

# Invariance Properties and Performance Evaluation of Bit Decoding Algorithms

by

Ali Abedi

A thesis

presented to the University of Waterloo

in fulfillment of the

thesis requirement for the degree of

Doctor of Philosophy

in

Electrical and Computer Engineering

Waterloo, Ontario, Canada, 2004

©Ali Abedi 2004

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

## Abstract

Certain properties of optimal bitwise APP (A Posteriori Probability) decoding of binary linear block codes are studied. The focus is on the Probability Density Function (*pdf*) of the bit Log-Likelihood-Ratio (*LLR*). A general channel model with discrete (not necessarily binary) input and discrete or continuous output is considered. It is proved that under a set of mild conditions on the channel, the *pdf* of the bit *LLR* of a specific bit position is independent of the transmitted code-word. It is also shown that the *pdf* of a given bit *LLR*, when the corresponding bit takes the values of zero and one, are symmetric with respect to each other (reflection of one another with respect to the vertical axis). In the case of channels with binary inputs, a sufficient condition for two bit positions to have the same *pdf* is presented. An analytical method for approximate performance evaluation of binary linear block codes using an Additive White Gaussian Noise (AWGN) channel model with Binary Phase Shift Keying (BPSK) modulation is proposed. The *pdf* of the bit *LLR* is expressed in terms of the Gram-Charlier series expansion. This expansion requires knowledge of the statistical moments of the bit *LLR*. An analytical method for calculating these moments which is based on some recursive calculations involving certain weight enumerating functions of the code is introduced. It is proved that the approximation can be as accurate as desired, using enough numbers of terms in the Gram-Charlier series expansion. A new method for the performance evaluation of Turbo-Like Codes is presented. The method is based on estimating the *pdf* of the bit *LLR* by using an exponential model. The moment matching method is combined with the maximum entropy principle to estimate

the parameters of the new model. A simple method is developed for computing the Probabilities of the Point Estimates (PPE) for the estimated parameters, as well as for the Bit Error Rate (BER). It is demonstrated that this method requires significantly fewer samples than the conventional Monte-Carlo (MC) simulation.

## Acknowledgements

I would like to thank my great advisor, Professor Amir Keyvan Khandani, whose guidance and patience during my studies was truly unique. It is really hard to take responsibility for the future career of someone with supervising him and I believe that he did an excellent job on showing me the true path at all different stages of my research. I highly appreciate his putting time and energy and his constructive criticism for training me. He did more than a supervisor for me, even more than a friend, he was like a brother to me.

Many thanks to the other members of my dissertation committee, professors George Freeman, Andrew Huenis, Mary Thompson at the University of Waterloo and Professor Marc Fossorier at the University of Hawaii for their efforts on reviewing my thesis and helping me improve the quality of the presentations. Taking time out of their busy schedule to come to my defense is greatly appreciated. I am truly honored to have such a great examining committee.

This research was supported by funds from Communications and Information Technology Ontario (CITO), Canadian Institute for Telecommunications Research (CITR), the University of Waterloo, and the Natural Sciences and Engineering Research Council of Canada (NSERC). Thank you for your financial support of the research. Special thanks goes to the staff of the Department of Electrical and Computer Engineering at the University of Waterloo, in particular Ms. Wendy Boles, for having been so helpful and supportive.

It is necessary to thank my parents and in-laws, for their endless support at all stages of my studies. None of these would have been possible without the love

of my life, my darling wife: Shahrzad, who sacrificed the best years of her life to support me during my Ph.D studies. Far away from home and parents, she took good care of me and tried to ease the pressure of the work. I would never be able to compensate all the things she did for me, but as a small token of appreciation, I dedicate my thesis to her.

Last but not the least, I would like to add a few words about my son, Ryan. He is truly the most significant phenomena during my Ph.D studies. Ryan has completely changed our life by his sweet presence and by providing us with the opportunity to witness the evolution of a human being. I feel humble in front of such a fast learning baby, and at the same time feel proud of being his father.

To my darling wife: *Shahrzad*

and

our sweet son: *Ryan*

# Contents

- 1 Introduction** **1**
  - 1.1 Digital Communications and Coding Theory . . . . . 1
  - 1.2 Review of the Literature . . . . . 3
  - 1.3 Overview of the Thesis . . . . . 8
  
- 2 Properties of Bit Decoding Algorithms** **10**
  - 2.1 Chapter Overview . . . . . 10
  - 2.2 Modeling . . . . . 11
    - 2.2.1 Channels with a Geometrical Representation . . . . . 13
    - 2.2.2 Channels without Geometrical Representation . . . . . 14
  - 2.3 Main Results . . . . . 17
  - 2.4 Summary . . . . . 23
  
- 3 Performance Evaluation of Binary Linear Block Codes** **24**
  - 3.1 Chapter Overview . . . . . 24
  - 3.2 Modeling . . . . . 25
  - 3.3 Gram-Charlier Expansion of *pdf* . . . . . 28

