_{t_p}(j)\beta_{t_p}(j)}{\sum_{t=1}^{T} \alpha_{t_p}(j)\beta_{t_p}(j)} \qquad (2.5.1.20)$$

$$\pi_i = \alpha_1(i)\beta_1(i) \qquad (2.5.1.21)$$

2.5.2 Scaling of the Forward and Backward Variables

Most of the earlier literature dedicated to estimating the parameters of HMMs using the BWA do not address the issue of scaling the Markov model parameters. The forward and backward vectors tend to zero exponentially especially for large data size and numerical underflow results if these variables are not properly scaled. We can scale the variables if we define

$$\bar{\alpha}_t(j) = \frac{\alpha_t(j)}{\sum_{i=1}^{N} \alpha_t(i)} \qquad (2.5.2.1)$$

and hence define the scaling vector C_t as

$$C_t = \sum_{i=1}^{N} \alpha_t(i) \qquad (2.5.2.2)$$

This implies that $$\sum_{i=1}^{N} \bar{\alpha}_t(i) = 1 \qquad (2.5.2.3)$$

and $$\bar{\beta}_t(i) = \frac{\beta_t(i)}{C_t} \qquad (2.5.2.4)$$

with the initialization of $$\bar{\beta}_T = \frac{\mathbf{1}}{C_T} \qquad (2.5.2.5)$$

where $\mathbf{1}$ denotes the column vector containing all ones. This method avoids numerical underflow in both the forward and the backward variable. Refer to [4] for more detail about scaling.

2.5.3 The Forward-only BWA for HMMs

In the forward-backward implementation, the algorithm requires the amount of memory that is proportional to the data size T, which can be very large. In contrast, unidirectional algorithms require an amount of memory that is independent of the data size and it also uses less memory than the forward-backward Baum-Welch algorithm, primarily because of the fact that the backward variable $\boldsymbol{\beta}$ is not used during re-estimation. We only need to keep the forward variable in memory and this also makes the model easily programmable in a DSP chip. The forward-only Baum-Welch algorithm can be applied online without waiting until the entire sequence of length T has been received. We can also discard previous sequences, and then estimate the model from the estimated parameters obtained from those sequences. Let us now represent the mathematical expressions for estimating the parameters of the forward-only BWA.

From the forward-backward BWA, it is evident that we have to compute sums of the form

$$S(y_1^T) = \sum_{t+1}^{T} \boldsymbol{\alpha}(y_1^T)\mathbf{U}(t, y_t)\boldsymbol{\beta}(y_{t+1}^T) \tag{2.5.3.1}$$

where $\mathbf{U}(t, y_t)$ is a matrix. We will try to find efficient ways of representing these matrices. In the forward-backward BWA the standard form for determining these matrices is used. It follows from the forward and backward probability definitions that

$$S = \boldsymbol{\pi}\, \mathbf{S}(y_1^T)\mathbf{1} \tag{2.5.3.2}$$

where

$$\mathbf{S}(y_1^T) = \sum_{t=1}^{T} \mathbf{P}(y_1^t)\mathbf{U}(t, y_t)\mathbf{P}(y_{t+1}^T) \tag{2.5.3.3}$$

and

$$\mathbf{P}(y_u^v) = \prod_{i=u}^{v} \mathbf{P}(y_i) \tag{2.5.3.4}$$

are the corresponding matrix probabilities. For $u \le v \le w$

$$\mathbf{P}(y_u^w) = \mathbf{P}(y_u^v)\mathbf{P}(y_v^w) \tag{2.5.3.5}$$

$$\mathbf{S}(y_u^w) = \mathbf{S}(y_u^v)\mathbf{P}(y_{v+1}^w) + \mathbf{P}(y_u^v)\mathbf{S}(y_{v+1}^w) \tag{2.5.3.6}$$

These equations form the basis for the recursive algorithm. Remember that the forward variable in the forward-backward BWA is computed using

$$\boldsymbol{\alpha}(y_1^{t_{k+1}}) = \boldsymbol{\alpha}(y_1^{t_k})\mathbf{P}(y_{t_k+1}^{t_{k+1}}) \tag{2.5.3.7}$$

If we define $$\boldsymbol{s}(y_1^{t_{k+1}}) = \boldsymbol{s}(y_1^{t_k})\mathbf{P}(y_{t_k+1}^{t_{k+1}}) + \boldsymbol{\alpha}(y_1^{t_k})\mathbf{P}(y_{t_k+1}^{t_{k+1}}) \quad (2.5.3.8)$$

where $$\boldsymbol{s}(y_1^t) = \boldsymbol{\pi}\mathbf{S}(y_1^t) \quad (2.5.3.9)$$

and if $t_k = t$ and $t_{k+1} = t_k$, we have the following

$$\boldsymbol{\alpha}(y_1^{t+1}) = \boldsymbol{\alpha}(y_1^t)\mathbf{P}(y_{t+1}) \quad (2.5.3.10)$$

$$\boldsymbol{s}(y_1^{t+1}) = \boldsymbol{s}(y_1^t)\mathbf{P}(y_{t+1}) + \boldsymbol{\alpha}(y_1^t)\mathbf{U}(t, y_{t+1}) \quad (2.5.3.11)$$

The numerator of the equation

$$p_{ij_{p+1}} = \frac{\sum_{t=1}^{T} \alpha_{t_p}(i)\, p_{ij_p} b_{j_p}(y_{t+1}) \beta_{t+1_p}(j)}{\sum_{t=1}^{T} \alpha_{t_p}(i) \beta_{t_p}(i)} \quad (2.5.3.12)$$

can be written $$S(y_1^T) = \sum_{t=1}^{T} \boldsymbol{\alpha}(y_1^{t-1})\mathbf{U}_{ij}\, p_{ij} b_{j,p}(y_t)\boldsymbol{\beta}(y_{t+1}^T) \quad (2.5.3.13)$$

where $\mathbf{U}_{ij}$ is the $N \times N$ matrix whose ij-th element is one, and the rest of its elements are zeros. It follows from this equation that

$$\boldsymbol{s}_{ij}(y_1^{t+1}) = \boldsymbol{s}_{ij}(y_1^t)\mathbf{P}(y_{t+1}) + \boldsymbol{\alpha}(y_1^t)\mathbf{U}_{ij}\, p_{ij,p} b_{j,p}(y_t) \quad (2.5.3.14)$$

The numerator of equation

$$b_{j_{p+1}}(k) = \frac{\sum_{\substack{t \\ \forall x_t = k}} \alpha_{t_p}(j)\beta_{t_p}(j)}{\sum_{t=1}^{T} \alpha_{t_p}(j)\beta_{t_p}(j)} \quad (2.5.3.15)$$

can be written as $$S_j(y) = \sum_{t=1}^{T} \boldsymbol{\alpha}(y_1^t)\mathbf{U}_{jj}\delta(\mathrm{y}, y_t)\boldsymbol{\beta}(y_{t+1}^T) \quad (2.5.3.16)$$

where $\delta(y, y_t) = 1$, if $y = y_t$, and $\delta(y, y_t) = 0$ if $y \neq y_t$. Hence we can write

$$\mathbf{s}_j(y_1^{t+1}, y) = \mathbf{s}_j(y_1^t, y)\mathbf{P}(y_{t+1}) + \boldsymbol{\alpha}(y_1^t)\delta(y, y_{t+1})\mathbf{U}_{ii} \quad (2.5.3.17)$$

Also $$\mathbf{s}_{ij}(y_1^{t+1}) = \mathbf{s}_{ij}(y_1^t)\mathbf{P}(y_{t+1}) + \boldsymbol{\alpha}(y_1^t)\mathbf{U}_{ij}\, p_{ij,p} b_{j,p}(y_t) \quad (2.5.3.18)$$

$$\mathbf{s}_j(y_1^{t+1}, y) = \mathbf{s}_j(y_1^t, y)\mathbf{P}(y_{t+1}) + \boldsymbol{\alpha}(y_1^t)\delta(y, y_{t+1})\mathbf{U}_{ii} \quad (2.5.3.19)$$

are the equations that replace the forward-backward BWA.

Note that we do not need to keep any of these variables in memory. Equations (2.5.3.18) and (2.5.3.19) are the only equations used in the parameter estimation of a general HMM using the forward-only BWA. The initial conditions for the algorithm are

$$\boldsymbol{s}_{ij}(y_1^1) = \mathbf{0} \tag{2.5.3.20}$$

$$\boldsymbol{s}_{j}(y_1^1, y) = \mathbf{0} \tag{2.5.3.21}$$

The parameters of this forward-only BWA are estimated as follows

$$p_{ij,p+1} = \frac{\boldsymbol{s}_{ij}(y_1^T)\mathbf{1}}{\sum_j \boldsymbol{s}_{ij}(y_1^T)\mathbf{1}} \tag{2.5.3.21}$$

$$b_{j,p+1}(y) = \frac{\sum_j \boldsymbol{s}_{j}(y_1^T)\mathbf{1}}{\sum_j \mathbf{b}_{j,p+1}(y)} \tag{2.5.3.22}$$

Note that by using only the $\boldsymbol{\alpha}$ variable for parameter estimations and not using any backward path reduces memory requirements significantly, but it might sometime give sub-optimal results.

2.5.4 The Forward-backward BWA for Semi-HMMs

In this section we discuss a modified version of the forward-backward BWA that is used to estimate the parameters of semi-HMMs. The state transition matrix of a semi-HMM can be written as

$$\mathbf{P} = \begin{bmatrix} \mathbf{P}_{11} & \mathbf{P}_{12} & \dots & \mathbf{P}_{1s} \\ \mathbf{P}_{21} & \mathbf{P}_{22} & \dots & \mathbf{P}_{2s} \\ \vdots & & & \\ \mathbf{P}_{s1} & \mathbf{P}_{s2} & \dots & \mathbf{P}_{ss} \end{bmatrix} \tag{2.5.4.1}$$

where s is the total number of observation symbols in the process. In case of error source modeling, s takes only two values, either 0 or 1. The input sequence is in the form of a runlength vector and hence the length of the input sequence is greatly reduced, especially when we have sequences involving long stretches of identical symbols. Refer to section 2.4 for details about the runlength vector. The dimensions of the individual sub-matrices in the diagonal depend upon the number of all the states that can generate that particular

observation symbol. The dimension of off-diagonal sub-matrices depends upon the number of states that generate the current observation symbol, and the number of states that can generate the next observation symbol.

Following the notations given in [4], define the fast forward variable $\boldsymbol{\alpha}(v_1^n, \tau)$ as

$$\boldsymbol{\alpha}(v_1^n, \tau) = \boldsymbol{\pi} \prod_{i=1}^{n_{t-1}} \mathbf{P}^{m_i}(X_i, \tau) \tag{2.5.4.2}$$

where, $\boldsymbol{\pi}$ is the initial state probability vector,

X_i is the observation symbol, 0 or 1 in case of error sequences,

m is the number of times a particular symbol is observed,

τ is the model parameter vector.

The fast forward variable $\boldsymbol{\alpha}(v_1^n, \tau)$ is a row vector and is obtained in the vector form during the implementation of the algorithm. Define

$$\boldsymbol{\alpha}_0(v_1^t, \tau) = \pi_{X_1} \prod_{i=1}^{t-1} [P_{X_i X_i}^{m_i - 1} P_{X_i X_{i+1}}] \tag{2.5.4.3}$$

and

$$\boldsymbol{\alpha}(v_1^t, \tau) = \pi_{X_1} \prod_{i=1}^{t-1} [P_{X_i X_i}^{m_i - 1} P_{X_i X_{i+1}}] P_{X_i X_i}^{m_i - 1} \tag{2.5.4.4}$$

and hence develop an algorithm for computing the fast forward variable

$$\begin{aligned} \boldsymbol{\alpha}_0(v_1^0, \tau) &= \boldsymbol{\pi}_{X_1} \\ \boldsymbol{\alpha}(v_1^t, \tau) &= \boldsymbol{\alpha}_0(v_1^t, \tau) \mathbf{P}_{X_i X_i}^{m-1} \\ \boldsymbol{\alpha}_0(v_1^{t+1}, \tau) &= \boldsymbol{\alpha}(v_1^t, \tau) \mathbf{P}_{X_i X_{i+1}} \end{aligned} \tag{2.5.4.5}$$

Note that all these variables are row vectors. The dimension of these variables at a particular time depends upon the observation symbols observed at that time and the number of states that can generate that observation symbol. For example, if at time t a symbol 1 is observed and there are k states that can output a symbol 1, then the dimension of the fast forward variable is k. Hence, the fast forward variable is of the order $1 \times k$.

The backward variable $\boldsymbol{\beta}(v_n^N, \tau)$ is shown as

$$\boldsymbol{\beta}(v_n^N, \tau) = \prod_{i-n_t}^{N} \mathbf{P}^{m_i}(X_i, \tau) \mathbf{1} \tag{2.5.4.6}$$

Note that this is a column vector. Define

$$\boldsymbol{\beta}_0(v_n^N,\tau)=\prod_{i=t}^{T}[\mathbf{P}_{X_{i-1}X_i}\mathbf{P}_{X_iX_i}^{m_i-1}]\mathbf{1} \tag{2.5.4.7}$$

$$\boldsymbol{\beta}(v_t^N,\tau)=\mathbf{P}_{X_tX_t}^{m_t-1}\prod_{i=t+1}^{T}[\mathbf{P}_{X_{i-1}X_i}\mathbf{P}_{X_iX_i}^{m_i-1}]\mathbf{1} \tag{2.5.4.8}$$

Hence we derive the backward algorithm

$$\begin{aligned}&\boldsymbol{\beta}_0(v_{N+1}^N,\tau)=\mathbf{1}\\&\boldsymbol{\beta}(v_t^N,\tau)=\Pr_{X_tX_t}^{m_t-1}\boldsymbol{\beta}_0(v_{t+1}^N,\tau)\\&\boldsymbol{\beta}_0(v_t^N,\tau)=\Pr_{X_tX_{t+1}}\boldsymbol{\beta}_0(v_{t+1}^N,\tau)\end{aligned} \tag{2.5.4.9}$$

Note that all of the above variables are in the form of a column vector. The dimension of these variables at a particular time depends upon the observation symbols observed at that time and the number of states that can generate that observation symbol. For example, if at time t a symbol 1 is observed and there are k states that can output a symbol 1, then the dimension of the fast backward variable is k and the order is $k\times 1$. More detail about this algorithm can be obtained from [3] and [4].

In order to estimate the model Λ, define

$$\Gamma_t(i,j)=\Pr(q_t=i,q_{t+1}=j\mid O,\Lambda) \tag{2.5.4.10}$$

where O denotes the observation sequence. This can be written in terms of the forward and backward variables as

$$\Gamma_{\boldsymbol{\mu}_c\boldsymbol{\mu}_{c+1},c}=\frac{\alpha_{c|\boldsymbol{\mu}_c}\Lambda_{\boldsymbol{\mu}_c\boldsymbol{\mu}_{c+1}}^{m(\boldsymbol{\mu}_{c+1})-1}\beta_{c+1|\boldsymbol{\mu}_{c+1}}}{\Pr(O\mid\Lambda)} \tag{2.5.4.11}$$

$\Gamma_{\varepsilon\boldsymbol{\mu}}=[0]$, unless $\boldsymbol{\varepsilon}=\boldsymbol{\mu}_c$ and $\boldsymbol{\mu}=\boldsymbol{\mu}_{c+1}$

Also

$$\Gamma_{\boldsymbol{\mu}_c\boldsymbol{\mu}_c,c}=\frac{(m(\boldsymbol{\mu}_c)-1)\alpha_{c|\boldsymbol{\mu}_c}\Lambda_{\boldsymbol{\mu}_c\boldsymbol{\mu}_c}^{m(\boldsymbol{\mu}_c)-1}\beta_{c+1|\boldsymbol{\mu}_{c+1}}}{\Pr(O\mid\Lambda)} \tag{2.5.4.12}$$

where [0] denotes the matrix containing all zeros. In simulations, the probability $P(O\mid\Lambda)$ is not necessarily required.

The expected number of transitions from i to j is $\sum_{t=1}^{C-1}\Gamma_t(i,j)$, and this can be used directly to estimate the state transition matrix $\mathbf{P}$. Note that the elements of transition

matrix can be estimated in the form of blocks, and hence the simulation runtime can be significantly reduced.

2.5.5 The Forward-only BWA for Semi-HMMs

We can use modifications in the algorithm given in [3] and [4] to obtain the forward-only version of the BWA for estimating semi-HMM parameters. This algorithm does not need the fast backward variable and hence it is capable of estimating the parameters of Markov models *on the fly*. As this model uses the runlength vector as input, it is fast as compared to the models that work on symbol-by-symbol basis. The forward-only estimation property also allows it to be easily programmed in a DSP chip for real time applications.

Consider the state transition matrix that was discussed for forward-backward semi-Markov models. The matrix has the form

$$\mathbf{P} = \begin{bmatrix} \mathbf{P}_{11} & \mathbf{P}_{12} & \dots & \mathbf{P}_{1s} \\ \mathbf{P}_{21} & \mathbf{P}_{22} & \dots & \mathbf{P}_{2s} \\ \vdots & & & \\ \mathbf{P}_{s1} & \mathbf{P}_{s2} & \dots & \mathbf{P}_{ss} \end{bmatrix} \tag{2.5.5.1}$$

where s is the total number of observation symbols in the process. Define the fast forward variable $\boldsymbol{\alpha}(v_1^n, \tau)$ as

$$\boldsymbol{\alpha}(v_1^n, \tau) = \boldsymbol{\pi} \prod_{i=1}^{n_{t-1}} \mathbf{P}^{m_i}(X_i, \tau) \tag{2.5.5.2}$$

where $\boldsymbol{\pi}$ is the initial state probability vector

X_i is the observation symbol, 0 or 1 in case of error sequences

m is the number of times a particular symbol is observed

The fast forward variable $\boldsymbol{\alpha}(v_1^n, \tau)$ is a row vector and is obtained in the vector form during the implementation. Define

$$\boldsymbol{\alpha}_0(v_1^t, \tau) = \pi_{X_1} \prod_{i=1}^{t-1} [\Pr_{X_i X_i}^{m_i - 1} \Pr_{X_i X_{i+1}}] \tag{2.5.5.3}$$

and $$\boldsymbol{\alpha}(v_1^t, \tau) = \pi_{X_1} \prod_{i=1}^{t-1} [\Pr_{X_i X_i}^{m_i - 1} \Pr_{X_i X_{i+1}}] \Pr_{X_i X_i}^{m_i - 1} \tag{2.5.5.4}$$

and hence develop an algorithm for estimating the fast forward variable

$$\begin{aligned} \boldsymbol{\alpha}_0(v_1^0, \tau) &= \boldsymbol{\pi}_{X_1} \\ \boldsymbol{\alpha}(v_1^t, \tau) &= \boldsymbol{\alpha}_0(v_1^t, \tau) \mathbf{P}_{X_i X_i}^{m-1} \\ \boldsymbol{\alpha}_0(v_1^{t+1}, \tau) &= \boldsymbol{\alpha}(v_1^t, \tau) \mathbf{P}_{X_i X_{i+1}} \end{aligned} \tag{2.5.5.5}$$

Note that all of these variables are row vectors. The dimension of these variables at a particular time is the number of states in the model.

We define $\mathbf{s}_{ij}(\boldsymbol{x}_1^{t+1})$ as

$$\mathbf{s}_{ij}(\boldsymbol{x}_1^{t+1}) = \mathbf{s}_{ij}(\boldsymbol{x}_1^t) \mathbf{P}^{m_{t+1}} + \boldsymbol{\alpha}(\boldsymbol{x}_1^{t_k}) \mathbf{V} \tag{2.5.5.6}$$

where the matrix $\mathbf{V}$ is calculated using the following procedure

$$\begin{aligned} &b = \text{rem}(m,2) && \mathbf{V} = b\mathbf{W} \\ &\mathbf{R} = \mathbf{P} \qquad \mathbf{Q} = \mathbf{P}^b && \mathbf{G} = \mathbf{W} \\ &\text{while } m > 0 \\ &\quad \mathbf{G} = \mathbf{R}\,\mathbf{G} + \mathbf{G}\,\mathbf{R} && \mathbf{R} = \mathbf{R}^2 \\ &\quad m = (m - b)/2 && b = \text{rem}(m,2) \\ &\quad \mathbf{V} = \mathbf{R}^b\ \mathbf{V} + b\mathbf{G}\,\mathbf{Q} && \mathbf{Q} = \mathbf{Q}\,\mathbf{R}^b \\ &\text{end} \end{aligned} \tag{2.5.5.7}$$

This allows us to use fast matrix exponentiation which is very useful in reducing the runtime of the algorithm because the value of m could be very large, as the lengths between two consecutive bursts could be very long. The vector $\mathbf{s}$ is used to get the state transition matrix

$$\mathbf{P}(i, j) = \sum_{i=1}^{N} \mathbf{s}_{ij}(x_1^{t+1}) \qquad t = 1, 2, ..., T \tag{2.5.5.8}$$

This algorithm might give us sub-optimal results as we are trading off the speed for computational complexity.

2.5.6 Phase-Type (PH) Distribution Approximation

The Phase-type distribution or PH distribution [4] is used to estimate the fade duration [10] occurring in wireless channels. The PH distribution is a special case of the matrix-geometric distribution with the form

$$g(x,\tau) = \boldsymbol{\pi}\,\mathbf{A}^{x-1}\,\mathbf{b} \tag{2.5.6.1}$$

where $\boldsymbol{\pi}$ is a row vector, $\mathbf{b}$ is a column vector, and $\mathbf{A}$ is a $u \times u$ square matrix.

$$\mathbf{P} = \begin{bmatrix} \mathbf{A} & \mathbf{b} \\ \boldsymbol{\pi} & 0 \end{bmatrix} \tag{2.5.6.2}$$

is a stochastic matrix. By definition,

$$\boldsymbol{\pi}\mathbf{1} = 1 \tag{2.5.6.3}$$

$$\mathbf{b} = (\mathbf{I} - \mathbf{A})\mathbf{1} \tag{2.5.6.4}$$

In this section we will develop an EM algorithm for approximating a probability distribution $f(x)$ with phase-type (PH) distribution.

To develop the EM algorithm, the matrix product is written as

$$g(x,\tau) = \sum_{z} \pi_{i_0}\, a_{i_0 i_1} \ldots a_{i_{x-2} i_{x-1}}\, a_{i_{x-1},u+1} \tag{2.5.6.5}$$

where $a_{i,u+1} = b_i$ and $z = \boldsymbol{i}_0^{x-1}$. Define

$$\psi(z;x,\tau) = \pi_{i_0}\, a_{i_0 i_1} \ldots a_{i_{x-2} i_{x-1}}\, a_{i_{x-1},u+1} \tag{2.5.6.6}$$

so that

$$g(x,\tau) = \sum_{z} \psi(z;\tau) \tag{2.5.6.7}$$

Dividing both sides of this equation by $g(x,\tau)$, and defining $\kappa(z;x,\tau) = \psi(z;x,\tau) / g(x,\tau)$, it can be rewritten that

$$\sum_{z} \kappa(z;x,\tau) = 1 \tag{2.5.6.8}$$

In this way $\kappa(z;x,\tau)$ can be interpreted as a probability distribution.

Define $\rho(x,\tau_p) = f(x) / g(x,\tau)$, and replace the $\log \psi(z;x,\tau)$ as the sum of logarithms

$$\log \psi(z;x,\tau) = \log \tau_{i_0} + \sum_{k=1}^{x} \log a_{i_{k-1},i_k} \tag{2.5.6.9}$$

Note that $$\sum_{z,i_0=i} \psi(z;x,\tau) = \sum_{i_1^{x-1}} a_{i,i_1}\, a_{i_1,i_2} \ldots a_{i_{x-2},i_{x-1}}\, a_{i_{x-1},u+1} \tag{2.5.6.10}$$

is the $i-$th element of the matrix

$$\boldsymbol{\beta}(x,\tau)=\mathbf{A}^{x-1}\mathbf{b} \tag{2.5.6.11}$$

Since $$\sum_{i=1}^{u} \pi_i \beta_i(x,\tau) = \boldsymbol{\pi}\,\boldsymbol{\beta}(x,\tau) = \boldsymbol{\pi}\,\mathbf{A}^{x-1}\mathbf{b} = g(x,\tau) \tag{2.5.6.12}$$

we have $$\sum_x \rho(x,\tau)\, g(x,\tau) = \sum_x f(x) = 1 \tag{2.5.6.13}$$

and hence

$$\pi_{i,p+1} = \pi_{i,p} \sum_x \beta_i(x,\tau_p)\rho(x,\tau_p) \tag{2.5.6.14}$$

This result can be used in simulations directly. Referring to [4], the BWA algorithm for fitting the PH distribution to $f(x)$ is given as follows

$$\pi_{i,p+1} = \pi_{i,p} \sum_x \rho(x,\tau_p)\beta_i(x,\tau_p) \tag{2.5.6.15}$$

$$a_{ij,p+1} = a_{ij,p} \frac{\sum_x \rho(x,\tau_p) \sum_{k=1}^{x-1} \alpha_i(k-1,\tau_p)\beta_j(x-k,\tau_p)}{\sum_x \rho(x,\tau_p) \sum_{k=1}^{x} \alpha_i(k-1,\tau_p)\beta_i(x-k+1,\tau_p)} \tag{2.5.6.16}$$

$$b_{i,p+1} = b_{i,p} \frac{\sum_x \rho(x,\tau_p)\, \alpha_i(x-1,\tau_p)}{\sum_x \rho(x,\tau_p) \sum_{k=1}^{x} \alpha_i(k-1,\tau_p)\beta_j(x-k+1,\tau_p)} \tag{2.5.6.17}$$

Define the recursive forward algorithm as

$$\boldsymbol{\alpha}(0,\tau) = \boldsymbol{\pi} \qquad \boldsymbol{\alpha}(x,\tau) = \boldsymbol{\alpha}(x-1,\tau)\,\mathbf{A} \qquad x = 1,2,\ldots \tag{2.5.6.18}$$

and $\boldsymbol{\beta}(x,\tau)$ can be evaluated recursively by the backward algorithm

$$\boldsymbol{\beta}(1,\tau) = \mathbf{b} \qquad \boldsymbol{\beta}(x+1,\tau) = \mathbf{A}\boldsymbol{\beta}(x,\tau) \qquad x = 1,2,\ldots \tag{2.5.6.19}$$

All the elements of the forward variable $\boldsymbol{\alpha}(1)$, $\boldsymbol{\alpha}(2)$, ... , $\boldsymbol{\alpha}(x-1)$ and the backward variable $\boldsymbol{\beta}(1)$, $\boldsymbol{\beta}(2)$, ..., $\boldsymbol{\beta}(x-1)$ do not need to be saved if the data is processed by defining a matrix

$$\mathbf{W}(x,\tau) = \sum_{k=1}^{x-1} \mathbf{A}^{x-k-1}\mathbf{b}\,\boldsymbol{\pi}\,\mathbf{A}^{k-1} = \sum_{k=1}^{x} \boldsymbol{\beta}(x-k,\tau)\,\boldsymbol{\alpha}(k-1,\tau) \tag{2.5.6.20}$$

$\boldsymbol{\beta}(x-k,\tau)\boldsymbol{\alpha}(k-1,\tau)$ is a square matrix whose $ij-$th element is $\alpha_j(k-1,\tau)\beta_i(x-k,\tau)$. We can write

$$\mathbf{W}(x+1,\tau)=\sum_{k=1}^{x}\mathbf{A}^{x-k}\mathbf{b}\boldsymbol{\pi}\mathbf{A}^{k-1}=\mathbf{A}\mathbf{W}(x,\tau)+\mathbf{b}\,\boldsymbol{\alpha}(x-1,\tau) \tag{2.5.6.21}$$

Hence the following forward-only algorithm can be obtained

$$\boldsymbol{\alpha}(0,\tau)=\boldsymbol{\pi} \qquad \boldsymbol{\beta}(1,\tau)=\mathbf{b} \qquad \mathbf{W}(1,\tau)=\mathbf{0} \tag{2.5.6.22}$$

Next we compute

$$A_{ij}(x,\tau_p)=a_{ij,p}w_{ji}(x,\tau_p) \tag{2.5.6.23}$$

and

$$\rho(x,\tau)=\frac{f(x)}{g(x,\tau)} \tag{2.5.6.24}$$

with

$$g(x,\tau)\ =\boldsymbol{\pi}\boldsymbol{\beta}(x,\tau) \tag{2.5.6.25}$$

We then update the sum of the estimated parameters using (2.5.6.15), (2.5.6.16) and (2.5.6.17). Note that $\alpha_j(k-1,\tau)\beta_i(x-k,\tau)$ can be replaced with $A_{ij}(x,\tau)$, and this result can be easily used in simulations to estimate $a_{ij,p+1}$.

Recalculate

$$\begin{aligned}&\mathbf{W}(k+1,\tau)=\mathbf{A}\mathbf{W}(k,\tau)+\mathbf{b}\,\boldsymbol{\alpha}(k-1,\tau)\\&\boldsymbol{\alpha}(k,\tau)=\boldsymbol{\alpha}(k-1,\tau)\,\mathbf{A}\quad k=1,2,...,x-1\\&\boldsymbol{\beta}(k+1,\tau)=\mathbf{A}\,\boldsymbol{\beta}(k,\tau)\end{aligned} \tag{2.5.6.26}$$

and hence estimate the model parameters as

$$\begin{aligned}a_{ij,p+1}&=C_i\sum_{m=1}^{T}A_{ij}(x_m,\tau_p)/\,g(x_m,\tau_p)\\b_{i,p+1}\ &=C_i\sum_{m=1}^{T}\alpha_i(x_m-1,\tau_p)/\,g(x_m,\tau_p)\end{aligned} \tag{2.5.6.27}$$

where C_i is the normalization factor given as

$$C_i^{-1}=\sum_{m=1}^{T}\left[A_{ij}(x_m,\tau_p)/\,g(x_m,\tau_p)+\alpha_i(x_m-1,\tau_p)/\,g(x_m,\tau_p)\right] \tag{2.5.6.28}$$

The memory requirements for this algorithm are very small because there is no need to keep track of forward and backward paths.

2.6 Other Direct Maximization Techniques

Some other methods for estimating HMM parameters use direct maximization of likelihood process using gradient search techniques. Some of these techniques are momentum smoothing, adaptive step size, conjugate gradients and quasi-Newton methods. The basic idea of these technique is that since likelihood of an HMM can be represented using a polynomial equation, these methods try to maximize the probability by selecting the optimal model parameters. It should be noted that none of these methods guarantees to achieve the global maxima in the parameter searching. The primary reason of not using them in our research is because of the typically long time series that we usually encounter in wireless communications.

Chapter 3

Order Estimation of Discrete Markov Channel Models

Even though the estimation of the parameters HMMs has been studied extensively, a consistent estimation of the number of hidden states still remains an unsolved problem [18], [19], and [27]. Recent advances in the order estimation of HMMs have given great hope in solving this problem in a general HMM setting. In this chapter, we discuss the order estimation problem of binary HMMs, which we encounter in wireless channel modeling, using a simulation based approach. This approach is adopted because of the lack of mathematical theory on the consistent estimation of the number of hidden states of an HMM. We first discuss the estimation of the number of states of MCs, since these two problems are closely related to one another. The chapter is organized as follows. First we formulate the order estimation of Markov models as a classical composite hypothesis testing problem. Next we discuss the likelihood (and also the log-likelihood) of binary MCs and HMMs. The Bayesian Information Criterion (BIC) and the Akaike's Information Criterion (AIC), that are used most extensively in solving this problem, are discussed later. Then we discuss the identifiability problem that we face in the case of binary HMM order estimation. The autocorrelation function, used as a performance evaluation method of different Markov models, is derived and some important information theoretic measures are discussed next. Simulation results are presented in the last section.

3.1 Hypothesis Testing in Markov Order Estimation

The order estimation of Markov processes is influenced by the theory of composite hypothesis testing. In this section, we give a brief overview of its theory and its use in estimating the order of different Markov models.

In Markov order estimation, we usually face the following composite hypothesis testing problem. The hypothesis H_0 is normally defined as the hypothesis that the process belongs to the set of models with order r_0, denoted as M^{r_0} against the hypothesis H_1 that a process belongs to the set of all models with orders other than r_0, here denoted as r with $r > r_0$. This can mathematically be written as $(\cup_r M^r) \setminus M^{r_0}$. Thus

H_0: The source belong to M^{r_0}

H_1: The source belongs to $(\cup_r M^r) \setminus M^{r_0}$

Here $M^{r_0} \subseteq M^r$. We usually think of having a set with all possible models of orders less than a specific order. We start with the lowest possible order and the test gradually rejects the order that fails in favor of the remaining set of the models. The test on the samples of length n is a function that maps this process on $\{0, 1\}$. If the function is 1, the test rejects H_0 in favor of H_1, otherwise the test does not reject H_0. The region K_n on which the test rejects H_0 is called the *critical region*.

When dealing with the composite hypothesis testing of Markov models, the *n*-th likelihood ratio test between M^{r_0} and $M^r \setminus M^{r_0}$ has a critical region defined as [20]

$$K_n = \left\{ y_1^n : \log \frac{\sup_{P \in M^r} \Pr(y_1^n)}{\sup_{P \in M^{r_0}} \Pr(y_1^n)} \leq \text{pen}(n) \right\} \tag{3.1.1}$$

where pen(*n*) is an *n*-dependent threshold and $\Pr(y_1^n)$ is the likelihood of process y_1^n given the model. The term pen(*n*) must be positive to give a non-trivial test level.

3.2 Binary Markov Chain Order Estimation

In this section we present the theory of Markov chain order estimation using the goodness of fit approach. We will focus on binary error sequences, i.e., sequences represented by zeros and ones, in which a binary 1 denotes a transmission error and a 0 denotes no error. Mathematically, an error sequence $e_0, e_1, ..., e_n$ with order k has a conditional probability of an error at the time n that can be written as

$$\Pr(e_n \mid e_0, e_1, ..., e_{n-1}) = \Pr(e_n \mid e_{n-k}, ..., e_{n-1}) \tag{3.2.1}$$

Here the states are defined as k-tuples $S_n = (e_{n-k+1}, e_{n-k+2}, ..., e_n)$ and the state transition matrix for a k-th order Markov chain has $2^k \times 2^k$ dimensions. In general, the multiple order Markov chain of order k having n symbols has $n^k \times n^k$ dimensions and in our case of binary sequences $n = 2$.

The states have physical meanings since different basis sequences constitute different sets of states. As the order of a Markov chain increases, the basis sequences that constitute these models also increases and this increases the number of states. However, note that increasing the order of a Markov chain exponentially increases the chain complexity.

In Markov model building we usually use two test criteria, namely the χ^2 criterion and the information criterion I. If $n(e_1, e_2, ..., e_{k+1})$ denotes the frequency of transitions of error patterns $e_1, e_2, ..., e_{k+1}$ and $\hat{p}_{e_1 e_2 ... e_{k+1}}$ are the corresponding probabilities, then we can write

$$I = \sum_{e_1, e_2, ..., e_{k+1}} n(e_1, e_2, ..., e_{k+1}) \ln(\hat{p}_{e_1 e_2 ... e_{k+1}} / \hat{p}_{e_2 e_3 ... e_{k+1}}) \tag{3.2.2}$$

Here I is asymptotically distributed as $0.5\chi^2$. The χ^2 criterion can be written as

$$\chi^2 = \sum_{e_1, e_2, ..., e_{k+1}} n^*(e_1, e_2, ..., e_{k+1})(\hat{p}_{e_1 e_2 ... e_{k+1}} - \hat{p}_{e_2 e_3 ... e_{k+1}})^2 / \hat{p}_{e_2 e_3 ... e_{k+1}} \tag{3.2.3}$$

It is shown in [4] that $2I$ and χ^2 are approximately equal for large data samples. Given an error sequence, the general steps for finding the optimum order of a Markov chain are as follows:

Start with the hypothesis that the sequence is obtained from a binary symmetric channel with no memory. Test this hypothesis against an alternate hypothesis that it is obtained from a first-order Markov chain. If the above hypothesis is rejected then test the hypothesis that the sequence is drawn from a fist-order Markov chain against the hypothesis that it is drawn from a second-order Markov chain. Continue testing until the correct order fits the experimental data reasonably well.

For finding the degrees of freedom, we refer to [4], [21] and [22]. According to [4] the total number of degrees of freedom is 2^k, where *k* denotes the order of Markov chains. [22] states that the testing of the null hypothesis that a chain is an *r*-th order Markov chain within the hypothesis that it is of *t*-th order is asymptotically χ^2 with $(2^{t+1}-2^t)-(2^{r+1}-2^r)$ degrees of freedom.

We now briefly mention the three possible test results that can be obtained using this goodness of fit approach. A test result is *pessimist* when the CDF of *I* is lower than the CDF of the theoretical $0.5\chi^2$. As shown in the Fig. 3.2.1, if we choose the information criterion to be 3, then the theoretical plot recommends selecting the corresponding order with approximately 88% confidence whereas the information criterion gives a confidence level of approximately 82%.

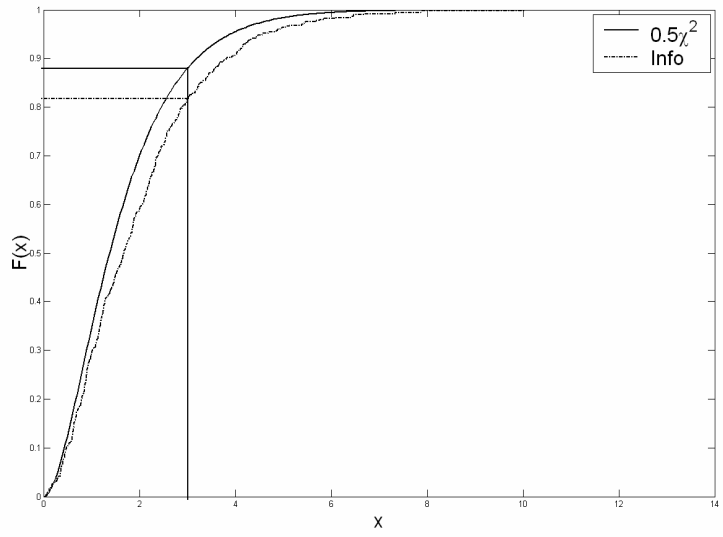


Figure 3.2.1: An example of pessimist test result of MC order estimation using goodness of fit approach. Here the CDF of *I* is lower than that of $0.5\ \chi^2$ plot.

In an *optimistic* case, the theoretical CDF is lower than the empirical CDF, as shown in Fig. 3.2.2. In this case, for an information criterion of 3 (on the *x*-axis), the theoretical $0.5\chi^2$ plot recommends to choose a confidence level of around 77% where from the information criterion we get around 82% confidence on our test.

In a *true* test, there is no difference in the CDFs of *I* and $0.5\chi^2$ plot and the two CDFs essentially overlap one another. This is shown in Fig. 3.2.3.

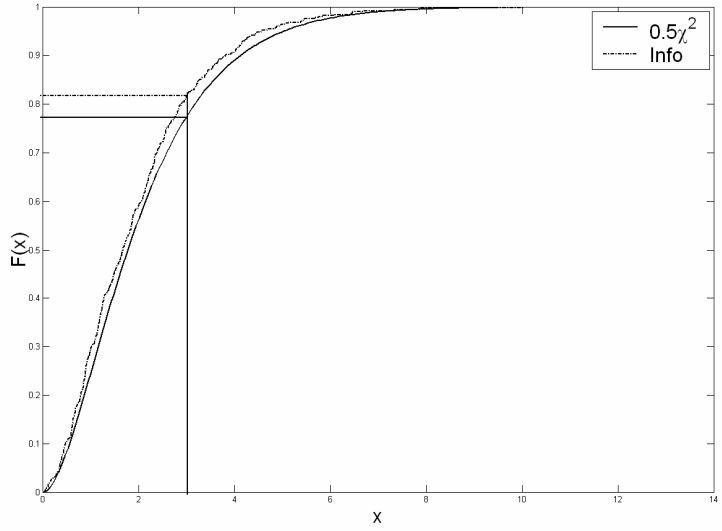


Figure 3.2.2: An example of optimist MC order estimation test result using goodness of fit approach. Here the theoretical CDF is lower than the empirical CDF.

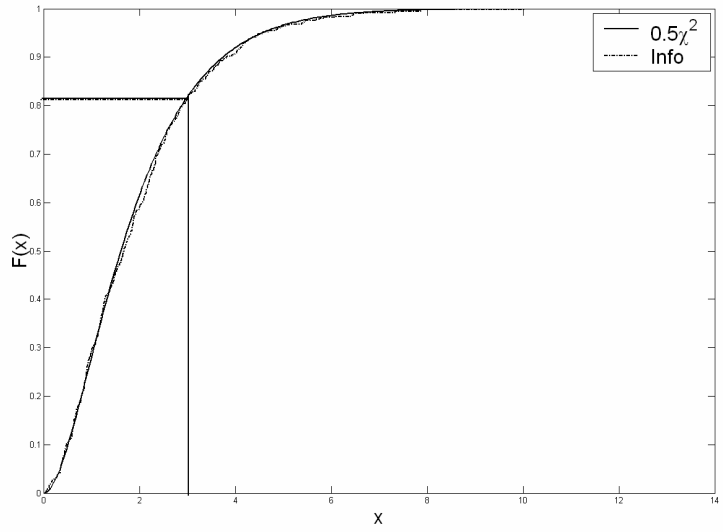


Figure 3.2.3: An example of true test MC order estimation result using goodness of fit. Here the two CDFs overlap one another.

3.3 Likelihood of Binary Markov Processes

If we denote by n_{ij} the number of transitions from state *i* to state *j* and a_{ij} as their corresponding probabilities, then the likelihood of process $e_1, e_2, ..., e_T$ (considering Markov chain modeling) can be written as

$$\Pr(e_1^T) = \sum_{i=1}^{N} \sum_{j=1}^{N} n_{ij} \log(a_{ij}) \tag{3.3.1}$$

This can be obtained by a simple computation and given an error sequence, the likelihood of MC of a certain order is found to be a fixed number, and can be obtained simply by assigning different basis sequences to different states and observing the number of times a particular state is visited. However, this is not the case with HMMs where the likelihood is calculated as,

$$\Pr(e_1^T) = \boldsymbol{\pi}\mathbf{B}(e_1)\ \mathbf{P}\mathbf{B}(e_2)\ \mathbf{P} \ldots \mathbf{P}\mathbf{B}(e_T)\mathbf{1}' \tag{3.3.2}$$

Here $\boldsymbol{\pi}$ is the initial distribution of states and based on the definition of output symbol probability matrix $\mathbf{B}$, we can write $\mathbf{B}(e_i) = \mathrm{diag}(\mathbf{b}_{e_i+1})$. A graphical illustration of the

BWA estimation of HMM parameters and its convergence on log-likelihood surface is shown in Figure 3.3.1.

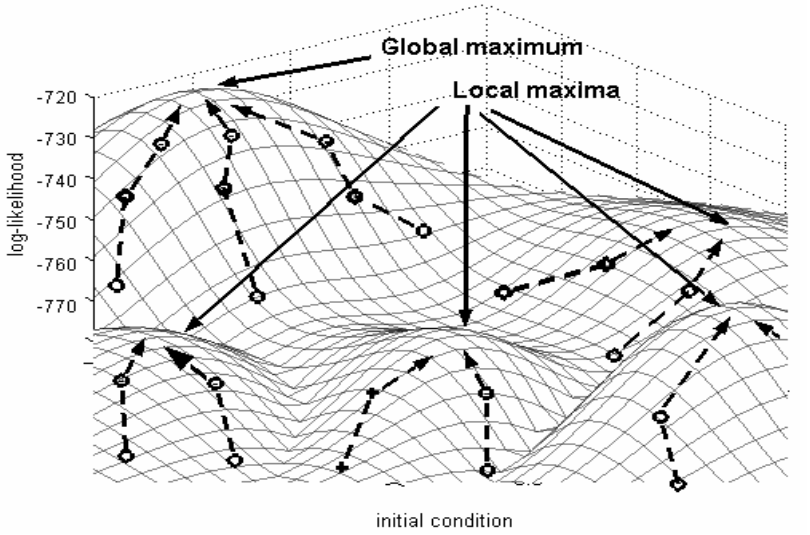


Figure 3.3.1: A conceptual profile of the log-likelihood surface during HMM parameter estimation using the BWA.

It is proved that the likelihood of a Markov process increases as we increase the number of states [8], [20]. This is always found to be true in all practical MC cases but not in HMM simulations as the Baum-Welch algorithm, used for estimating the parameters of HMMs, can be stuck at local maxima of likelihood, and the likelihood in this case could be less than the likelihood found for HMMs of smaller order. To resolve this problem, we run the BWA several times with different initial conditions and choose the model (of a certain order) that gives us the maximum value of the log-likelihood. However, note that this still doesn't guarantee to give us the optimum model (the global maximum) because there are infinite possibilities of initial conditions and checking all of them for the global maximum is quite impossible.