3.4	Computing Moments Using Taylor Expansion of $LLR$ . . . . .	29
3.5	Computing Probability of Error . . . . .	33
3.6	Convergence properties . . . . .	35
3.7	Numerical Results . . . . .	36
3.8	Summary . . . . .	38
<b>4</b>	<b>Performance Evaluation of Turbo-Like Codes</b>	<b>41</b>
4.1	Chapter Overview . . . . .	41
4.2	Modeling the $pdf$ of the Bit $LLR$ . . . . .	42
4.3	Moment Matching Using the Maximum Entropy Principle . . . . .	43
4.4	Probabilities of the Point Estimates (PPE) . . . . .	47
4.4.1	PPE for the Estimated Parameters . . . . .	47
4.4.2	PPE for the Estimated BER . . . . .	49
4.5	Numerical Results . . . . .	51
4.6	Summary . . . . .	54
<b>5</b>	<b>Concluding Remarks</b>	<b>56</b>
5.1	Summary of the Contributions . . . . .	56
5.2	Future Work . . . . .	57
<b>A</b>	<b>Proofs of Theorems</b>	<b>60</b>
A.1	Proof of Theorem 4 . . . . .	60
A.2	Proof of Theorem 5 . . . . .	61
A.3	Proof of Property (3.44) . . . . .	62

<b>B</b>	<b>Moment Estimation</b>	<b>64</b>
B.1	Error in Moment Estimation . . . . .	64
B.2	Covariance Matrix of the Moments . . . . .	66
<b>C</b>	<b>Properties of the Monte-Carlo (MC) Simulation</b>	<b>67</b>
C.1	Variance of the MC Simulation . . . . .	67
C.2	Computing PPEs for the MC Simulation . . . . .	68
<b>D</b>	<b>Cumulant Method</b>	<b>70</b>

# List of Tables

4.1	Comparison of the proposed method and the MC simulation. . . . .	51
4.2	Relation between $n$ and PPE at $E_b/N_0=2\text{dB}$ . . . . .	53

# List of Figures

2.1	Channel Model . . . . .	12
2.2	Mapping of points with an isometry . . . . .	14
2.3	Channel model for example 1. . . . .	16
2.4	Channel model for example 2. . . . .	17
2.5	Channel model for example 3 . . . . .	18
3.1	BER for (15,11,3) Cyclic code. . . . .	37
3.2	BER for (12,11,2) single parity check code. . . . .	38
3.3	BER for binary extended (24,12,8) Golay code. . . . .	39
3.4	Flow chart of the proposed method. . . . .	40
4.1	BER curves for Turbo-Code of the length 100 and rate 1/2. . . . .	54

# List of Acronyms

<b>AWGN</b>	: Additive White Gaussian Noise
<b>BER</b>	: Bit Error Rate
<b>BPSK</b>	: Binary Phase Shift Keying
<b>CDF</b>	: Cumulative Distribution Function
<b>COV</b>	: Covariance
<b>FER</b>	: Frame Error Rate
<b>GT</b>	: Gaussian Tail
<b>IFT</b>	: Inverse Fourier Transform
<b>IS</b>	: Importance Sampling
<b>ISI</b>	: Inter Symbol Interference
<b>LDPC</b>	: Low Density Parity Check Codes
<b>LFSR</b>	: Linear Feedback Shift Register
<b>LLR</b>	: Log Likelihood Ratio
<b>MAP</b>	: Maximum a Posteriori Probability
<b>MC</b>	: Monte-Carlo
<b>ML</b>	: Maximum Likelihood
<b>MT</b>	: Mean Translation

**PAM** : Pulse Amplitude Modulation  
**pdf** : Probability Density Function  
**pmf** : Probability Mass Function  
**PPE** : Probabilities of the Point Estimates  
**QAM** : Quadrature Amplitude Modulation  
**RT** : Rayleigh Tail  
**SNR** : Signal to Noise Ratio  
**VS** : Variance Scaling

# Chapter 1

## Introduction

### 1.1 Digital Communications and Coding Theory

Digital Communications is an essential ingredient of our modern life. The main steps toward developing a practical system are: requirement analysis, design, and performance analysis. There have been numerous research works in designing a high performance digital communication system, but very few in the area of performance evaluation. The reason is that, while one can perform a simulation to get an approximate idea of the performance of conventional communications systems, this is not the case when dealing with high performance systems in very low bit error rate regions. Simulation methods for state of the art codes are time intensive and costly. This is why analytical performance evaluation techniques are becoming more important. These methods are meant to be precise, fast and simple to significantly reduce the cost of designing a communication system.

The demand for having reliable communications systems is the main motivation

behind the channel coding theory. Channel coding allows for the exchange of signal power and signal bandwidth without performance loss. In other words, adaptation to transmission conditions is possible with channel coding. Channel coding is applied to ensure adequate transmission quality by adding redundancy to the data stream. This redundancy enables the receiver to detect/correct transmission errors.

Shannon published his fundamental work on channel capacity in 1948 [1], which describes the maximum data rate that can be reliably transmitted over a certain channel. Since then, there have been numerous efforts to reach the channel capacity by designing a good code. One of the main obstacles in designing a high performance code is performance evaluation. Detecting a few possible errors at high Signal to Noise Ratio (SNR) regions requires millions of samples.

The importance of this work in the analysis of a coded system is very crucial as it will reduce the number of samples required for performance analysis and will address some hidden properties of the channel coder which may help in designing more efficient codes and less complex decoders. The conventional simulation methods are very time consuming and require a huge amount of computational power. These are the main reasons that make them expensive and in some cases infeasible to be performed. The proposed methods in this thesis will have a great impact on the communications field by saving a huge amount of time and energy in the design and test of high quality systems.

## 1.2 Review of the Literature

In the application of channel codes, one of the major problems is the development of an efficient decoding algorithm for a given code. The class of Maximum Likelihood (ML) decoding algorithms are designed to find a valid code-word with the maximum likelihood value. The ML algorithms are known to minimize the Frame Error Rate (FER) under conditions where code-words occur with equal probability.

Another class of decoding algorithms, known as bit decoding, compute the probability of the individual bits and decide on the corresponding bit values independent of each other. The straightforward approach to bit decoding is based on summing up the probabilities of different code-words according to the value of their component in a given bit position of interest. Reference [2] provides an efficient method (known as BCJR) to compute the bit probabilities of a given code using its trellis diagram. There are some special methods for bit decoding based on the coset decomposition principle [3], sectionalized trellis diagrams [4], and using the dual code [5,6].

Maximum Likelihood decoding algorithms have been the subject of numerous research activities, while bit decoding algorithms have received much less attention in the past. More recently, bit decoding algorithms have received increasing attention, mainly due to the fact that they deliver bit reliability information. This reliability information has been effectively used in a variety of applications including Turbo decoding.

In 1993, a new class of channel codes, called Turbo-Codes, were announced [7], which have an astonishing performance and at the same time allow for a sim-

ple iterative decoding method using the reliability information produced by a bit decoding algorithm. Due to the importance of Turbo-Codes, there has been a growing interest among communication researchers to work on the bit decoding algorithms.

Probability density function (*pdf*) of the bit Log-Likelihood-Ratio (*LLR*) can be used as a tool for analysis of bit decoding algorithms. A recent work [8] on analysis of Sum-Product decoding of Low-Density-Parity-Check (LDPC) codes takes advantage of certain symmetry properties for *pdf* of bit *LLR* over binary input channels with Additive White Gaussian Noise (AWGN) interference. It is shown in [9] that for a binary input, *output – symmetric* channel defined in [10] (assuming that the all zero code-word is transmitted), the bit *LLR* at each node of the code graph has a symmetric *pdf* (refer to [9] for the definition of symmetry) and this symmetry is preserved under belief propagation decoding. Note that the definition of “symmetry” in the current thesis is different from [9]. In [10], it is shown that for a binary input, *output – symmetric* channel, the conditional probability of error is independent of the transmitted code-word. A more general result concerning the invariance property of the *pdf* of the bit *LLR* is proved in theorem 1 of section 2 (note that a more general channel model is used as compared to [10]).

The analytical performance evaluation of symbol by symbol decoders is considered a hard task in [11, 12]. Although there is a method for calculating exact performance (in the sense of expected Hamming distortion) of Viterbi decoding of convolutional codes over Binary Symmetric Channels [13], there has been no method for performance evaluation of bit decoding in general. Some asymptotic

expressions are derived in [14] for bit error probability of binary linear block codes in the AWGN channel with bit decoding. The bit error probabilities of convolutional codes over Binary Symmetric Channels is considered in [12] with ML decoding. An upper bound is presented in [15] for the performance of finite-delay symbol-by-symbol decoding of trellis codes over discrete memoryless channels.

In this thesis, Gram-Charlier series expansion is employed to find the *pdf* of the bit *LLR*. This method is used in some other communications applications, including calculation of *pdf* of the sum of Log-Normal variates [16], evaluation of the error probability in PAM (Pulse Amplitude Modulation) digital data transmission systems with correlated symbols in the presence of inter-symbol interference and additive noise [17], computing nearly Gaussian distributions [18], and computation of the error probability of equal-gain combiners with partially coherent fading signals [19]. Reference [20] presents a method for computing an unknown *pdf* using infinite series (also refer to [21]). Reference [22] computes moments of phase noise and uses maximum entropy criterion [23] to find the corresponding *pdf*.

One of the most important issues in coding theory is to develop an efficient method for performance evaluation, since the Monte-Carlo (MC) simulation is extremely time consuming for low Bit Error Rate (BER) values. There have been numerous efforts devoted to the performance evaluation of Turbo-Codes. These approaches derive some bounds on the average performance of Turbo-Codes by assuming ML decoding [24, 25, 26].