Due to large data size, the true likelihood of the process tends to zero and we use split data likelihood [23] to estimate the true value of likelihood. Because of extremely

low numerical values of these probabilities, we use the logarithm of the likelihood values. In this process, the observation sequence is segmented into blocks of equal length and these blocks are considered mutually independent. The overall likelihood of the observed process is then the product of individual likelihoods. Mathematically, for block size of m and a total of n blocks, we can write

$$\Pr(e_1^T \mid \theta) = \prod_{k=1}^{n} \Pr(e_{1+m(k-1)}^{mk} \mid \theta) \tag{3.3.2}$$

Fig. 3.3.2 shows the log-likelihood values of different HMMs that are obtained by running the BWA multiple times on an observation sequence with different initial conditions. The maximum value of the log-likelihood is also mentioned in the figure. We choose the model that gives this maximum value of the log-likelihood. In Figure 3.3.3, we present the improvements that can be achieved by running the BWA multiple times with different initial conditions. An error sequence is given as input to the BWA that is run with different initial conditions. Each initial condition produces an HMM and the optimum model is achieved by selecting the one that produces the maximum value of the log-likelihood. Another model is selected that produces the minimum value of the log-likelihood. The autocorrelation function of the input sequence is then compared with autocorrelations of these two HMMs. The autocorrelation of the optimum HMM is found to be much closer to the autocorrelation of the original sequence.

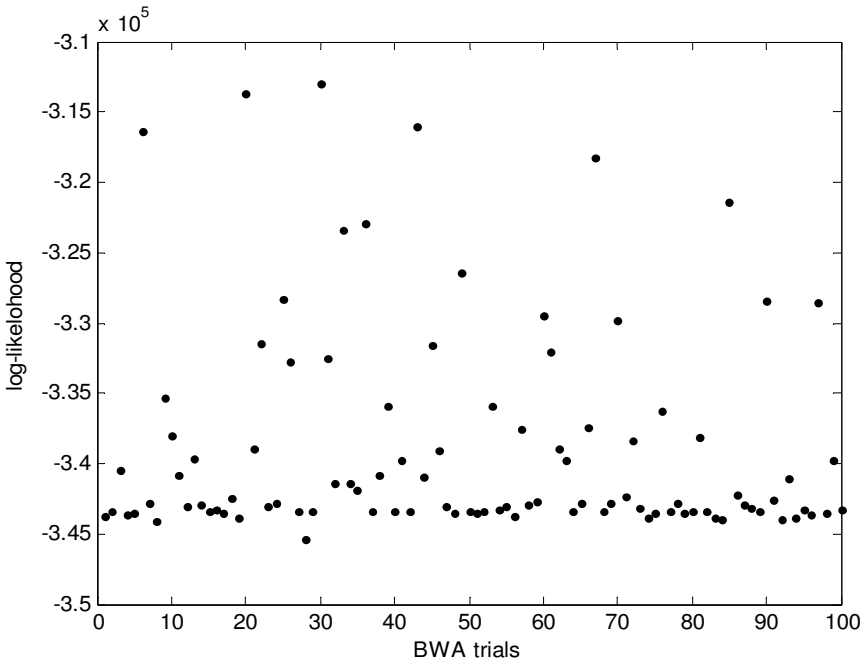


Figure 3.3.2: Log-likelihood values obtained by running the BWA 100 times. Significant improvement can be achieved by running the BWA multiple times with different initial conditions.

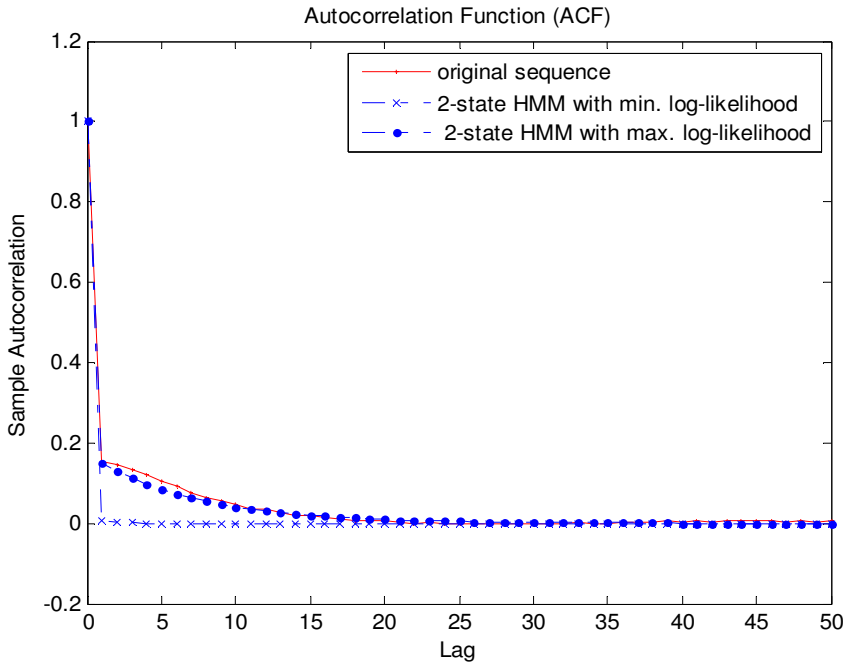


Figure 3.3.3: Autocorrelation of the original input sequence with the autocorrelations of the two HMMs.

3.4 MC and HMM Order Estimation Using AIC and BIC

According to [18], "in the case of HMM, the problem of model selection (and in particular the choice of the number of states in the Markov chain component model) has yet to be satisfactorily solved."

Let us now give a brief introduction to the theory of order estimation for HMMs. We know that the likelihood values in case of HMM, given observed data, increase by increasing the number of states of the models. Hence, the problem of order estimation of Markov processes deals with the compensation of the likelihood function. A penalty term is added to the maximum likelihood and the resulting compensated likelihood is maximized. The general form of an order estimator is [20]

$$r_n \triangleq \arg\max_r \left[\sup_{P \in M^r} \log \Pr(y_1^n) - \text{pen}(n, r) \right] \tag{3.4.1}$$

Proper choice of the penalty term allows the strongly consistent estimation of the model order.

For the case of a Markov chain, the order estimation problem is relatively easier. Finesso uses the Law of Iterated Logarithm (LIL) to get an estimate on the order of a Markov chain [8]. Since such knowledge is not available from LIL in the case of HMMs and we rely on universal coding [24] when dealing with Markov models with hidden state space. The universal code maps the integers (representing messages) onto prefix-free (also called self-delimiting or instantaneously decodable) binary codes. These codes do not require the receiver or the transmitter to know the maximum integer (number of potential messages) ahead of time. Examples of such codes are unary coding, Fibonacci coding and Rice coding etc. [24], [25]. The use of universal coding arises when we need to understand the behavior of the ratio of likelihood,

$$\sup_{P \in M^r} \log \Pr(y_1^n) - \sup_{P \in M'^0} \log \Pr(y_1^n) \tag{3.4.2}$$

We now briefly discuss different HMM order estimators that are available. An excellent survey of these estimators is provided in [19] and [20] and we borrow some important results from them. As pointed out earlier, HMM order estimator theory is influenced by universal source coding and composite hypothesis testing. Different HMM order estimators, that we have studied so far, mainly rely on the theory of universal

source coding and among them the Bayesian Information Criterion (BIC), Minimum Description Length (MDL) and Penalized Maximum Likelihood (PML) [20] are the most popular. All these order estimators are related to the generalized likelihood ratio testing approach. To prove their consistency, we need the Asymptotic Equipartition Principles (AEP) or the Shannon-McMillan-Breiman theorem [24], which will be discussed in some detail shortly. Most of the order estimators satisfy a generalized Neyman-Pearson criterion of optimality. Specifically, the optimal estimator minimizes the probability of underestimation among all estimators whose probability of overestimation lies below a specific value.

Another main topic of HMM order estimation is the study of *error exponents* for different estimators. Error exponents or Bahadur efficiency [20] has proved to be a relevant measure of asymptotic relative efficiency. It is the best rate of exponential decay of the probability of underestimation among all estimators whose probability of overestimation is uniformly bounded away from unity over the entire range of (unknown) measures that could generate the observations.

A different estimation technique of HMM order is inspired by the idea of penalized minimum distance that was proposed by Chen and Kalbfleish [26] and is known in the literature as the CK criterion. The work of CK is extended recently by MacKay from one dimension to multiple dimensions for the case of HMMs [27], [28]. The CK criterion estimates the parameters of a model (including the number of components) by minimizing a form of penalized minimum distance function given by

$$D(\overline{F}_n, F) = d_{KS}(\overline{F}_n, F) - c_n \sum_{j=1}^{K} \log \pi_n \tag{3.4.3}$$

over all F. Here F is the finite mixing distribution with K components and mixing probabilities π_j. The function d_{KS} is the Kolmogorov-Smirnov distance [11] of the two functions. MacKay uses the penalized minimum-distance method to consistently estimate the number of hidden states in a stationary hidden Markov model for finite mixture models. The author proves the consistency of the proposed estimator for the observed data and uses computer simulations to evaluate the performance of this estimator on real world data. However, for the case of binary sequences, this estimator cannot be used in its current form as the proposed method requires that the family of finite mixtures of the

conditional distribution be identifiable, which is not true for the binary distribution (see [29], pp. 185). However, it is true for the binomial(n, p) distribution when $n > 2K$, where K is the number of hidden states.

Let's now discuss the Bayesian Information Criterion (BIC) and Akaike's Information Criterion (AIC), arguably the two most important and frequently used order estimation techniques currently. Let k be the dimension of a Markov model and n be the length of observation sequence, then the BIC order estimator is defined as

$$\hat{r}_n \triangleq \arg\min\left[-2(\sup_{P\in M^r}\log\Pr(y_1^n)) + k\log(n)\right] \tag{3.4.4}$$

The BIC is a special case of penalized maximum likelihood order estimators. It was introduced by Schwarz in late 1970s using Bayesian reasoning. For MCs, the following theorem is established by Csiszár and Shields [30]

> For a stationary irreducible Markov chain process with distribution P^* and of order r^*, the BIC order estimator converges to r^*- almost surely.

For a complete proof of this theorem, refer to [19], [30]. At this point, we don't have a proof for the consistency of BIC for HMM order estimator, but we are hopeful to follow the theory of BIC estimator, already proved for MC, to establish the consistency of this order estimator technique in a general HMM setting. In this report we have performed several HMM simulations to study the performance of BIC applied on binary error sequences.

The AIC is defined as

$$\hat{r}_n \triangleq \arg\min\left[-2(\sup_{P\in M^r}\log\Pr(y_1^n)) + 2k\right] \tag{3.4.5}$$

3.5 Identifiability

When dealing with the order estimation problem of Markov processes, the first difficulty we normally face is the identifiability problem. Generally speaking, the identifiability of a function means that if the function evaluates identical values then the

arguments must be the same. In other words a function is "identifiable" if and only if it is one-to-one. Mathematically, for any function $f(.)$ to be identifiable,

$$f(x) = f(y) \Leftrightarrow x = y \qquad \forall x, y \tag{3.5.1}$$

An example of a function that is unidentifiable is the sine function within the range $[0, \infty)$. This is because of the fact that this function can have the same value for more than one argument (e.g., $\sin(0) = \sin(2\pi)$). Note that the same function is identifiable if the arguments are restricted to $[-\pi/2, \pi/2]$.

The identification of the function of Markov chains was discussed by Blackwell and Koopmans [31] and then was later studied by Gilbert [32] who expanded the work of Blackwell and showed a necessary identifiability condition. This problem was further studied by Dharmadhikhari who provided the sufficient condition for identifiability for a function of finite Markov chains [33]. Later Heller completed the characterization of identifiability of finite Markov chains [34].

In the identification problem related to Markov processes, one usually wishes to learn whether or not a distribution of $Y_1, Y_2, \ldots$ lies in a given model class, for example, we may wish to determine whether a distribution has certain characteristics, e.g., whether it is independent and identically distributed or it is generated by a Markov chain, etc.

3.5.1 Identifiability of a Binary Hidden Markov Process

A random process *Y* is said to have a mixture distribution if its distribution depends on a quantity that also has a distribution. The density of a random process *Y* given the density of *g* mixture components can be written as

$$f_Y(y) = \sum_{i=1}^{g} m_i f_i(y) \tag{3.5.1.1}$$

where $f_Y(y)$ are the densities, m_i are nonnegative quantities that sum to one and *g* is the number of mixture components. It is interesting to note that the marginal distribution of *Y* (where *Y* is a HMP) given *X*, the underlying state sequence, is a finite mixture process [35].

$$f_Y(y_t) = \sum_{k=1}^{N} \pi_k f_Y(Y_t = y_t \mid X_t = k) \qquad (3.5.1.2)$$

Here π_k is the stationary distribution of states and N is the total number of states of the mixture model.

Most of the order estimation techniques that we have studied so far are based on the fact that the marginal distribution of the observation sequence is a finite mixture model and estimating the order of the HMM corresponds to the estimation of the number of mixture components. This requires the finite mixture model to be identifiable which is not true for binary processes and can be shown by a simple example. If a process Y can be modeled as a mixture of binomial distributions with parameters p_1 and p_2, then we can write

$$\begin{aligned} f_Y(y) &= \Pr(Y = y) \\ &= \Pr(Y = y \mid p_1)\Pr(p = p_1) + \Pr(Y = y \mid p_2)\Pr(p = p_2) \qquad y = 0,1 \\ &= \binom{1}{y} p_1^y (1-p_1)^{1-y} \Pr(p = p_1) + \binom{1}{y} p_2^y (1-p_2)^{1-y} \Pr(p = p_2) \end{aligned}$$

Let $\Pr(p_1) = \alpha$, then $\Pr(p_2) = 1-\alpha$

$$\begin{aligned} \Pr(Y=0) &= \binom{1}{0} p_1^0 (1-p_1)^{1-0}\alpha + \binom{1}{0} p_2^0 (1-p_2)^{1-0}(1-\alpha) \\ &= \alpha(1-p_1) + (1-\alpha)(1-p_2) \\ \Pr(Y=1) &= \binom{1}{1} p_1^1 (1-p_1)^{1-1}\alpha + \binom{1}{1} p_2^1 (1-p_2)^{1-1}(1-\alpha) \\ &= \alpha p_1 + (1-\alpha) p_2 \end{aligned}$$

Here there are two equations and three unknowns, so we can have an infinite number of solutions.

3.6 Autocorrelation Functions of Markov Processes

In this section we will discuss the autocorrelation functions of MCs and HMMs. First we will discuss the autocorrelation of MCs and then obtain an expression to evaluate the autocorrelation of HMMs.

Let $\{X_t : t \in \mathbb{N}\}$ be an irreducible homogeneous Markov chain on the first m positive integers, with transition probability matrix $\mathbf{P}$. Suppose that $\{X_t\}$ is stationary so that $\mathbf{P}$ is for all t the distribution of X_t. By the irreducibility, there exists a unique, strictly positive, stationary distribution, which we shall denote by the vector $\boldsymbol{\pi} = [\pi_1\ \pi_2 ... \pi_N]$. The irreducibility implies that there exists 1 as a simple eigenvalue of $\mathbf{P}$ and the corresponding right and left eigenvectors are unique up to constant multiples. It then follows that such eigenvectors are multiples of column vector $\mathbf{1} = [1\ 1\ 1 ... 1]'$.

If we define a vector $\mathbf{v} = [1\ 2\ 3\ ...\ N]$, a diagonal matrix $\mathbf{V}$ with $\mathbf{v}$ on the principal diagonal, i.e., $\mathbf{V} = diag(\mathbf{v})$, and $p_{ij}(k) = (\mathbf{P}^k)_{ij}$, we have the following results for the mean of X_t and the covariance of X_t and X_{t+k}, for all nonnegative integers k. Note that all such results are not meaningful unless the states 1, 2, 3,…, m are quantitative.

$$E(X_t) = \sum_{i=1}^{N} i\pi_i = \boldsymbol{\pi}\mathbf{v}' \tag{3.6.1}$$

where $\Pr(X_t = i) = \pi_i$ and $\boldsymbol{\pi} = [\pi_1\ \pi_2 ... \pi_N]$.

$$E(X_t X_{t+k}) = \sum_{i=1}^{N}\sum_{j=1}^{N} i\ j\ \pi_i \Pr(X_{t+k} = j \mid X_t = i) \tag{3.6.2}$$

$$= \sum_{i,j} \left(i\pi_i\right) p_{ij}(k)_j = \boldsymbol{\pi}\mathbf{V}\mathbf{P}^k\mathbf{v}' \tag{3.6.3}$$

$$Cov(X_t, X_{t+k}) = \boldsymbol{\pi}\mathbf{V}\mathbf{P}^k\mathbf{v}' - \left(\boldsymbol{\pi}\mathbf{v}'\right)^2 \tag{3.6.4}$$

Even if $\mathbf{P}$ is not diagonalizable, some simplification of the expression for the covariance can be achieved by writing $\mathbf{P}$ in Jodan canonical form [36], [37]. The matrix $\mathbf{P}$ can be written as $\mathbf{U}\boldsymbol{\Omega}\mathbf{U}^{-1}$, where $\mathbf{U}$, $\mathbf{U}^{-1}$ and $\boldsymbol{\Omega}$ are of the forms

$$\mathbf{U} = [\mathbf{1}'\ \mathbf{R}], \qquad \mathbf{U}^{-1} = \begin{bmatrix} \boldsymbol{\delta} \\ \mathbf{W} \end{bmatrix} \quad \text{and} \quad \boldsymbol{\Omega} = \begin{bmatrix} \mathbf{1} & \mathbf{0} \\ \mathbf{0}' & \boldsymbol{\Psi} \end{bmatrix}$$

Hence

$$\mathbf{P}^k = \mathbf{U}\boldsymbol{\Omega}^k\mathbf{U}^{-1} = \mathbf{1}'\boldsymbol{\delta} + \mathbf{R}\boldsymbol{\Psi}^k\mathbf{W} \tag{3.6.5}$$

and

$$Cov(X_t, X_{t+k}) = \boldsymbol{\delta}\mathbf{V}\left(\mathbf{1}'\boldsymbol{\delta} + \mathbf{R}\boldsymbol{\Psi}^k\mathbf{W}\right)\mathbf{v}' - \left(\boldsymbol{\delta}\mathbf{v}'\right)^2 = \left(\boldsymbol{\delta}\mathbf{V}\mathbf{R}\right)\boldsymbol{\Psi}^k\left(\mathbf{W}\mathbf{v}'\right) \tag{3.6.6}$$

Since this is true for $k = 0$, we have also the variance of X_t by using $k = 0$ in (3.6.6), and hence the ACF of $\{X_t\}$ can be written as

$$\rho_k = \frac{Cov(X_t, X_{t+k})}{Var(X_t)} = \frac{\left(\boldsymbol{\delta}\mathbf{V}\mathbf{R}\right)\boldsymbol{\Psi}^k\left(\mathbf{W}\mathbf{v}'\right)}{\boldsymbol{\delta}\mathbf{V}\mathbf{R}\mathbf{W}\mathbf{v}'} \tag{3.6.7}$$

Equation (3.6.7) is used to obtain the ACFs of different MCs that we will discuss in section 3.9. It is a faster way of obtaining the autocorrelation of Markov Chains. We can directly obtain the autocorrelation without first having the data generated from the model and then obtain its sample autocorrelation. MC reverse simulation, i.e., generating data from the model, is initially used to obtain the autocorrelation from the generated data in order to validate the mathematical expression that we obtained for the MC ACF.

Now we derive the expression for HMM autocorrelation function. As a preliminary step we state two results which we will use in deriving moment properties of a hidden Markov model. If we denote $f(Y)$ as the function of Markov chain and assume that its expectation exist, we can write,

$$E(f(Y_t)) = \sum_{i=1}^{N} E(f(Y_t) \mid X_t = i)\pi_i \tag{3.6.8}$$

This is proved by conditioning on X_t and noting that $P(X_t = i) = \pi_i$. Secondly, provided again that the relevant expectations exist, we have for $k \in \mathbb{N}$ that

$$E(f(Y_t, Y_{t+k})) = \sum_{i,j=1}^{N} E(f(Y_t, Y_{t+k}) \mid X_t = i, X_{t+k} = j)\pi_i p_{ij}(k) \tag{3.6.9}$$

where $p_{ij}(k) = (\mathbf{P}^k)_{ij}$. We can also write (3.6.9) as [18]

$$E(f(Y_t, Y_{t+k})) = \sum_{i,j} E(f(Y_t, Y_{t+k}) \mid X_t = i, X_{t+k} = j)\Pr(X_t = i, X_{t+k} = j) \tag{3.6.10}$$

The result (3.6.9) then follows since $\Pr(X_t = i, X_{t+k} = j) = \pi_i p_{ij}(k)$.

For the case of binomial-hidden Markov model, the conditional distribution of Y_t is binomial with the parameters n_t and $p(t)$. We shall assume throughout this section that $\mathbf{P}$ is diagonalizable, since that is the case of most practical interest. We use notations

$\mathbf{h} = [p_1 \ p_2 \ ... p_N]$ with p_i being the probability of generating an error in state *i*, $\mathbf{H} = diag(\mathbf{h})$ and $\mathbf{D} = diag(\boldsymbol{\pi})$.

From equation (3.6.8) we have

$$E(Y_t) = \sum_{i=1}^{m} (n_t p_i)\pi_i = n_t \boldsymbol{\pi}\mathbf{h}' \tag{3.6.11}$$

and

$$E(Y_t^2) = \sum_{i=1}^{N} \left(n_t p_i (1 - p_i) + n_t^2 p_i^2 \right) \pi_i \tag{3.6.12}$$

$$= n_t \boldsymbol{\pi}\mathbf{h}' + n_t (n_t - 1)\mathbf{h}\mathbf{D}\mathbf{h}' \tag{3.6.13}$$

$$= n_t \boldsymbol{\pi}\mathbf{h}' + n_t (n_t - 1)\boldsymbol{\pi}\mathbf{H}\mathbf{h}' \tag{3.6.14}$$

Hence

$$Var(Y_t) = n_t (n_t - 1)\boldsymbol{\pi}\mathbf{H}\mathbf{h}' + n_t \boldsymbol{\pi}\mathbf{h}' - n_t^2 \left(\boldsymbol{\pi}\mathbf{h}' \right)^2 \tag{3.6.15}$$

$$= n_t^2 \left(\boldsymbol{\pi}\mathbf{H}\mathbf{h}' - \left(\boldsymbol{\pi}\mathbf{h}' \right)^2 \right) + n_t \left(\boldsymbol{\pi}\mathbf{h}' - \boldsymbol{\pi}\mathbf{H}\mathbf{h}' \right) \tag{3.6.16}$$

Now we have for $k \in \mathbb{N}$

$$E(Y_t Y_{t+k}) = \sum_{i,j=1}^{N} (n_t p_i)(n_{t+k} p_j)\pi_i p_{ij}(k) = n_t n_{t+k} \boldsymbol{\pi}\mathbf{H}\mathbf{P}^k\mathbf{h}' \tag{3.6.17}$$

The covariance can therefore be written as

$$Cov(Y_t, Y_{t+k}) = n_t n_{t+k} \left(\boldsymbol{\pi}\mathbf{H}\mathbf{P}^k\mathbf{h}' - \left(\boldsymbol{\pi}\mathbf{h}' \right)^2 \right) \tag{3.6.18}$$

Hence the autocorrelation is given by

$$Corr(Y_t, Y_{t+k}) = \frac{n_t n_{t+k} \left(\boldsymbol{\pi}\mathbf{H}\mathbf{P}^k\mathbf{h}' - \left(\boldsymbol{\pi}\mathbf{h}' \right)^2 \right)}{\left(Var(Y_t) Var(Y_{t+k}) \right)^{1/2}} \tag{3.6.19}$$

provided that $\boldsymbol{\pi}\mathbf{H}\mathbf{h}' - \left(\boldsymbol{\pi}\mathbf{h}' \right)^2 = Var\left(p(t) \right)$ is nonzero, we can define

$$\alpha = \frac{\boldsymbol{\pi}\mathbf{h}' - \boldsymbol{\pi}\mathbf{H}\mathbf{h}'}{\boldsymbol{\pi}\mathbf{H}\mathbf{h}' - \left(\boldsymbol{\pi}\mathbf{h}' \right)^2} \tag{3.6.20}$$

and write the autocorrelation as

$$\rho_k = \left(1 + \alpha / n_t \right)^{-\frac{1}{2}} \left(1 + \alpha / n_{t+k} \right)^{-\frac{1}{2}} \frac{\boldsymbol{\pi}\mathbf{H}\mathbf{P}^k\mathbf{h}' - \left(\boldsymbol{\pi}\mathbf{h}' \right)^2}{\boldsymbol{\pi}\mathbf{H}\mathbf{h}' - \left(\boldsymbol{\pi}\mathbf{h}' \right)^2} \tag{3.6.21}$$

Since it can be shown that $\boldsymbol{\pi}\mathbf{h}' \geq \boldsymbol{\pi}\mathbf{H}\mathbf{h}'$, it follows that α is nonnegative, and the factors $(1+\alpha/n_t)^{-1/2}$ and $(1+\alpha/n_{t+k})^{-1/2}$ are at most 1. If n_t is constant with respect to t, expression (3.6.21) shows that the autocorrelation depends on k only through $\mathbf{P}^k$. It is further shown in [18] that in general for N and $\mathbf{P}$ diagonlizable we can express the covariance of Y_t and Y_{t+k} in terms of the k-th powers of the eigenvalues w_i (other than 1) of $\mathbf{P}$. With $\mathbf{U}$ as defined earlier, $c = \boldsymbol{\pi}\mathbf{H}\mathbf{U}$ and $\mathbf{d}' = \mathbf{U}^{-1}\mathbf{h}'$, we have for $k \in \mathbb{N}$

$$Cov(S_t, S_{t+k}) = n_t n_{t+k} \sum_{i=2}^{N} c_i w_i^k d_i \tag{3.6.22}$$

and the corresponding expression for the autocorrelation can be obtained by dividing it with the variance.

The case $n_t = 1$ provides models for binary time series and the results derived above simplified to the following:

$$E(Y_t) = \boldsymbol{\pi}\mathbf{h}' \tag{3.6.23}$$

$$Var(Y_t) = \boldsymbol{\pi}\mathbf{h}' - (\boldsymbol{\pi}\mathbf{h}')^2 \tag{3.6.24}$$

$$Corr(Y_t, Y_{t+k}) = \frac{\boldsymbol{\pi}\mathbf{H}\mathbf{P}^k\mathbf{h}' - (\boldsymbol{\pi}\mathbf{h}')^2}{\boldsymbol{\pi}\mathbf{h}' - (\boldsymbol{\pi}\mathbf{h}')^2} \tag{3.6.25}$$

3.7 Information Theoretic Measures for Markov Processes

Some information theoretic measures related to Markov processes are briefly discussed in this section. Information theoretic measures such as some measures of distance between two Markov models are useful if these models have the same number of states. Our understanding is that these measures are not useful in this work as in models having different number of states, the likelihood of Markov models increase, given an observed sequence, as we increase its number of states.

3.7.1 Kullback-Leibler and Divergence-Rate Distance

If λ_1 and λ_2 are two probability measures on a data set Y_1^n for a given n, the Kullback-Leibler distance (also called Kullback-Leibler divergence, relative entropy or KLD) between λ_1 and λ_2 is defined by

$$D(\lambda_1 \| \lambda_2) = \sum_{x \in Y_1^n} \lambda_1(x) \log[\lambda_1(x)/\lambda_2(x)] \tag{3.7.1.1}$$

with the usual convention that a term in the above sum is taken to be zero whenever $\lambda_1(x) = 0$ and is taken to be ∞ whenever $\lambda_1(x) > 0$ and $\lambda_2(x) = 0$. The following properties should be noted:

$$0 \le D(\lambda_1 \| \lambda_2) \le \infty \tag{3.7.1.2}$$

$$D(\lambda_1 \| \lambda_2) = 0, \text{ if } \lambda_1 = \lambda_2 \tag{3.7.1.3}$$

The asymptotic relative entropy or divergence-rate distance between the two processes, denoted by $D_\infty(\lambda_1 | \lambda_2)$, is defined mathematically as

$$D_\infty(\lambda_1 | \lambda_2) \triangleq \lim_{n \to \infty} \frac{1}{n} D(\lambda_1 \| \lambda_2) \tag{3.7.1.4}$$

It should be noted that

$$0 \le D_\infty(\lambda_1 | \lambda_2) \le \infty \tag{3.7.1.5}$$

$$D_\infty(\lambda_1 | \lambda_2) = 0, \text{ if } \lambda_1 = \lambda_2 \tag{3.7.1.6}$$

In 1993, Kieffer used an information-theoretic approach to estimate the order of a constrained finite state model class [38]. The two theorems presented in his paper related to code-based model identification and order estimation were extensively used later.

3.7.2 Shannon-McMillan-Breiman Theorem

In ergodic theory, the Shannon-McMillan-Breiman theorem (also denoted as the SBC theorem) is one of the most fundamental limit theorems for classical discrete dynamic systems. The Shannon-McMillan (SM) theorem is also known as the Asymptotic Equipartition Property (AEP) in classical information theory. The SMB theorem is a stronger version of the SM Theorem and can be interpreted as a special case of the individual ergodic theorem.

SBC is a strong and almost sure convergence statement. Recall that the Shannon entropy of a probability distribution P on a finite set $\mathbb{A}$ is defined by

$$H(P) = -\sum_{a \in \mathbb{A}} P(a) \log P(a) \tag{3.7.2.1}$$

where log generally denotes the natural logarithm. In the information theory setting it is more common to use the base 2 algorithm, relating to the bit as the fundamental information unit.

Let Y_t be a finitely valued ergodic process with probability distribution $Q(.)$. Then

$$\frac{1}{n} \log Q(Y_1^n) \to E_Q[\log Q(Y_0 \mid Y_{-\infty}^{-1})] \quad \text{almost surely} \tag{3.7.2.2}$$

This form of the theorem has direct application in information theory because it allows the estimation of the entropy rate of a finite alphabet stationary ergodic source.

Now assume Q to be the stationary ergodic distribution of Y_t and M be a finite order stationary Markov measure on the sequence space, with the assumption of the absolute continuity of Q with respect to the n-th order marginal of M and denoting the corresponding density by $q(y_0^{n-1})$, Barron defined the divergence rate of a true distribution Q with respect to the reference measure M as [39]

$$D_\infty(Q \mid M) = \lim_k E_Q[\log q(Y_k \mid Y_0^{k-1})] \tag{3.7.2.3}$$

Then $D_\infty(Q \mid M)$ is well defined and

$$\frac{1}{n} \log q(y_1^n) \to D_\infty(Q \mid M) \quad \text{almost surely} \tag{3.7.2.4}$$

Finesso generalized the theorem [8] for the case of HMP and stated that if Y_t is stationary ergodic under the probability distribution Q and a HMP under the alternate distribution P for some fixed q then for $q(Y_1^k) = \frac{Q(Y_1^k)}{P(Y_1^k)}$ and defined

$$D_\infty(Q \mid M) = \lim_k E_Q[\log q(Y_k \mid Y_0^{k-1})] \tag{3.7.2.5}$$

then $D_\infty(.)$ is well defined and moreover

$$\frac{1}{n}\log\frac{Q(Y_1^k)}{P(Y_1^k)} \to D_\infty(Q \mid P) \quad \text{almost surely} \tag{3.7.2.6}$$

For the detailed proof of this theorem, refer to [8]. In [20] it is shown that for θ and θ' being two HMP, then for all initial distributions q and q' on the hidden state space of θ and θ' respectively,

$$\lim_k \frac{1}{n}\log\frac{P_{q,\theta}\{Y_1^n\}}{P_{q',\theta'}\{Y_1^n\}} = D_\infty(P_\theta \mid P_{\theta'}) \quad \text{almost surely} \tag{3.7.2.7}$$

where $D_\infty(P_\theta \mid P_{\theta'}) = \lim_{n\to\infty}\frac{1}{n}E\left[\log\frac{dP^n}{dQ^n}\right]$ is defined as the relative entropy rate or divergence rate between two processes.

3.8 Modeling Quantized Fading Statistics

We present PH distribution simulation results using Clarke's model [10] of the Rayleigh fading channel with Doppler parameter $f_d = 10\,\text{Hz}$, and with sampling rates of 0.5Kbps and 10Kbps.

Case 1 - PH Approximation of Rayleigh Fading with f_d = 10Hz and f_s = 0.5Kbps

The probability density function of the resulting Rayleigh fading envelope is shown in Fig. 3.8.1.

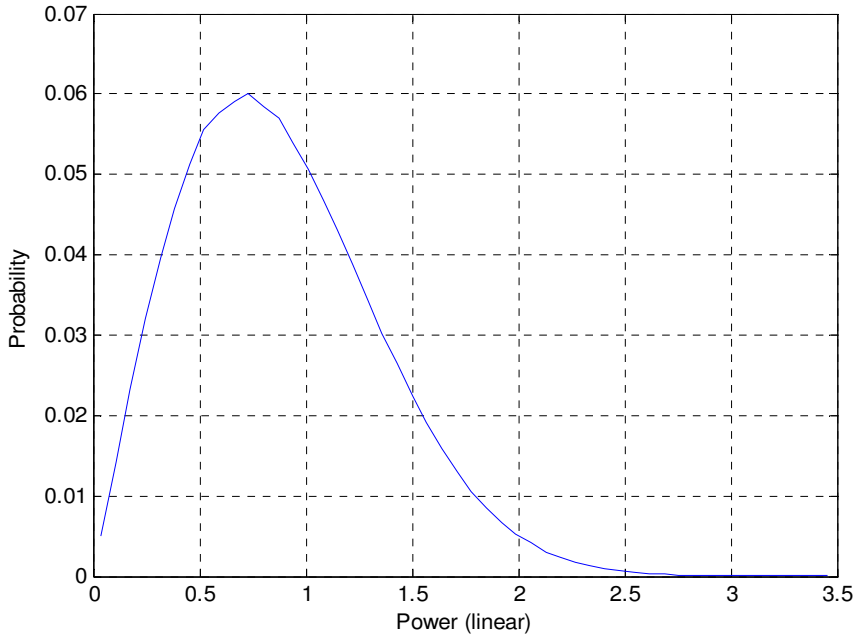


Figure 3.8.1: Probability density function of Rayleigh fading envelope. Rayleigh fading parameters are $f_d = 10\text{Hz}$ and $f_s = 0.5\text{Kbps}$.

We simulated 1 million steps of the fading amplitude to estimate the state transition probability matrix where the states are represented by the quantization intervals (0, 0.0149), (0.0149, 0.0329), (0.0329, 0.0725), (0.0725, 0.1601), (0.1601, 0.3535), (0.3535, 0.7805), (0.7805, 1.7234) and $(1.7234, \infty)$. The quantized levels of the fading envelope are shown in Figure 3.8.2. There are 8 quantization levels, and the sixth, the seventh, and the eighth quantization levels are selected in this simulation. The primary reason for selecting these quantization levels is because of sufficient data samples contained by these levels.

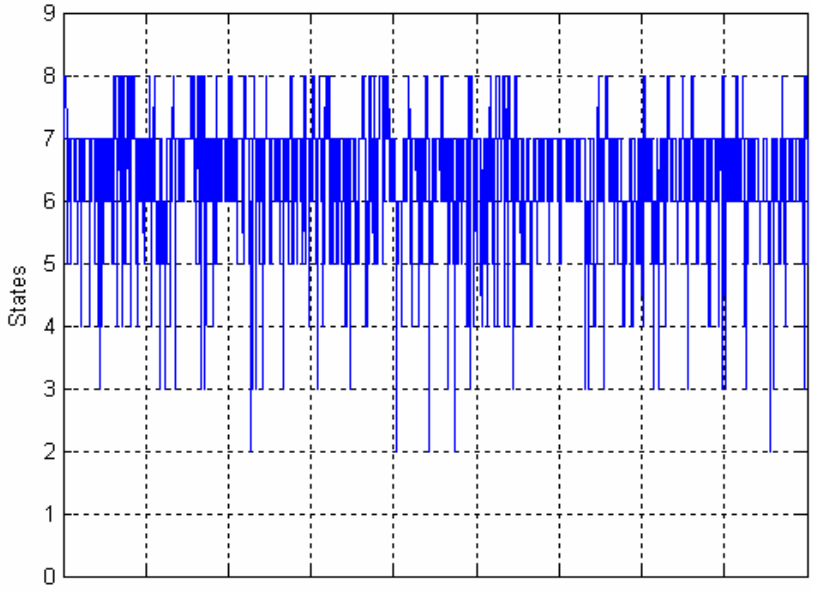


Figure 3.8.2: Fading envelope quantized to different fading levels. Rayleigh fading parameters are $f_d = 10\text{Hz}$ and $f_s = 0.5\text{Kbps}$.

This sixth quantization level is then approximated using a 10-state PH distribution approximation model. The quantized state distribution vector is given as input to the model that approximates the PH-distribution, i.e. the model discussed in Section 2.5.6. The matrix that is obtained from the PH-distribution approximation algorithm (for quantization state 6) is shown below.

$$\mathbf{P} = \begin{bmatrix} 0.0509 & 0.0425 & 0.0146 & 0.0266 & 0.0063 & 0.1203 & 0.0769 & 0.5836 & 0.0049 & 0.0736 \\ 0.1095 & 0.1390 & 0.0326 & 0.1439 & 0.1878 & 0.1283 & 0.0953 & 0.1307 & 0.0200 & 0.0130 \\ 0.0784 & 0.0508 & 0.2106 & 0.0391 & 0.0036 & 0.0159 & 0.2870 & 0.0180 & 0.2962 & 0.0003 \\ 0.0362 & 0.0866 & 0.0459 & 0.3886 & 0.3557 & 0.0210 & 0.0081 & 0.0364 & 0.0181 & 0.0036 \\ 0.0016 & 0.0261 & 0.0117 & 0.3209 & 0.6108 & 0.0000 & 0.0102 & 0.0139 & 0.0007 & 0.0039 \\ 0.1465 & 0.0204 & 0.0248 & 0.0418 & 0.0060 & 0.1537 & 0.0362 & 0.0995 & 0.0162 & 0.4549 \\ 0.2531 & 0.0412 & 0.1223 & 0.0519 & 0.0015 & 0.1601 & 0.0794 & 0.0574 & 0.2227 & 0.0103 \\ 0.0905 & 0.0383 & 0.0054 & 0.0298 & 0.0197 & 0.0978 & 0.0195 & 0.1755 & 0.0094 & 0.5140 \\ 0.3791 & 0.0133 & 0.0368 & 0.0022 & 0.0021 & 0.1018 & 0.0079 & 0.1676 & 0.2886 & 0.0007 \\ 0.0003 & 0.0173 & 0.8782 & 0.0490 & 0.0039 & 0.0021 & 0.0255 & 0.0003 & 0.0235 & 0 \end{bmatrix}$$

The fade duration distribution for the sixth quantization state and its approximation are shown in Figure 3.8.3. The plots show a very close match.

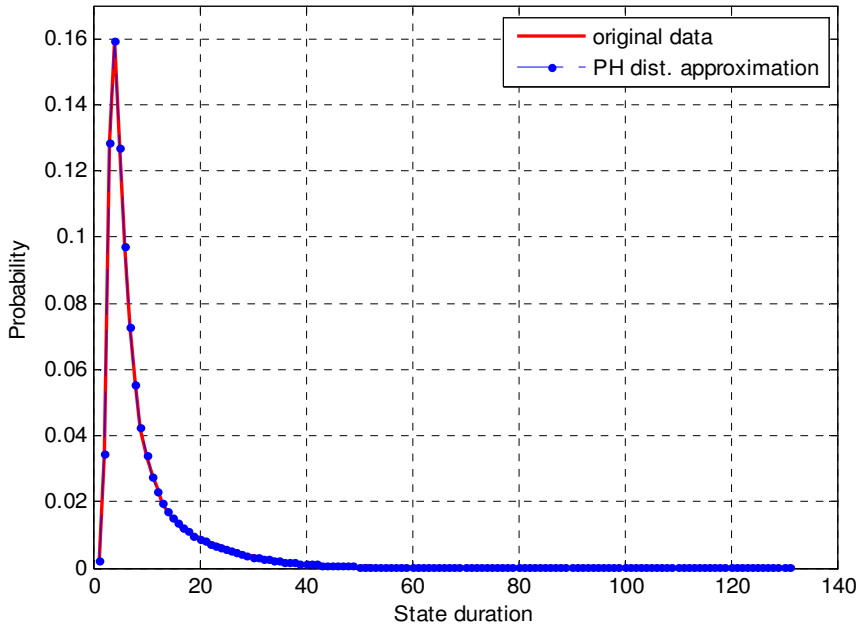


Figure 3.8.3: Fade duration distribution and its approximation for the sixth quantization level. Rayleigh fading parameters are $f_d = 10\text{Hz}$ and $f_s = 0.5\text{Kbps}$.

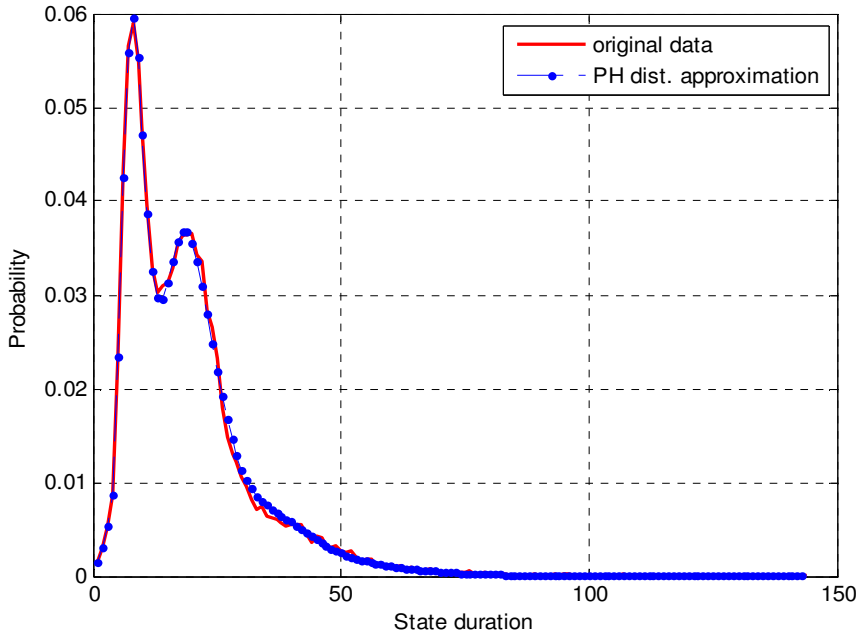


Figure 3.8.4: Fade duration distribution and its approximation for the seventh quantization level. Rayleigh fading parameters are $f_d = 10\text{Hz}$ and $f_s = 0.5\text{Kbps}$.

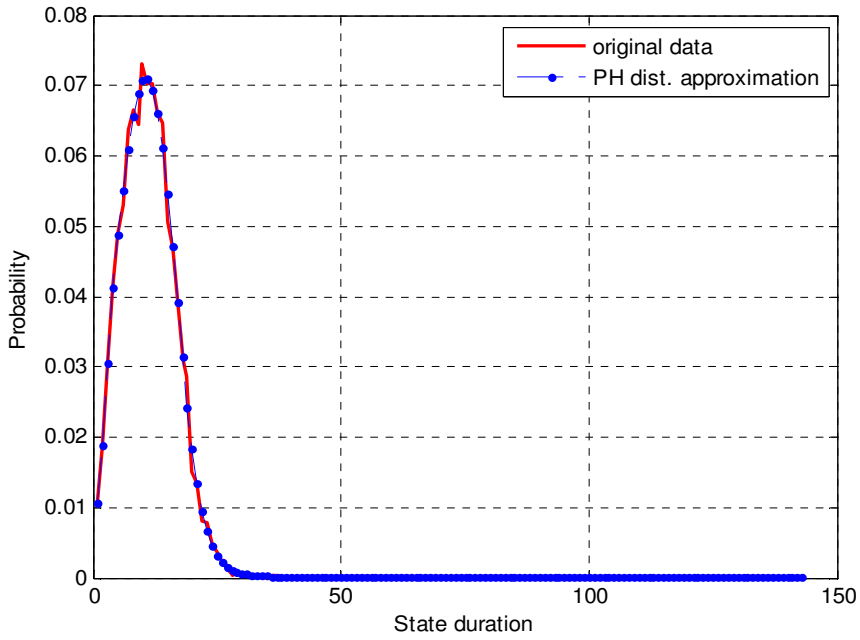


Figure 3.8.5: Fade duration distribution and its approximation for the eighth quantization level. Rayleigh fading parameters are $f_d = 10\text{Hz}$ and $f_s = 0.5\text{Kbps}$.