The relations between the frame error probability and the decoding algorithms for block codes are reviewed in [27]. Decoding a linear block code over binary symmetric channels is considered in [28]. This reference also provides a concise

literature survey. The bit error probability for maximum-likelihood decoding of binary linear block codes over AWGN channels is investigated in [29], where for randomly generated codes, it is shown that the conventional approximation of the block error probability at high SNR, holds for systematic encoding only. Also systematic encoding provides the minimum bit error probability when the inverse mapping corresponding to the generator matrix of the code is used to retrieve the information sequence. Some researchers have considered simplified cases of analytical BER calculations. Exact analysis of bit error probability for 4-state soft decision Viterbi decoding is considered in [30]. An analytical method for computing the bit error probability of a two-state convolutional code with Maximum a Posteriori Probability (MAP) decoding is presented in [31]. The Pearson system of distributions is adopted in [32] to compute the error probability expectations, where moment matching is used to estimate the parameters of the model. Estimating the parameters of the generalized Gaussian *pdf* by using entropy matching is considered in [33].

Other researchers have employed the Importance Sampling (IS) method to improve the performance of the MC simulation by increasing the weight of the rare error events. In this method, instead of choosing the samples from the original distribution, the samples are selected from a modified distribution which concentrates the points where the rare error events occur. This modified distribution is obtained from the original distribution by the application of a biasing function. This ensures a variance reduction if the biasing function is appropriately selected. The Gaussian Tail (GT) and Rayleigh Tail (RT) biasing functions are investigated in [34]. The IS method is applied to evaluate the performance of a digital com-

munications system with Inter-Symbol Interference (ISI) in [35], and is extended to evaluate the performance of multi-hop satellite links in [36]. A general formulation of the IS method in probability space notation is introduced in [37]. The IS method is used in [38] to simulate the Viterbi decoder by examining the trellis structure in relation to the rare error events. A comparison between the Mean Translation (MT) technique and the Variance Scaling (VS) technique is performed in [39], where it is shown that the structure of the error regions determines the better method. Recently, [40] has revisited the IS method with the strategy to increase the rate by which the variance approaches zero, instead of reducing the variance itself.

The Turbo-Product Codes (of a small block length) are simulated in [41] by partitioning the error regions and by using MT for each sub region independently. This method becomes inefficient as the complexity of the code increases. In the case of Turbo-Codes with a large block length, the search for the appropriate biasing functions may be lengthy, which renders this method even more complicated than the conventional MC simulation.

It is observed in [7, 26, 42] that the *pdf* of the bit *LLR* is nearly Gaussian. This motivates us to propose an exponential model which has a polynomial in the exponent. The aforementioned model has the ability to efficiently capture the deviation of the desired *pdf* from Gaussian. The moments of the bit *LLR* are used to estimate the parameters for the proposed model.

### 1.3 Overview of the Thesis

Chapter 2 investigates certain properties of the bit decoding algorithms for the case of linear binary block codes. The focus is on the *pdf* of the bit *LLR* using a general channel model with discrete input and discrete or continuous output. It is proved that under a set of mild conditions on the channel, the *pdf* of the bit *LLR* of a specific bit position is independent of the transmitted code-word. It is also shown that the *pdf* of a given bit *LLR* when the corresponding bit takes the values of zero and one are symmetric with respect to each other (reflection of one another with respect to the vertical axis). For the case of channels with binary input, a sufficient condition for two bit positions to have the same *pdf* is presented. Such a condition can be found by examining the code automorphism group of the code. It is shown that for the class of Cyclic codes this sufficient condition is always satisfied. This means that any given two bit positions in a Cyclic code have the same *pdf* for their bit *LLR*.

An analytical method for approximate performance evaluation of binary linear block codes using an AWGN channel model with BPSK modulation is presented in Chapter 3. The *pdf* of the bit *LLR* is expressed in terms of the Gram-Charlier series expansion. This expansion requires knowledge of the statistical moments of the bit *LLR*. An analytical method for calculating these moments is introduced. This is based on some recursive calculations involving certain weight enumerating functions of the code. It is proved that the approximation can be as accurate as desired by using enough number of terms in the Gram-Charlier series expansion. Numerical results are provided for some examples, which demonstrate close

agreement with simulation results.

Chapter 4 presents a new method for the performance evaluation of Turbo-Like Codes. The method is based on estimating the *pdf* of the bit *LLR* by using an exponential model. It is widely known that the *pdf* of the bit *LLR* is close to the normal density, and the proposed approach takes advantage of this property to simplify the calculations. The moment matching method is combined with the maximum entropy principle to estimate the parameters of the new model. A simple method is developed for computing the probabilities of the point estimates for the estimated parameters, as well as for the BER. The corresponding results are adopted to compute the number of samples that are required for a given precision of the estimated values. It is demonstrated that this method requires significantly fewer samples than the conventional MC simulation. A summary of the contributions and future work can be found in Chapter 5. This thesis is prepared using *hyperref* package, all the equations and literature reference links in the document are active to make it easy to navigate.

# Chapter 2

## Properties of Bit Decoding Algorithms

### 2.1 Chapter Overview

This chapter is organized as follows. In Section 2.2, the model used to analyze the problem is presented. All notations and assumptions are given in this Section. Some theorems<sup>1</sup> are proved on bit decoding algorithms in Section 2.3. The summary is presented in Section 2.4.

---

<sup>1</sup>This work is a continuation of [43], in which the case of AWGN channel with BPSK modulation is considered.

## 2.2 Modeling

Assume that a binary linear code  $\mathcal{C}$  with code-words of length  $N$  is given. Notation  $\mathbf{c}^i = (c_1^i, c_2^i, \dots, c_N^i)$  is used to refer to the  $i^{\text{th}}$  code-word and its elements. The code is partitioned into a sub-code  $C_k^0$  and its coset  $C_k^1$  according to the value of the  $k^{\text{th}}$  bit position of its code-words. i.e.,

$$C_k^i = \{\mathbf{c} \in \mathcal{C} : c_k = i\}, \quad i = 0, 1 \quad (2.1)$$

Bit wise binary addition of two code-words on the code book is denoted as,  $\mathbf{c}^i \oplus \mathbf{c}^j$ . Note that the sub-code  $C_k^0$  is closed under binary addition. Each code-word is partitioned into  $L$  blocks of  $m$  bits, assuming  $N = mL$ , to be transmitted over a channel with a discrete input alphabet set composed of  $2^m$  elements. Notation  $\mathbf{I}_j^i$ ,  $i = 1, \dots, |\mathcal{C}|$ ,  $j = 1, \dots, L$ , is used for these blocks, which will be called  $m$ -blocks hereafter. For example, code-word  $\mathbf{c}^i$  is composed of  $(\mathbf{I}_1^i, \mathbf{I}_2^i, \dots, \mathbf{I}_L^i)$ . Assume that there exists a one to one correspondence between the  $2^m$  possible  $m$ -blocks and the input symbols of the channel. The set of  $m$ -blocks referred as  $\mathcal{I}$  forms a group under binary addition.

The channel has  $2^m$  discrete input and discrete or continuous output as shown in Figure 2.1. For channels with discrete output,  $\mathcal{O}$  is a set of discrete alphabets and  $p(\cdot)$  stands for the probability mass function (*pmf*). Without loss of generality, assume  $\mathcal{O} = \{1, 2, \dots, v\}$ , set of integer numbers. For the continuous output channels,  $\mathcal{O} \subset \mathfrak{R}^n$ , where  $\mathfrak{R}$  is the set of real numbers,  $n$  is the size of vector  $\mathbf{x}$ , and  $p(\cdot)$  stands for *pdf*.

Consider the situation of sending a code-word  $\tilde{\mathbf{c}} = (\tilde{\mathbf{I}}_1, \dots, \tilde{\mathbf{I}}_L)$  through the

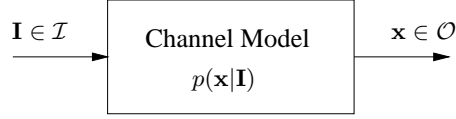


Figure 2.1: Channel Model

channel. Each  $m$ -block,  $\tilde{\mathbf{I}}_j$ ,  $j = 1 \dots L$ , is transmitted and a symbol  $\mathbf{x}_j$ ,  $j = 1 \dots L$ , is received at the channel output. A common tool to express the bit probabilities in bit decoding algorithms is based on using the so-called Log-Likelihood-Ratio ( $LLR$ ). The  $LLR$  of the  $k^{th}$  bit position is defined by the following equation,

$$LLR_{\tilde{\mathbf{c}}}(k) = \log \frac{P(\tilde{c}_k = 1 | \mathbf{x}_1 \dots \mathbf{x}_L)}{P(\tilde{c}_k = 0 | \mathbf{x}_1 \dots \mathbf{x}_L)}, \quad (2.2)$$

where  $\tilde{c}_k$  is the value of the  $k^{th}$  bit in the transmitted code-word and  $\log$  stands for natural logarithm. Assuming,

$$P(\tilde{c}_k = 0) = P(\tilde{c}_k = 1) = \frac{1}{2}, \quad (2.3)$$

for a memoryless channel we have,

$$LLR_{\tilde{\mathbf{c}}}(k) = \log \frac{\sum_{\mathbf{c}^i \in C_k^1} p(\mathbf{x}_1 \dots \mathbf{x}_L | \mathbf{c}^i)}{\sum_{\mathbf{c}^i \in C_k^0} p(\mathbf{x}_1 \dots \mathbf{x}_L | \mathbf{c}^i)} = \log \frac{\sum_{\mathbf{c}^i \in C_k^1} \prod_{j=1}^L p(\mathbf{x}_j | \mathbf{I}_j^i)}{\sum_{\mathbf{c}^i \in C_k^0} \prod_{j=1}^L p(\mathbf{x}_j | \mathbf{I}_j^i)}. \quad (2.4)$$

Assuming a linear code, a set of conditions on the channel is derived for which the choice of  $\tilde{\mathbf{c}}$  does not have any impact on the  $pdf$  of  $LLR_{\tilde{\mathbf{c}}}(k)$  as long as the value of the  $k^{th}$  bit remains unchanged. This is a generalization of the distance invariance property at the bit level. It will be also shown that, under the same

set of conditions, the *pdf* of a given bit *LLR* when the corresponding bit takes the values of zero and one are symmetric with respect to each other. For the case of channels with binary input ( $m = 1$ ) a sufficient condition for the *LLR* of two bit positions to have the same *pdf* is presented.