Case 2 - PH Approximation of Rayleigh Fading with f_d = 10Hz and f_s = 10Kbps

Another Rayleigh fading channel is tested for PH approximation. The channel has $f_d = 10$ and $f_s = 10$ KHz. The seventh, eighth, and ninth quantization levels are chosen for the PH approximation. The quantization levels are (0, 0.0149), (0.0149, 0.0329), (0.0329, 0.0725), (0.0725, 0.1601), (0.1601, 0.3535), (0.3535, 0.7805), (0.7805, 1.7234) and $(1.7234, \infty)$. Figures 3.8.6, 3.8.7, and 3.8.7 show the PH distribution approximation for the seventh, eighth, and ninth quantization levels respectively. The PH distribution for the ninth state does not match with the original distribution.

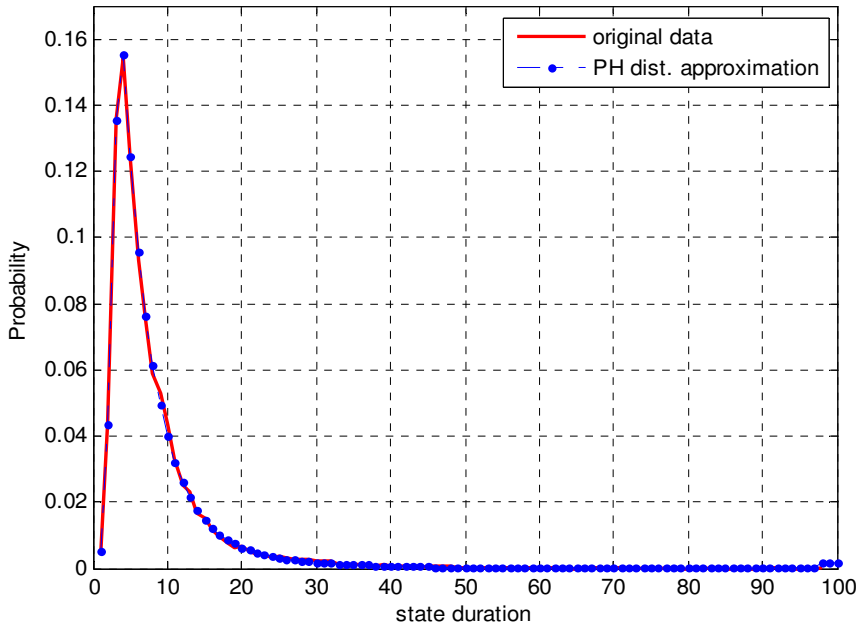


Figure 3.8.6: Fade duration distribution and its approximation for the seventh quantization level. Rayleigh fading parameters are $f_d = 10$Hz and $f_s = 10$Kbps .

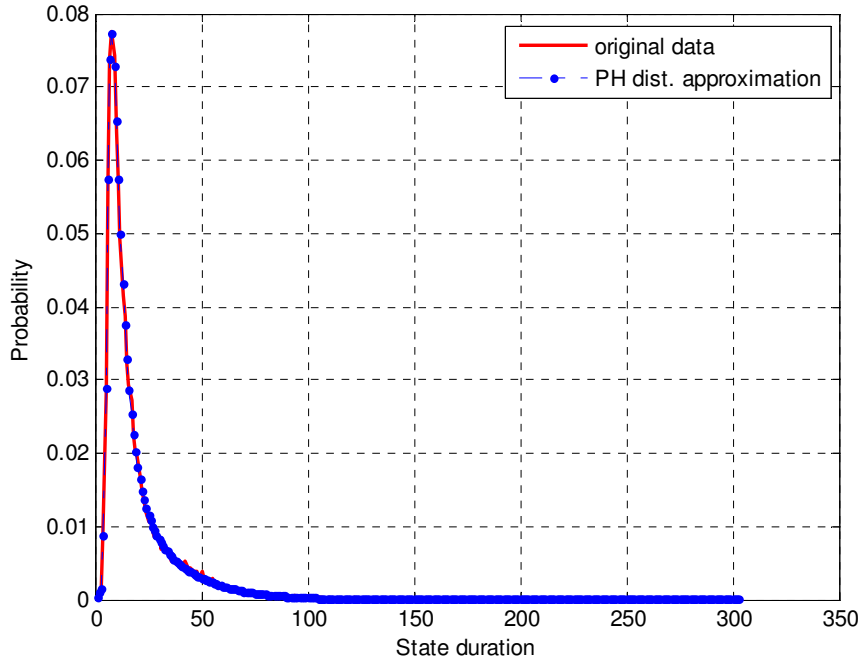


Figure 3.8.7: Fade duration distribution and its approximation for the eighth quantization level. Rayleigh fading parameters are $f_d = 10\text{Hz}$ and $f_s = 10\text{Kbps}$.

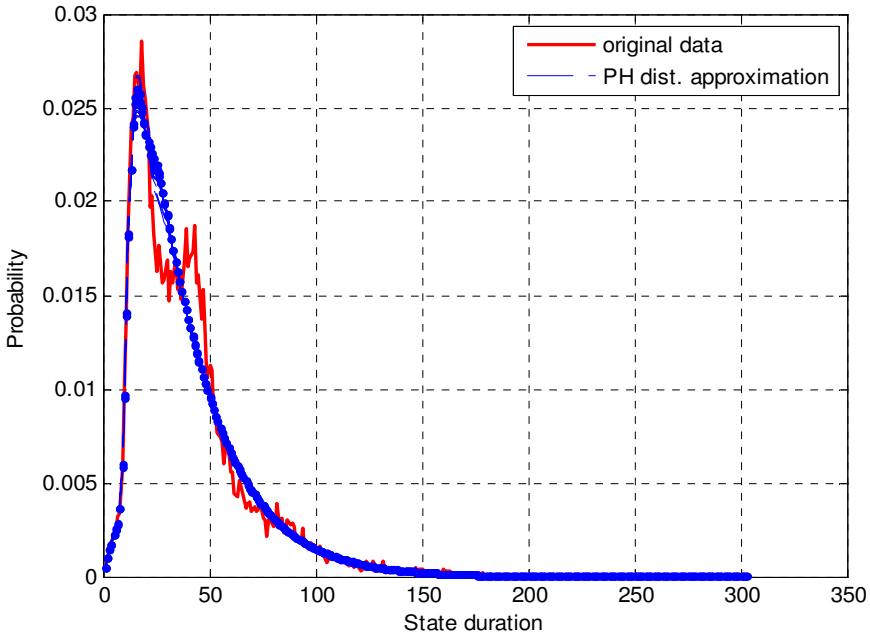


Figure 3.8.8: Fade duration distribution and its approximation for the ninth quantization level. Rayleigh fading parameters are $f_d = 10\text{Hz}$ and $f_s = 10\text{Kbps}$.

3.9 Order Estimation of Binary Markov Chain Using Goodness of Fit

Case 1- Zero-Order Markov Chain

We perform order estimation test on a binary memoryless chain with the BER of 0.1. Fig. 3.9.1 shows the CDFs of different test statistics. The top plot shows the test result to be optimistic. From the result we found out that the three information criteria are very close to the theoretical χ^2 plots. Hence we reject these hypotheses and conclude that the given sequence is from a binary symmetric channel with no memory.

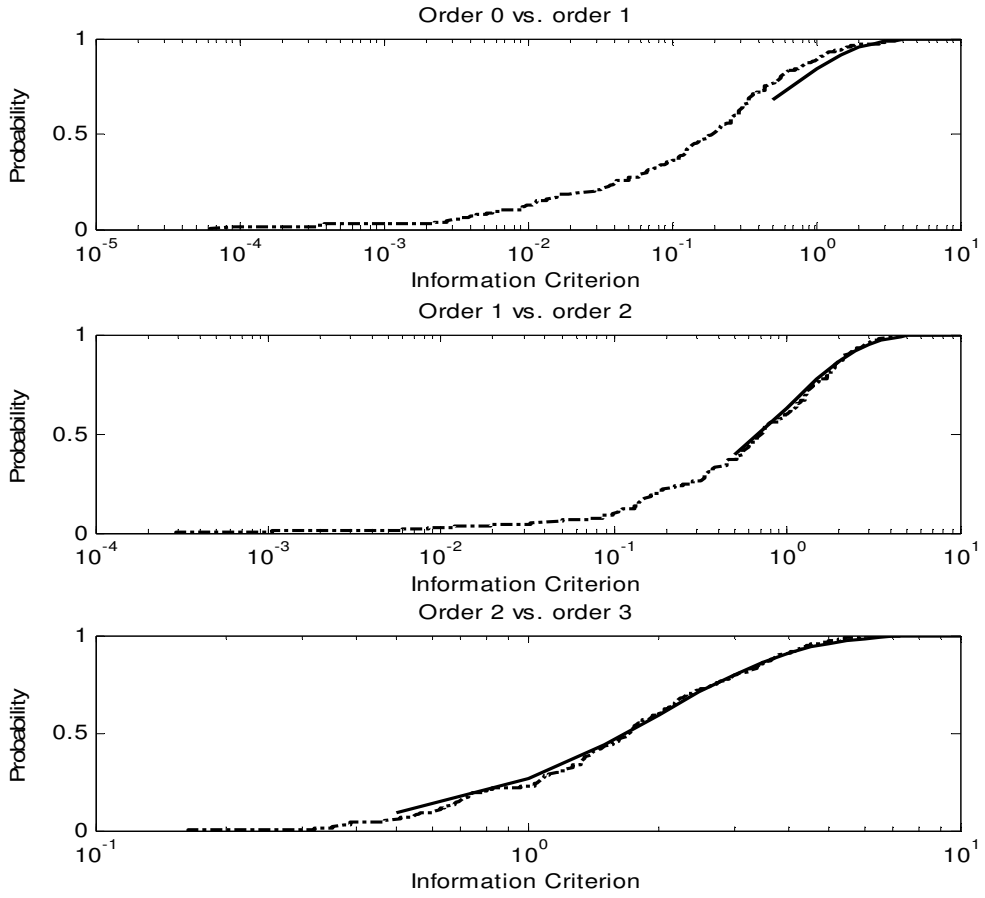


Figure 3.9.1: Markov chain order estimation using goodness of fit. The original sequence is a memoryless chain.

Case 2 - First-Order Markov Chain

A first-order Markov chain for binary error sequence takes the form

$$\Pr(e_n \mid e_0, e_1, ..., e_{n-1}) = \Pr(e_n \mid e_{n-1}) \tag{3.9.1}$$

The state transition matrix can be written as

$$\mathbf{P} = \begin{bmatrix} \Pr(0|0) & \Pr(1|0) \\ \Pr(0|1) & \Pr(1|1) \end{bmatrix} \tag{3.9.2}$$

We generate many error sequences based on the following state transition matrix

$$\mathbf{P} = \begin{bmatrix} 0.90 & 0.10 \\ 0.73 & 0.27 \end{bmatrix} \tag{3.9.3}$$

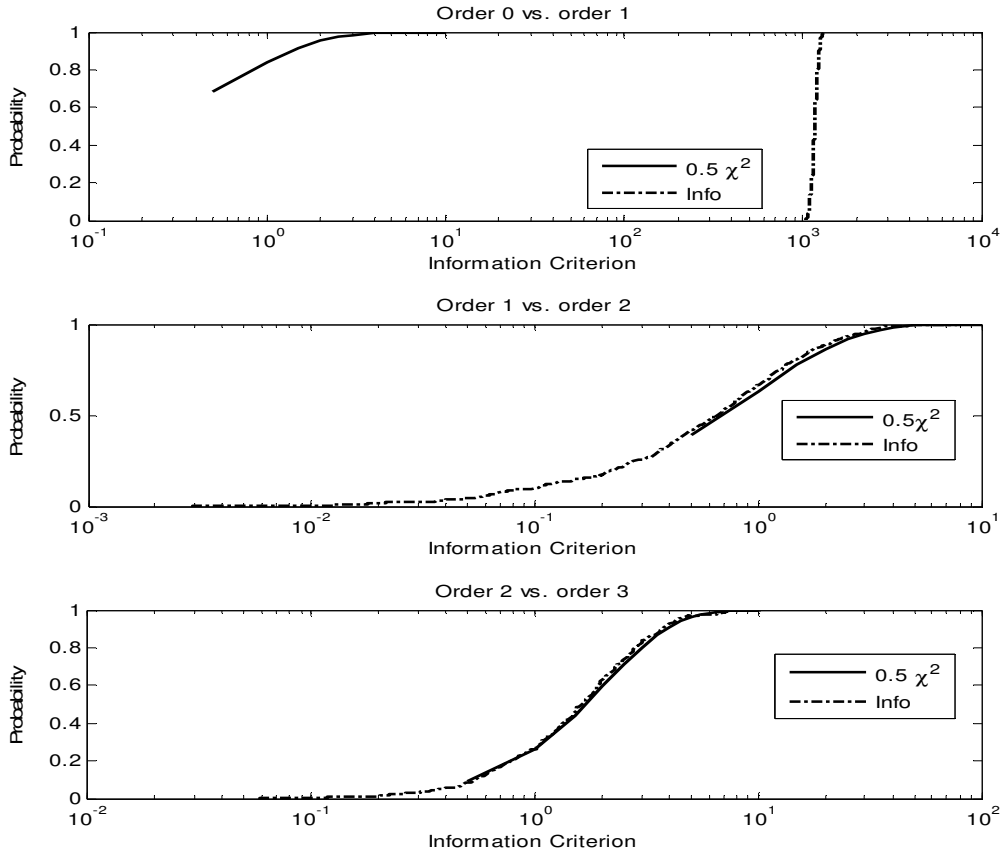


Figure 3.9.2: Markov chain order estimation using goodness of fit. The true order of Markov chain is 1.

The length of each error sequence generated for these tests is 100,000 bits and the tests are performed 1000 times to get the statistics of information criteria. The first test, order 0 vs. 1 (Fig. 3.9.2 top plot), shows that the information criterion doesn't fit the theoretical $0.5\chi^2$ CDF. In fact the sample values of the information criterion are much higher than the theoretical $0.5\chi^2$ CDF. However for order 1 vs. order 2 and order 2 vs. order 3 we find *I* to be distributed as $0.5\chi^2$. Therefore from these plots, we infer that the given data is drawn from a first-order Markov chain.

Case 3 - Second-Order Markov Chain

Let us now consider order estimation of a second-order Markov chain with a transition matrix having the form

$$\mathbf{P}=\begin{bmatrix} \Pr(0|0,0) & 0 & \Pr(1|0,0) & 0 \\ \Pr(0|1,0) & 0 & \Pr(1|1,0) & 0 \\ 0 & \Pr(0|0,1) & 0 & \Pr(1|0,1) \\ 0 & \Pr(0|1,1) & 0 & \Pr(1|1,1) \end{bmatrix} \tag{3.9.4}$$

We consider the following state transition matrix

$$\mathbf{P}=\begin{bmatrix} 0.90 & 0.00 & 0.10 & 0.00 \\ 0.60 & 0.00 & 0.40 & 0.00 \\ 0.00 & 0.95 & 0.00 & 0.05 \\ 0.00 & 0.70 & 0.00 & 0.30 \end{bmatrix} \tag{3.9.5}$$

Fig. 3.9.3 shows that first two information criteria are greater than the theoretical $0.5\chi^2$ plots whereas in the third plot, (test for order 2 vs. order 3), *I* is found to be distributed as $0.5\chi^2$. Hence from the plots we infer that the Markov chain is of order 2.

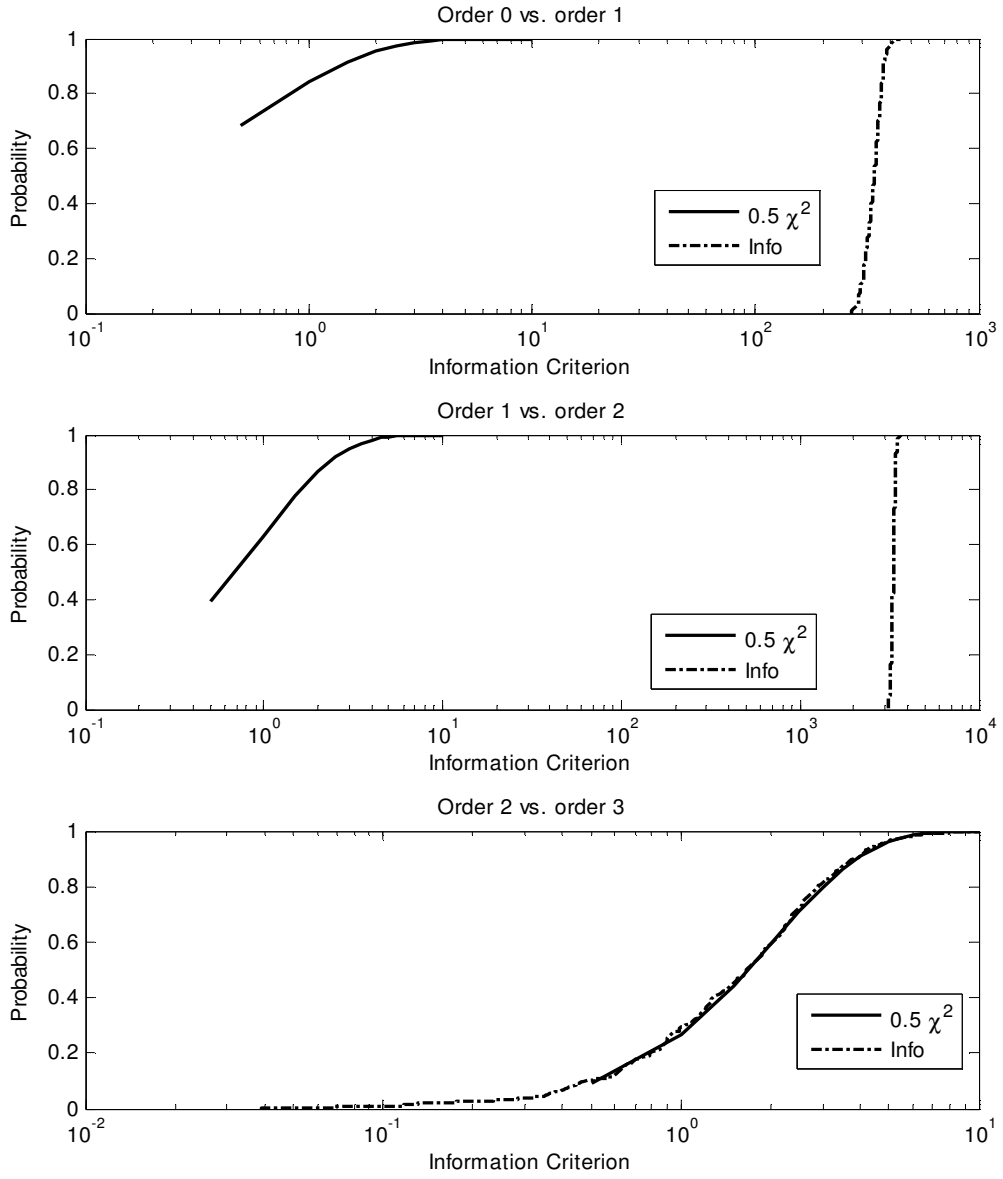


Figure 3.9.3: Markov chain order estimation using goodness of fit. The true order of the Markov Chain is 2.

Case 4 - Third-Order Markov Chain

The state transition matrix for a third-order MC is an extension of the second-order Markov chain given in [4]. The state transition matrix takes the form

$$\mathbf{P} = \begin{bmatrix} \Pr(0|0,0,0) & 0 & 0 & 0 & \Pr(1|0,0,0) & 0 & 0 & 0 \\ \Pr(0|1,0,0) & 0 & 0 & 0 & \Pr(1|1,0,0) & 0 & 0 & 0 \\ 0 & \Pr(0|0,1,0) & 0 & 0 & 0 & \Pr(1|0,1,0) & 0 & 0 \\ 0 & \Pr(0|1,1,0) & 0 & 0 & 0 & \Pr(1|1,1,0) & 0 & 0 \\ 0 & 0 & \Pr(0|0,0,1) & 0 & 0 & 0 & \Pr(1|0,0,1) & 0 \\ 0 & 0 & \Pr(0|1,0,1) & 0 & 0 & 0 & \Pr(1|1,0,1) & 0 \\ 0 & 0 & 0 & \Pr(0|0,1,1) & 0 & 0 & 0 & \Pr(1|0,1,1) \\ 0 & 0 & 0 & \Pr(0|1,1,1) & 0 & 0 & 0 & \Pr(1|1,1,1) \end{bmatrix}$$

(3.9.6)

We generate Markov sequences according to the following state transition matrix

$$\mathbf{P} = \begin{bmatrix} 0.9000 & 0 & 0 & 0 & 0.1000 & 0 & 0 & 0 \\ 0.6000 & 0 & 0 & 0 & 0.4000 & 0 & 0 & 0 \\ 0 & 0.7800 & 0 & 0 & 0 & 0.2200 & 0 & 0 \\ 0 & 0.9900 & 0 & 0 & 0 & 0.0100 & 0 & 0 \\ 0 & 0 & 0.8500 & 0 & 0 & 0 & 0.1500 & 0 \\ 0 & 0 & 0.8000 & 0 & 0 & 0 & 0.2000 & 0 \\ 0 & 0 & 0 & 0.7500 & 0 & 0 & 0 & 0.2500 \\ 0 & 0 & 0 & 0.5500 & 0 & 0 & 0 & 0.4500 \end{bmatrix}$$

(3.9.7)

Fig. 3.9.4 shows that the information criteria in first three tests are greater than the corresponding theoretical $0.5\chi^2$ plots. We infer from the tests performed above that the information criterion remains greater than the theoretical $0.5\chi^2$ plots till the optimum order is estimated. After estimating the correct order, the information criterion *I* is found to be $0.5\chi^2$ distributed. Hence the order of Markov chain is found to be 3.

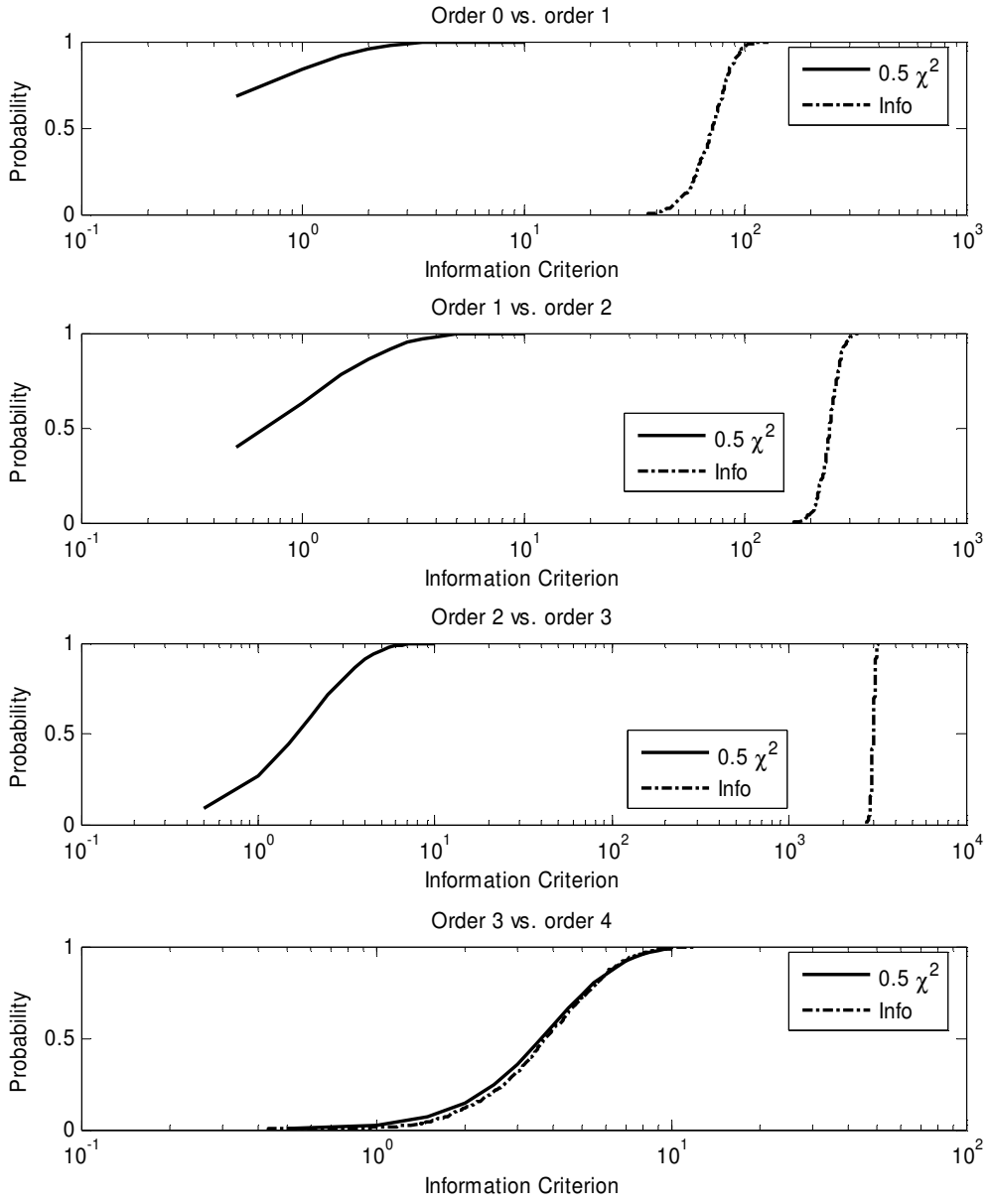


Figure 3.9.4: Markov chain order estimation using goodness of fit. The true order true is 3.

We have shown how to estimate the correct order of a Markov chain before fitting the Markov models to experimental data. Note that for using this technique, we need to

have a sufficient number of sequences to obtain the statistics of information criterion. Also the simulation time to process so many sequences is practically not possible in the real world scenarios. In later sections, we will use BIC and AIC to estimate the order of Markov chains. BIC and AIC provide fast estimation of the order of MCs without the need of requiring a large number of data sequences to obtain the test results.

3.10 Forward-backward and Forward-only BWA Simulation

In this section we test the accuracy of two forms of the BWA (forward-backward and forward-only) that are used to estimate the parameters (**A, B** and $\boldsymbol{\pi}$) of HMM.

An error sequence is generated using the two-state HMM of (3.10.1) and the resulting sequence is used to train two different HMMs, one using the FB-BWA and the other using the FO-BWA.

$$\mathbf{A} = \begin{bmatrix} 0.5890 & 0.4110 \\ 0.1149 & 0.8851 \end{bmatrix} \mathbf{B} = \begin{bmatrix} 0.9409 & 0.9634 \\ 0.0591 & 0.0366 \end{bmatrix} \quad \boldsymbol{\pi} = \begin{bmatrix} 0.2184 & 0.7816 \end{bmatrix} \qquad (3.10.1)$$

Different initial conditions are used in both cases and the best model (based on the log-likelihood values) is chosen. The log-likelihood using the FB-BWA is found to be -721.0073 whereas using the FO-BWA, the best model is found to have a log-likelihood of -721.2814. The convergence of model based on log-likelihood during the search process for the case of the FB-BWA is shown in Fig. 3.10.1. Similar plot is obtained with the FO-BWA simulation.

Fig. 3.10.2 and 3.10.3 compares the error-free intervals of original and HMM generated data using both the FB-BWA and the FO-BWA. The two plots show very a close match. Table 3.10.1 summarizes the comparison of error-interval probabilities of sample sequence and sequences generated using the two HMMs, obtained from two different forms of the BWA. The HMM that is obtained by using the FB-BWA is found to be slightly better for the error intervals of lengths 1, 2 and 3.

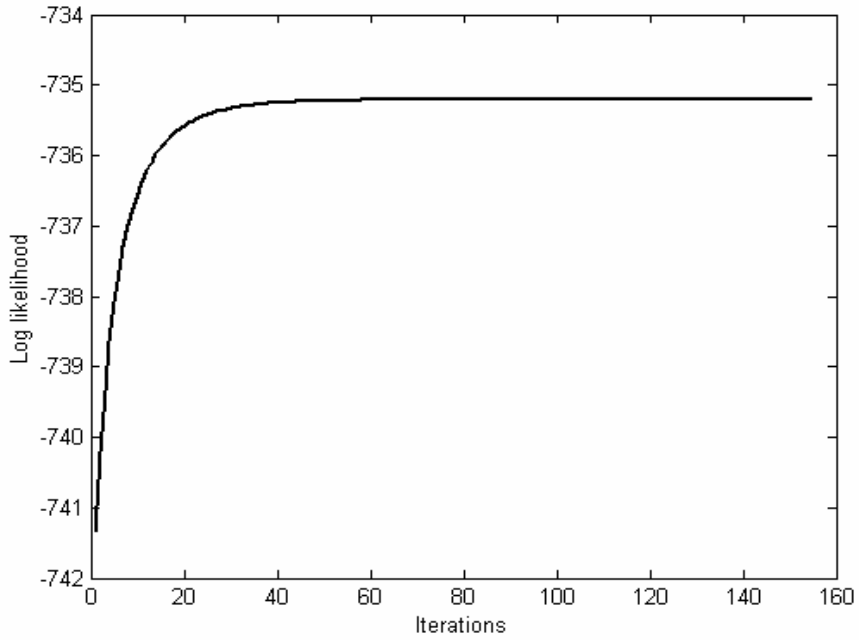


Figure 3.10.1: Convergence of log-likelihood values using the FB-BWA.

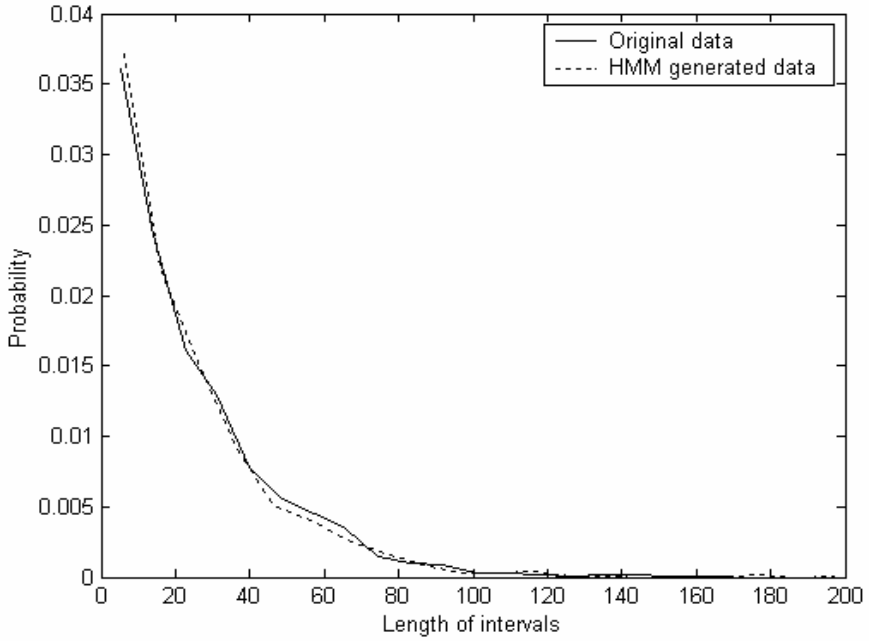


Figure 3.10.2: Comparison of error-free intervals of original data and HMM generated data using the FB-BWA.

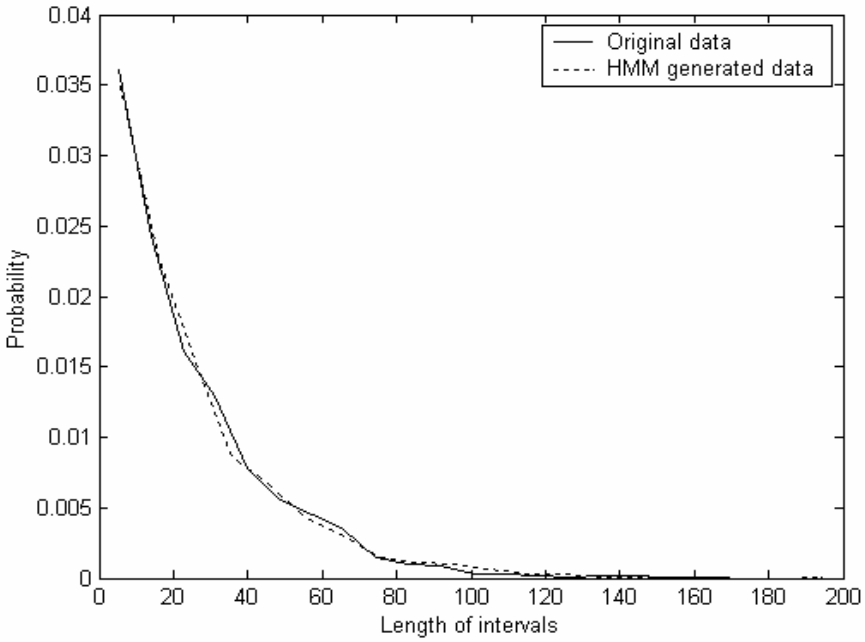


Figure 3.10.3: Comparison of error-free intervals of original data and HMM generated data using FO-BWA.

	length-1	length-2	length-3	length-4
original	0.9512	0.0488	0	0
FB-BWA	0.9597	0.0389	0.0010	0.0005
FO-BWA	0.9639	0.0341	0.0020	0

Table 3.10.1: Probability of error intervals in original sample and HMM generated data obtained using the FB-BWA and the FO-BWA.

It is also found that the variance of the log-likelihood values obtained from using the FB-BWA is larger than that of HMMs the variance obtained using the FO-BWA. There could be two reasons. Either the log-likelihood profile of the FO-BWA is unimodal

and all initial conditions converge to same peak or there are many peaks with essentially the same maximum values of the log-likelihood.

3.11 MC and HMM Order Estimation Using AIC and BIC

Case 1 - First-Order Markov Chain

An error sequence is generated from a second-order Markov chain. Different MCs and HMMs are developed and AIC and BIC for each of these models are then calculated, based on their respective log-likelihood values. The results are shown in Table 3.11.1.

model	k	$-l$	AIC	BIC
independent	1	768.9626	1.5339×10^3	1.5455×10^3
first-order MC	2	732.9876	$\mathbf{1.4661\times10^3}$	$\mathbf{1.4773\times10^3}$
second-order MC	4	732.1630	1.4701×10^3	1.4925×10^3
third-order MC	8	726.8199	1.4781×10^3	1.5229×10^3
2-state HMM	4	733.8313	1.4757×10^3	1.4981×10^3
3-state HMM	9	731.6999	1.4814×10^3	1.5318×10^3
4-state HMM	16	731.0626	1.4941×10^3	1.5837×10^3

Table 3.11.1: AIC and BIC results for different Markov models (original sequence is a first-order binary MC).

It is found out the AIC and BIC both select the first-order Markov chain as the best candidate to model the given data. The ACFs of the first-order MC, the 2-state HMM and the 3-state HMM are plotted in Figs. 3.11.1, 3.11.2, and 3.11.3 and compared with the sample autocorrelation.

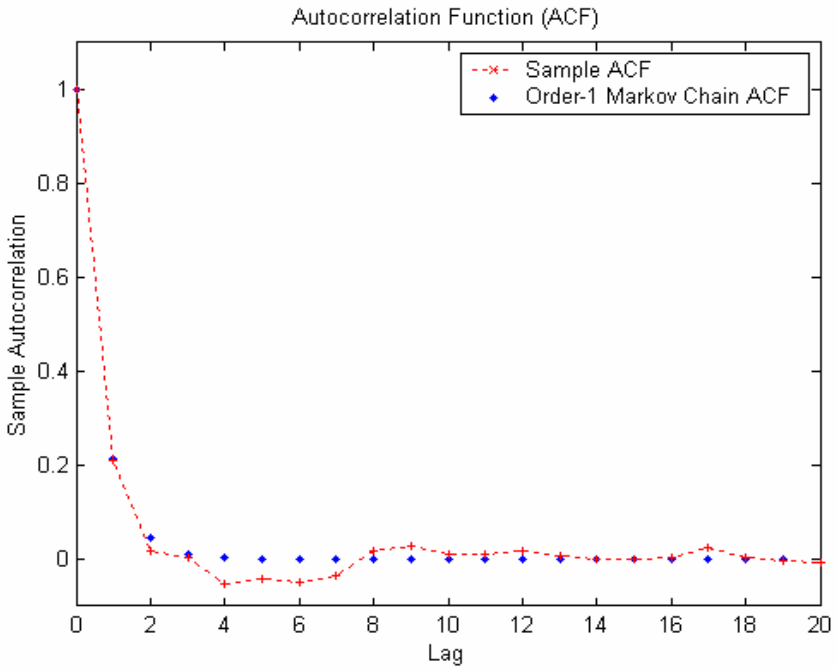


Figure 3.11.1: First 20 lags of sample ACF and order-1 MC ACF. Sample ACF is obtained from the data generated by a first-order MC.

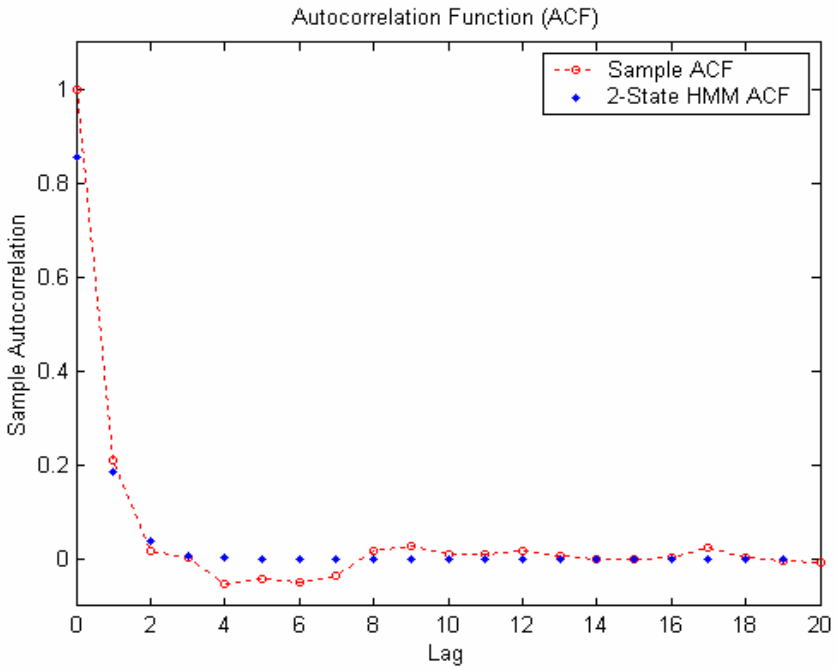


Figure 3.11.2: First 20 lags sample ACF and 2-state HMM ACF. Sample ACF is obtained from the data generated by a first-order MC.

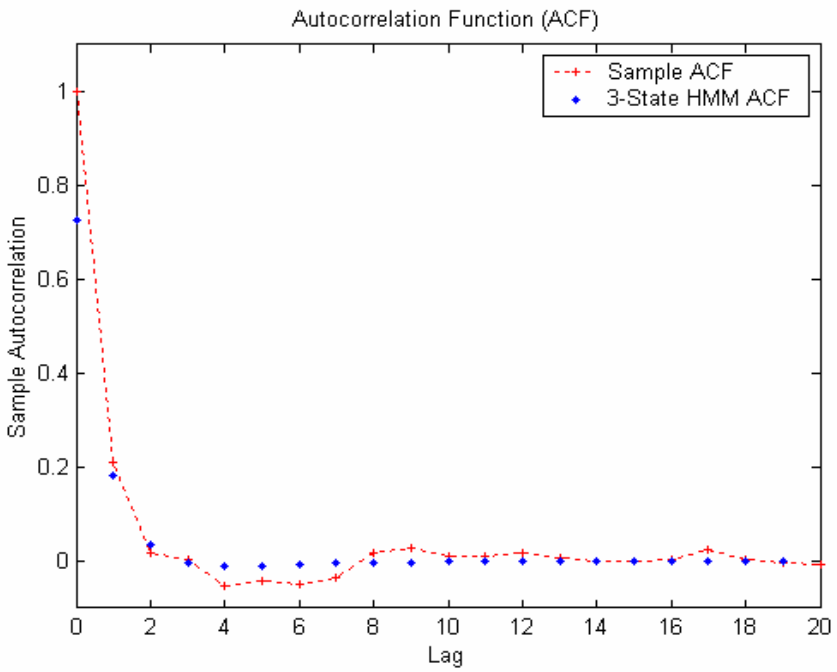


Figure 3.11.3: First 20 lags of sample ACF and 3-state HMM ACF. Sample ACF is obtained from the data generated by a first-order MC.

Case 2 - Second-Order Markov Chain

In case 2, a second-order MC is generated and different Markov models are developed. Table 3.11.2 shows the values of AIC and BIC obtained from these models. Both AIC and BIC select the second-order MC as the best candidate.

model	k	$-l$	AIC	BIC
independent	1	735.7232	1.4374×10^3	1.4789×10^3
first-order MC	2	734.7704	1.4735×10^3	1.4844×10^3
second-order MC	4	672.4514	$\mathbf{1.3529\times10^3}$	$\mathbf{1.3747\times10^3}$
third-order MC	8	670.2456	1.3565×10^3	1.4000×10^3
2-state HMM	4	724.1406	1.4563×10^3	1.4780×10^3
3-state HMM	9	697.5232	1.4130×10^3	1.4620×10^3

4-state HMM	16	677.4835	1.3870×10^{3}	1.4740×10^{3}

Table 3.11.2: AIC and BIC values of different Markov models (original sequence is a second-order binary Markov Chain).

First-order MC, second-order MC and 4-state HMM ACFs are calculated and plotted along with the ACF generated from the observation sequence. The second-order MC seems to be the best fit here.

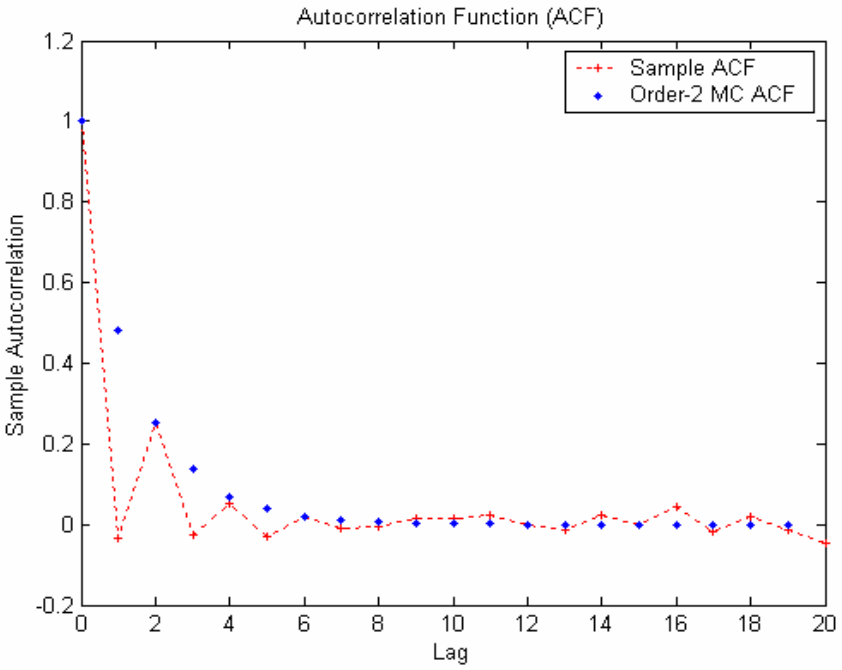


Figure 3.11.4: First 20 lags of sample ACF and second-order MC ACF. Sample ACF is obtained from the data generated by a second-order MC.

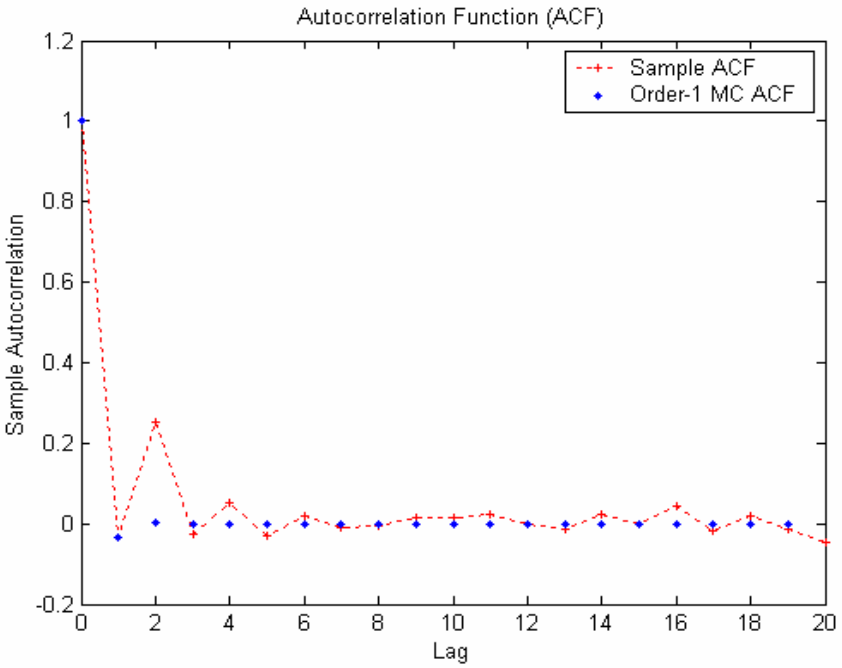


Figure 3.11.5: First 20 lags of sample ACF and first-order MC ACF. Sample ACF is obtained from the data generated by a second-order MC.

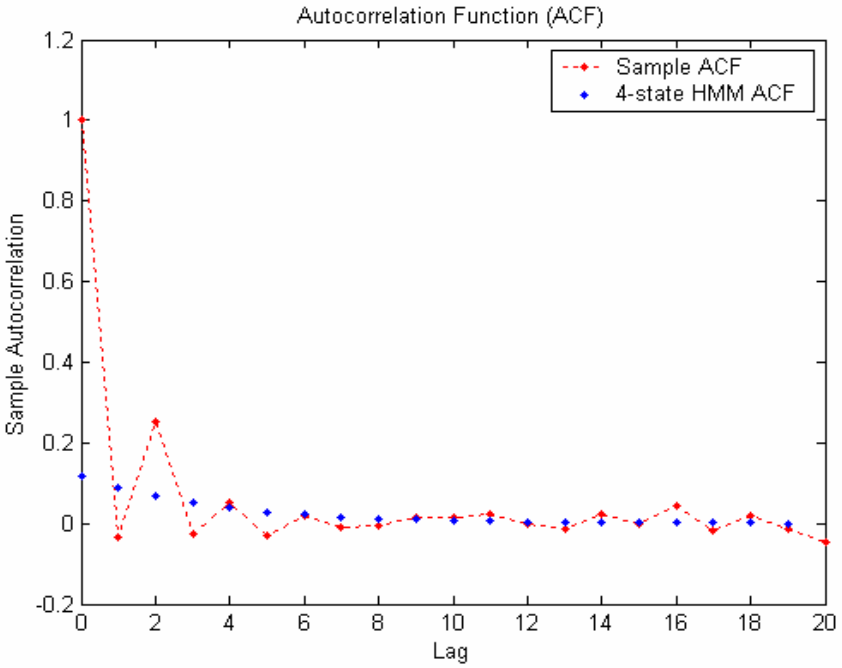


Figure 3.11.6: First 20 lags of sample ACF and 4-state HMM ACF. Sample ACF is obtained from the data generated by a second-order MC.

Case 3 - Three-State HMM

In case 3, a three-state HMM is generated and in this case, both AIC and BIC pick a 2-state HMM instead of a 3-state HMM as the best candidate. This is a consistent result that we have observed that in case of binary HMMs that AIC and BIC both usually pick a 2-state HMM as the best candidate. We are further investigating this in more detail and we wish to find the theoretical reason for this behavior of AIC and BIC order estimator.

model	k	$-l$	AIC	BIC
independent	1	776.6481	1.5553×10^3	1.5605×10^3
first-order MC	2	760.2741	1.5245×10^3	1.5349×10^3
second-order MC	4	752.1174	1.5122×10^3	1.5329×10^3
third-order MC	8	734.0367	1.4841×10^3	1.5254×10^3
2-state HMM	4	734.4222	$\mathbf{1.4768\times10^3}$	$\mathbf{1.4975\times10^3}$
3-state HMM	9	734.4220	1.4868×10^3	1.5334×10^3
4-state HMM	16	729.6323	1.4913×10^3	1.5740×10^3

Table 3.11.3: AIC and BIC values of different Markov models (original sequence is a 3-state binary hidden Markov sequence).

ACFs are plotted in Figures 3.11.7, 3.11.8 and 3.11.9. The first-order MC (having relatively high values of AIC and BIC) doesn't seem to fit the sample data reasonably well.

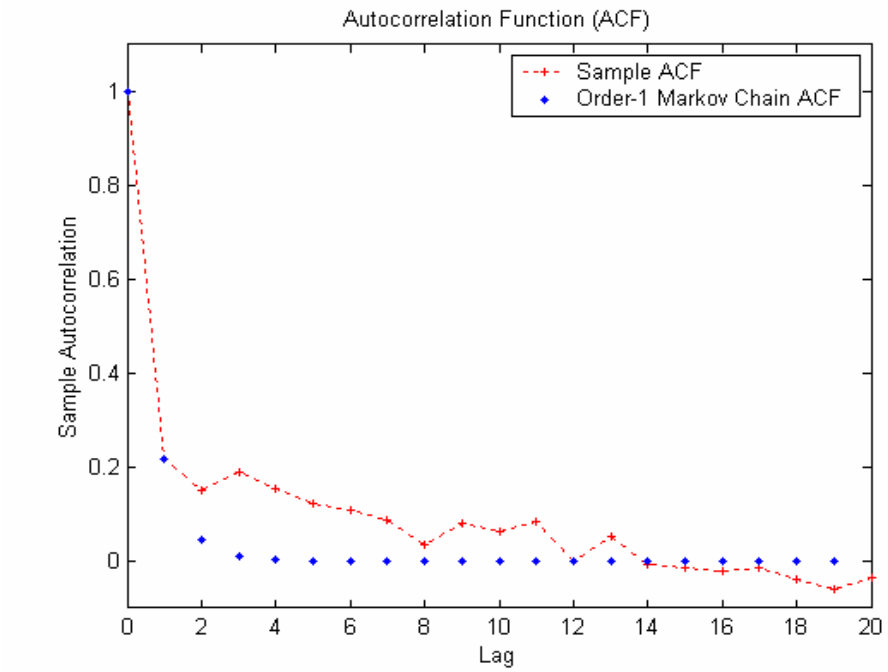


Figure 3.11.7: First 20 lags of sample ACF and first-order MC ACF. Sample ACF is obtained from the data generated by a three-state HMM.

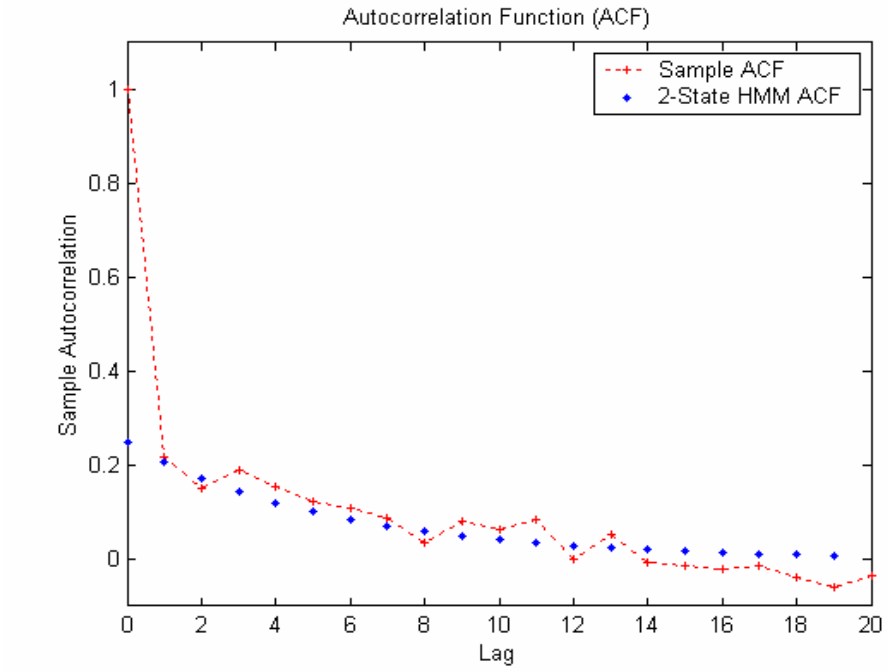


Figure 3.11.8: First 20 lags of sample ACF and 2-state HMM ACF. Sample ACF is obtained from the data generated by a three-state HMM.

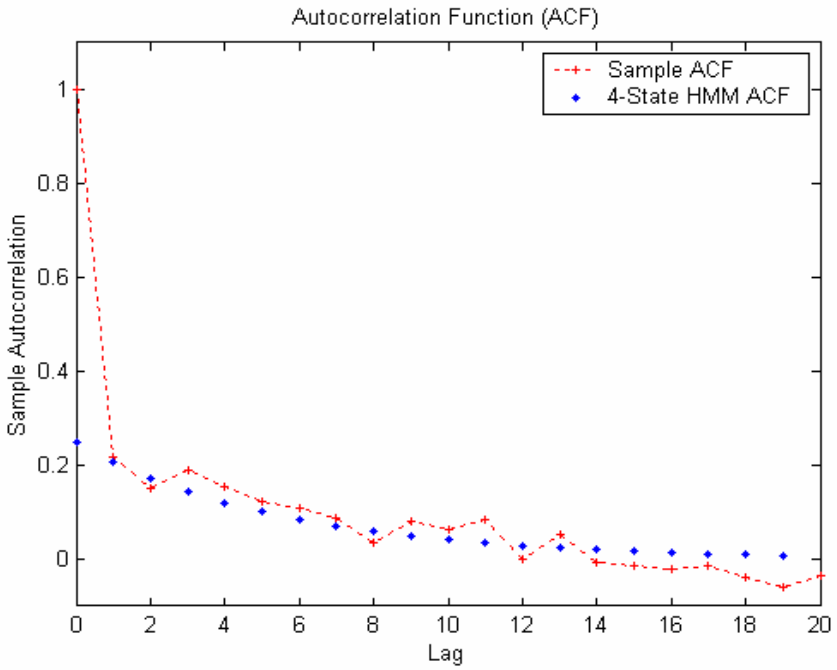


Figure 3.11.9: First 20 lags of sample ACF and 4-state HMM ACF. Sample ACF is obtained from the data generated by a three-state HMM.

Case 4 - Five-state HMM

In case 4, we use a 5-state HMP and develop different HMMs using the BWA. Table 3.11.4 summarizes AIC and BIC results.

model	k	$-l$	AIC	BIC
2-state HMM	4	734.4950	$\mathbf{1.4656\times10^3}$	$\mathbf{1.4895\times10^3}$
3-state HMM	9	734.4852	1.4753×10^3	1.5291×10^3
4-state HMM	16	734.4815	1.4896×10^3	1.5848×10^3
5-state HMM	25	734.4400	1.5070×10^3	1.6563×10^3

Table 3.11.4: AIC and BIC values of different Markov models (original sequence is a five-state binary HMP).

As seen from the table, the 2-state HMM is chosen to be the best model for the given binary data. We now plot different ACFs to observe the closeness of original data and developed models. Figs. 3.11.10, 3.11.11, 3.11.12 and 3.11.13 show the ACFs of original and model generated data. It can be inferred that the performance (with respect to autocorrelation) of all these models is essentially the same.

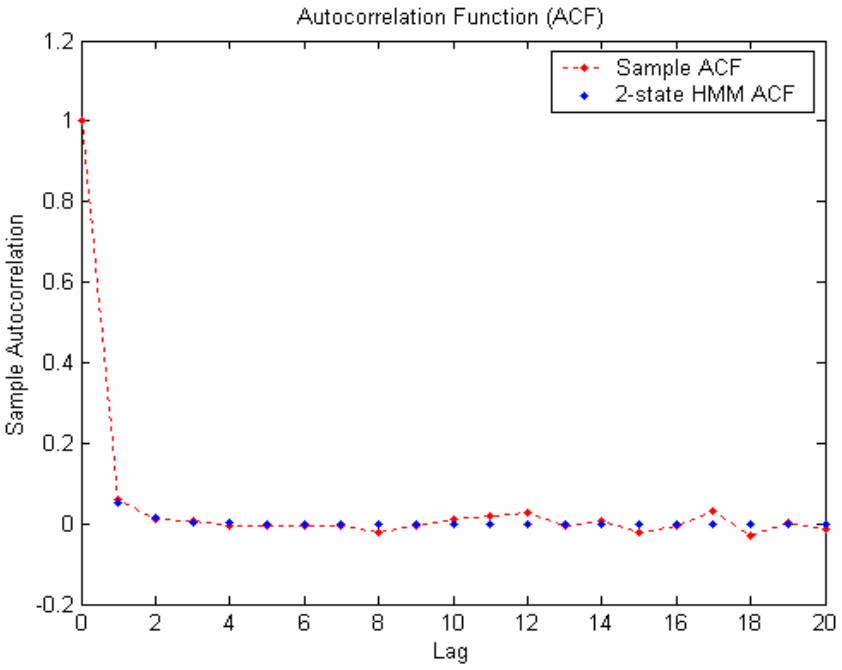


Figure 3.11.10: First 20 lags of sample ACF and 2-state HMM ACF. Sample ACF is obtained from the data generated by a five-state HMM.

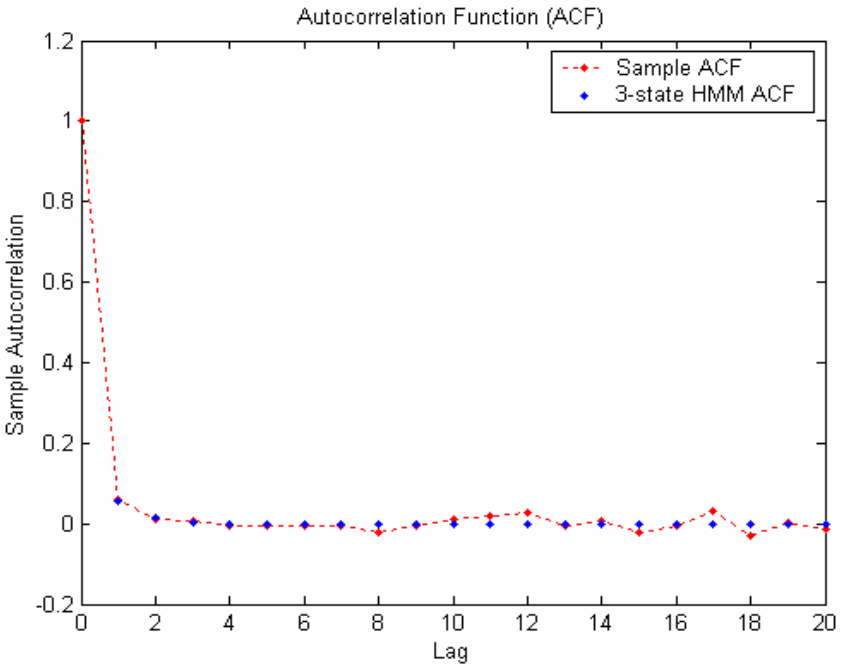


Figure 3.11.11: First 20 lags of sample ACF and 3-state HMM ACF. Sample ACF is obtained from the data generated by a five-state HMM.

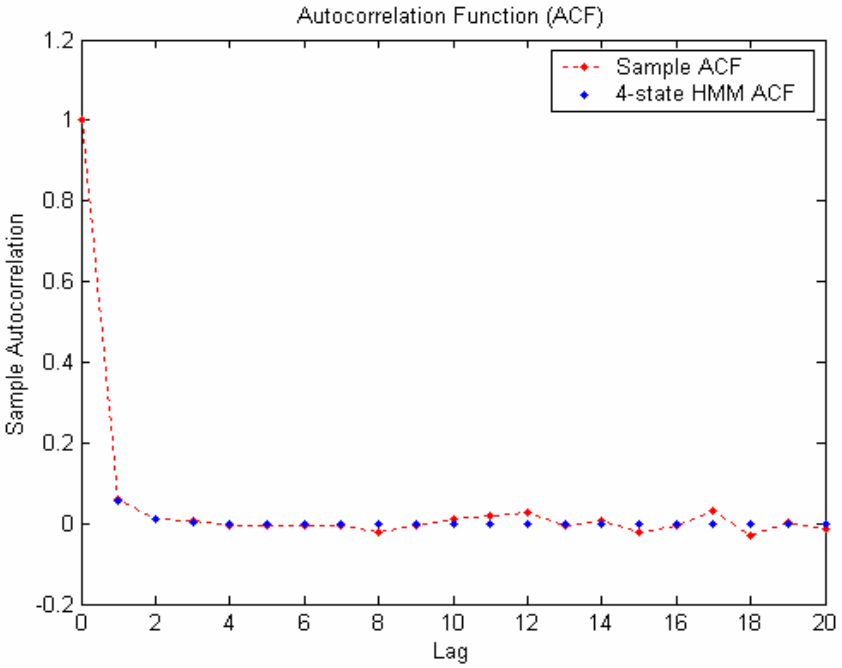


Figure 3.11.12: First 20 lags of sample ACF and 4-state HMM ACF. Sample ACF is obtained from the data generated by a five-state HMM.

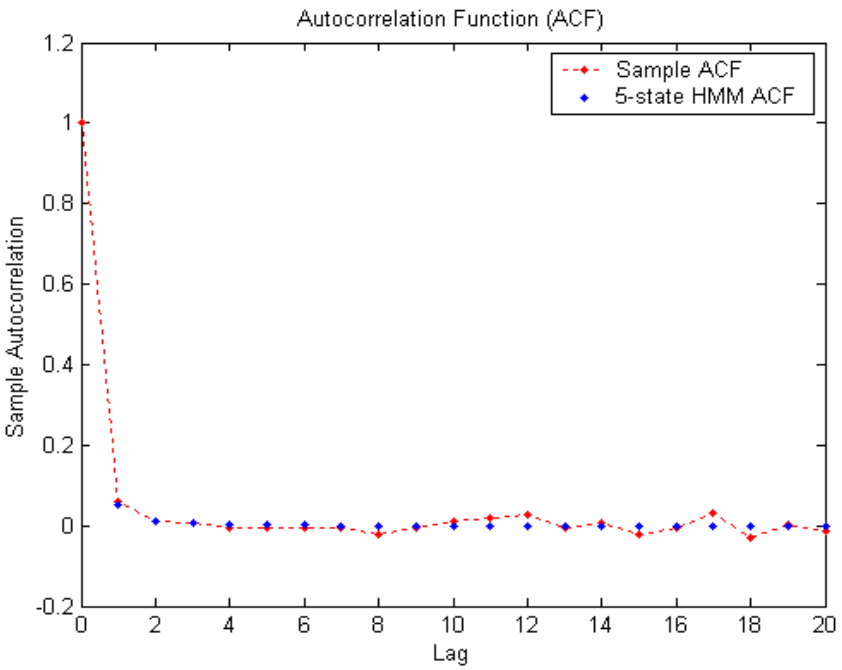


Figure 3.11.13: First 20 lags of sample ACF and 5-state HMM ACF. Sample ACF is obtained from the data generated by a 5-state HMM.

Case 5 - Six-State HMM

BIC and AIC order estimation is performed on an error sequence that is drawn from a six-state HMM. The result is summarized in the Table 3.11.5.

model	k	$-l$	AIC	BIC
2-state HMM	4	736.9440	$\mathbf{1.4819\times10^3}$	$\mathbf{1.5047\times10^3}$
3-state HMM	9	736.8543	1.4917×10^3	1.5430×10^3
4-state HMM	16	736.7852	1.5056×10^3	1.5967×10^3
5-state HMM	25	736.7299	1.5235×10^3	1.6659×10^3
6-state HMM	36	736.7299	1.5455×10^3	1.7505×10^3

Table 3.11.5: AIC and BIC values of different Markov models (original sequence is a six-state binary HMP).

As clear from the table, both BIC and AIC choose a 2-state HMM as the best choice even though the data is generated from a six-state HMM. Now the ACFs of of 2-state and 6-state HMMs are compared with the sample data ACF. We don't see any improvement in the model by increasing the number of states from 2 to 6.

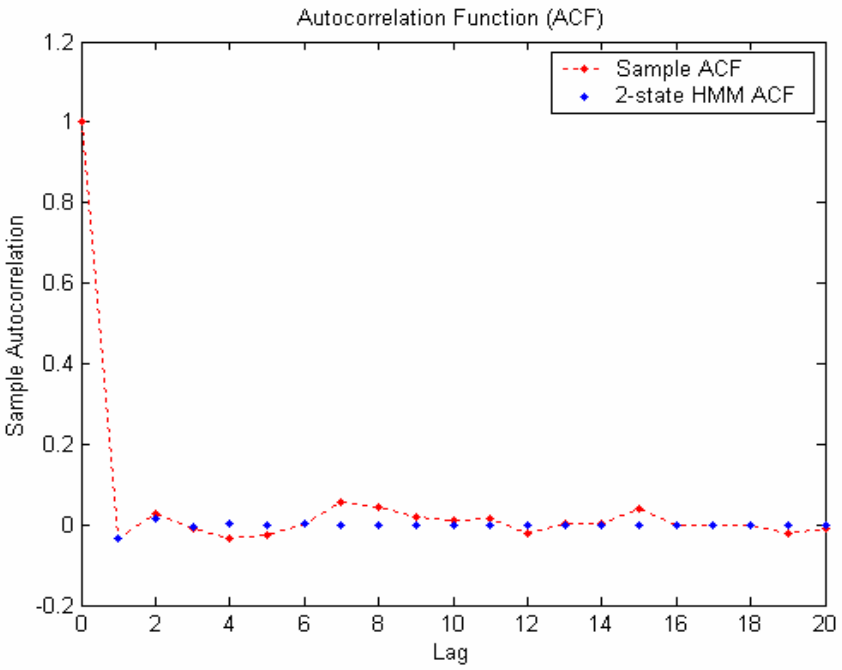


Figure 3.11.14: First 20 lags of sample ACF and 2-state HMM ACF. Sample ACF is obtained from the data generated by a six-state HMM.

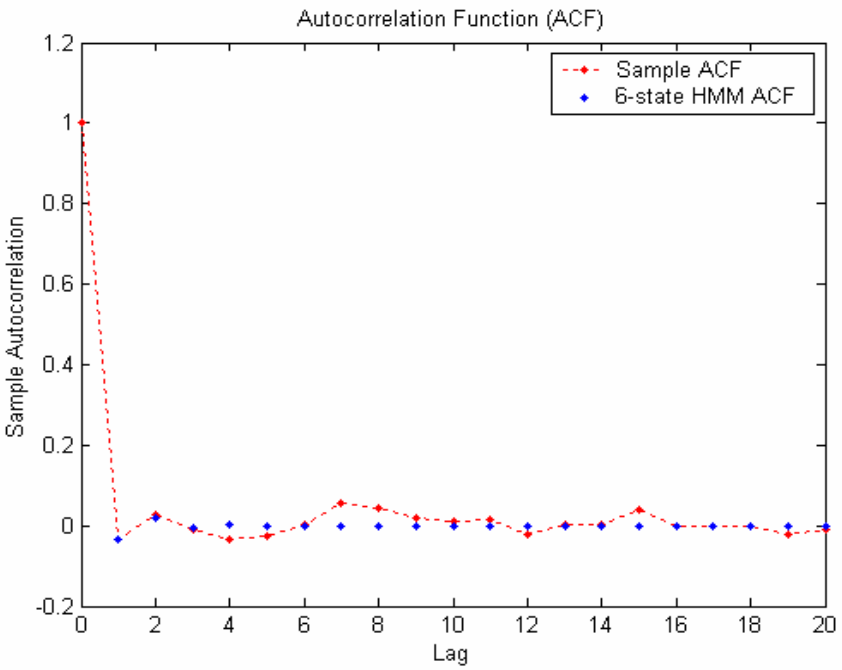


Figure 3.11.15: First 20 lags of sample ACF and 6-state HMM ACF. Sample ACF is obtained from the data generated by a six-state HMM.

Case 6 - Seven-State HMM

As a final test, we perform BIC and AIC order estimation on data generated by a seven-state HMM. Table 3.11.5 summarizes the results. Again AIC and BIC choose a 2-state HMM as the best candidate among other HMMs. We now plot some ACFs using the models from the Table 3.11.6. There is no significant difference between the ACFs plots of 2-state and 7-state HMMs.

model	k	$-l$	AIC	BIC
2-state HMM	4	734.4950	$\mathbf{1.4772\times10^3}$	$\mathbf{1.4974\times10^3}$
3-state HMM	9	734.4852	1.4872×10^3	1.5326×10^3
4-state HMM	16	734.4815	1.5012×10^3	1.5819×10^3

5-state HMM	25	734.4400	1.5176×10^3	1.6438×10^3
6-state HMM	36	734.4232	1.5396×10^3	1.7213×10^3
7-state HMM	49	734.4136	1.5672×10^3	1.8145×10^3
8-state HMM	64	734.4105	1.5956×10^3	1.9186×10^3

Table 3.11.6: AIC and BIC values of different Markov models (original sequence is a seven-state binary HMP).

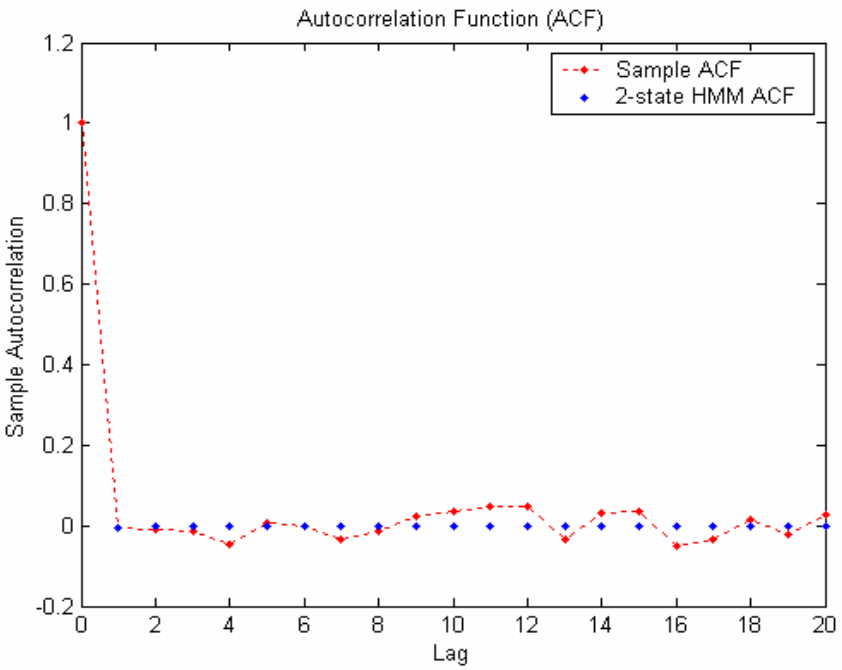


Figure 3.11.16: First 20 lags of sample ACF and 2-state HMM ACF. Sample ACF is obtained from the data generated by a seven-state HMM.

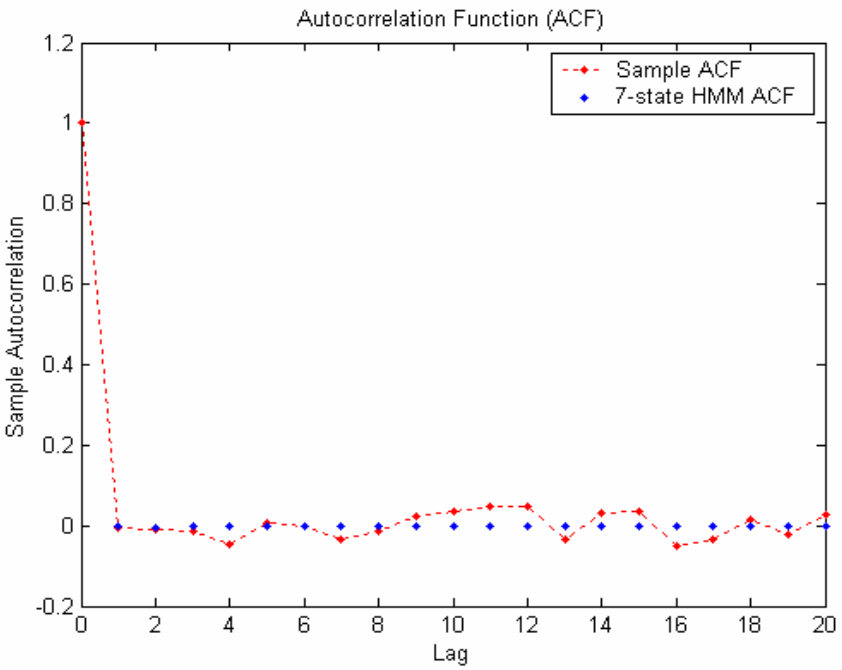


Figure 3.11.17: First 20 lags of sample ACF and 7-state HMM ACF. Sample ACF is obtained from the data generated by a six-state HMM.

3.12 Order Estimation of Binary Hidden Markov Rayleigh Fading Channel Models

In this section we discuss the order estimation of binary error sequences that are produced by wireless fading channels. BPSK modulation is used and channel fading is modeled as Rayleigh distributed. The noise is modeled as AWGN. We will show the effect of varying the sampling rate on a constant fading rate of 7Hz and the SNR of 10dB. Three different cases are presented with the varying sampling frequency of 0.5Kbps, 1Kbps, and 1Mbps.

First, let us introduce the test of stationarity (also called Runs Test) on a given observation sequence. This test is used in different fields of science and engineering to see if the observed data is stationary or not. A process Y_n is called strictly stationary if the distribution of $(Y_{p+1},...,Y_{p+k})$ is the same as that of $(Y_1,...,Y_k)$ for each values of p and k. A process Y_n is called second-order stationary if its mean $m_n = E(Y_n)$ is constant

(independent of n), and the autocovariance only depends on the time difference k for all n, i.e., $(Cov(k,n) = Cov(Y_n, Y_{n-k}) = Cov(k))$. The Runs Test uses simple hypothesis testing approach to make a decision. Here the hypothesis H_0 is the hypothesis that the given sequence is stationary where as the alternate hypothesis H_1 is the hypothesis that it is non-stationary. The Runs Test on a binary sequence is performed according to the following procedure [40]:

1. Define a run as a number of consecutive ones in the given binary sequence (usually referred as an error burst in our case).
2. Divide the trace into segments of equal lengths. The segment length can be arbitrarily chosen, as long as all the segments are of equal length.
3. Compute the lengths of runs in each segment.
4. Find the median value of the run length in the trace and count the number of runs of length above and below the median value.
5. Plot a histogram for the number of runs.

For a stationary sequence, the number of runs distribution between the 0.05 and 0.95 cut-offs should be close to 90% [98].

Case 1 - Rayleigh Fading with 7Hz Doppler Frequency, 0.5Kbps Sampling Frequency and 10dB SNR.

BPSK error statistics is obtained with the sampling frequency of 0.5Kbps operating in a Rayleigh fading channel having the fading rate of 7Hz, and the SNR is set to be 10dB. The fading envelope is shown in Fig. 3.12.1. A portion of this fading envelope for a period of 0.1sec is shown in Fig. 3.12.2.

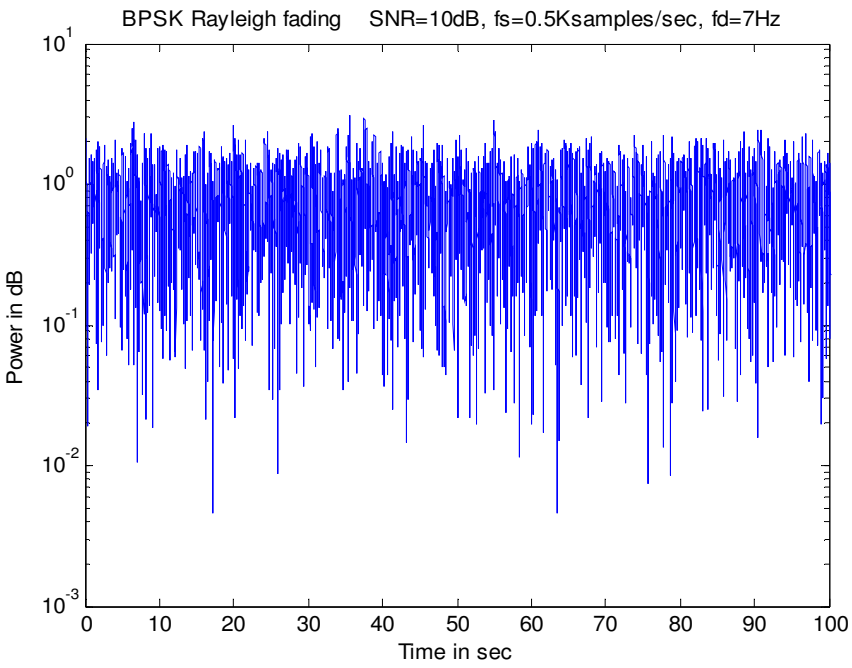


Figure 3.12.1: Rayleigh fading envelope, $f_d = 7\text{Hz}$, $f_s = 0.5\text{Kbps}$, and $\text{SNR} = 10\text{dB}$.

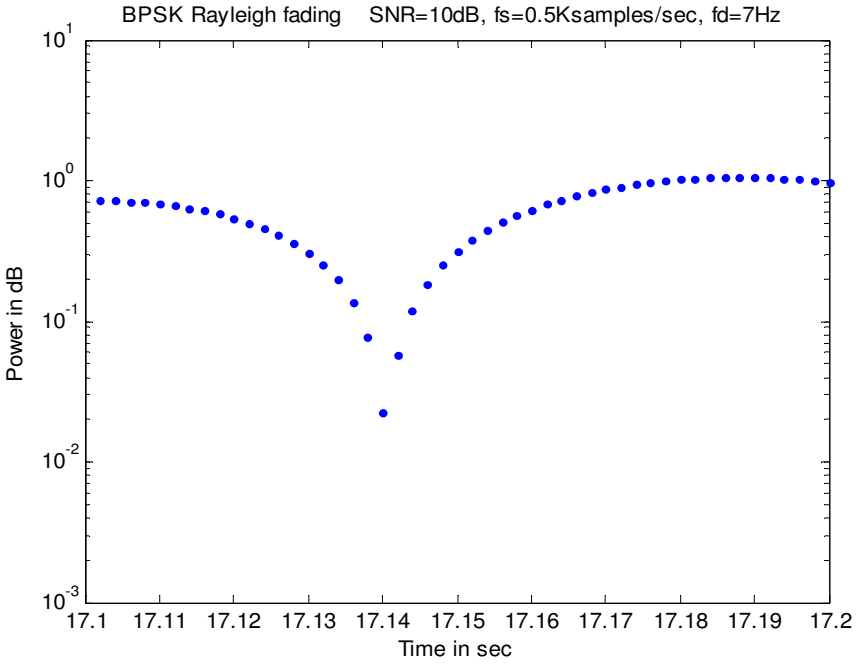


Figure 3.12.2: A segment of Rayleigh fading envelope from 17.1sec to 17.2sec, $f_d = 7\text{Hz}$, $f_s = 0.5\text{Kbps}$, and $\text{SNR} = 10\text{dB}$.

The error trace is obtained and is tested for stationarity. Fig. 3.12.3 shows that the test accepts the hypothesis that the given data is stationary as the PDF lies between 5% and 95% cutoffs (1.00 and 19.00 in this case).

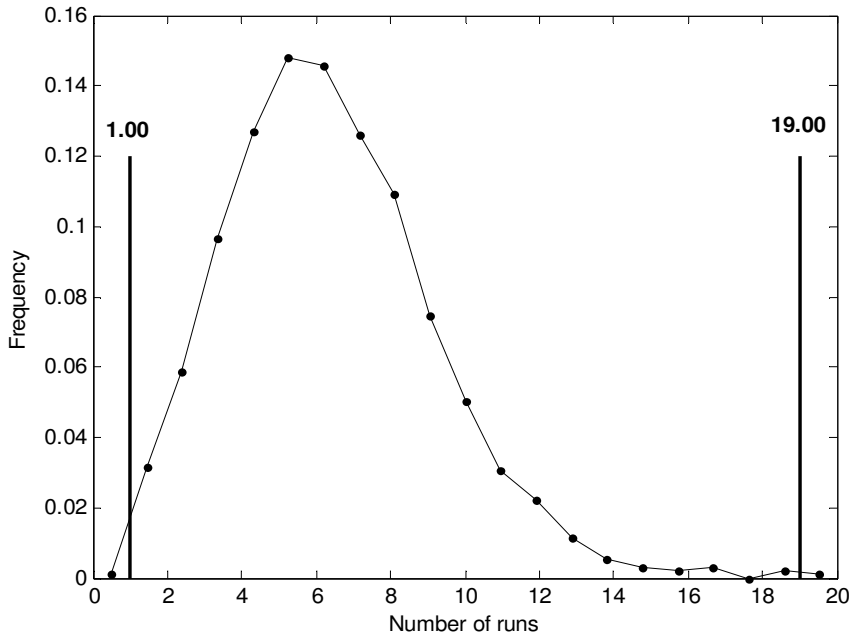


Figure 3.12.3: The Runs Test applied to the error trace obtained using BPSK with $f_d = 7\text{Hz}$, $f_s = 0.5\text{Kbps}$, and $\text{SNR} = 10\text{dB}$.

The error trace obtained from waveform level simulation is trained using the BWA. The algorithm is run multiple times to avoid local maxima and saddle points. The best model is chosen on the basis of the log-likelihood values obtained using the procedure discussed in section 3.3. Split data log-likelihood is used to evaluate the performance of different models. The sample autocorrelation of the original error trace is compared with the autocorrelation of the 2-state and 3-state HMMs. The resulting plots in Fig. 3.12.4 show a very close match. Table 3.12.1 summarizes AIC and BIC simulation results of 2-state and 3-state HMMs. AIC does not differentiate between order-2 and order-3 whereas BIS chooses order-2 as the optimum order.

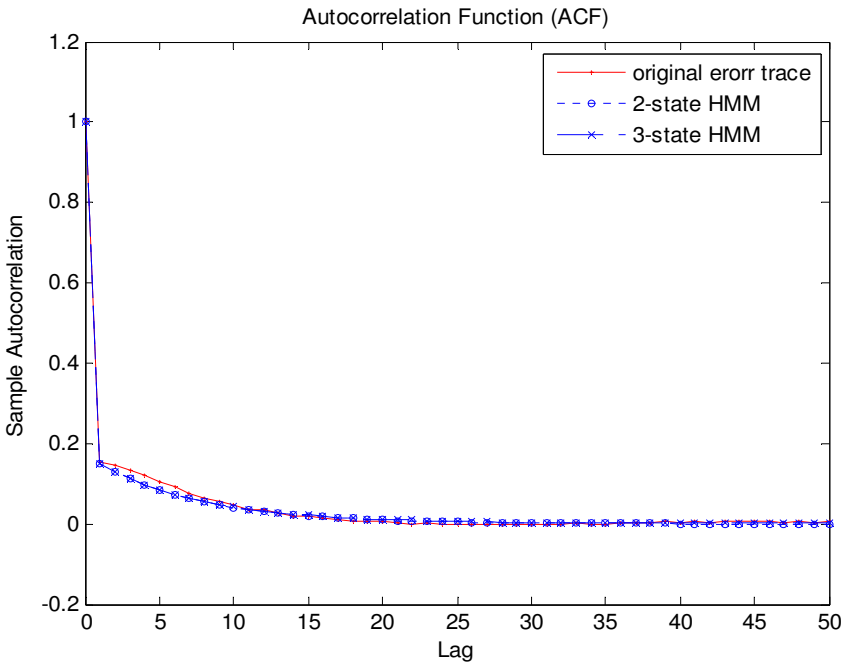


Figure 3.12.4: Comparison of autocorrelations of the original data with a 2-state and a 3-state HMM. Original error trace is obtained using BPSK simulation with $f_d = 7\text{Hz}$, $f_s = 0.5\text{Kbps}$, and $\text{SNR} = 10\text{dB}$.

model	k	$-l$	AIC	BIC
2-state HMM	4	1.57280×10^5	$\mathbf{3.14568\times10^5}$	$\mathbf{3.14615\times10^5}$
3-state HMM	9	1.57275×10^5	$\mathbf{3.14568\times10^5}$	3.14674×10^5

Table 3.12.1: AIC and BIC values of different HMMs (original sequence is an error trace obtained from BPSK with Rayleigh fading simulation with 7Hz doppler frequency, 0.5Kbps sampling frequency, and 10dB SNR).

Case 2 - BPSK Rayleigh Fading with 7Hz Doppler Frequency, 1Kbps Sampling Frequency and 10dB SNR.

We simulated a BPSK Rayleigh fading scenario with the fading rate of 7Hz, the sampling frequency of 1Kbps and with the SNR of 10dB. Fig. 3.12.5 shows a portion of the fading envelope from 18.85 to 19.85 sec. The Runs Test is performed on the data to check the stationarity in the error sequence. The test accepts the hypothesis that the data is stationary against the alternate hypothesis that the given data is non-stationary.

HMMs of order-2 and order-3 are used to model the channel error statistics and the resulting autocorrelations are shown in Fig. 3.12.7. The plots show a close match. Table 3.12.2 summarizes AIC and BIC results.

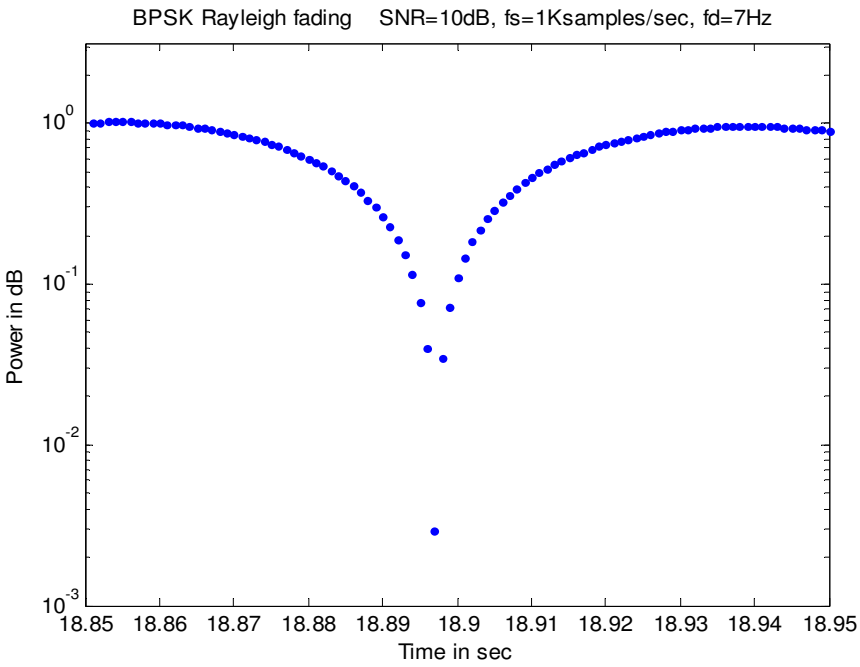


Figure 3.12.5: A segment of Rayleigh fading envelope from 18.85sec to 18.95sec, $f_d = 7\text{Hz}$, $f_s = 1\text{Kbps}$, and $\text{SNR} = 10\text{dB}$.

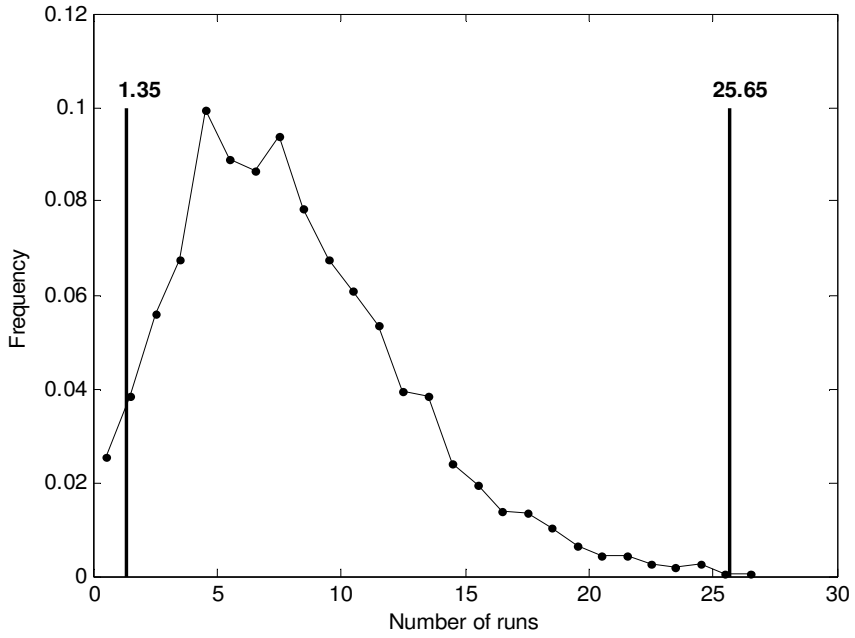


Figure 3.12.6: The Runs Test applied to the error trace obtained using BPSK with $f_d = 7\text{Hz}$, $f_s = 1\text{Kbps}$, and $\text{SNR} = 10\text{dB}$.