The following sufficient condition is required to carry out the proofs. If for any  $\tilde{\mathbf{I}} \in \mathcal{I}$  and any  $\mathbf{x} \in \mathcal{O}$  one can find a  $\mathbf{y} \in \mathcal{O}$  such that for all  $\mathbf{I} \in \mathcal{I}$ ,

$$p(\mathbf{x}|\mathbf{I} \oplus \tilde{\mathbf{I}}) = p(\mathbf{y}|\mathbf{I}), \quad (2.5)$$

i.e.,

$$\forall \tilde{\mathbf{I}} \in \mathcal{I}, \forall \mathbf{x} \in \mathcal{O}, \exists \mathbf{y} \in \mathcal{O} : p(\mathbf{x}|\mathbf{I} \oplus \tilde{\mathbf{I}}) = p(\mathbf{y}|\mathbf{I}), \quad \forall \mathbf{I} \in \mathcal{I}. \quad (2.6)$$

This is obviously equivalent to,

$$\forall \tilde{\mathbf{I}}^1, \tilde{\mathbf{I}}^2 \in \mathcal{I}, \forall \mathbf{x} \in \mathcal{O}, \exists \mathbf{y} \in \mathcal{O} : p(\mathbf{x}|\mathbf{I} \oplus \tilde{\mathbf{I}}^1 \oplus \tilde{\mathbf{I}}^2) = p(\mathbf{y}|\mathbf{I}), \quad \forall \mathbf{I} \in \mathcal{I}, \quad (2.7)$$

however, the form given in (2.7) is more convenient to use in the proof of the theorems.

### 2.2.1 Channels with a Geometrical Representation

Notation  $\mathbf{P}_{\mathbf{I}^i} \in \mathfrak{R}^n$  is used to refer to the channel input symbols representing  $\mathbf{I}^i$ . In this case, the  $m$ -blocks are just labels of the points in an Euclidean space. Assume that the signal set at the channel input is geometrically uniform [44]. This means that for any given pair of signal points, say  $\mathbf{P}_{\tilde{\mathbf{I}}^1}$  and  $\mathbf{P}_{\tilde{\mathbf{I}}^2}$ , there exists an isometry which transforms  $\mathbf{P}_{\tilde{\mathbf{I}}^1}$  to  $\mathbf{P}_{\tilde{\mathbf{I}}^2}$  while leaving the signal set unchanged. In addition, assume that transmission of  $\tilde{\mathbf{I}}^1$  and  $\tilde{\mathbf{I}}^2$  resulted in receiving  $\mathbf{x}$  and  $\mathbf{y}$  at the receiver,

respectively. This scenario is shown in Figure 2.2, and it is assumed to be also valid for the corresponding labels.

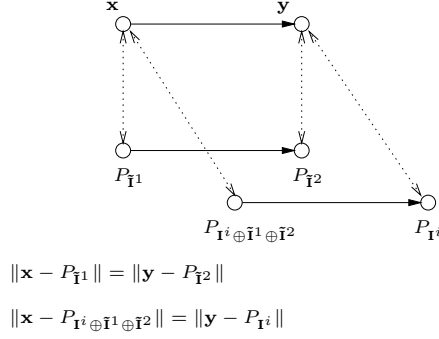


Figure 2.2: Mapping of points with an isometry

It is easy to see that under the following conditions,

- (i)  $\mathbf{y}$  is selected as the image of  $\mathbf{x}$  under the isometry  $\mathbf{P}_{\mathbf{I}^1} \implies \mathbf{P}_{\mathbf{I}^2}$
- (ii)  $p(\mathbf{x}|\mathbf{P}_{\mathbf{I}^1})$  is a function of  $\|\mathbf{x} - \mathbf{P}_{\mathbf{I}^1}\|$ ,  $\forall \mathbf{x}, \forall \mathbf{P}_{\mathbf{I}^1}$

the condition given in (2.7) will be satisfied. A well known example for a channel satisfying condition (ii), is the AWGN channel.

## 2.2.2 Channels without Geometrical Representation

In this section, assume that the channel output set is a discrete set composed of elements  $\mathbf{x}^j \in \mathcal{O}$ . The channel is characterized by a matrix of transition probabilities,  $\mathbf{A}$ .

$$\mathbf{A}_{u \times v} = [a_{ij}], \quad a_{ij} = p(\mathbf{x}^j | \mathbf{I}^i), \quad u = 2^m = |\mathcal{I}|, \quad v = |\mathcal{O}|. \quad (2.8)$$

The condition given in (2.7) can be satisfied, if after permuting all input symbols by adding an arbitrary  $m$ -block  $\mathbf{I}$  to them, for each column in  $\mathbf{A}_{u \times v}$ , there exists

another column for which the probability values are shuffled in the same order as the corresponding  $m$ -blocks. It appears that in this case the proposed channel model is equivalent to a *Regular* channel. Reference [45] defines the concept of the *Regular* channel as follows. Assume that permutation  $\psi_{\mathbf{I}}$  acts on the set  $\mathcal{O}$  with the property,

$$\forall \mathbf{I}^1, \mathbf{I}^2 \in \mathcal{I}, \forall \mathbf{x}^j \in \mathcal{O} \quad \psi_{\mathbf{I}^1}(\psi_{\mathbf{I}^2}(\mathbf{x}^j)) = \psi_{\mathbf{I}^1 \oplus \mathbf{I}^2}(\mathbf{x}^j). \quad (2.9)$$

The channel is called a *Regular* channel, if the probability  $p(\mathbf{x}^j | \mathbf{I}^i)$  only depends on  $\psi_{\mathbf{I}^i}(\mathbf{x}^j)$ . It can be verified easily that a *Regular* channels is always *Symmetric* in sense of Gallager [46], where in [46] the symmetry condition only involves the channel symbols and not the underlying labeling. For a recent introduction to the *Regular* channels refer to [47].

Here are some examples for the discrete case.

**Example 1:** For the channel shown in Figure 2.3 we have,

$$\mathbf{A} = \left[ \begin{array}{c|cccc} & \mathbf{x}^1 & \mathbf{x}^2 & \mathbf{x}^3 & \mathbf{x}^4 \\ \hline 0 & 1/2 - \epsilon_1 & \epsilon_1 & 1/2 - \epsilon_2 & \epsilon_2 \\ 1 & \epsilon_1 & 1/2 - \epsilon_1 & \epsilon_2 & 1/2 - \epsilon_2 \end{array} \right] \quad (2.10)$$

**Example 2:** For the channel shown in Figure 2.4 we have,

$$\mathbf{A} = \left[ \begin{array}{c|ccccc} & \mathbf{x}^1 & \mathbf{x}^2 & \mathbf{x}^3 & \mathbf{x}^4 & \mathbf{x}^5 \\ \hline 00 & e_0 & e_1 & \epsilon & 0 & 0 \\ 01 & e_1 & e_0 & \epsilon & 0 & 0 \\ 10 & 0 & 0 & \epsilon & e_0 & e_1 \\ 11 & 0 & 0 & \epsilon & e_1 & e_0 \end{array} \right] \quad (2.11)$$

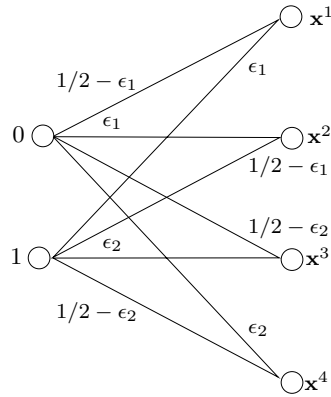


Figure 2.3: Channel model for example 1.

where  $e_0 + e_1 + \epsilon = 1$ .

**Example 3:** For the channel shown in Figure 2.5 we have,

$$\mathbf{A} = \left[ \begin{array}{c|cccccccc} & \mathbf{x}^1 & \mathbf{x}^2 & \mathbf{x}^3 & \mathbf{x}^4 & \mathbf{x}^5 & \mathbf{x}^6 & \mathbf{x}^7 & \mathbf{x}^8 \\ \hline 00 & e_0 & e_1 & e_2 & e_3 & e_4 & e_3 & e_2 & e_1 \\ 01 & e_2 & e_1 & e_0 & e_1 & e_2 & e_3 & e_4 & e_3 \\ 10 & e_4 & e_3 & e_2 & e_1 & e_0 & e_1 & e_2 & e_3 \\ 11 & e_2 & e_3 & e_4 & e_3 & e_2 & e_1 & e_0 & e_1 \end{array} \right] \tag{2.12}$$

where  $e_0 + 2(e_1 + e_2 + e_3) + e_4 = 1$ . The values of error probabilities which are not shown follow the same pattern as the values specified on the figure. It is easy to see that the required condition for the columns of the probability matrix are satisfied in all of the above examples.

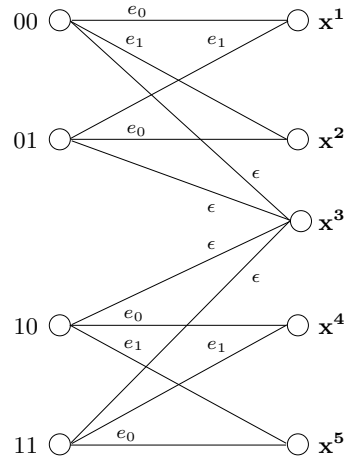


Figure 2.4: Channel model for example 2.

## 2.3 Main Results

In this Section the symmetry properties of the bit  $LLR$  are presented in form of some theorems.