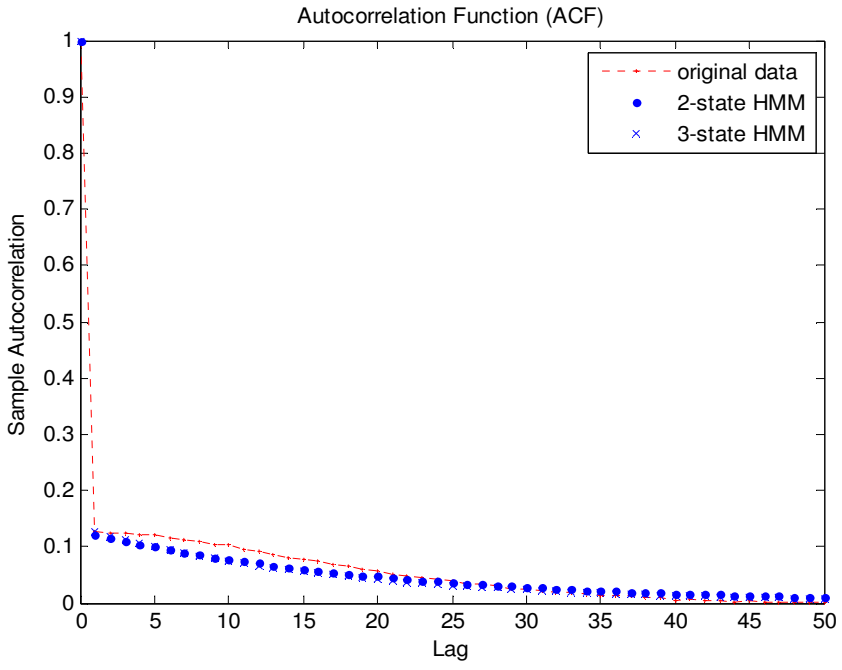


Figure 3.12.7: ACF comparison of the original data with a 2-state HMM and a 3-state HMM. Original trace is obtained using BPSK simulation with $f_d = 7\text{Hz}$, $f_s = 1\text{Kbps}$, and $\text{SNR} = 10\text{dB}$.

model	k	$-l$	AIC	BIC
2-state HMM	4	3.12710×10^5	$\mathbf{6.25428\times10^5}$	$\mathbf{6.25475\times10^5}$
3-state HMM	9	3.12932×10^5	6.25882×10^5	6.25989×10^5

Table 3.12.2: AIC and BIC values of different HMMs (original error trace is obtained using BPSK simulation with $f_d = 7\text{Hz}$, $f_s = 1\text{Kbps}$, $\text{SNR} = 10\text{dB}$).

Case 3 - BPSK Rayleigh Fading with 7Hz Doppler Frequency, 1Mbps Sampling Frequency and 10dB SNR.

In case 3 we simulate a BPSK communications system with 1Mbps sampling frequency. The channel is modeled as Rayleigh fading having the fading rate of 7Hz and the noise is assumed AWGN. The SNR is set to be 10dB. Two portions of the fading envelopes, each of duration 0.1sec, are shown in Figs. 3.12.8 and 3.12.9. We can see that the system in this case processes a very large amount of data, as compared to previous cases, in a short period of 0.1sec. This results in very long stretches of zeros when the system is in good state. This is because of the slow fading rate and using very high sampling frequency with the SNR of 10dB. Similarly when the channel is in fade, the system experiences long stretches of traces having zeros and ones, distributed according to a certain probability distributions. Since these segments of the error trace contain memory, and because of slow fading and high sampling frequency, the channel in good state is expected to remain in a good state for a very long time before transitioning to a bad state and vice versa.

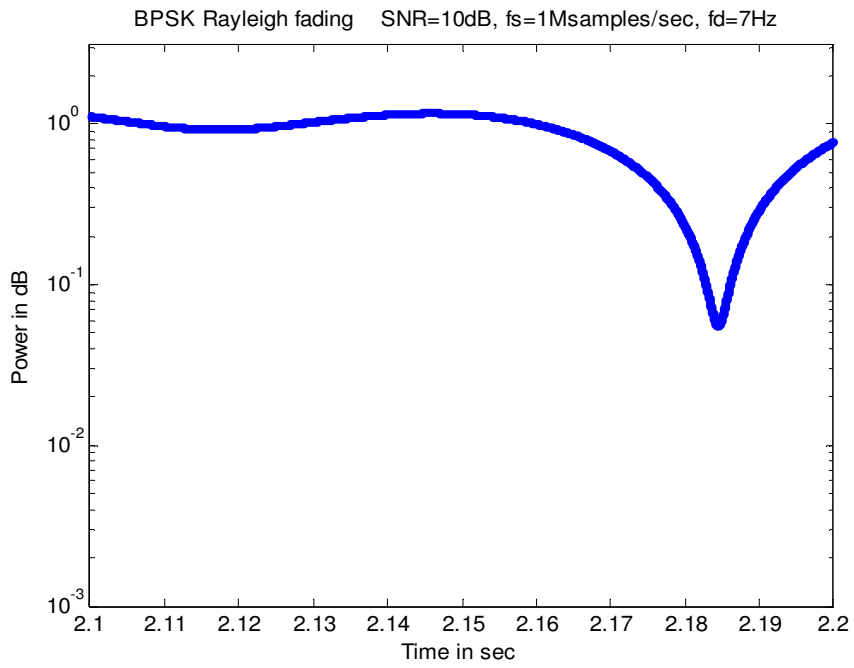


Figure 3.12.8: A segment of Rayleigh fading envelope from 2.1sec to 2.2sec, $f_d = 7\text{Hz}$, $f_s = 1\text{Mbps}$, and $\text{SNR} = 10\text{dB}$.

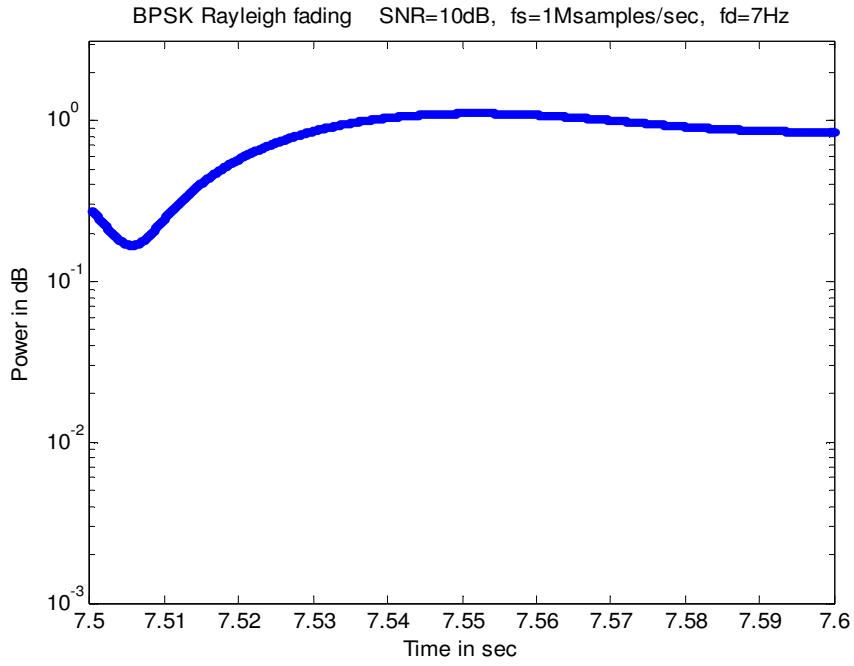


Figure 3.12.9: A segment of Rayleigh fading envelope from 7.5sec to 7.6sec, $f_d = 7\text{Hz}$, $f_s = 1\text{Mbps}$, and $\text{SNR} = 10\text{dB}$.

The error sequence generated from the waveform level simulation is tested for stationarity and the resulting histogram is shown in Fig. 3.12.10. As seen from the plot, the test rejects the hypothesis that the given error sequence is a stationary sequence.

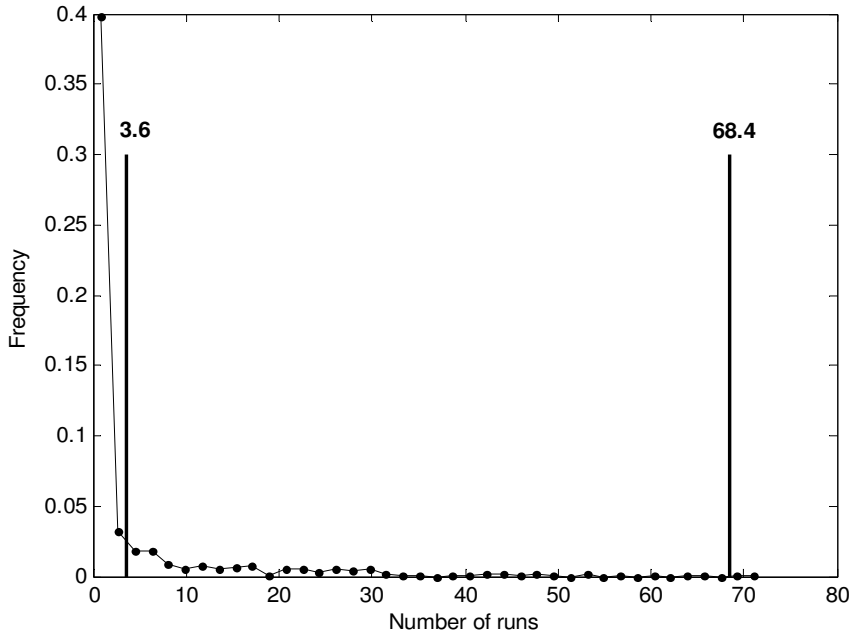


Figure 3.12.10: The Runs Test applied to the error trace obtained using BPSK with $f_d = 7\text{Hz}$, $f_s = 1\text{Mbps}$, and $\text{SNR} = 10\text{dB}$.

We obtained different HMMs from this error sequence and plot their autocorrelations to compare them with the autocorrelation of the original sequence. The results are shown in Fig. 3.12.11. We found that no HMM fit the given data with considerable accuracy. Table 3.12.3 summarizes the simulation results. This could be because of overestimating the number of states of the HMM. It is sometimes found that large number of states doesn't collect the statistics of the observation sequence.

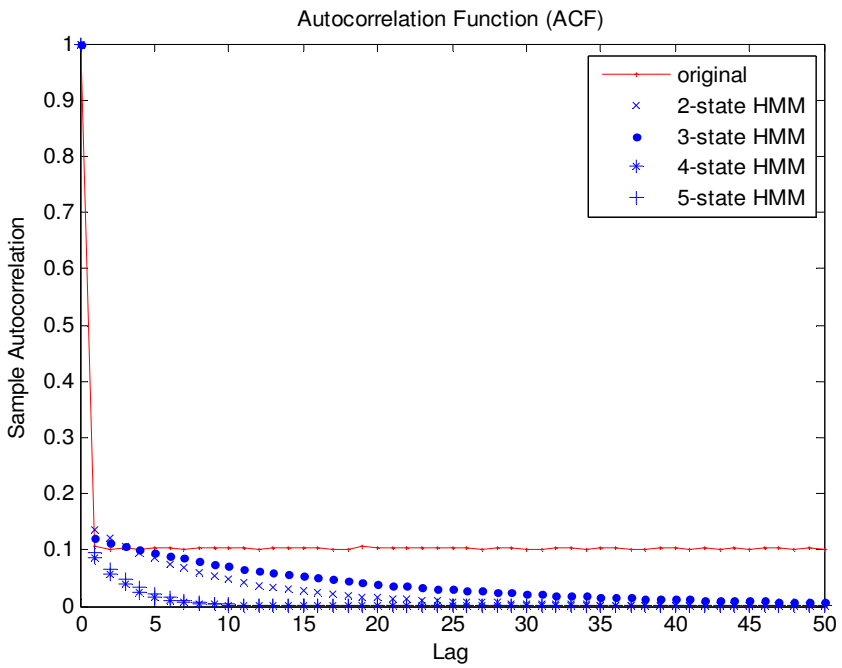


Figure 3.12.11: Comparison of autocorrelations of the original data with 2-state, 3-state, 4-state, and 8-state HMMs. Original error trace is obtained using BPSK simulation with $f_d = 7\text{Hz}$, $f_s = 1\text{Mbps}$, and $\text{SNR} = 10\text{dB}$.

model	k	$-l$	AIC	BIC
2-state HMM	4	1.41891×10^5	**2.83786×10^5**	**2.83809×10^5**
3-state HMM	9	1.41887×10^5	2.83792×10^5	2.83898×10^3
4-state HMM	16	1.58530×10^5	3.17093×10^5	3.17282×10^3
5-state HMM	25	1.53818×10^5	3.07687×10^5	3.07982×10^3

Table 3.12.3: AIC and BIC values of different Markov models (original error trace is obtained using BPSK simulation with $f_d = 7\text{Hz}$, $f_s = 1\text{Mbps}$, and $\text{SNR} = 10\text{dB}$).

Fig. 3.12.12 shows the comparison of the probability of the error-free intervals of the original and HMM generated error sequences. The distributions obtained from the

HMM generated error sequences do not follow the distribution of the error-free intervals of the original trace.

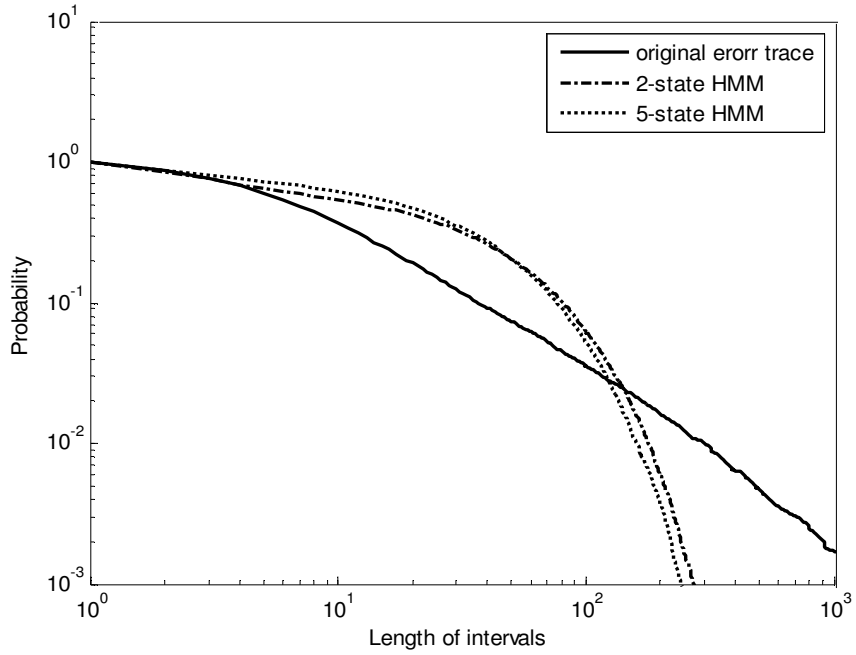


Figure 3.12.12: Distribution of the error-free intervals of the original and HMM generated error sequences.

Table 3.12.4 shows the frequency of error bursts of different lengths in the original trace and the traces generated by the 2-state and the 5-state HMM. It can be seen that the frequencies of the error bursts of the original trace vary widely with those generated by HMMs.

	length 1	length 2	length 3	length 4
original	34389	3686	420	43
2-HMM	26940	4300	838	171
5-HMM	27649	3920	1086	268

Table 3.12.4: Comparison of error bursts of different lengths in the original and HMM-generated trace.

As mentioned before that semi-HMMs are gaining popularity in modeling telecommunication systems, we train two semi-HMMs, one with four states (two good and two bad) and the other having eight states (four good and four bad), using the runlength vector of the original error trace. Several semi-HMMs are obtained and the best models are chosen based on the split-data likelihood values. ACF comparison is shown in Fig. 3.12.13. We see that neither model can fit the data with reasonable accuracy. This is primarily because of the non-stationarity in the data.

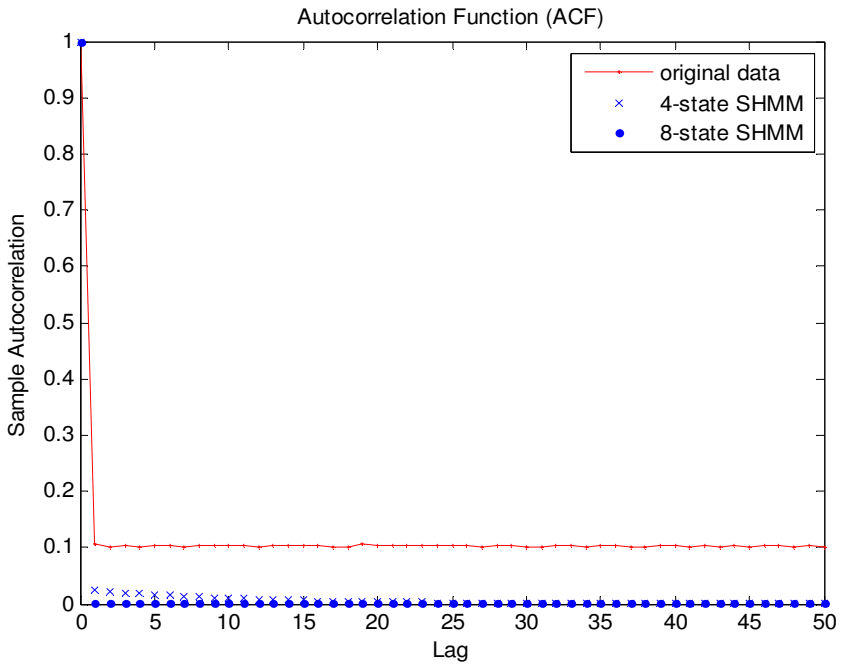


Figure 3.12.13: Comparison of autocorrelations of the original data with a 4-state semi-HMM and a 8-state semi-HMM. Original error trace is obtained using BPSK simulation with $f_d = 7\text{Hz}$, $f_s = 1\text{Mbps}$, and $\text{SNR} = 10\text{dB}$.

The error trace is then split into two traces, a good trace that contains zeros in runlength vector form and a bad trace that is obtained from the data when it is in fade. This procedure is further explained in Fig. 3.12.14. The length of bad trace is chosen by a

factor *C*, where *C* is the sum of mean and standard deviation of the error burst runlengths. This method of data splitting can be found in more detail in [40].

0000 … 0000000110 … 10111000000 … 000000001101 … 010111000…

| good trace | bad trace | *C* | good trace | bad trace | *C* |

Figure 3.12.14: Illustration of good and bad traces obtained from the original error trace.

Runs Test is performed on the bad trace and the test accept the stationary hypothesis. The resulting histogram is shown in Fig. 3.12.15.

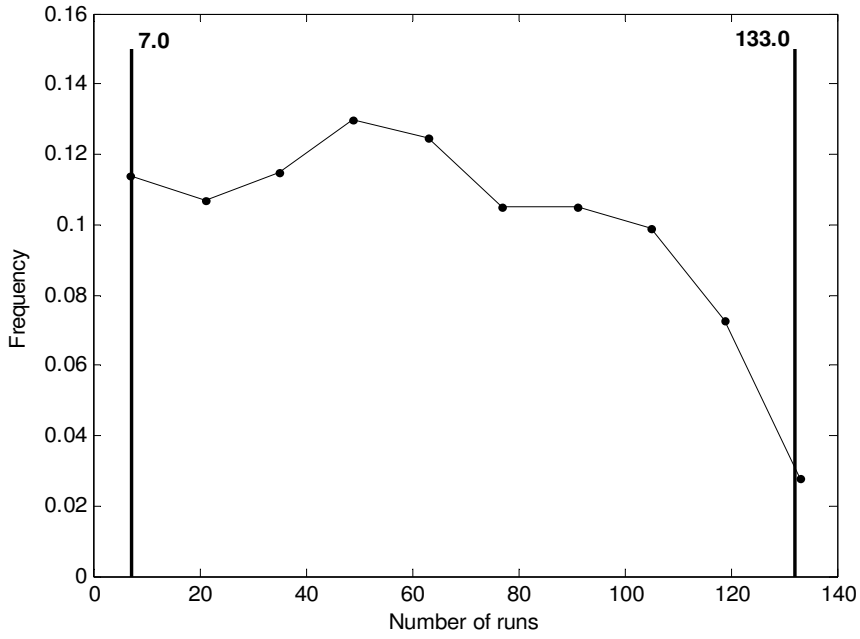


Figure 3.12.15: The Runs Test applied to the bad trace that is obtained from spitting the original error trace into good and bad traces.

We then train different HMMs to fit the bad trace and found out that a 2-state HMM is not an appropriate model to reproduce such data. It is also observed that the models perform better as we increase the number of hidden states. The 5-state, 10-state and 15-state HMMs are found to be better models, among all others, to reproduce the bad trace. ACF comparison is shown in Figs. 3.12.16 – 20. The runlengths of good trace is found to be best modeled by an inverse Gaussian random process. The comparison between the distribution of runlengths of the good trace and inverse Gaussian random process is shown in Fig. 3.12.21.

A binary sequence is produced by generating the runlengths of the good trace from the inverse Gaussian process and by generating the segments of bad traces using the 15-state HMM. ACF comparison of the two data sequences is shown in Fig. 3.12.22. Comparing with Figs. 3.12.11 and 3.12.13, the two plots show a close match. Table 3.12.5 summarizes the simulation results. BIC and AIC both choose the 5-state HMM as the best model.

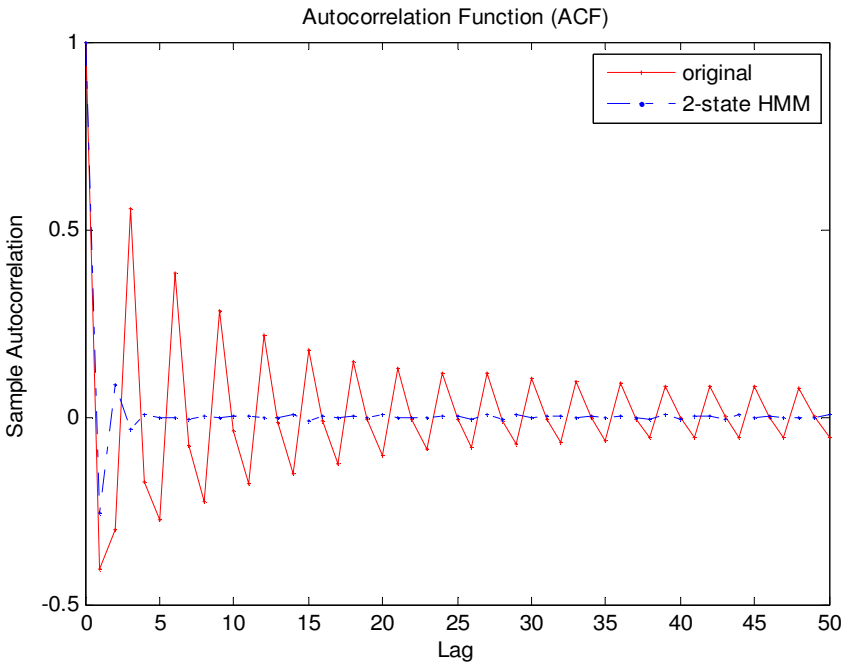


Figure 3.12.16: Comparison of autocorrelations of the bad data trace with a 2-state HMM.

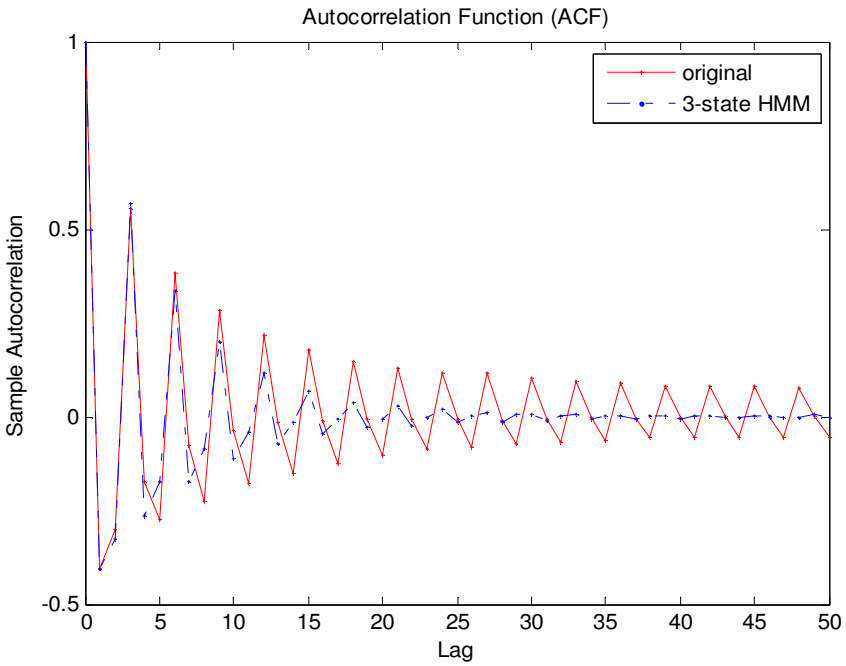


Figure 3.12.17: Comparison of autocorrelations of the bad data trace with a 3-state HMM

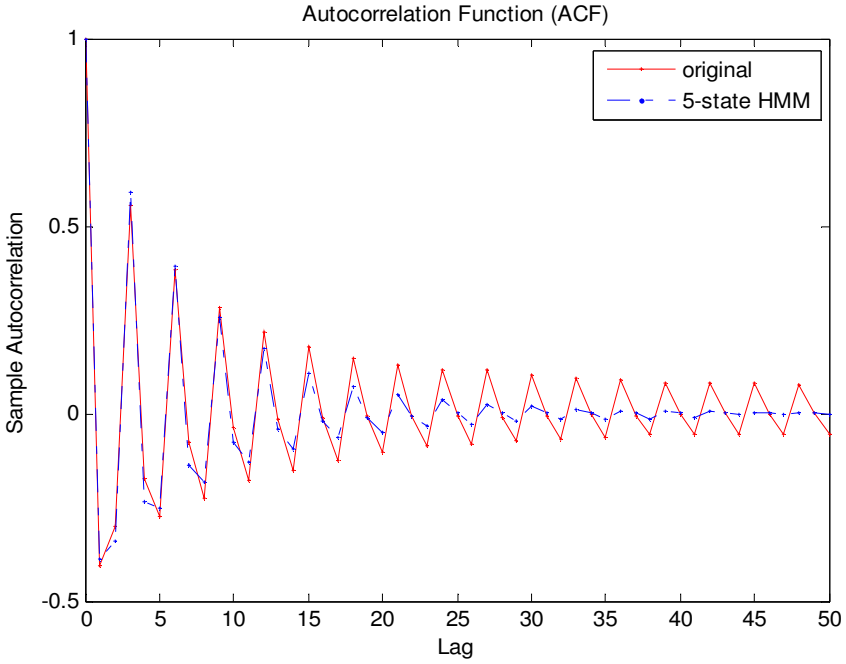


Figure 3.12.18: Comparison of autocorrelations of the bad data trace with a 5-state HMM.

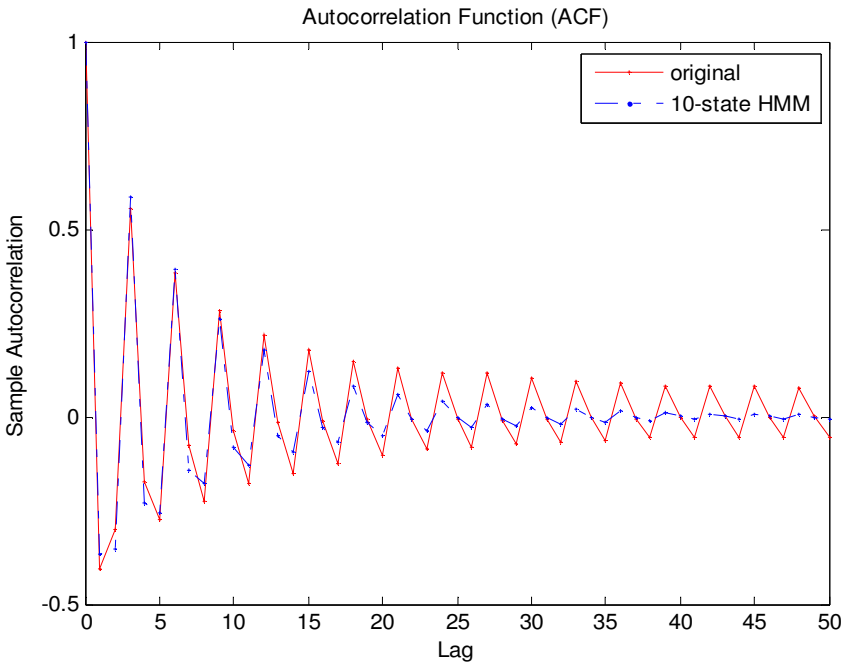


Figure 3.12.19: Comparison of autocorrelations of the original data with a 10-state HMM.

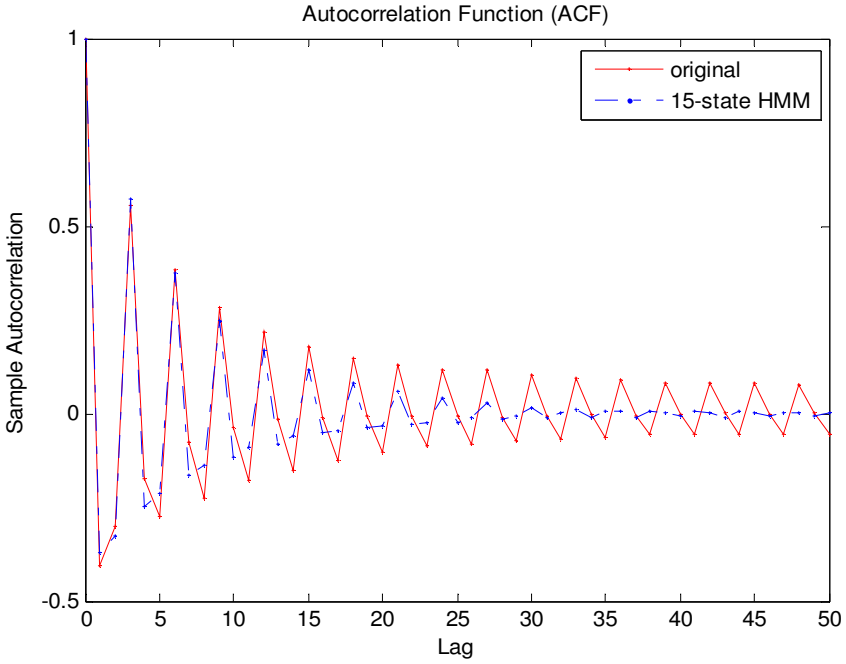


Figure 3.12.20: Comparison of autocorrelations of the original data with a 15-state HMM.

model	k	$-l$	AIC	BIC
2-state HMM	4	5.7699×10^4	1.1541×10^5	1.1546×10^5
3-state HMM	9	3.1382×10^4	6.2783×10^4	6.2890×10^4
5-state HMM	25	2.9490×10^4	$\mathbf{5.9031\times10^4}$	$\mathbf{5.9327\times10^4}$
10-state HMM	100	3.0719×10^4	6.1639×10^4	6.2820×10^4
15-state HMM	225	3.1172×10^4	6.2795×10^4	6.5454×10^4

Table 3.12.5: AIC and BIC values of different Markov models (original sequence is binary bad trace).

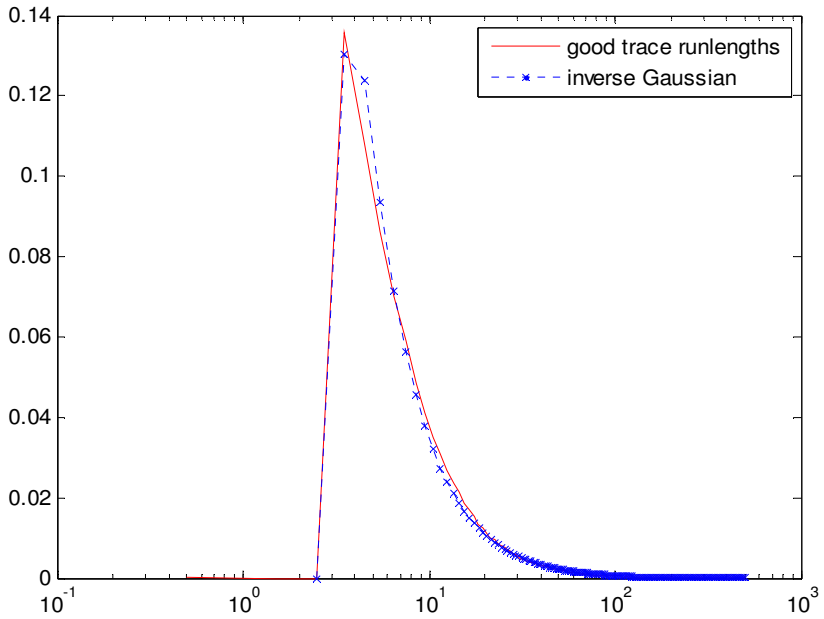


Figure 3.12.21: Comparison of runlength distributions of the good trace with the distribution of the inverse Gaussian process.

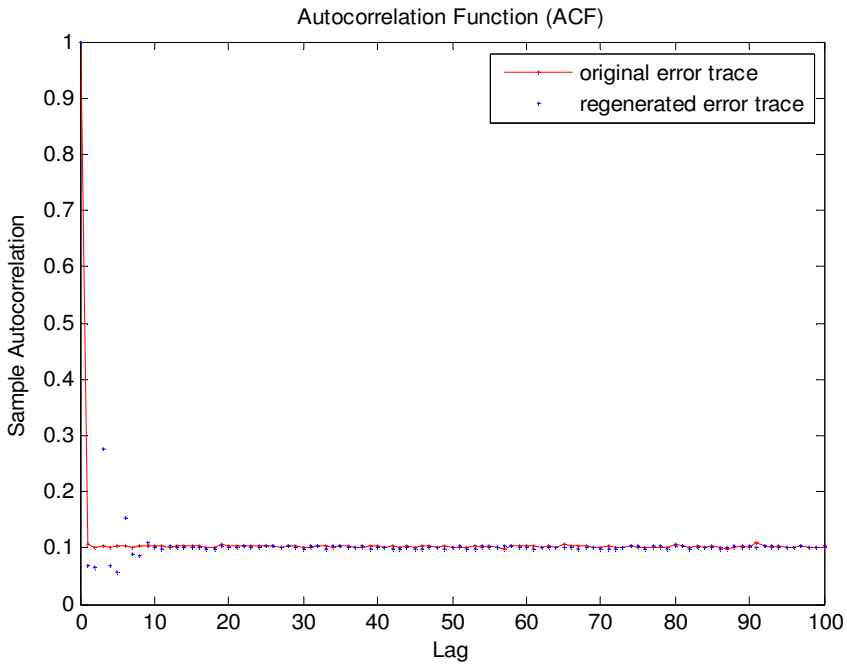


Figure 3.12.22: Comparison of autocorrelations of the original data with the data generated using an inverse Gaussian process and a 15-state HMM.

Fig. 3.12.23 shows the distribution of the error-free intervals of the original and regenerated sequences. The plots show a close match. Table 3.12.6 shows the frequency of error bursts in the two sequences. Comparing with Table 3.12.4, the frequencies of these intervals are found to be much closer in this case. Semi-HMMs are also used to model the bad trace. Fig 3.12.24 shows the ACFs of the bad trace and the ACFs of the 4-state and 10-state semi-HMMs. Even though we run the forward-backward fast BWA multiple time to avoid local maxima in both cases, It is interesting to find that both semi-HMMs fail to model the bad trace.

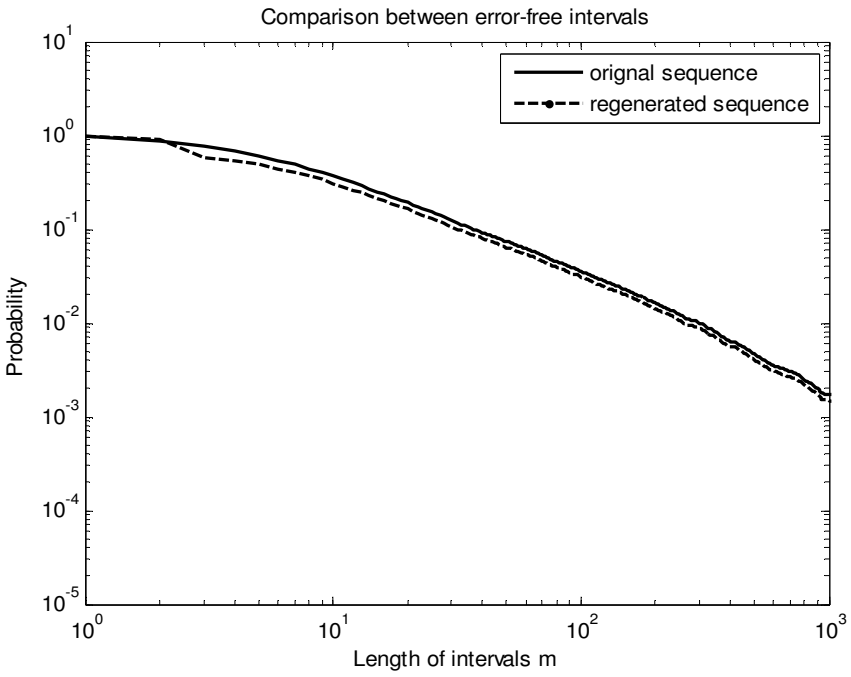


Figure 3.12.23: Distribution of the error-free intervals of the original trace and the trace produced by the inverse Gaussian process and the 15-state HMM.

	length 1	length 2	length 3	length 4
original	34389	3686	420	43
regen.	35213	3711	432	78

Table 3.12.6: Comparison of error bursts of different lengths in original and the trace generated from the inverse Gaussian process and the 15-state HMM.

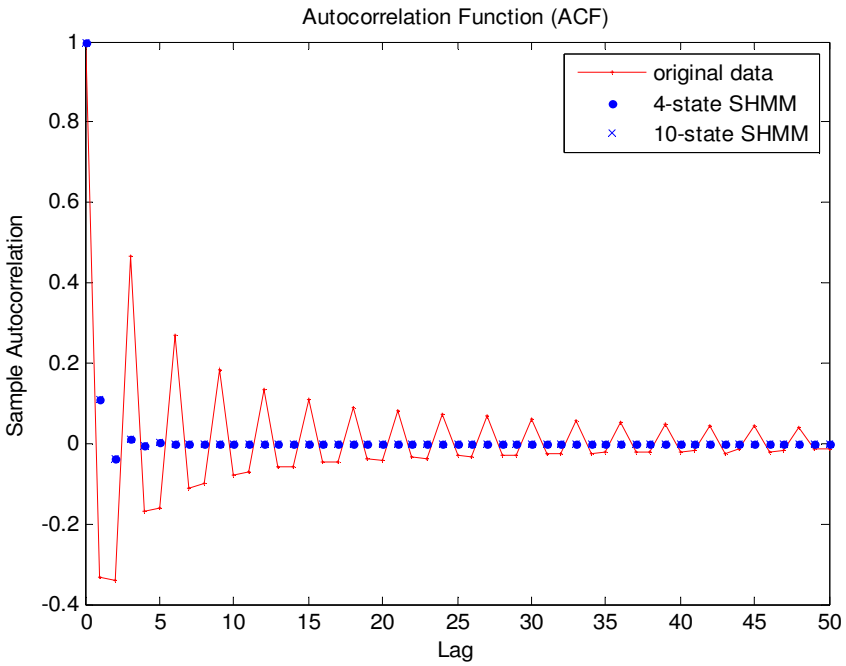


Figure 3.12.24: Comparison of autocorrelations of the bad data trace with the 4-state and 10-state semi-HMMs.

3.13 Neural Network and its Connection With the HMM

In [43], Chapter 17, the authors have studied some connections between Artificial Neural Networks (ANNs) [41] and HMMs. ANN, also known as a Simulated Neural Network (SNN) or just Neural Network (NN) is an interconnected group of artificial neurons that uses a mathematical or computational model for information processing based on a connectionist approach to computation. Connectionist approaches base their models upon the known neurophysiology of the brain and attempt to incorporate those functional properties thought to be required for cognition. Connectionist systems rely on parallel processing of sub-symbols, using statistical properties instead of logical rules to transform information. In most cases the ANN is an adaptive system that changes its structure based on external or internal information that flows through the network. They are nonlinear statistical data modeling tools that can be used to model complex relationships between inputs and outputs or to find patterns in data.

ANNs, like people, learn by example. They are configured for a specific application, such as pattern recognition or data classification, through a learning process. Since ANNs work very well in identifying patterns or trends in data, they are used in variety of applications such as sales forecasting, industrial process control, customer research, data validation, risk management, and target marketing. A typical ANN with four inputs and one output is shown in Fig. 3.13.1. For more details about ANN, refer to [41].

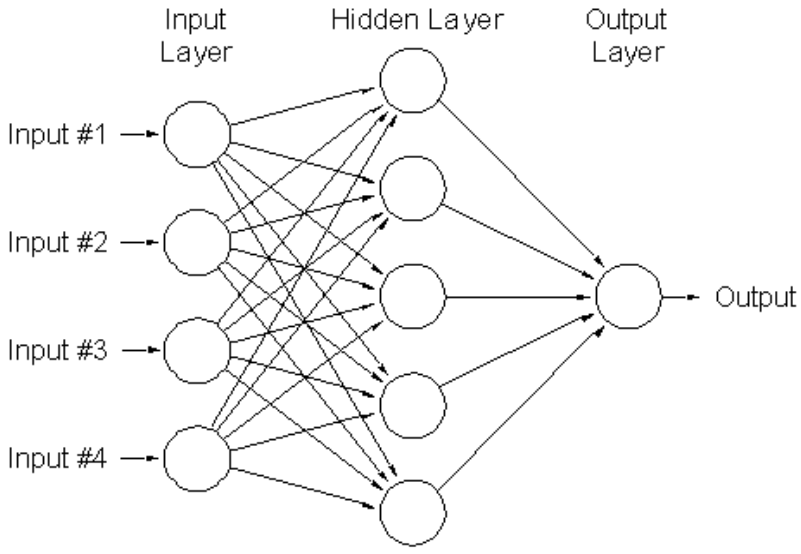


Figure 3.13.1: A four inputs, one output and a single hidden layer ANN.

ANNs are strikingly similar to HMMs. [43] talks about the relationships between HMMs and feed-forward ANNs. This paper actually discusses some HMMs that are defined by ANNs and also certain stochastic ANNs from a hidden variables model in which a threshold logic unit is hidden along with each state of the process. ANNs are usually trained using the back propagation algorithm, which is very similar to the BWA. For further study on the similarities between back propagation and the BWA, refer to [42].

We are not aware of any mathematical proof in literature but it is a well known fact that we generally do not require more than two hidden layers to train an ANN. In this work on HMM order estimation of stationary data sequence, we have determined that usually only two hidden states are required to model any arbitrary binary time series. Case 3 of section 3.12 (the case with the sampling rate of 1Mbps) should be treated as a special case where we constructed a bad trace by concatenating short segments of error sequences and the models with higher number of states gives a better fit. Hence there is likely to be a strong relationship between the number of hidden layers of ANN and the number of hidden states of HMM.

3.14 Inference from the Simulation Results

Our study so far concludes that we can model the fade duration of fading envelopes (both flat and frequency selective) by using Markov processes. A thorough performance evaluation of HMMs that was developed using the FB-BWA and the FO-BWA shows that the accuracy of the FO-BWA (with respect to BER, error-free and error intervals) is comparable to that of the FB-BWA. However, the computational complexity of the FO-BWA is found to be significantly less than that of the FB-BWA.

The goodness of fit approach to estimate the order of the MC is found to be consistent but requires a prohibitive amount of simulation runtime to process all the data sequences. Also, we need a large number of sequences to obtain reliable test results and cannot use this method with the limited number of sequences. BIC and AIC, on the other hand, provide accurate MC order estimation and can accurately estimate the order based on a single observation sequence.

The lack of mathematical theory for HMM order estimation has led us to rely on computer simulations to estimate the order of binary HMMs. BIC and AIC seem to differentiate between Markov chain and hidden Markov sources and both methods usually select the 2-state HMM as the best choice if the original source is a stationary hidden Markov source (of any arbitrary order). For non-stationary source, no HMM and semi-HMM is found to model such data and we split the data into good trace and bad

trace to obtain a reliable mathematical model. Bad trace is found to be approximated best by the models with higher number of states.

3.15 Conclusion

We present a simulation based approach to the order estimation of discrete hidden Markov Rayleigh fading channels. It is found that for a stationary channels, a 2-state HMM is sufficient to represent the error trace. For non-stationary error-trace, a technique [40] of splitting the data into two different traces is used. By splitting the data, we can successfully represent the bad trace using HMMs and the good trace can be modeled by a standard random process.

We also study the order estimation of MCs using goodness of fit approach and found that the method can consistently estimate the correct order of MCs. The behavior of the BIC and AIC is also studied for MC and HMM order estimation.

Chapter 4

Signal Classification and Detection Using HMMs

This chapter deals with the sensing and classification of low power signals by a cognitive radio device. These problems (signal detection and signal classification) are among the two most challenging in CR [50], [51], [52]. Detection and classification of signals with very low SNR are crucial for proper CR functionality as they enable the CR to react and adapt to changes in the radio environment. In this work, spectral coherence, a key property of second-order cyclostationary spectral analysis, is investigated as a tool for detection and classification of signals in a CR environment. Many processes exhibit cyclostationarity, i.e., their statistical properties vary periodically with time. For example, in telecommunications, telemetry, radar, and sonar applications, periodicity is due to modulation, sampling, multiplexing, and coding operations. In mechanics it is due, for example, to gear rotation. In radio astronomy, periodicity results from revolution and rotation of planets and on the energy pulsation of stars. In econometrics, it is due to seasonality; and in atmospheric science it is due to the rotation and revolution of the earth.

A two-dimensional cycle frequency domain profile (also known as α-domain profile) is used here for detecting and classifying signals. A simple threshold based technique is employed for detection and the hidden Markov model, due of its well known pattern recognition capabilities, is used for classifying the type of signal after its successful detection. A similar approach is used in [95] where Artificial Neural Networks

(ANNs) are used to classify signals. We present the effects of varying the observation length in signal detection and classification. It is found that this cyclostationary based system can detect very weak signals and at the same time can correctly classify signal type.

4.1 Introduction

The radio spectrum is a precious resource that should be used carefully. It is however found that most of the radio spectrum has been utilized very inefficiently and over time, this has resulted in an apparent lack of radio spectrum for new technologies to be implemented [45]. Realizing the fact that the licensed spectrum remains unused most of the time, the Federal Communication Commission (FCC) is now considering to adopt unlicensed, rule-based strategies for spectrum allocation in certain frequency bands [46]. In May 2003, the FCC convened a workshop to examine the impact that cognitive radio could have on spectrum utilization and to study the practical regulatory issues that CR would raise. The FCC then expressed its desire to improve access to the radio spectrum through better utilization of time, frequency, power, bandwidth, and geographical space and its strong belief that CR holds the potential to accomplish these goals [47],[48].

Another challenge posed as wireless communication technology progresses and systems are being deployed is interoperability among systems operated by different entities, for example, different government agencies communicating with each other in emergency situations or communications between armed forces from different countries that are deployed at remote locations, etc.

Hence, we need to solve two challenging problems in order to respond to the increasing demand of more efficient communication systems, i.e., better spectrum utilization (through opportunistic spectrum access) and making seamless communications possible among different systems. Since most of the time, this opportunistic spectrum access and seamless communication exist without a centralized coordination, a radio needs to be able to continuously observe its environment, change its parameters, and at the same time make sure that its actions improves not only its own performance but also the performance of the network in which the radio operates. Cognitive radio is believed

to have the potential to solve these challenging problems due to its capability of opportunistic spectrum access and adeptly responding to the change in its operating environment.

Since the diversity and complexity of digital communication system is expected to increase significantly, the future CR should allow communications between wireless devices having heterogeneous radio access technologies over the unoccupied spectrum as an opportunistic access to the channel. To initiate communication by CR devices, the classification of the received signal becomes very important after successful signal detection, since the knowledge of signal types occupying spectrum bands, on which a CR attempts to operate, provides more comprehensive awareness to the CR. For example, in IEEE 802.22 WRAN systems [49], the CR is required to take actions differently according to the type of incumbent signal it detects. The CR is required to evacuate not only the channel whenever a TV signal appears, but it also must evacuate the two adjacent TV channels as well, and apply a spectral mask which specifies the allowable transmit EIRP levels for other TV channels it can use. On the other hand, in the case of a wireless microphone, a CR is required to evacuate only the channel where the wireless microphone signal appears and is allowed to use adjacent channels.

The rest of this chapter is organized as follows. First we briefly discuss the cyclostationary spectral analysis and provide a mathematical formulation of spectral autocoherence. The α-domain profile is then discussed. We next study the spectral autocoherence and the α-domain profile of different types of modulated signals. Simulation results are presented at the end of this chapter.

4.2 Cyclostationary Spectral Analysis

A cyclostationary process [53], [54], [57] [58], [59] is a random process with probabilistic parameters, such as the autocorrelation function, that vary periodically with time. Most signals (analog as well as digital) exhibit cyclostationarity because of the periodicities in these signals due to sampling, modulation, multiplexing, coding, etc [55], [56]. Cyclostationary spectral analysis is the study of the periodicities hidden in such signals.

The concept of cyclostationarity has been applied in many areas of signal processing including synchronization [60] - [63], interference cancellation [64], [65], filter design [66], [67], adaptive filtering [68], system identification [69], coding [70], [71], queueing [72], [73], and detection, etc. [74], [75]. The same concept is used in other fields such as biomedical engineering [76], climatology [77], hydrology [78], and prediction [79], [80], [81]. It is found that the spectral correlation is a characteristic property of wide-sense cyclostationarity (cyclostationarity of the autocorrelation) and the spectral correlation function is a generalization of the power spectral density function [55]. It is also found that different types of modulated signals (such as BPSK and QPSK) that have identical power spectral density can have distinct spectral correlation functions that can help in the identification and classification of these signals. Stationary noise and interference are found to exhibit no spectral correlation. Also, the spectral correlation function contains phase and frequency information related to timing parameters in modulated signals. For more information, refer to [91], [92], [93], [94].

A process $x(t)$ has the spectral correlation given by

$$S_{x_T}^{\alpha}(t,f)_{\Delta t} \triangleq \frac{1}{\Delta t}\int_{t-\Delta t/2}^{t+\Delta t/2}\frac{1}{T}X_T\left(w, f+\frac{\alpha}{2}\right)X_T^*\left(w, f-\frac{\alpha}{2}\right)dw \tag{4.2.1}$$

where

$$X_T(t,f) \triangleq \frac{1}{\Delta t}\int_{t-T/2}^{t+T/2}x(w)e^{-j2\pi fw}dw \tag{4.2.2}$$

for all values of the *spectral location parameter*

$$f = \frac{1}{2}\left[\left(f+\frac{\alpha}{2}\right)+\left(f-\frac{\alpha}{2}\right)\right] \tag{4.2.3}$$

and the *spectral separation parameter*

$$\alpha = \left(f+\frac{\alpha}{2}\right)+\left(f-\frac{\alpha}{2}\right) \tag{4.2.4}$$

of interest. The spectral location parameter is the frequency of the second-order periodicity and is commonly referred to as the *cycle frequency*.

Let's consider two functions $u(t)$ and $v(t)$ such that

$$u(t) \triangleq x(t)e^{-j\pi\alpha t} \tag{4.2.5}$$

$$v(t) \triangleq x(t)e^{j\pi\alpha t} \tag{4.2.6}$$

Their time-variant finite-time complex spectra $U_T(t,f)$ and $V_T(t,f)$ are given by

$$U_T(t,f) \triangleq \int_{t-T/2}^{t+T/2} u(w)e^{-j2\pi fw}dw \tag{4.2.7}$$

and

$$V_T(t,f) \triangleq \int_{t-T/2}^{t+T/2} v(w)e^{-j2\pi fw}dw\,. \tag{4.2.8}$$

The spectral correlation (4.2.1) is identical to the statistical (time-smoothed) cross-spectrum

$$S_{x_T}^{\alpha}(t,f) \equiv S_{uv_T}(t,f)_{\Delta t} \triangleq \frac{1}{\Delta t}\int_{t-\Delta t/2}^{t+\Delta t/2}\frac{1}{T}U_T(w,f)V_T^*(w,f)dw \tag{4.2.9}$$

Therefore, the parameter f is also referred to as the *spectrum frequency*.

The time-variant cross periodogram

$$S_{uv_T(t,f)} \triangleq \frac{1}{T}U_T(t,f)V_T^*(t,f) \equiv \frac{1}{T}X_T\left(t,f-\frac{\alpha}{2}\right)X_T^*\left(t,f-\frac{\alpha}{2}\right) \triangleq S_{x_T}^{\alpha}(t,f) \tag{4.2.10}$$

is time-smoothed to obtain the spectral correlation (4.2.1). It is also referred to as the *time-variant cyclic periodogram* (or the *time-variant finite-time cyclic spectrum*). As mentioned in [54] Chapter 7, the time variant cyclic-correlogram can be written as

$$R_{x_T}^{\alpha}(t,\tau) \triangleq \frac{1}{T}\int_{t-(T-|\tau|)/2}^{t-(T-|\tau|)/2} x\left(w+\frac{\tau}{2}\right)x\left(w-\frac{\tau}{2}\right)e^{-j2\pi\alpha w}dw \tag{4.2.11}$$

and the cyclic periodogram $S_{x_T}^{\alpha}(t,.)$ is the Fourier transform

$$S_{x_T}^{\alpha}(t,.) = F\{R_{x_T}^{\alpha}(t,.)\} \tag{4.2.12}$$

Equation (4.2.12) is called the *cyclic-periodogram-cyclic-correlogram relation*. Since the spectral correlation (4.2.1) is a time-smoothed version of the time-variant finite-time cyclic spectrum (4.2.10),

$$S_{x_T}^{\alpha}(t,f)_{\Delta t} = \frac{1}{\Delta t}\int_{t-\Delta t/2}^{t+\Delta t/2} S_{x_T}^{\alpha}(w,f)dw, \tag{4.2.13}$$

it is called a *statistical cyclic spectrum*. Using the approximate equivalence [54] (Chapter 11) the spectral correlation $S_{x_T}^{\alpha}(t,f)_{\Delta t}$, which is the temporal cross correlation (with temporal separation $\tau = 0$) of the time-functions

$$\frac{1}{\sqrt{T}}X_T\left(.,f+\frac{\alpha}{2}\right)\text{and }\frac{1}{\sqrt{T}}X_T\left(.,f-\frac{\alpha}{2}\right), \tag{4.2.14}$$

can be interpreted (approximately) as the spectral autocorrelation (with spectral separation α) of the frequency function

$$\frac{1}{\sqrt{T}} X_T(t,.) \tag{4.2.15}$$

namely

$$S_{x_{\Delta t}}^{\alpha}(t,f)_{\Delta f} \cong \frac{1}{T} \int_{f-\Delta f/2}^{f+\Delta f/2} \frac{1}{\Delta t} X_{\Delta t}\left(t, v+\frac{\alpha}{2}\right) X_{\Delta t}^{*}\left(t, v-\frac{\alpha}{2}\right) dv \tag{4.2.16}$$

here $\Delta f = 1/T$. Note that in order to idealize the spectral correlation, it is required that $\Delta t \to \infty$, whereas $\Delta f \to 0$. The two spectral cyclic spectra have a cycle resolution given by the reciprocal of the temporal resolution

$$\Delta\alpha \cong \frac{1}{\Delta t} \tag{4.2.17}$$

Therefore, the cyclic spectra can resolve second-order periodicity at two cycle frequencies, say α_1 and α_2, if and only if these two cycle frequencies are separated by more than the cycle resolution (4.2.17), i.e.,

$$|\alpha_1 - \alpha_2| > \Delta\alpha \tag{4.2.18}$$

Using the limit $\Delta t \to \infty$, we can write

$$\lim_{\Delta t\to\infty} S_{x_{\Delta t}}^{\alpha}(t,f)_{\Delta f} = \lim_{\Delta t\to\infty} S_{x_{1/\Delta f}}^{\alpha}(t,f)_{\Delta f} \tag{4.2.19}$$

$$= \frac{1}{\Delta f} \int_{-\infty}^{\infty} v_{1/\Delta f}(\tau) \hat{R}_x^{\alpha}(\tau) e^{j2\pi f\tau} d\tau \tag{4.2.20}$$

where $\hat{R}_x^{\alpha}(\tau)$ is the limit cross correlation. Note that

$$\hat{S}_x^{\alpha}(.) = F\{\hat{R}_x^{\alpha}(.)\} \tag{4.2.21}$$

The cyclic autocorrelation, according to [54] can be written as

$$\hat{S}_x^{\alpha}(f) = \lim_{\Delta f\to 0} \lim_{\Delta t\to\infty} \hat{S}_{x_{1/\Delta f}}^{\alpha}(t,f)_{\Delta t} \tag{4.2.22a}$$

or

$$\hat{S}_x^{\alpha}(f) = \lim_{\Delta f\to 0} \lim_{\Delta t\to\infty} \hat{S}_{x_{\Delta t}}^{\alpha}(t,f)_{\Delta f} \tag{4.2.22b}$$

Eq. (4.2.22a) gives the interpretation of the limit cyclic spectrum as an idealized version of the spectral correlation (4.2.1). Equation (4.2.22b) gives its alternative interpretation as an idealized version of a spectrally smoothed cross-spectrum.

According to [54] the limiting cross spectrum for two time series $x(t)$ and $y(t)$ is defined by

$$\hat{S}_{xy}^{\alpha}(f) = \lim_{\Delta f \to 0} \lim_{\Delta t \to \infty} \hat{S}_{xy_{1/\Delta f}}^{\alpha}(t, f)_{\Delta t} \tag{4.2.23a}$$

and

$$\hat{S}_{xy}^{\alpha}(f) = \lim_{\Delta f \to 0} \lim_{\Delta t \to \infty} \hat{S}_{xy_{\Delta t}}^{\alpha}(t, f)_{\Delta f} \tag{4.2.23b}$$

The *time-variant cyclic cross periodogram* (or *cyclic finite-time cross spectrum*) is therefore given by

$$S_{xy_T}^{\alpha}(t, f) \triangleq \frac{1}{T} X_T\left(t, f + \frac{\alpha}{2}\right) Y_T^*\left(t, f - \frac{\alpha}{2}\right) \tag{4.2.24}$$

and, by using the analogy, $S_{xy_T}^{\alpha}(t, f)$ can be characterized by the Fourier transform

$$S_{xy_T}^{\alpha}(t, .) = F\{R_{xy_T}^{\alpha}(t, .)\} \tag{4.2.25}$$

for which the function

$$R_{xy_T}^{\alpha}(t, \tau) \triangleq \frac{1}{T} \int x\left(w + \frac{\tau}{2}\right) y\left(w - \frac{\tau}{2}\right) e^{-j2\pi\alpha w} dw \tag{4.2.26}$$

is the *time-variant cyclic cross correlogram*. The limit statistics exhibit the symmetries given by

$$\hat{R}_{xy}^{\alpha}(-\tau) = \hat{R}_{yx}^{\alpha}(\tau) \tag{4.2.27a}$$

$$\hat{R}_{xy}^{-\alpha}(\tau) = \hat{R}_{yx}^{\alpha}(\tau)^* \tag{4.2.27b}$$

$$\hat{S}_{xy}^{\alpha}(-f) = \hat{S}_{yx}^{\alpha}(f) \tag{4.2.27c}$$

$$\hat{S}_{xy}^{-\alpha}(f) = \hat{S}_{yx}^{\alpha}(f)^* \tag{4.2.27d}$$

The limit cyclic spectrum $\hat{S}_x^{\alpha}(f)$ in (4.2.22a) is the limit as $\Delta f \to 0$

$$\lim_{\Delta t \to \infty} \frac{1}{\Delta t} \int_{-\Delta t/2}^{\Delta t/2} \Delta f \; X_{1/\Delta f}\left(t, f + \frac{\alpha}{2}\right) X_{1/\Delta f}^*\left(t, f - \frac{\alpha}{2}\right) \tag{4.2.28}$$

of the two spectral components

$$\sqrt{\Delta f}\, X_{1/\Delta f}\left(t, f + \frac{\alpha}{2}\right) \text{ and } \sqrt{\Delta f}\, X_{1/\Delta f}\left(t, f - \frac{\alpha}{2}\right). \tag{4.2.29}$$

In fact, since the mean of each of these frequency components is zero,

$$\lim_{\Delta t\to\infty}\frac{1}{\Delta t}\int_{-\Delta t/2}^{\Delta t/2}\Delta f\; X_{1/\Delta f}\left(t, f\pm\frac{\alpha}{2}\right)dt = 0\,. \tag{4.2.30}$$

Assuming only that either $f\pm\alpha/2\neq 0$ or $x(t)$ contains no first-order periodicity with frequencies $f\pm\alpha/2$, then $\hat{S}_x^{\alpha}(f)$ is actually the limit, as $\Delta f\to 0$, of the temporal *covariance* of the two spectral components (4.2.29). Normalization of this temporal covariance by the geometric mean of the two temporal variances

$$\lim_{\Delta t\to\infty}\frac{1}{\Delta t}\int_{-\Delta t/2}^{\Delta t/2}\Delta f\left|X_{1/\Delta f}\left(t, f\pm\frac{\alpha}{2}\right)\right|^2 dt \tag{4.2.31}$$

gives the temporal *correlation coefficient* of the two spectral components (4.2.27). But, the limit as $\Delta f\to 0$ of these two temporal variances yields the two limit spectra

$$\hat{S}_x\left(f\pm\frac{\alpha}{2}\right)=\lim_{\Delta f\to 0}\lim_{\Delta t\to\infty}\frac{1}{\Delta t}\int_{-\Delta t/2}^{\Delta t/2}\Delta f\left|X_{1/\Delta f}\left(t, f\pm\frac{\alpha}{2}\right)\right|^2 dt \tag{4.2.32}$$

Consequently, the limit as $\Delta f\to 0$ of the temporal correlation coefficient of the two spectral components (4.2.29) is given by

$$\hat{C}_x^{\alpha}(f)\triangleq\frac{\hat{S}_x^{\alpha}(f)}{[\hat{S}_x^{\alpha}(f+\alpha/2)\hat{S}_x^{\alpha}(f-\alpha/2)]^{1/2}} \tag{4.2.33}$$

which is called the *spectral autocoherence* (or simply *autocoherence*) of $x(t)$ at the cycle frequency α and spectrum frequency *f*. It follows from the relation (4.2.33) that

$$|\hat{C}_x^{\alpha}(f)|\leq 1\,. \tag{4.2.34}$$

Moreover, since the magnitude of the correlation coefficient can equal unity if and only if the two variables (time series) are linearly dependent, then the autocoherence magnitude can equal unity at some values of *f* and α,

$$|\hat{C}_x^{\alpha}(f)|=1 \tag{4.2.35}$$

if and only if the two time series (4.2.30) are linearly dependent (in the mean square sense) in the limit $\Delta f\to 0$ for these values of f and α.

The magnitude of the spectral coherence ranges between 0 and 1. It is the measure of the strength of cyclostationarity in a signal. Its maximum over the spectral frequency f gives the cycle frequency domain profile (also known as the α-domain profile).

$$I(\alpha) = \max_{f} |\hat{C}_x^{\alpha}(f)| \tag{4.2.36}$$

The cyclic frequency domain profile is used here in the detection and classification of signals having low SNRs.

4.3 Cyclic Spectral Analysis of Modulated Signals

In this section, the spectral coherences and cycle frequency domain profiles of the following modulation techniques are discussed in the presence of AWGN:

- Double sideband suppressed carrier amplitude modulation (DSB-SC AM)
- Binary phase shift keying (BPSK)
- Binary frequency shift keying (B-FSK)
- Minimum phase shift keying (MSK)
- Quadrature phase shift keying (QPSK)

The common simulation parameters for generating spectral coherence are

- Sampling frequency F_s = 1, every frequency is normalized to F_s
- Intermediate carrier frequency (IF) $F_c = 0.25$
- Noise source: additive white Gaussian noise
- Signal-to-noise radio (SNR) of training signals = -3dB and -5dB, defined as the ratio of signal power over one symbol period to the average noise power
- Data samples captured per observation at IF, T = 1024 samples
- Number of observations N = 100

4.3.1 SCF and Cycle Frequency Domain Profile of DSB-SC AM

The DSB-SC AM can be expressed mathematically as [82],[83], [84]

$$s(t) = A_c \cos(2\pi F_c t)\, m(t) \tag{4.3.1}$$

The AM specific simulation parameters are,

- The carrier amplitude $A_c = 1$

- The binary random message sequence $m(t)$ assumed to be +1 and -1 equally likely per sample
- The SNR for AM is defined as the ratio of instantaneous signal power to the average noise power

Figure 4.3.1 shows the cyclic spectral coherence and the corresponding cycle frequency domain profile of the AM signal with SNR of 0dB and observation length of 100.

4.3.2 SCF and Cycle Frequency Domain Profile of BPSK

DSB-SC AM can be expressed mathematically [82],[83], [84]

$$s(t) = \sqrt{\frac{2E_b}{T_b}}\cos(2\pi f_c t + q_c)m(t) \tag{4.3.2}$$

- The bit energy E_b is normalized to have unity energy,
- Binary random message sequence $m(t)$ having +1 and -1 equally likely per symbol duration T_b
- Bit rate $F_d = 1/T_b = 0.1$
- Roll-off factor of root raised cosine filter (RRC) $\beta = 0.35$

Figure 4.3.2 shows the cyclic spectral coherence and the corresponding cycle frequency domain profile of BPSK signal with SNR of 0dB and observation length of 100.

4.3.3 SCF and Cycle Frequency Domain Profile of B-FSK

The binary FSK modulation can be expressed mathematically as [82],[83], [84]

$$\begin{aligned} s_1(t) &= \sqrt{\frac{2E_b}{T_b}}\cos(2\pi F_c t + 2\pi \Delta f\, t), \quad 0 \le t \le T_b \quad \text{(binary 1)} \\ s_0(t) &= \sqrt{\frac{2E_b}{T_b}}\cos(2\pi F_c t - 2\pi \Delta f\, t), \quad 0 \le t \le T_b \quad \text{(binary 0)} \end{aligned} \tag{4.3.3}$$

- The bit energy E_b is normalized to unity energy over one bit duration
- The constant frequency offset $\Delta f = 0.1$
- Bit rate $F_d = 1/T_b = 0.1$

Figure 4.3.3 shows the cyclic spectral coherence and the corresponding cyclic domain profile of the binary FSK signal with 0dB SNR and 100 observation length.

4.3.4 SCF and Cycle Frequency Domain Profile of MSK

MSK modulation can be expressed mathematically as [82],[83], [84]

$$s(t) = \sum_{i=0}^{\infty} m_I(t)\, p(t - 2iT_b)\cos 2\pi F_c + \sum_{i=0}^{\infty} m_Q(t)\, p(t - 2iT_b - T_b)\sin(2\pi F_c) \qquad (4.3.4)$$

- $m_I(t)$ and $m_Q(t)$ are the odd and even bits of $m(t)$ respectively
- Bit rate $F_d = 1/T_b = 0.1$

The digital modulation index, the maximum frequency deviation during one bit period, is defined as,

$$h = 2\Delta f \times T_b \qquad (4.3.5)$$

MSK is continuous-phase FSK with a minimum modulation index h = 0.5 that will produce orthogonal signaling [82]. Therefore the frequency offset can be obtained from (4.3.5) where Δf is set to be 0.025. The cyclic features will be shown at $2(F_c \pm 0.025) = 0.5 \pm 0.05$. This is confirmed by the spectral coherence and cycle frequency domain profile of MSK, shown in Fig. 4.3.4.

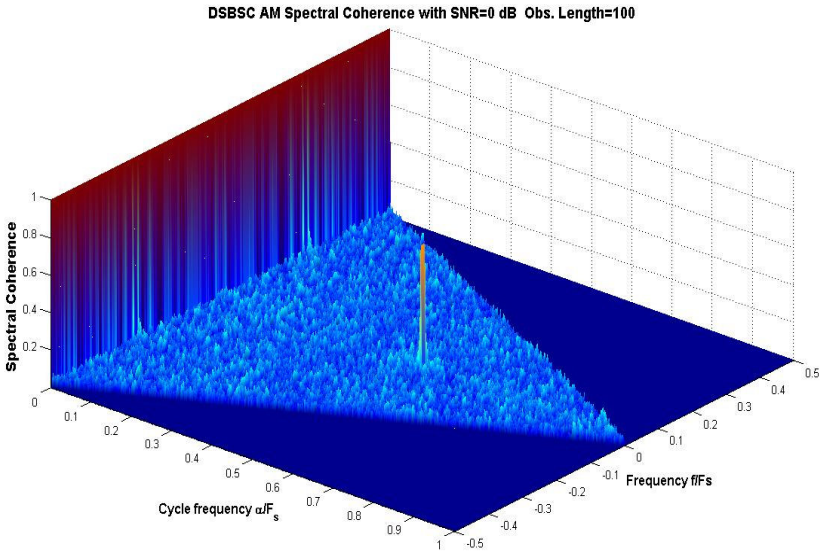
4.3.5 SCF and Cycle Frequency Domain Profile of QPSK

The QPSK modulation can be expressed mathematically as [82],[83], [84]

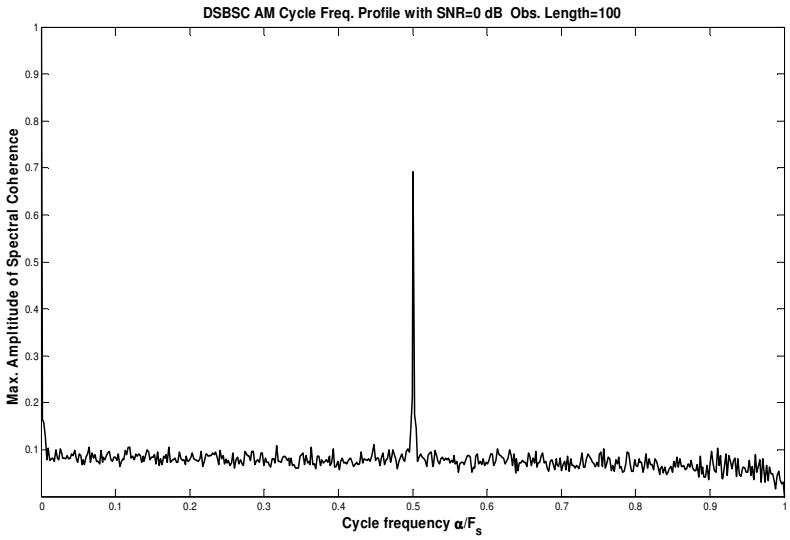
$$s(t) = \begin{bmatrix} \sqrt{\frac{2E_s}{T_s}}\cos\left((i-1)\frac{\pi}{2}\right)\cos(2\pi F_c t) \\ \sqrt{\frac{2E_s}{T_s}}\sin\left((i-1)\frac{\pi}{2}\right)\cos(2\pi F_c t) \end{bmatrix} \quad 0 \le t \le T_s \quad i = 1,\ 2,\ 3,\ 4 \qquad (4.3.6)$$

- The symbol energy E_s is normalized to unit energy over one symbol duration
- Symbol rate $F_d = 1/T_s = 0.1$
- Roll-off factor of root raised cosine filter (RRC) $\beta = 0.35$
- The index *i* is chosen from 1, 2, 3, and 4 with an equal probability of 0.25

Figure 4.3.5 shows the SCF and corresponding cycle frequency domain profile.

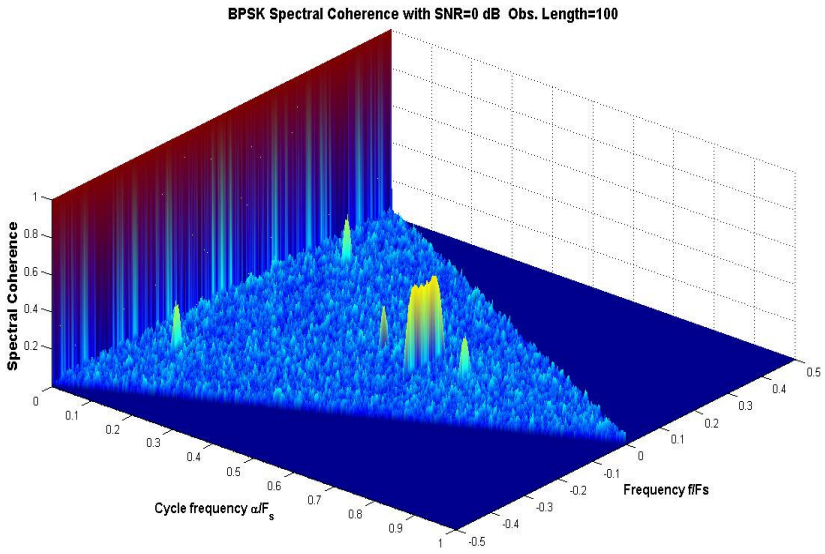


Spectral Coherence of DSB-SC AM

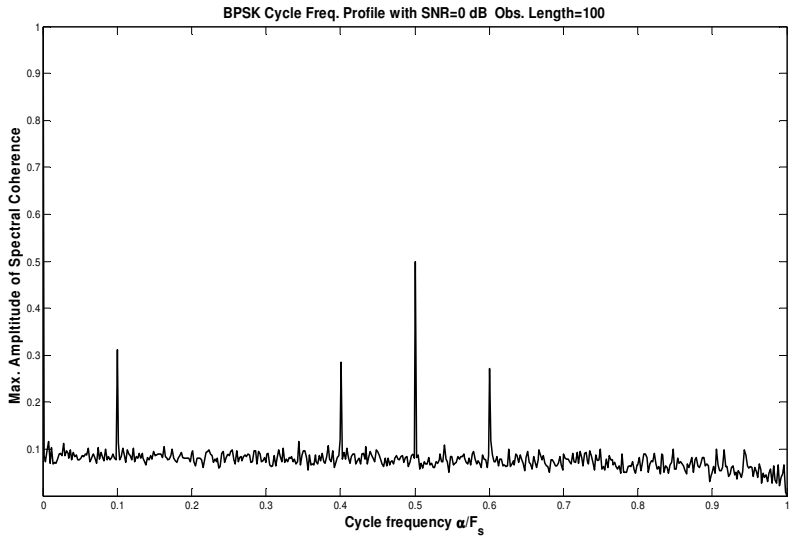


Cycle Frequency Domain Profile of DSB-SC AM

Figure 4.3.1: Spectral Coherence and cycle frequency domain profile of DSB-SC AM.

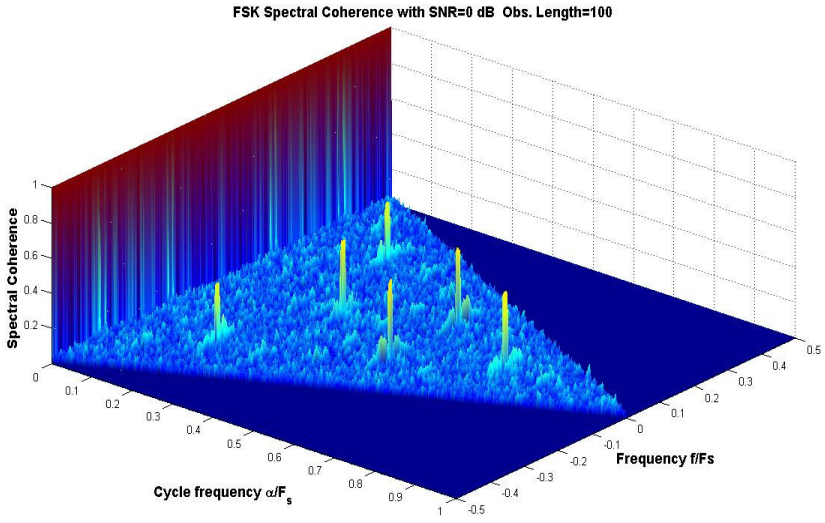


Spectral Coherence of BPSK

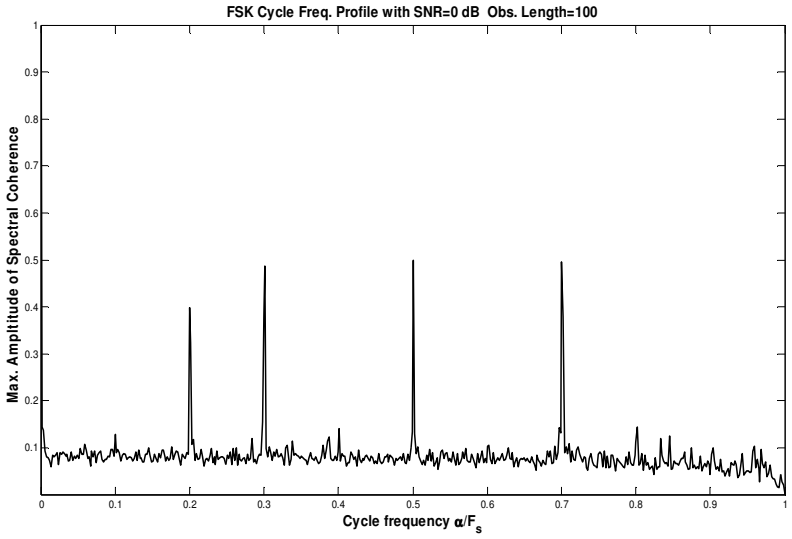


Cycle Frequency Domain Profile of BPSK

Figure 4.3.2: Spectral Coherence and cycle frequency domain profile of BPSK.

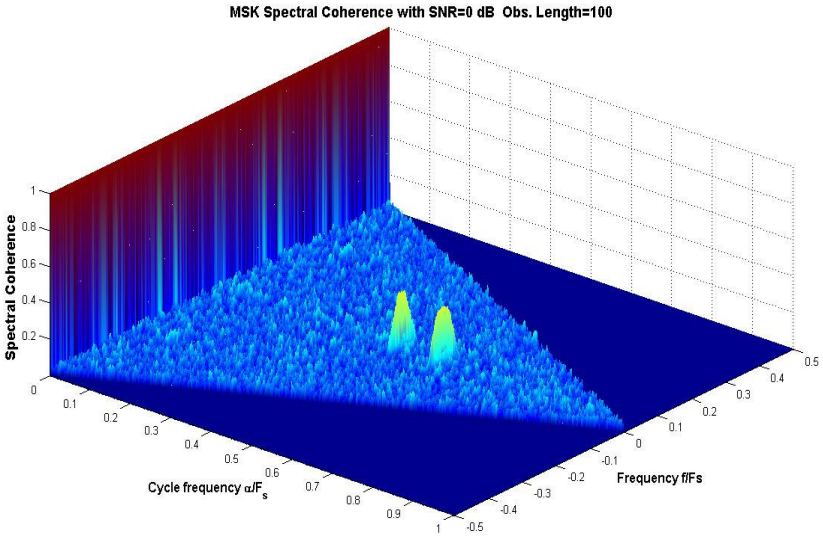


Spectral Coherence of binary FSK

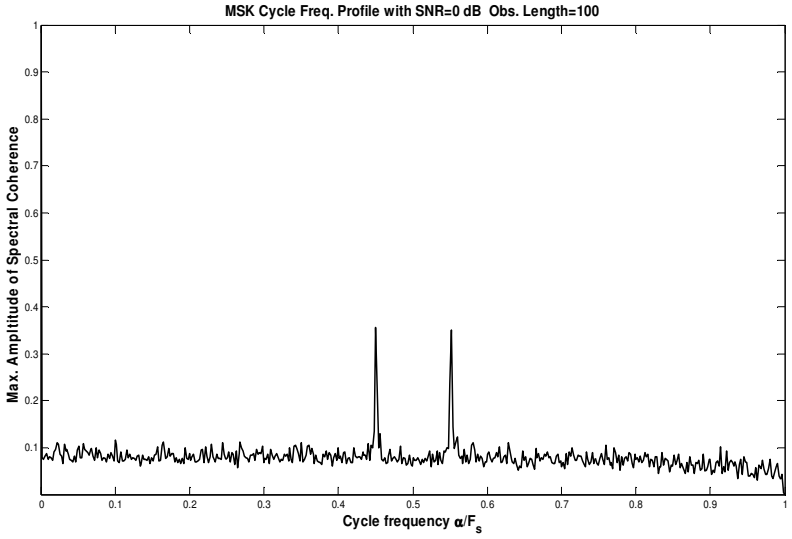


Cycle Frequency Domain Profile of binary FSK

Figure 4.3.3: Spectral Coherence and cycle frequency domain profile of binary FSK.

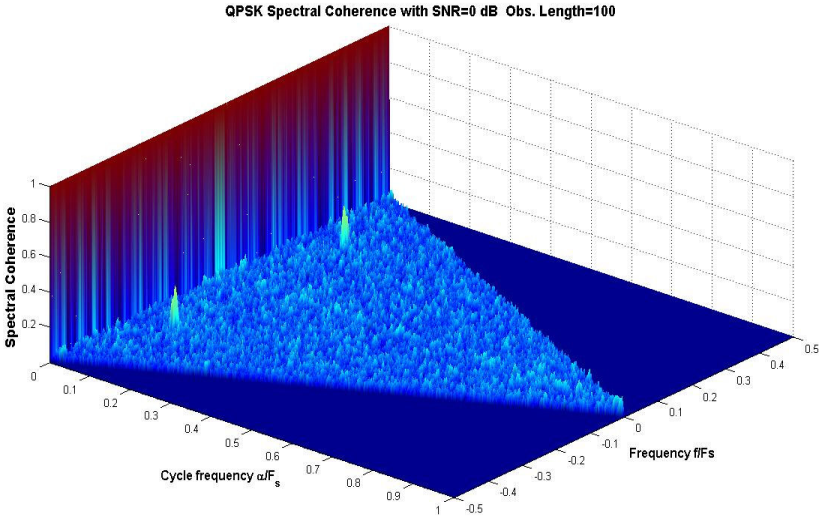


Spectral Coherence of MSK

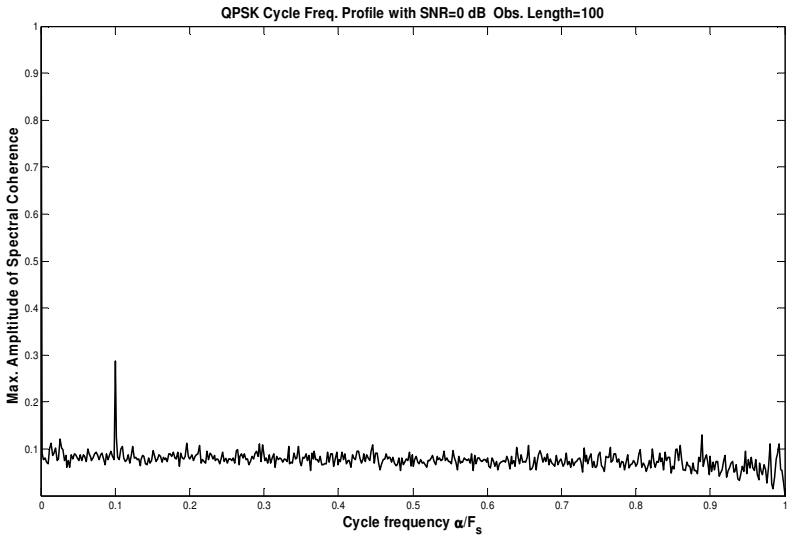


Cycle Frequency Domain Profile of MSK

Figure 4.3.4: Spectral Coherence and cycle frequency domain profile of MSK.



Spectral Coherence of QPSK



Cycle Frequency Domain Profile of QPSK

Figure 4.3.5: Spectral Coherence and cycle frequency domain profile of QPSK.

4.4 Signal Detection

In this section we discuss the method that is used for signal detection in the cycle frequency domain. The crest factor, discussed briefly in this section, is used for detecting

the peaks of the signals. Monte Carlo simulations are used to evaluate the performance of this proposed technique.

4.4.1 The Crest Factor as a Detection Threshold

The crest factor (CF) is proposed for finding dominant peaks in the cycle frequency domain. The crest factor of a waveform is equal to its peak amplitude divided by its root mean square (RMS) value, i.e., for the a waveform $x(t)$ with peak x_{max} and x_{RMS} being its RMS value, we can write the crest factor C as

$$C = \frac{x_{max}}{x_{RMS}} \tag{4.4.1.1}$$

In the case of cyclic spectral coherence, we can write the crest factor C_I as,

$$C_I = \frac{\max(I(\alpha))}{\sqrt{\left(\sum_{\alpha=0}^{N} I^2(\alpha)\right)/N}} \tag{4.4.1.2}$$

where $I(\alpha)$ is the cycle frequency domain profile and N is the total number of discrete cyclic frequencies.

The use of the crest factor in signal detection is necessary because usually the system is operating at low SNR and also because of using finite numbers signal blocks in averaging. It is found that when the observation length is not sufficient to suppress the noise in cycle frequency domain, it is very difficult to find dominant peaks. In such cases, the noise floor in the cycle frequency domain profile changes by varying the observation length and the SNR. Therefore, it is difficult to set up a unified threshold value for signal detection.

Fig. 4.4.1.1 shows the flowchart for finding dominant peaks. As an example, the dominant peaks for BPSK signal with 50 observation block is shown in Fig. 4.4.1.2, evaluated using the steps shown in Fig. 4.4.1.1.

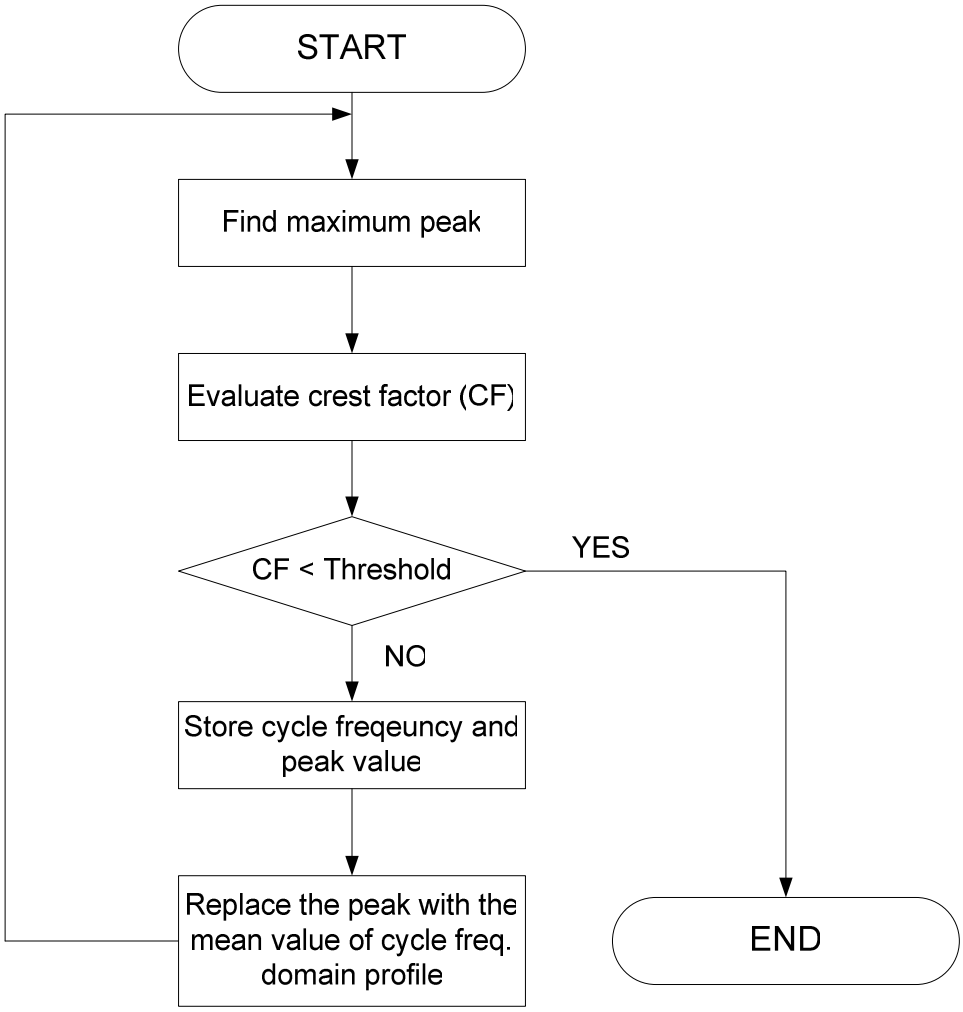


Figure 4.4.1.1: Flowchart of finding dominant peaks using crest factor.

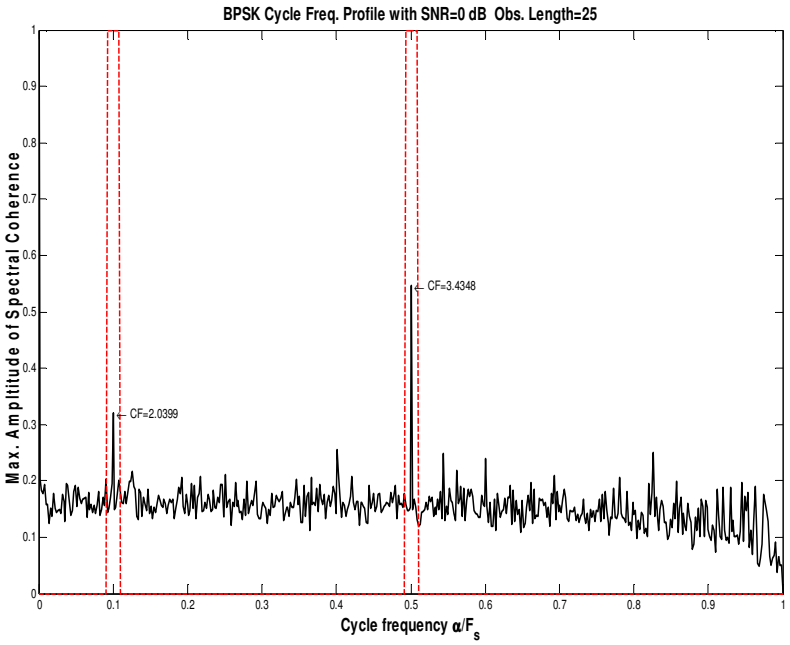


Figure 4.4.1.2: Determination of crest factors of BPSK signal.

4.4.2 Threshold Test for Signal Detection

Hypothesis testing is applied for determining the existence of signals when AWGN is present. The following simple hypotheses are used,

$$\begin{cases} H_0 : x(t) = n(t) \\ H_1 : x(t) = s(t) + n(t) \end{cases} \tag{4.4.2.1}$$

For signal detection, a threshold C_{TH} is defined as the CF for the case when the signal does not exist. The proposed threshold-based signal detection test be performed as follows

$$\begin{cases} C_T < C_I : \text{Declare } H_0 \\ C_T \geq C_I : \text{Declare } H_1 \end{cases} \tag{4.4.2.2}$$

The distribution of CF when no signal is present in the cycle frequency domain profile is shown in Fig. 4.4.2.1. The CF for the probability of false alarm of 10% is found to be 1.71 in this case.

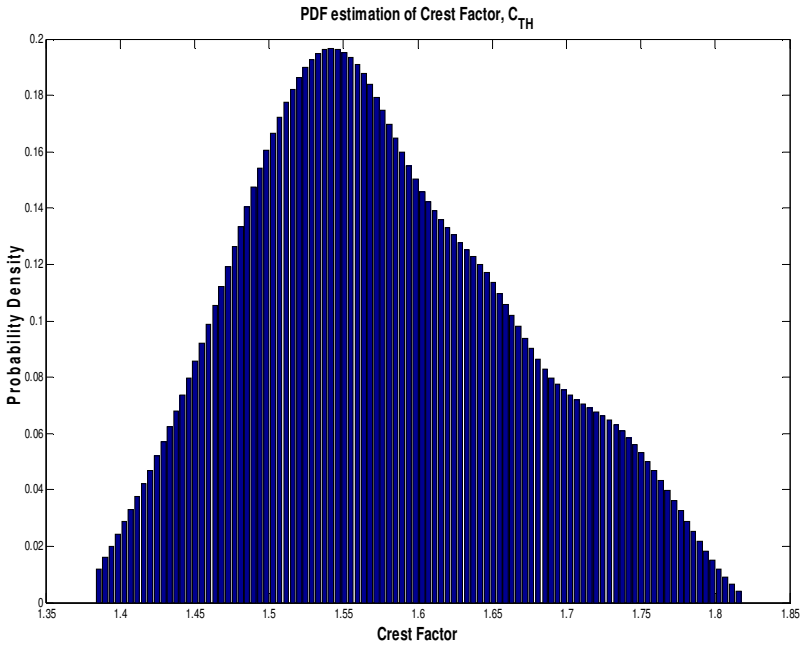


Figure 4.4.2.1: Distribution of crest factor when no signal is present.

4.5 HMM As Signal Classifier

Hidden Markov Models have been successfully used for pattern recognition in different fields of science and engineering. Some examples are speech recognition, handwriting recognition, face recognition, etc. Due to the need for an online estimation technique in real world applications, the forward-only BWA (FO-BWA), that can estimate HMM parameters on the fly, is used for building HMMs.

Since the peak signal components of a particular modulation type, for a given SNR and observation length, can vary slightly and degrade the performance of a signal classifier, we use binary sequences that are obtained from cycle frequency domain profile using the crest factor.

For the case of binary sequences, the probability of generating the observation sequence given the model ζ can be written mathematically as

$$\Pr(y_1^T \mid \zeta) = \boldsymbol{\pi}\mathbf{B}(y_1)\ \mathbf{P}\mathbf{B}(y_2)\ \mathbf{P}\ldots\ \mathbf{P}\mathbf{B}(y_T)\mathbf{1}' \tag{4.5.1}$$

Here $\mathbf{B}(y_k)$ with $k = 1, 2, 3, \ldots, T$ denotes the probability of generating symbol y_k from different states. As mentioned before, because of the long data size, we use the logarithm of $\Pr(y_1^T \mid \zeta)$, usually known as the log-likelihood.

Hence five different binary sequences, each for a specific modulation type, are used for training and incoming signal is also used in binary form. The SNR used for training is 0dB and the observation blocks are set to be 100. Different initial conditions are used and the best model, based on the value of split-data log-likelihood, is selected. In this way, we obtain five HMMs, one for each

During the recognition phase, the cycle frequency domain profile of the unknown incoming signal, after its successful detection, is then sent to the HMMs. The log-likelihood values generated are compared with the log-likelihoods of the original training sequences and the closest match is selected as the signal type. A simplified block diagram of signal classification is shown in Fig. 4.5.1.

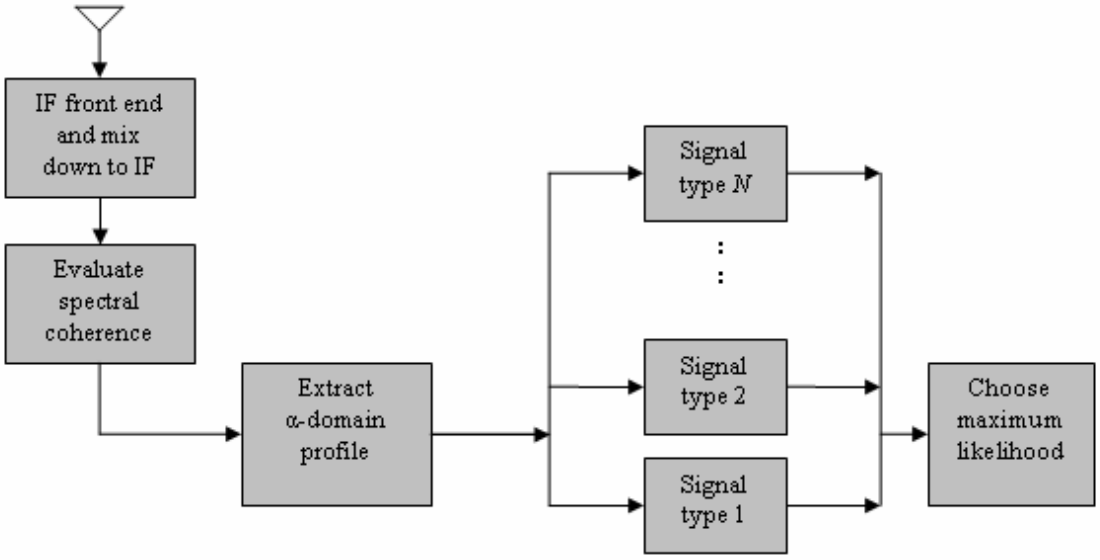


Figure 4.5.1: Block diagram of HMM based signal classification system.

4.6 Simulation Result

4.6.1 Signal Detection

We analyze the performance of SCF based signal detection technique by varying the observation length and the SNR of the incoming signal. The block lengths used in this analysis are 25, 50, 75, and 100. The SNRs used for the performance analysis are -9dB, -6dB, -3dB, and 0dB. Figs. 4.6.1.1, 4.6.1.2, 4.6.1.3, 4.6.1.4, and 4.6.1.5 show the detection probabilities of AM, BPSK, FSK, MSK, and QPSK respectively. As seen from the figures, all signals with the exception of QPSK processed with only 25 blocks gives 100% probability of detection. The detection probability increases significantly with the increase of observation length and also with increasing SNR.

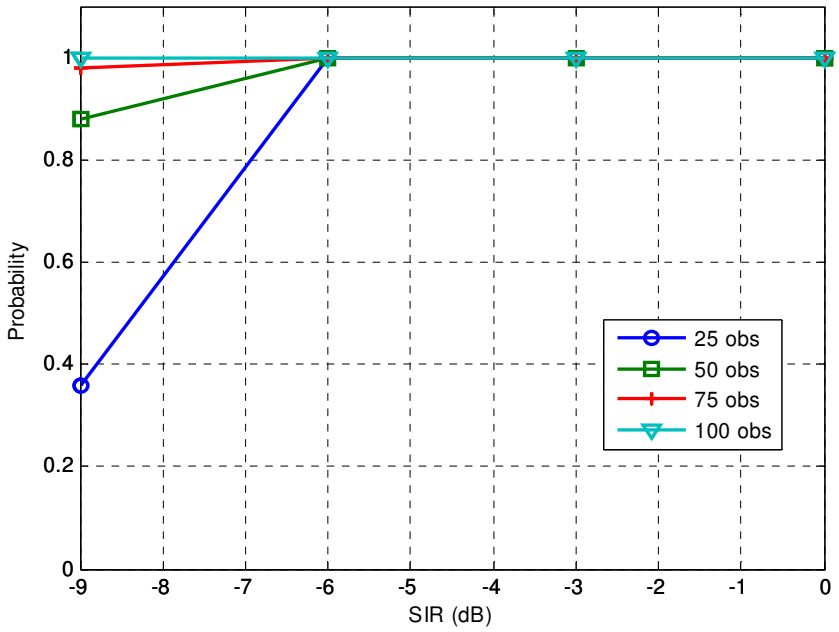


Figure 4.6.1.1: Probability of detection of AM signals with 10% false alarm.

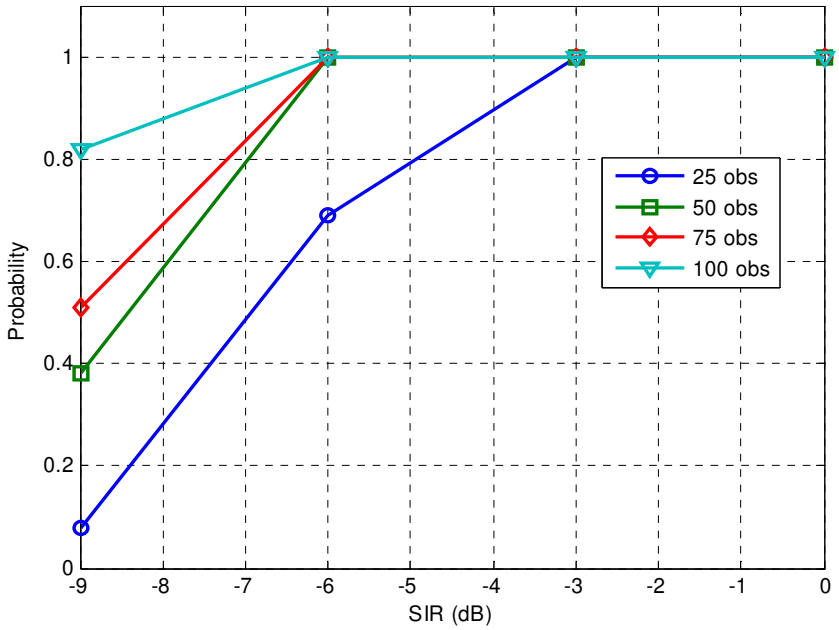


Figure 4.6.1.2: Probability of detection for BPSK signals with 10% false alarm.

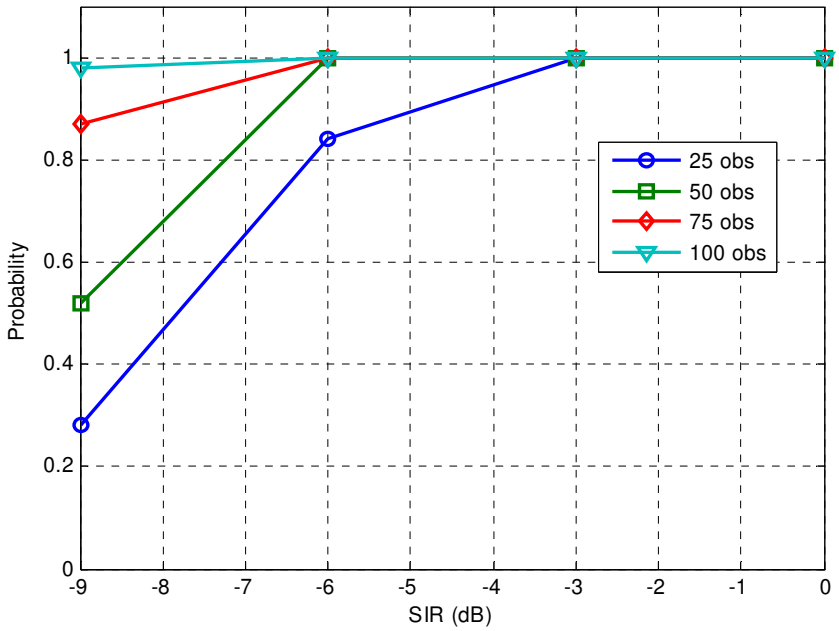


Figure 4.6.1.3: Probability of detection of FSK signals with 10% false alarm.

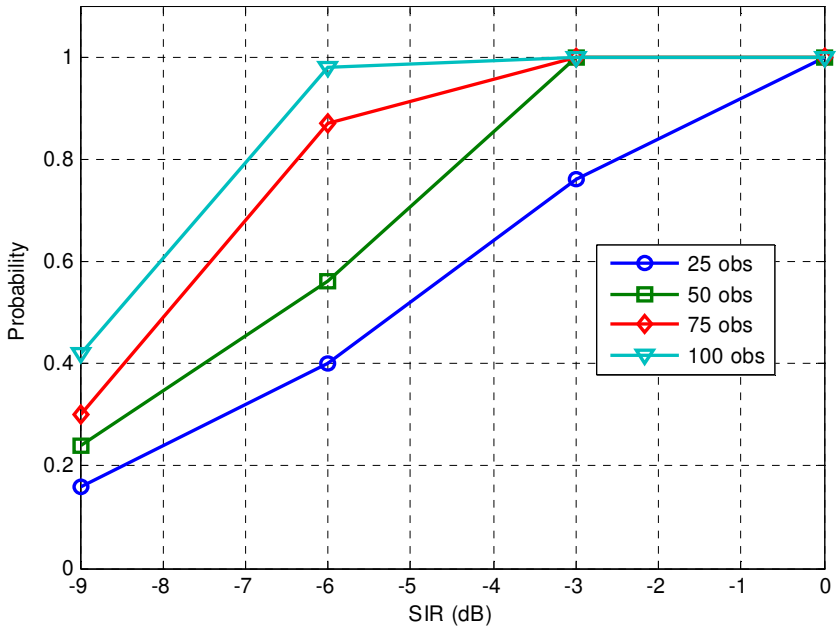


Figure 4.6.1.4: Probability of detection of MSK signals with 10% false alarm.

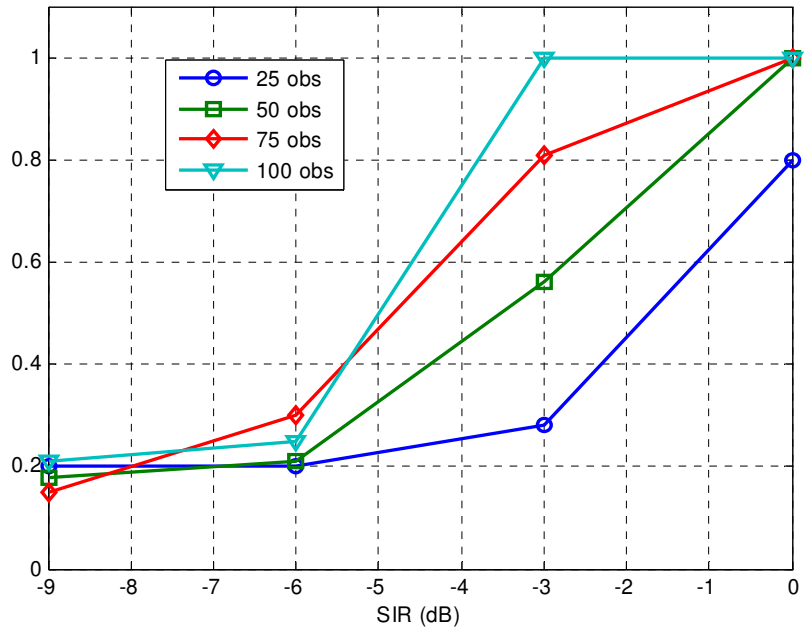


Figure 4.6.1.5: Probability of detection of QPSK signals with 10% false alarm.

4.6.2 Signal Classification

Monte Carlo simulations are performed for evaluating the performance of the HMM based signal classifier. The HMMs (in Fig. 4.5.1) are trained with SNR of -3dB and 300 observation blocks. Different incoming signals with SNR of -3dB and -5dB are received at the receiver front end and are processed with varying observation lengths to obtain the percentage of successful classification. The results are summarized in Fig. 4.6.2.1, 4.6.2.2, 4.6.2.3, 4.6.2.4, and 4.6.2.5 respectively. Note that the signals having a -5dB SNR usually require more observation blocks to achieve a given performance than when the signals have a -3dB SNR. It is interesting to note that for the case of FSK signals, the number of blocks required for more than 80% correct classification is the same for these two cases.

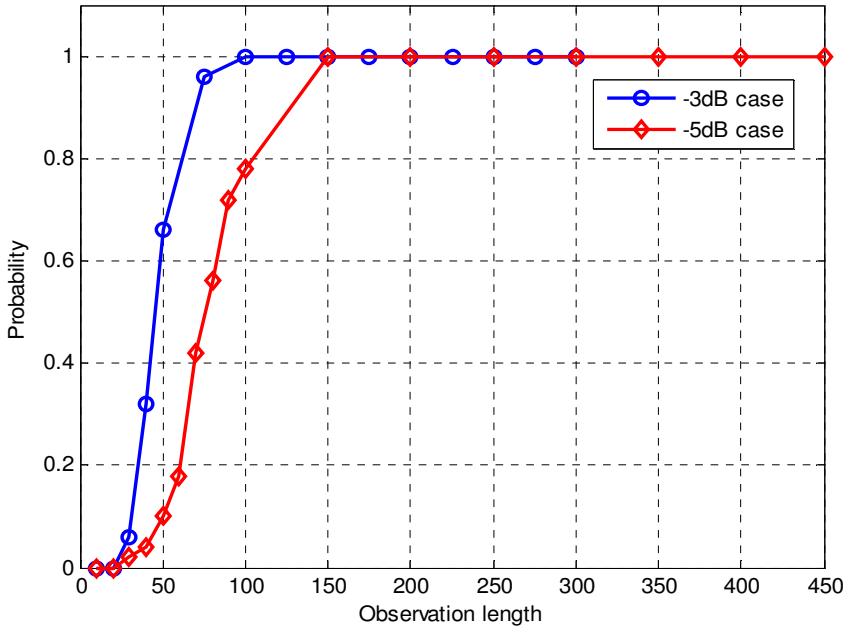


Figure 4.6.2.1: Percentage of successful classification of AM when incoming signals have SNRs of -3dB and -5dB. Training signal has -3dB SNR and 300 observation length.

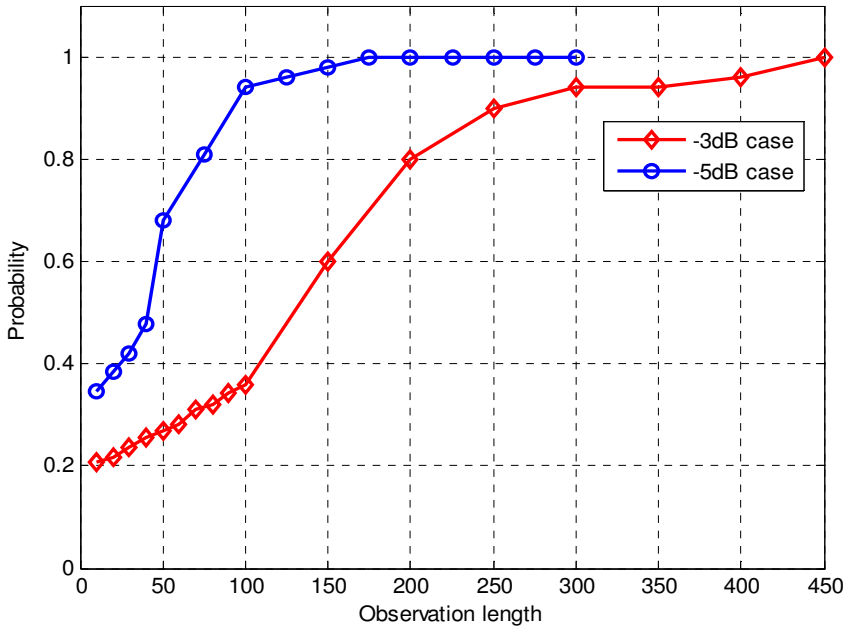


Figure 4.6.2.2: Percentage of successful classification of BPSK when the incoming signals have -3dB and -5dB SNRs. Training signal has -3dB SNR and 300 observation length.

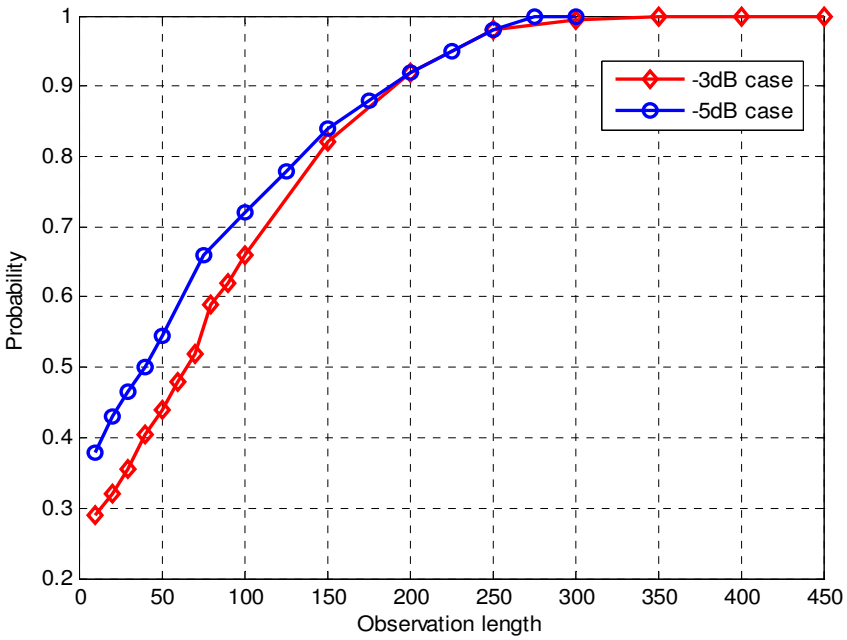


Figure 4.6.2.3: Percentage of successful classification of FSK when the incoming signals are coming at -3dB and -5dB. Training signal has SNR of -3dB and observation length of 300.

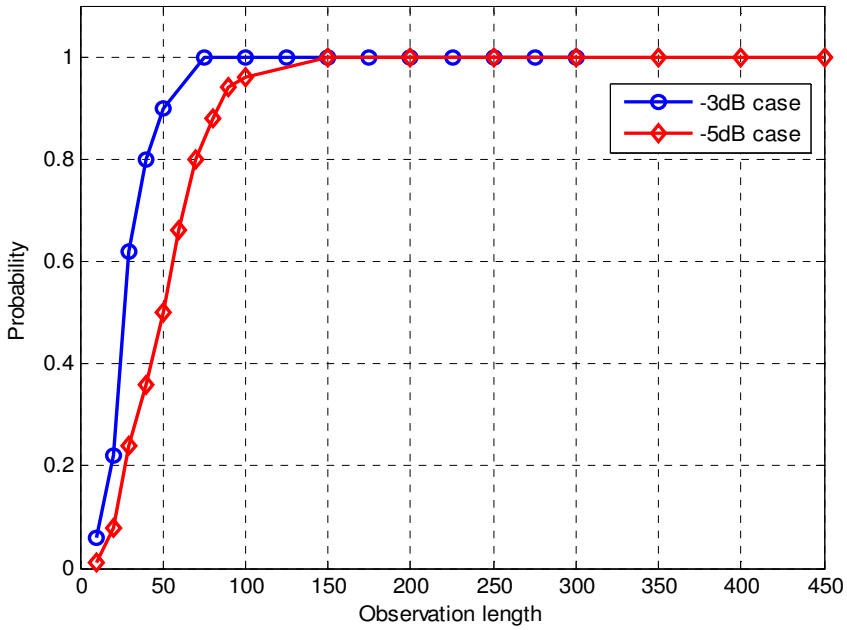


Figure 4.6.2.4: Percentage of successful classification of MSK when the incoming signals have SNRs of -3dB and -5dB. Training signal has -3dB SNR and 300 observation length.

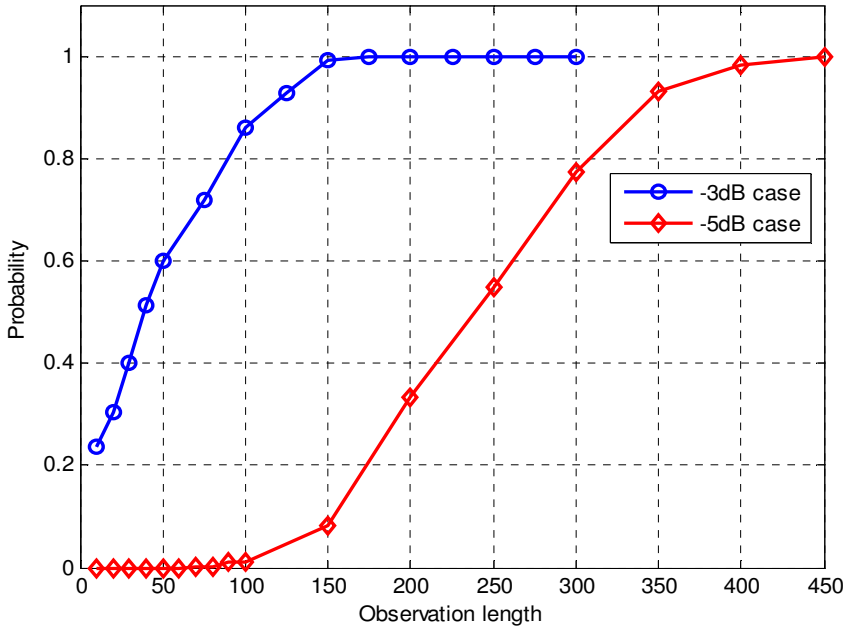


Figure 4.6.2.5: Percentage of successful classification of QPSK when the incoming signals have -3dB and -5dB SNRs. Training signal has SNR of -3dB and observation length of 300.

4.7 Conclusion

We have investigated SCF based techniques for the detection and classification of radio signals in a cognitive radio environment. Simulation results have shown a high probability of correct detection, even at very low SNR, if the number of observation blocks is increased accordingly. The system can be easily expanded to accommodate new signal types. One of the advantages that this proposed signal detection method offers is that it does not require any prior knowledge about the transmitted signal. This allows the CR device, which tries to access to a specific channel in an opportunistic manner, to blindly detect signals from active or licensed users that are operating in a particular channel.

Performance evaluation of the novel HMM based signal classification technique is also presented. It is found that signal classification can be performed with 100% accuracy if the incoming signals, that usually have low SNRs, are processed with large number of observation blocks.

Chapter 5

Dynamic Spectrum Sharing in Cognitive Radio Using HMMs

Cognitive radio networks can be designed to manage the radio spectrum more efficiently by utilizing the spectrum holes in primary users' licensed frequency bands [97]. Recent studies have shown that the radio spectrum is poorly utilized by licensed users even in urban geographical areas. This spectrum utilization can be improved significantly by making it possible for secondary users (who are not being served by the primary system) to access spectrum holes, i.e., frequency bands not used at a given time, by primary users. In this chapter we propose to use hidden Markov Models (HMMs) to model and predict the spectrum occupancy of licensed radio bands. This novel approach can dynamically select different licensed bands for its own use with significantly less interference to and from the licensed users. It is found that by predicting the duration of spectrum holes of primary users, the CR can utilize them more efficiently by leaving the band, that it currently occupies, before the start of traffic from the licensed user of that band.

The impact of CR transmission on the licensed users is also presented. It is shown that significant SIR improvements can be achieved using a HMM based dynamic spectrum allocation as compared to the traditional memoryless CSMA based approach. The results obtained using the HMM are very promising and HMM techniques can offer

a new paradigm for predicting channel behavior, an area that has been of much research interest lately.

Recent measurements have shown that there are frequency bands, even in urban areas that are largely unoccupied. Some of these frequency bands are only partially occupied while others are heavily used. Spectrum usage statistics give us a notion of *spectrum holes* which are defined in [85] as bands of frequencies assigned to a primary user, but at a particular time and specific geographical location, the bands are not being utilized by that user. The concept of the utilization of spectrum holes in CR using CSMA based band allocation is shown in Fig. 5.1. As seen from the figure, the CSMA based spectrum allocation would always result in collision whenever the primary user, whose band is currently occupied by the CR, starts using a currently used band for its data transmission. This could significantly degrade the performance of both CR network as well as the data communication of primary users. This is especially true if both networks have base stations that are located in close proximity.

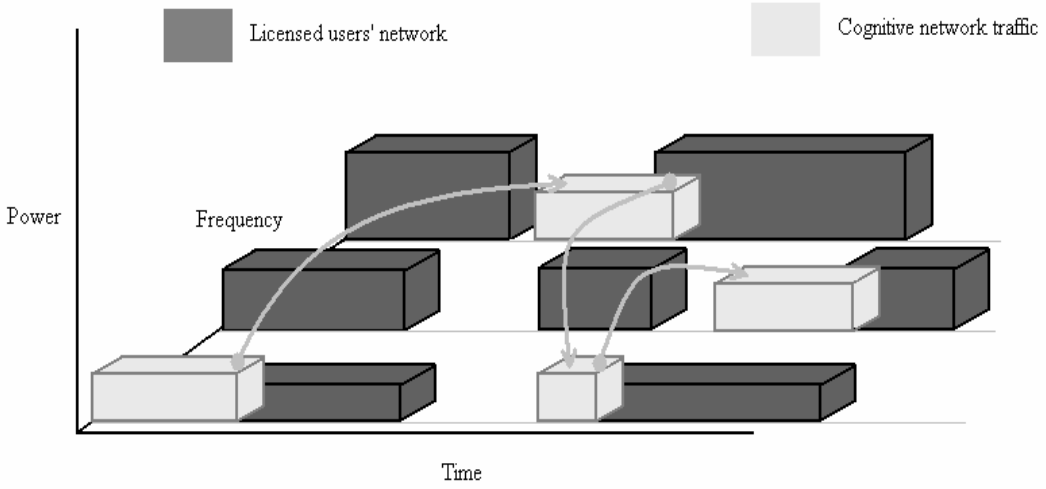


Figure 5.1: Illustration of spectrum holes in a cognitive radio network. Spectrum allocation is performed using the conventional CSMA based technique.

Cognitive radio [50], [51], [52] has recently been proposed as a method to promote the efficient use of the spectrum by exploiting the existence of spectrum holes.

The CR observes the current spectrum conditions in a given band, and if it detects no traffic, it may start its data transmission using the same band. While engaged in the transmission, the radio may observe the signal of the licensed user. In conventional CR design, if such a signal is detected, the radio's cognition cycle should direct itself to jump out of the spectrum in order to minimize interference. It should then look for other available spectrum to continue its data transmission. In this work, we propose a new way of utilizing the spectrum holes. Instead of jumping out from a given frequency band after detecting the presence of a signal from the licensed user, we propose to perform a prediction on usage behavior and then decide to remain in the same frequency band or to move to another band. If a correct prediction is performed, the CR can leave the current frequency band before detecting any signal from the primary user. We hope that our method will provide minimal collisions with the signal from the licensed users.

Our methodology can be explained as follows. Different HMMs first train themselves using spectrum usage patterns of the incumbent users and once reliable models are developed, the spectrum manager decides which frequency to use based on the likelihood of spectrum holes of these bands. In this work, usage patterns are assigned binary bits with zeros indicating no traffic and ones indicating that the spectrum is being used by the licensed user at that particular time. Hence we obtain binary vectors for each frequency band and use these vectors as training sequences for HMMs assigned for different radio bands. The HMMs will then predict the occurrence of spectrum holes in different bands and the cognitive engine will dynamically choose available frequencies for its use.

The rest of the chapter is organized as follows. First we will briefly introduce time-series prediction using hidden Markov models. Next we discuss the Markov-based Channel Prediction Algorithm (MCPA), the algorithm used for dynamic spectrum allocation. Simulation results and conclusions are given later.

5.1 Theory

For the case of a binary sequence, the probability of generating the observation sequence given the model ζ can be written mathematically as

$$\Pr(y_1^T \mid \zeta) = \boldsymbol{\pi}\mathbf{B}(y_1)\ \mathbf{P}\mathbf{B}(y_2)\ \mathbf{P} \ldots \mathbf{P}\mathbf{B}(y_T)\mathbf{1}' \tag{5.1.1}$$

Here $\mathbf{B}(y_k)$ for $k = 1, 2, 3, \ldots, T$ denotes the probability of generating symbol y_k from different states. Because of the significantly long data size, the logarithm of $\Pr(y_1^T \mid \zeta)$, usually known as the log-likelihood function is used. In this work of dynamic spectrum allocation, we predict the next behavior of channel by finding the joint probabilities $\Pr(y_1^T, 0 \mid \zeta)$ and $\Pr(y_1^T, 1 \mid \zeta)$ and then make a decision on which frequency band to use for data transmission.

There seems to be a recent interest in data prediction/forecasting using HMMs recently. Some of the recent work includes stock market forecasting [86], [87], forecasting early conflict in the southern Balkans [88] and predicting earthquakes are examples. The main reason of such a wide use of HMM in prediction problems is mainly because of its strong theoretical foundations and tractability. This is mainly because of the fact that using the HMM to forecast a time series is easily explainable and has solid statistical foundation, properties that are not usually available in Artificial Neural Networks (ANNs) [89] and [90], which are also used frequently for predicting time series.

5.1.1 Markov-based Channel Prediction Algorithm (MCPA)

The spectrum occupancy is modeled as binary sequence with one signifying that the spectrum is occupied at the particular instant by the primary user and a zero denoting that the spectrum is unoccupied at that instant. The spectrum usage pattern is modeled as exponentially distributed (in simulation case 1) and as Poisson distributed (in simulation case 2) with channel occupancy ratio of more than 50% in both cases.

Let us now explain the Markov-based Channel Prediction Algorithm (MCPA) used to predict the behavior of different channels and performs spectrum allocation dynamically. Binary sequences are obtained based on channel statistics as explained earlier. Different HMMs are then trained for each frequency band of interest using these binary sequences. The Forward-only BWA is used to obtain the parameters of the HMMs, namely the state transition matrix $\mathbf{P}$, output symbol probability matrix $\mathbf{B}$, and

the initial state probability vector $\boldsymbol{\pi}$. The flowchart of the algorithm in case of single primary user is shown in Fig. 5.1.1.

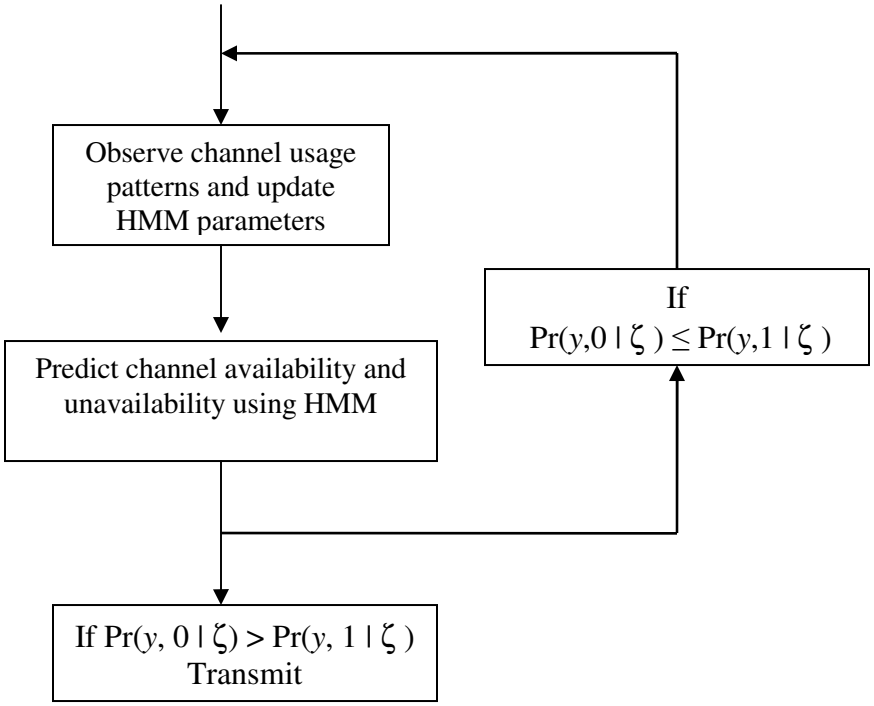


Figure 5.1.1: Flowchart of the MCPA for a single primary user case. Multiple channel case is an extension of this algorithm.

We predict channel behavior by finding the joint probabilities $\Pr(y_1^T, 0 \mid \zeta)$ and $\Pr(y_1^T, 1 \mid \zeta)$. If at a particular instance we observe that $\Pr(y_1^T, 0 \mid \zeta) > \Pr(y_1^T, 1 \mid \zeta)$, which means that the probability of a particular frequency band being unoccupied is higher than the probability of it being occupied, the cognitive radio can use the frequency band of interest for its data transmission. We compare the two probabilities with a threshold δ such that the channel is assumed in use when

$$\Pr(y_1^T,\ 0 \mid \zeta) - \Pr(y_1^T,\ 1 \mid \zeta) \geq \delta \qquad (5.1.1)$$

Simulations are performed using this criterion with different values of δ.

The graphical illustration of the MCPA technique is shown in Fig. 5.1.2. Note that the CR traffic leaves a particular band before the detection of data transmission from the primary user. In this way, we can significantly reduce the impact of the CR network on licensed users. Simulation results have shown significant improvement in the SIR of licensed users by performing spectrum allocation using HMMs.

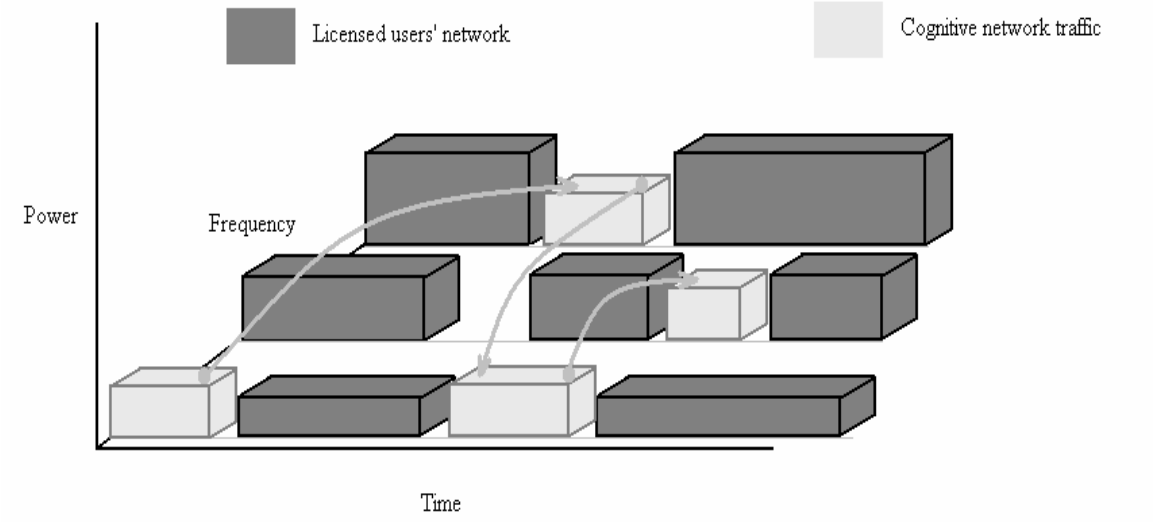


Figure 5.1.2: Illustration of dynamic spectrum allocation using MCPA.

The case of multiple primary users is just an extension of single user case. Different log-likelihood values are calculated and the channel that gives the highest probability of being unoccupied among all channels is selected for data transmission.

5.2 Simulation Results

We present the two simulation cases with four licensed users (here called interferers) and one unlicensed user. Each interferer has a unique spectrum usage pattern

that has an occupancy probability of not less that 50%. The first case assumes the spectral usage patterns of the primary users to be exponentially distributed. In Case 2, we study the effectiveness of the MCPA when the spectral patterns are Poisson distributed. Fig. 5.2.1 shows the graphical representation of the simulation setup. The slot duration of both the primary user and the CR network is assumed to be same. Also, the system is assumed to be interference limited and noise is ignored in this study.

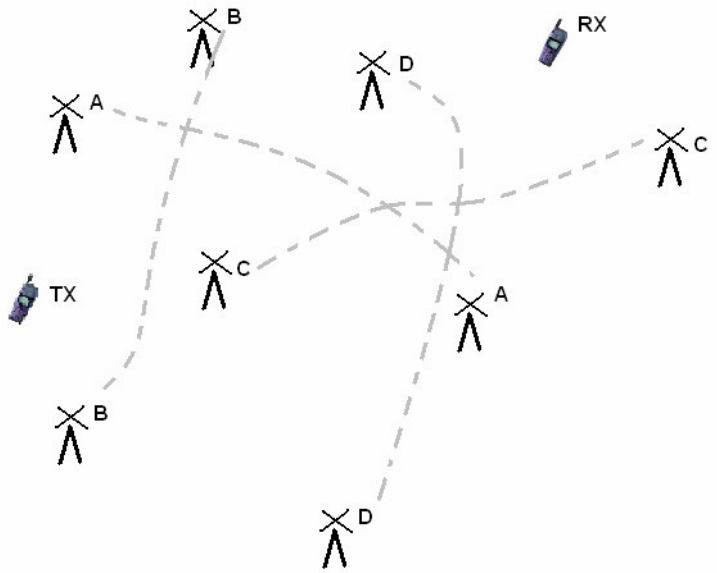


Figure 5.2.1: Simulation scenario with four licensed users and one cognitive radio transceiver labeled as 'TX' and 'RX'. The primary users are labeled using upper case alphabets.

Case 1 - MCPA Performance when the Spectral Usage Patterns of the Licensed Users are Exponentially Distributed

Figure 5.2.2(a), 5.2.2(b), 5.2.2.(c) and 5.2.2(d) show the spectrum usage statistics by the first, the second, the third and the fourth primary users respectively. The first plot shows the frequency of consecutive time slots when a particular spectrum is occupied whereas the second plot shows the frequency of consecutive unoccupied slots.

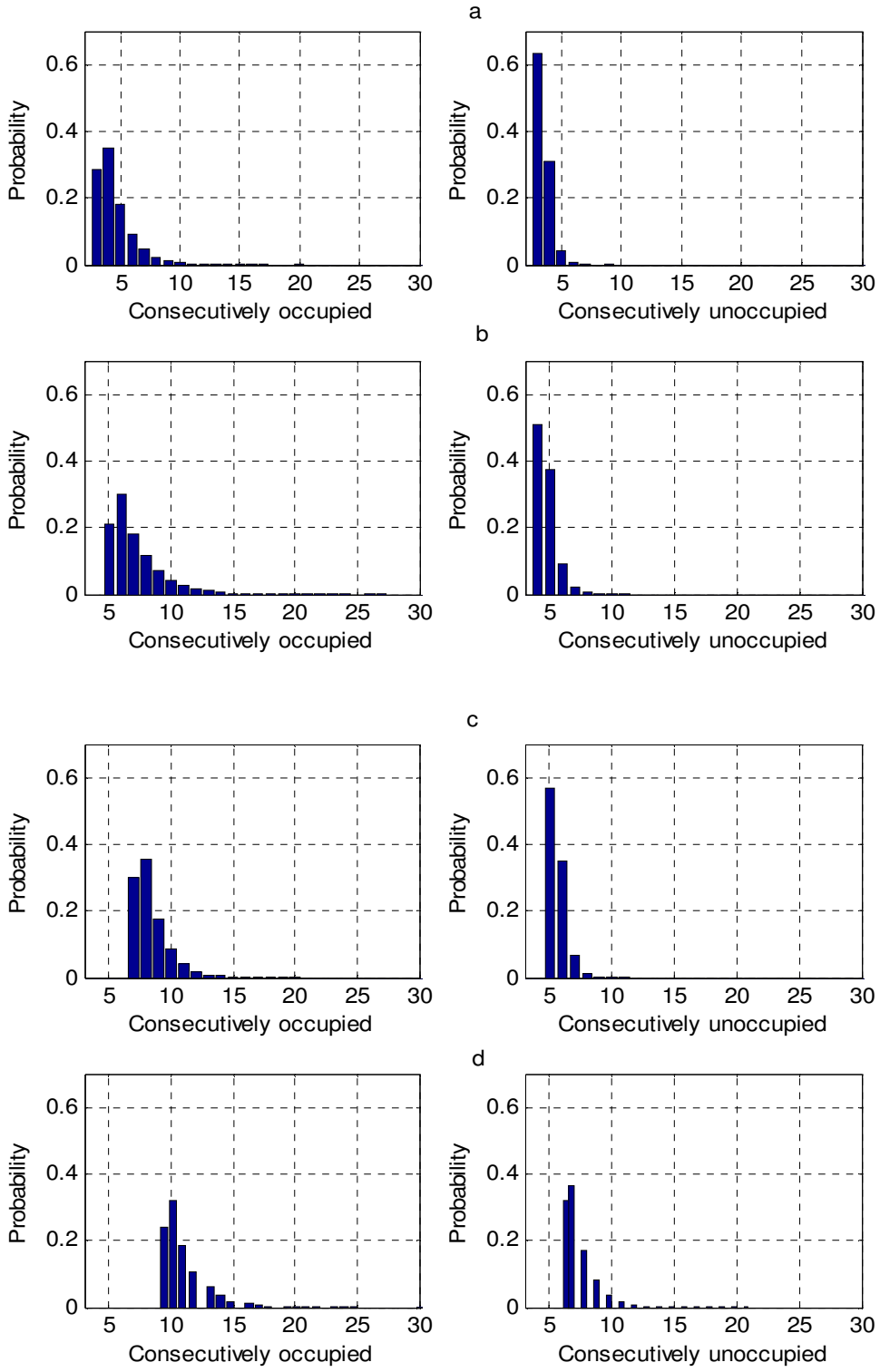


Figure 5.2.2: Spectrum usage patterns of (a) - first licensed users, (b) - second licensed user, (c) - third licensed user and (d) - fourth licensed user. The spectral patterns are assumed exponentially distributed.

We now present a comparison with the conventional CSMA based spectrum allocation technique and our proposed method of HMM based dynamic spectrum allocation. In the CSMA based spectrum allocation scheme that we adopt, the CR uses the channel when it observes no activity by the primary user and quits it whenever it senses that the primary user has started sending data through that channel (which may be very difficult to perform in practice). We therefore expect some degradation in terms of the SIR of the primary user because of the interference from the CR network that can not be completely avoided using the CSMA based approach. However, these possible collisions can be avoided by using the MCPA approach as the CR leaves the channel before the start of data transmission by the primary user. We perform a study to observe the impact on the licensed users by using these two techniques. We vary the power of the CR signal while keeping the powers of licensed signals constant and calculate the SIR at the receiver of the primary users. The value of δ used in this simulation is 0.0005. The CDFs of the SIRs in both cases (CSMA and MCPA) are plotted in Fig. 5.2.3. We can see an improvement of approximately 12dB by using HMM based dynamic spectrum allocation.

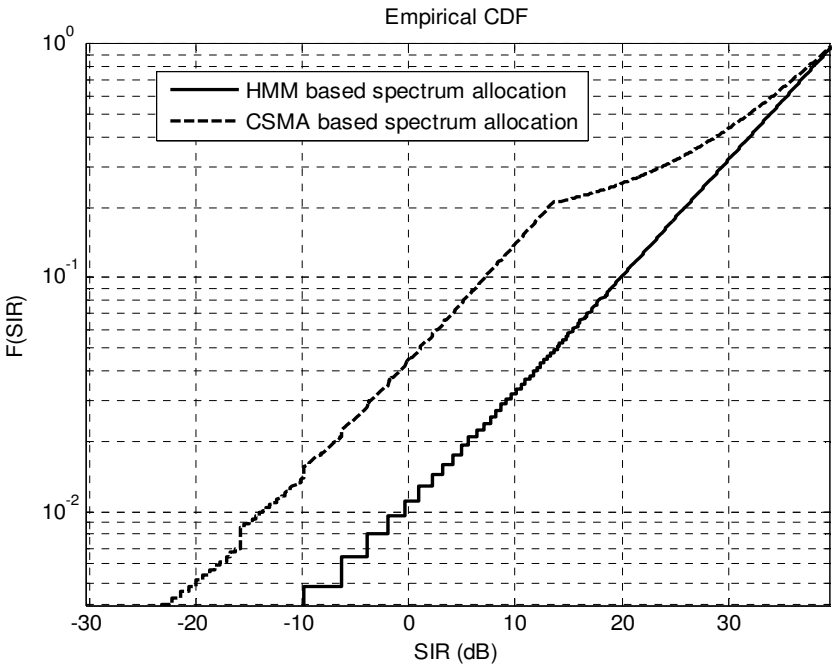


Figure 5.2.3: Comparisons of the primary user's SIR using CSMA based and MCPA based schemes.

Table 5.2.1 shows the BERs for different values of δ and their corresponding throughputs. We can see that an increase in δ gives better performance in terms of the BER. At the same time, the throughput decreases because of tougher constraints that the radio has to fulfill in order to transmit its data. Fig. 5.2.4 shows the BER plot for different values of δ used in the simulation. Figure 5.2.5 shows the percentage of time the CR uses the four licensed bands for its data transmission. As seen from the figure, the CR utilizes all the possible frequency bands, using the MCPA, based on the prediction results.

Threshold δ	No. of bits transmitted	CR BER	Throughput %
~0	4352	0.0294	98.25
0.0005	7905	0.0248	81.66
0.0030	4384	0.0212	69.55
0.0040	3016	0.0196	63.98
0.0059	5579	0.0151	48.15
0.0060	21794	0.00059	32.05
0.0063	47074	0	27.47

Table 5.2.1: Performance of the MCPA in terms of bit error rate and throughput.

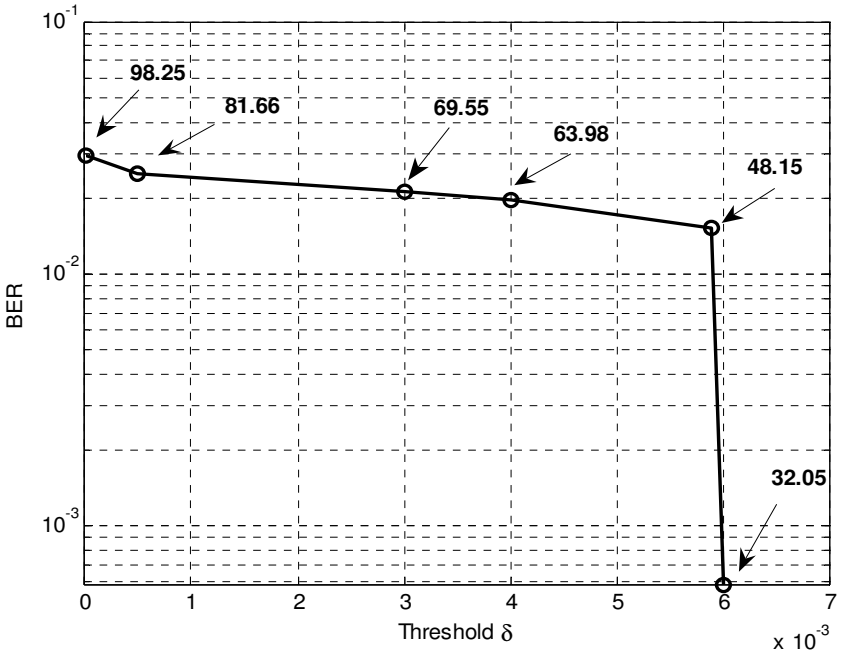
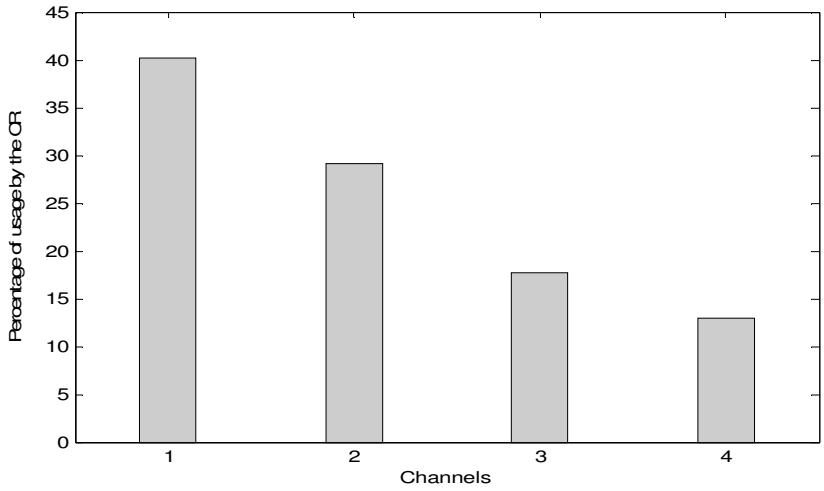


Figure 5.2.4: BER performance of the MCPA with four primary users.

(a)



(b)

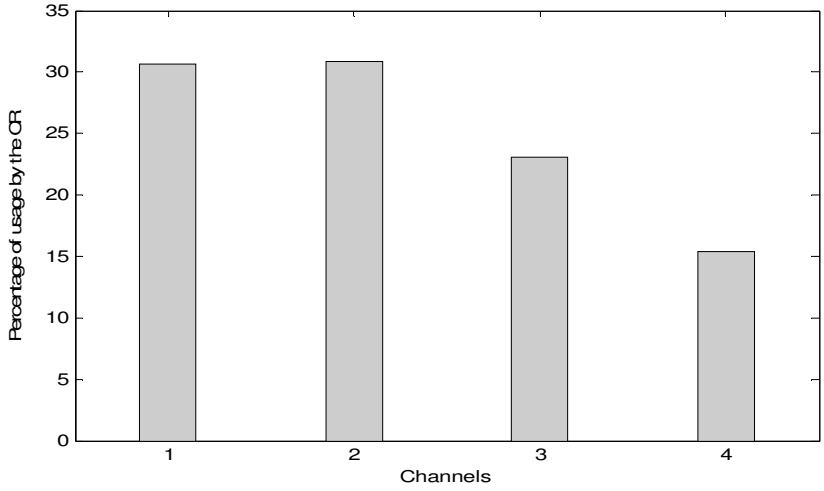


Figure 5.2.5: Percentage of time the cognitive radio uses four radio bands using (a) - CSMA based scheme and (b) - MCPA based scheme.

Case 2 - MCPA Performance When the Spectral Usage Patterns of the Licensed Users are Poisson Distributed

In the second case we assume the spectral allocation statistics of the primary users follow Poisson distribution. The distributions used in this case are shown in Fig. 5.2.6.

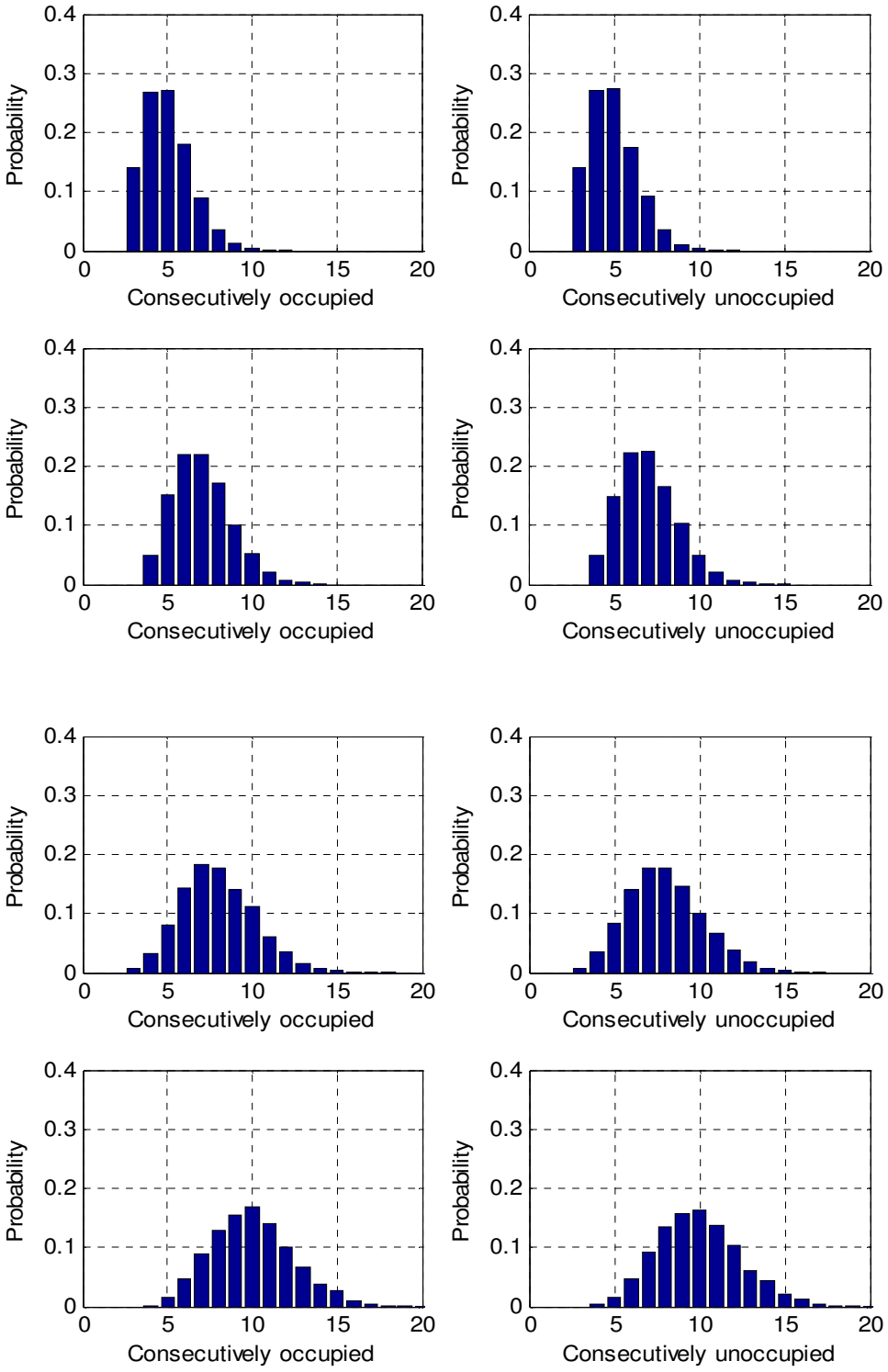


Figure 5.2.6: Spectrum usage patterns of (a) - first licensed users, (b) - second licensed user, (c) - third licensed user and (d) - fourth licensed user. The usage patterns are assumed to be Poisson distributed.

The CDFs of the SIRs in both cases (CSMA and MCPA) are plotted in Fig. 5.2.7 for $\delta = 0.0050$. We can see an improvement of approximately 20dB by using the HMM based dynamic spectrum allocation. Table 5.2.2 shows the BERs for different values of δ and their corresponding throughput. As in case 1, we can observe significant improvements in terms of the BER as the threshold δ increases. At the same time, the throughput of the system decreases because of the increase in δ, due to tougher constraints that the radio has to fulfill in order to transmit data. Fig. 5.2.8 is the BER plot for the different values of δ used in the simulation. Figure 5.2.9 shows the percentage of time CR uses these four licensed bands for its data transmission.

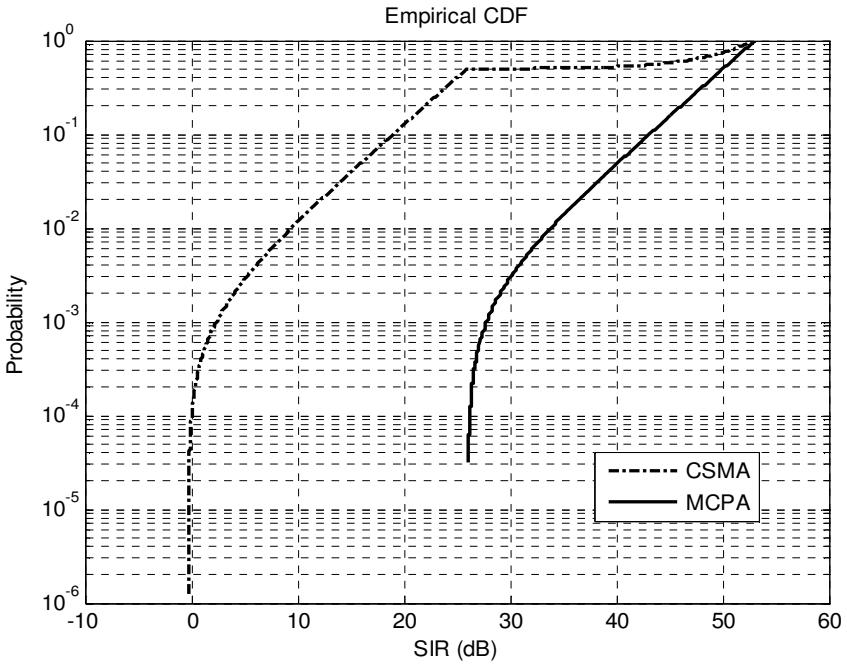


Figure 5.2.7: Comparisons of the primary user's SIR using CSMA based and MCPA based schemes.

Threshold δ	No. of bits transmitted	CR BER	Throughput %
~0	45578	0.0300	93.64
0.0030	43905	0.0260	89.82
0.0050	26412	0.022	54.03
0.0065	7173	0.00766	14.57
0.0070	2034	0	5.34

Table 5.2.2: Performance of the MCPA in terms of bit error rate and throughput.

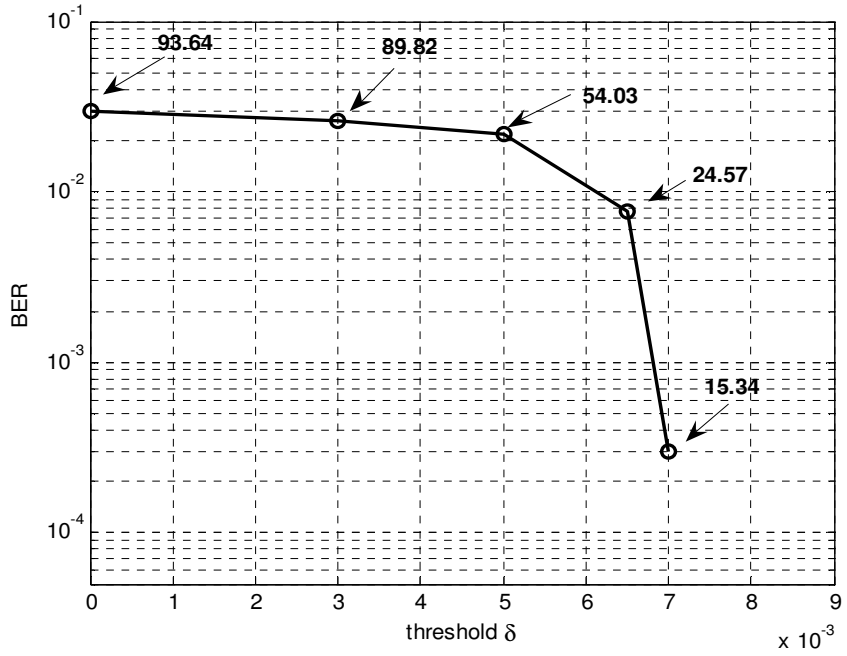
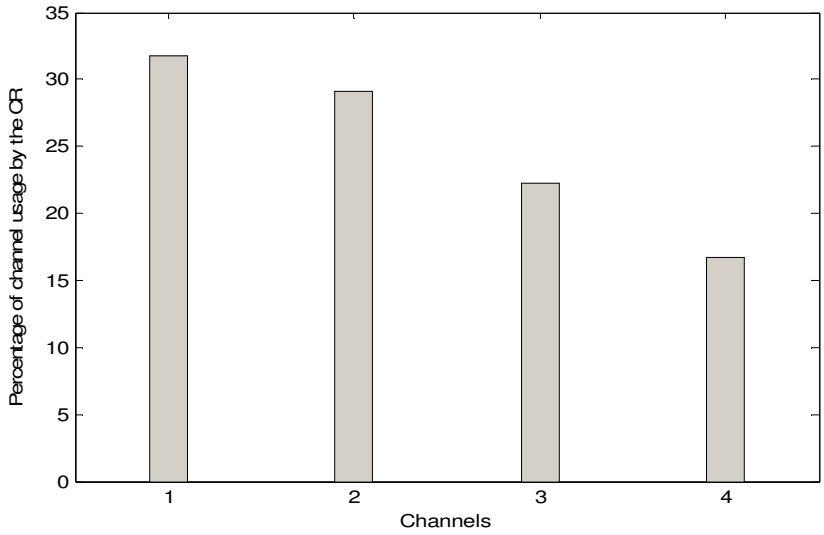


Figure 5.2.8: Performance of the MCPA with four primary users. Throughput is also shown with the corresponding values of the threshold.

(a)



(b)

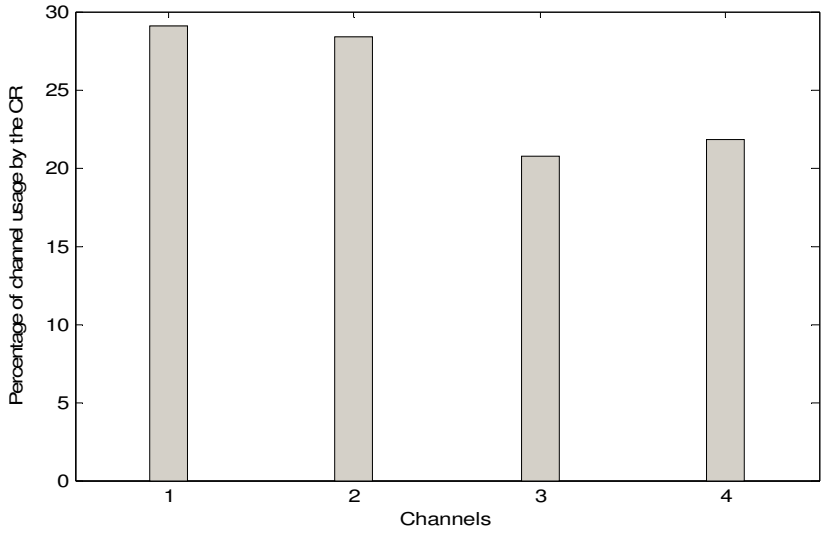


Figure 5.2.9: Percentage of time the cognitive radio uses each of the four radio bands using (a) - CSMA based scheme and (b) - MCPA based scheme.

5.3 Conclusion

We conclude that HMMs can significantly aid in dynamic spectrum allocation in cognitive radio by performing accurate predictions of the channel states of licensed users. We observe that by properly tuning HMM parameters we can obtain the performance comparable to BPSK case with just AWGN noise present in the system even though the CR is using frequency bands that are used by the primary users more than 50% of the time. This obviously comes with the cost of lowering the throughput as we put more strict conditions on the use of licensed channels by increasing the threshold.

Chapter 6

Conclusion and Future Work

This dissertation deals with the statistical analysis/modeling of discrete wireless channels and two novel applications of HMMs in cognitive radio.

The fade duration of Rayleigh fading envelope is modeled using the PH distribution approximation. The technique is found to approximate the unimodal as well as bimodal fade duration distributions of the sixth, the seventh, and the eighth quantization levels of the fading envelope that has Doppler frequency of 10 Hz and sampling rate of 0.5Kbps. The PH distribution can also approximate the seventh and the eighth quantization levels of the envelope that has $f_d = 10\,\text{Hz}$ and $f_s = 10\,\text{Kbps}$. It however fails to approximate the fade duration of the ninth quantization level. An extension of this work includes the fade duration approximation of other channel models using PH distribution.

The order estimation problem of binary MCs using BIC and goodness of fit approach is investigated. It is found that both techniques can consistently estimate the correct order of the MCs. The goodness of fit approach, however, is found to be difficult to implement practically, as it requires prohibitive amount of sequences to collect the statistics of χ^2 or information criteria. The BIC and AIC order estimation is then investigated for binary HMPs and it is found that both estimators usually select a second-order HMM as the model of choice, when the data is obtained from a stationary binary HMP. The autocorrelation function is used to find the closeness between the original data

and the model. The split-data log-likelihood is used in all simulations to avoid local maxima and saddle points. It is found that the BIC and AIC both select second-order if the channel produces stationary error traces. It is also found that non-stationary error traces, that contain very long stretches of zeros, cannot be modeled by hidden or semi-hidden Markov models. For modeling such traces, we adopt the procedure, recently proposed in [40], which splits the data into two traces. The runlength of good trace is then approximated using a discrete random process, inverse-Gaussian in our case, and the bad trace is approximated using HMMs. It is found that a 2-state HMM can not model such data and we require higher-order HMMs for modeling such trace. It is interesting to note that no semi-HMP is found to model the bad trace. An error sequence is produced using the inverse-Gaussian process and HMM that is found to be statistically similar to the original error trace. This detailed analysis on the order estimation of binary HMMs has revealed some interesting findings and these results can contribute in building reliable models to approximate the wireless channels.

The second part of this dissertation deals with the application of HMMs in cognitive radio networks. First, the problem of the classification of signals in cognitive radio using HMMs is investigated. The proposed pattern recognition based approach uses cyclostationarity of signals to classify different modulated signals after successful detection. The crest factor is used to obtain the peaks in the α-domain profile. It is found that signals with low SNRs can be detected, with very low probability of false alarm, if the sufficient number of blocks is used to obtain the α-domain profile. The signal classification method is also tested for signals having low SNRs and the results are found to be very promising. In future, we intend to study the effect of changing the IF frequency on signal detection and classification. The effect of multipath on signal detection/classification will also be investigated. Also, SNR estimation using HMMs will also be studied. These are some important issues that are faced by cognitive radio networks currently.

We proposed a novel technique of spectrum allocation using HMMs. The proposed method can reduce the interference to and from the licensed users significantly. The system is tested in two different channel conditions and the results are found to be very promising. Using this technique we can obtain the performance of the CR, in terms

of BER, comparable to the performance of the system working in just AWGN environment with no interference. Future research will include the extension of this scheme in non-binary channel state statistics, where signal powers from different licensed users will be used to predict the availability of different channels.

References

[1] R. A. Howard, *Dynamic Probabilistic Systems, Volume I: Markov Models*, John Wiley and Sons Inc, 1971.

[2] L. Rabiner, and B.-H. Jaung, *Fundamentals of Speech Recognition*, Prentice Hall, Englewood Cliffs, New Jersey, 1993.

[3] S. Sivaprakasam, and K. S. Shanmugan, "An equivalent model for burst errors in digital channels," *IEEE Trans. Commun.*, vol. 43, pp. 1345-1355, April 1995.

[4] W. Turin, *Digital Transmission Systems: Performance Analysis and Modeling*, McGraw-Hill, 1998.

[5] W. H. Tranter, K. S. Shanmugan, T. S. Rappaport, and K. L. Kosbar, *Principles of Communication Systems Simulation with Wireless Applications*, Prentice Hall, 2003.

[6] W. Feller, *Introduction to Probability Theory and Its Applications*, volume 1, John Wiley & Sons, 1950.

[7] R. A. Howard, *Dynamic Probabilistic Systems, Volume. II: SemiMarkov and Decision Processes*, John Wiley and Sons Inc, 1971.

[8] L. Finesso, *Consistent Estimation of the Order for Markov and Hidden Markov Models*, Ph.D. Dissertation, University of Maryland, 1990.

[9] S. Khundapur, and P. Narayan, "Order Estimation of Hidden Markov Sources and Binary Renewal Processes," *IEEE Trans. Inform. Theory*, 48, pp. 1704-1713, 2005.

[10] T. S. Rappaport, *Wireless Communications: Principles and Practice*, Prentice-Hall, 1996.

[11] A. Papoulis, and S. Pillai, *Probability, Random Variables and Stochastic Processes*, McGraw-Hill, 4th ed., 1973.

[12] W. A. Gardner, *Statistical Spectral Analysis: A Non-Probabilistic Theory*, Prentice Hall, 1988.

[13] S. Haykin, "Cognitive Radio: Brain-Empowered Wireless Communications," *IEEE Journ. on Sel. Areas in Commun.*, vol. 23, pp. 201-220, February 2005.

[14] S. M. Ross, *Stochastic Processes*, Second Edition, Wiley, 1996.

[15] L. E. Baum, T. Petrie, G. Soules, and N. Weiss, "A Maximization Technique Occurring in the Statistical Analysis of Probabilistic Functions of Markov Chains," *Ann. Math. Stat.*, vol. 41, issue 1, pp. 164-171, Feb. 1970.

[16] J. K. Baker, "Stochastic Modeling for Automatic Speech Understanding," in *Speech Recognition*, Reddy R. ed., New York: Academic Press, 1975.

[17] I. A. Akbar, *Markov Modeling of Third Generation Wireless Channels*, M.S. Thesis, Virginia Tech, 2003.

[18] I. L. MacDonald, and W. Zucchini, *Hidden Markov and Other Models for Discrete-valued Time Series*, Chapman & Hill, 1997.

[19] O. Cappé, E. Moulines, and T. Rydén, *Inference in Hidden Markov Models*, Springer, 2005.

[20] E. Gassiat, and S. Boucheron, "Optimal error exponent in hidden Markov models order estimation," *IEEE Trans. Inform. Theory*, vol. 49, pp. 964-980, 2003.

[21] P. Billingsley, "Statistical methods in Markov chains," *Ann. Math. Stat.*, vol. 32, pp. 12-40, 1961.

[22] T. W. Anderson and L. A. Goodman, "Statistical inference about Markov chains," *Ann. Math. Stat.*, vol. 28, pp. 89-110, 1957.

[23] T. Rydén, "Consistent and Asymptotically Normal Parameter Estimates for Hidden Markov Models," *Ann. Stat.*, vol. 22, no. 4, pp. 1884-1895, Dec. 1994.

[24] T. Cover, and J. Thomas, *Elements of Information Theory*, Wiley and Sons, New York, 1991.

[25] P. Elias. "Universal codeword sets and representations of the integers," *IEEE Trans. Inform. Theory*, IT-21:194-203, March 1975.

[26] J. Chen, and J. D. Kalbfleisch, "Penalized minimum-distance estimates in finite mixture models," *The Canadian Journal of Statistics*, vol. 24, pp. 167–175, 1996.

[27] R. J. MacKay, *Hidden Markov Models: Multiple Processes and Model Selection*, Ph.D. Thesis, University of British Columbia, Vancouver, 2003.

[28] R. J. MacKay, "Estimating the order of a hidden Markov models," *The Canadian Journal of Statistics*, vol. 30, no. 4, pp. 575-589, 2002.

[29] B. L. S. P. Rao, *Identifiability in Stochastic Process: Characterization of Probability Distributions*, Academic Press, 1992.

[30] I. Csiszár, and P. Shields, "The consistency of BIC Markov order estimator," *Ann. Stat.*, vol. 28, pp. 1601-1619, 2000.

[31] D. Blackwell, and L. Koopmans, "On the identifiability problem for functions of finite Markov chains," *Ann. Math. Stat.*, vol. 28, pp. 1011–1015, 1957.

[32] E. J. Gilbert, "On the identifiability problem for functions of finite Markov chains," *Ann. Math. Stat.*, vol. 30, pp. 688–697, 1959.

[33] S. W. Dharmadhikari, "Functions of finite Markov chains," *Ann. Math. Stat.*, vol. 34, pp. 1022–1032, 1963.

[34] A. Heller, "On stochastic processes derived from Markov chains," *Ann. Math. Stat.*, vol. 36, pp. 1286–1291, 1965.

[35] J. McLachlan, and D. Peel, *Finite Mixture Models*, New York, John Wiley & Sons Ltd, 2000.

[36] R. A. Horn, and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, 1985.

[37] F. R. Gantmacher, *The Theory of Matrices*, Chelsea Publishing Co., New York, 1959.

[38] J. C. Kieffer, "Strongly consistent code-based identification and order estimation for constrained finite-state model classes," *IEEE Trans. Inform. Theory*, vol. 39, pp. 893–902, May 1993.

[39] A. R. Barron, "The strong ergodic theorem for densities: Generalized Shannon–McMillan-Breiman theorem," *Ann. Probab.*, vol. 13, no. 4, pp. 1292–1303, 1985.

[40] A. Konrad, A. D. Joseph, and R. Ludwig, "A Markov-Based Channel Model Algorithm for Wireless Networks," In Proc. of *Fourth ACM International Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM)*, 2001.

[41] M. Gupta, L. Jin, and N. Homma, *Static and Dynamic Neural Networks: From Fundamentals to Advanced Theory*, John Wiley & Sons, Inc, 2003.

[42] J. S. Bridle, "Training stochastic model recognition algorithms as networks can lead to maximum mutual information estimation of parameters," In D. Touretzky, editor, *Advances in Neural Information Processing Systems 2*, pp. 211–217, San Mateo, CA.

[43] P. Smolensky, M. C. Mozer, and D. E. Rumelhart, *Mathematical Perspectives on Neural Networks*, Erlbaum, 1996.

[44] FCC 03-322, "NPRM - Facilitating Opportunities for Flexible, Efficient, and Reliable Spectrum Use Employing Cognitive Radio Technologies," *FCC*, Dec. 2003.

[45] M. A. McHenry, "NSF Spectrum Occupancy Measurements Projects Summary," *Shared Spectrum Company Report*, Aug. 2005.

[46] P. Kolodzy, "Spectrum Policy Task Force: Findings and Recommendations," in *International Symposium on Advanced Radio Technologies*, March 2003.

[47] Federal Communications Commission (FCC), "Spectrum Policy Task Force," *ET Docket no. 02-135*, November 15, 2002.

[48] Federal Communications Commission (FCC), "Notice of Proposed Rule Making," *ET Docket no. 04-113*, May 25, 2004.

[49] IEEE P802.22/D0.1, *Draft Standard for Wireless Regional Area Networks Part 22: Cognitive Wireless RAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Policies and procedures for operation in the TV Bands*, May 2006.

[50] J. Mitola, III, "Cognitive radio for flexible mobile multimedia communications," presented at *IEEE International Workshop on Mobile Multimedia Communications*, 1999.

[51] J. Mitola, III, and G. Q. Maguire, Jr., "Cognitive radio: making software radios more personal," *IEEE Personal Communications*, vol. 6, pp. 13-18, 1999.

[52] J. Mitola, III, *Cognitive Radio: An Integrated Agent Architecture for Software Defined Radio*, PhD Dissertation, Royal Institute of Technology, Sweden, May 2000.

[53] W. A. Gardner, *Introduction to Random Process with Applications to Signals and Systems*, MacMillan, 1986.

[54] W. A. Gardner, *Statistical Spectral Analysis: A Nonprobabilistic Theory*, Prentice Hall, 1988.

[55] W. A. Gardner, "Spectral Correlation of Modulated Signals: Part I-Analog Modulation," *IEEE Trans. Commun.*, vol. 35, no. 6, pp. 584- 594, June 1987.

[56] W. Gardner, W. Brown, C-K Chen, "Spectral Correlation of Modulated Signals: Part II-Digital Modulation," *IEEE Trans. on Commun.*, vol.35, no.6pp. 595- 601, June 1987.

[57] C. M. Spooner, *Theory and Application of High-Order Cyclostationarity*, Ph.D. Dissertation, 1992.

[58] W. Gardner, and C. M. Spooner, "The cumulant theory of cyclostationary time-series. I. Foundation," *IEEE Trans. Sig. Proc.*, vol. 42, no. 12, pp.3387-3408, December 1994.

[59] C. M. Spooner, and W. A. Gardner, "The cumulant theory of cyclostationary time-series. II. Development and applications," *IEEE Trans. Sig. Proc.*, vol. 42, no. 12, pp.3409-3429, December 1994.

[60] S. Pupolin, and C. Tomasi, "Spectral analysis of line regenerator time jitter," *IEEE Trans. Commun.*, vol. COM-32. pp. 561-566, May 1984.

[61] J. J. O'Reilly, "Timing extraction for baseband digital transmission," in *Problems of Randomness in Communication Engineering*, K. W. Cattermole and J. J. O'Reilly, Ed. London: Plymouth, 1984.

[62] L. E. Franks, and J. Bubrouski, "Statistical properties of timing jitter in a PAM timing recovery scheme," *IEEE Trans. Commun.*, vol. COM22, pp. 913-930, July 1974.

[63] L. E. Franks. "Carrier and bit synchronization in data communication-A tutorial review," *IEEE Trans. Commun.*, vol. COM-18, pp. 1107-1121, 1980.

[64] J. C. Campbell, A. J. Gibbs, and B. M. Smith, "The cyclostationary nature of crosstalk interference from digital signals in multipair cable- Part I: Fundamentals," *IEEE Trans. Commun.*, vol. COM-31, pp. 629-637, May 1983.

[65] J. P. A. Albuquerque, O. Shimbo, and L. N. Nygugen, "Modulation transfer noise effects from a continuous digital carrier to FDM/FM carriers in memoryless nonlinear devices," *IEEE Trans. Commun.*, vol. COM-32, pp. 337-353, April 1984.

[66] M. F. Mesiya, P. J. McLane, and L. L. Campbell, "Optimal receiver filters for BPSK transmission over a bandlimited nonlinear channel," *IEEE Trans. Commun.*, vol. COM-26, pp. 12-22, January 1978.

[67] T. H. E. Ericson, "Modulation by means of linear periodic filtering," *IEEE Trans. Inform. Theory*, vol. IT-27, pp. 322-327, May 1981.

[68] E. R. Ferrara Jr., and B. Widrow, "The time-sequenced adaptive filter," *IEEE Trans. Acoust. Speech Signal Processing*, vol. ASSP-29, pp. 679-683, June 1981.

[69] W. A. Gardner, "Optimization and adaptation of linear periodically time-variant digital systems," *Signal and Image Processing Lab*., Dept. Electrical and Computer Engineering, Univ. California, Davis, Tech. Rep. SIPL-85-9, 1985.

[70] G. L. Cariolaro, G. L. Pierobon, and G. P. Tronca, "Analysis of codes and spectra calculations," *Int. J. Electron.*, vol. 55, pp. 35-79, 1983.

[71] C. A. French, and W. A. Gardner, "Spread-spectrum dispreading without the code," *IEEE Trans. Commun.*, vol. COM-34, pp. 404-407, April 1986.

[72] M. H. Ackroyd, "Stationary and cyclostationary finite buffer behaviour computation via Levinson's method," *AT&T Bell Lab. Tech. J.*, vol. 63, pp. 2159-2170, December 1984.

[73] M. Kaplan, "Single-server queue with cyclostationary arrivals and arithmetic service," *Oper. Res.*, vol. 31, pp. 184-205, 1983.

[74] W. A. Gardner, "Signal interception: A unifying theoretical framework for feature detection," in Proc. *Onzieme Colloque sur le Traitement du Signal et des Images*, Nice, France, June 1-5, 1987.

[75] W. A. Gardner and L. Paura, "Signal interception: Performance advantages of cycle detectors," in Proc. *Onzieme Colloque sur le Traitement du Signal et des Images*, Nice, France, June 1-5, 1987.

[76] W. K. Johnson, "The dynamic pneumocardiogram: An application of coherent signal processing to cardiovascular measurement," *IEEE Trans. Biomed. Eng.*, vol. BME-298, pp. 471-475, 1981.

[77] K. Hasselmann and T. P. Barnett, "Techniques of linear prediction for systems with periodic statistics," *J. Atmos. Sci.,* vol. 38, pp. 2275-2283, 1981.

[78] A. V. Vecchia, "Periodic autoregressive-moving average (parma) modeling with applications to water resources," *Water Resource Bull.*, vol 21, pp. 721-730, 1985.

[79] A. G. Miamee, and H. Salehi, "On the prediction of periodically correlated stochastic processes," in *Multivariate Analysis–V, P. R. Krishnaiah*, Ed. Amsterdam: North-Holland, pp. 167-179, 1980.

[80] M. Pagano, and E. Parzen, "An approach to modeling seasonally stationary time-series," *J. Econometrics*, vol. 9, pp. 137-153, 1979.

[81] K. Hasselmann, and T. P. Barnett, "Techniques of linear prediction for systems with periodic statistics," *J. Atmos. Sci.*, vol. 38, pp. 2275-2283, 1981.

[82] L. W. Couch, II, *Digital and Analog Communication System*, Prentice Hall, 6th edition, 2001.

[83] J. G. Proakis, *Digital Communications*, fourth edition, McGraw Hill, 2001.

[84] R. E. Ziemer, and R. W. Peterson, *Introduction to Digital Communication*, second edition, Prentice Hall, 2001.

[85] S. Haykin, "Cognitive Radio: Brain-Empowered Wireless Communications," *IEEE Journ. Sel. Commun.*, vol. 23, pp. 201-220, Febuary 2005.

[86] M. R. Hassan, and B. Nath, "Stock Market Forecasting Using Hidden Markov Model: A New Approach," Proceedings of the 2005 *5th International Conference on Intelligent Systems Design and Applications*, ISDA'05, 2005.

[87] M. Bicego, E. Grosso and E. Otranto, "Recognising and Forecasting Increases and Decreases in Financial Markets via Hidden Markov Models," Dipartimento di Economia, Impresa e Regolamentazione, Via Torre Tonda 34, 07100 Sassari; University of Sassari, Italy.

[88] P. A. Schrodt, "Forecasting Conflict in the Balkans using Hidden Markov Models," paper presented at the American Political Science Association meetings Washington, DC.

[89] A. L. Blum, and R. L. Rivest, "Training a 3-node Neural Networks is NP-complete," *Neural Networks*, vol. 5, pp. 117-127, 1992.

[90] J. S. Judd, *Neural Network Design and Complexity of Learning*, MIT Press, USA, 1990.

[91] W. A. Gardner, "The role of spectral correlation in design and performance analysis of synchronizers," *IEEE Trans. Commun.*, vol. COM-34, pp. 1089-1095, Nov. 1986.

[92] W. M. Gardner, "Measurement of Spectral Correlation," *IEEE Trans. Acoust., Speech, Sig. Proc.*, vol. ASSP-34, no. 5, October 1986.

[93] ____, "The spectral correlation theory of cyclostationary time-series," *Signal Process.*, 1986.

[94] W. A. Brown III, and H. H. Loomis Jr., "Digital implementations of spectral correlation analyzers," *IEEE Trans. Sig. Proc.*, vol.41, no.2, pp.703-720, February 1993.

[95] A. Fehske, J. Gaeddert, and J. H. Reed, "A New Approach to Signal Classification Using Spectral Correlation and Neural Networks," *IEEE Int. Symp. on New Frontiers in DySpan*, November 2005.

[96] X-C. Yin, and N. Hirata, "Simulation based prediction, preliminary forecasting models and forecast tests and evaluations," second *ACES Workshop*, Tokyo, Japan, 2000.

[97] I. F. Akyildiz, W. Y. Lee, M. C. Vuran, and S. Mohanty, "NeXt generation/dynamic spectrum access/cognitive radio wireless networks: A survey," *Computer Networks*, 50, pp. 2127–2159, 2006.

[98] J. Bendat and A. Piersol, *Random Data: Analysis and Measurement Procedures*, Wiley, 1986.

Vita

Ihsan Ali Akbar was born in Karachi on March 16, 1978. He received his Bachelor of Engineering (B.E.) degree in Electrical Engineering from NED University of Engineering and Technology, Karachi, Pakistan in June 1999. He was enrolled as an M.S. Student at Virginia Polytechnic Institute and State University during spring 2000, and later joined the Mobile and Portable Radio Research Group (MPRG) of Virginia Tech as a research assistant during the summer of 2000. From summer 2000 to December 2001, he remained actively involved in wireless channel modeling using discrete channel modeling techniques, and a major portion of his research involved the study of error statistics for third generation wireless channels. He later joined the Ph.D. program of Virginia Tech in spring 2003 and is pursuing a Ph.D. in electrical engineering. His research interests are in the areas of wireless communications including channel models, Markov models, cognitive radio, statistical signal processing, ultra-wideband (UWB) communications, OFDM, and spread spectrum communications.