**Theorem 1** *The pdf of  $LLR(k)$  is not affected by the choice of the transmitted code-word  $\tilde{\mathbf{c}}$ , as long as the value of the  $k^{th}$  bit remains unchanged and the channel satisfies condition (2.7).*

*Proof:* Consider two code-words,  $\tilde{\mathbf{c}}^1, \tilde{\mathbf{c}}^2$  which have the same value in their  $k^{th}$  bit position. Let us assume that  $\tilde{\mathbf{c}}^1$  is transmitted through the channel and  $(\mathbf{x}_1 \dots \mathbf{x}_L)$  is received. This results in a realization of random variable  $LLR_{\tilde{\mathbf{c}}^1}(k)$

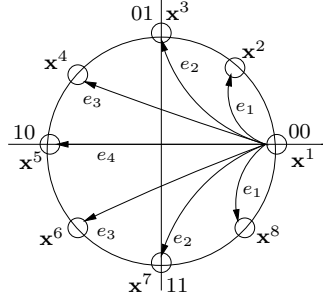


Figure 2.5: Channel model for example 3

with a value of,

$$LLR_{\tilde{\mathbf{c}}^1}(k) = \log \frac{\sum_{\mathbf{c}^i \in C_k^1} p(\mathbf{x}_1 \dots \mathbf{x}_L | \mathbf{c}^i)}{\sum_{\mathbf{c}^i \in C_k^0} p(\mathbf{x}_1 \dots \mathbf{x}_L | \mathbf{c}^i)} = \log \frac{\sum_{\mathbf{c}^i \in C_k^1} \prod_{j=1}^L p(\mathbf{x}_j | \mathbf{I}_j^i)}{\sum_{\mathbf{c}^i \in C_k^0} \prod_{j=1}^L p(\mathbf{x}_j | \mathbf{I}_j^i)}, \quad (2.13)$$

that occurs with probability  $p(\mathbf{x}_1 \dots \mathbf{x}_L | \tilde{\mathbf{c}}^1)$ . Noting that  $\tilde{\mathbf{c}}^1 \oplus \tilde{\mathbf{c}}^2 \in C_k^0$ , it is easy to show that,

$$LLR_{\tilde{\mathbf{c}}^1}(k) = \log \frac{\sum_{\mathbf{c}^i \in C_k^1} p(\mathbf{x}_1 \dots \mathbf{x}_L | \mathbf{c}^i \oplus \tilde{\mathbf{c}}^1 \oplus \tilde{\mathbf{c}}^2)}{\sum_{\mathbf{c}^i \in C_k^0} p(\mathbf{x}_1 \dots \mathbf{x}_L | \mathbf{c}^i \oplus \tilde{\mathbf{c}}^1 \oplus \tilde{\mathbf{c}}^2)} = \log \frac{\sum_{\mathbf{c}^i \in C_k^1} \prod_{j=1}^L p(\mathbf{x}_j | \mathbf{I}_j^i \oplus \tilde{\mathbf{I}}_j^1 \oplus \tilde{\mathbf{I}}_j^2)}{\sum_{\mathbf{c}^i \in C_k^0} \prod_{j=1}^L p(\mathbf{x}_j | \mathbf{I}_j^i \oplus \tilde{\mathbf{I}}_j^1 \oplus \tilde{\mathbf{I}}_j^2)}, \quad (2.14)$$

where  $\tilde{\mathbf{I}}_j^1, \tilde{\mathbf{I}}_j^2, \mathbf{I}_j^i$  are the  $j^{\text{th}}$   $m$ -blocks of the code-words  $\tilde{\mathbf{c}}^1, \tilde{\mathbf{c}}^2, \mathbf{c}^i$ , respectively.

If  $\tilde{\mathbf{c}}^2$  is transmitted, assuming condition (2.7) is satisfied, there exists a  $\mathbf{y} = (\mathbf{y}_1 \dots \mathbf{y}_L)$ ,  $\mathbf{y}_j \in \mathcal{O}$ ,  $j = 1, \dots, L$  such that,

$$p(\mathbf{y}_j | \tilde{\mathbf{I}}_j^2) = p(\mathbf{x}_j | \tilde{\mathbf{I}}_j^1). \quad (2.15)$$

Noting that the channel is memoryless, from (2.15), we conclude that,

$$\prod_{j=1}^L p(\mathbf{y}_j | \tilde{\mathbf{I}}_j^2) = \prod_{j=1}^L p(\mathbf{x}_j | \tilde{\mathbf{I}}_j^1), \quad (2.16)$$

$$p(\mathbf{y}_1 \dots \mathbf{y}_L | \tilde{\mathbf{c}}^2) = p(\mathbf{x}_1 \dots \mathbf{x}_L | \tilde{\mathbf{c}}^1). \quad (2.17)$$

This  $(\mathbf{y}_1 \dots \mathbf{y}_L)$  results in a realization of random variable  $LLR_{\tilde{\mathbf{c}}^2}(k)$  with a value of,

$$LLR_{\tilde{\mathbf{c}}^2}(k) = \log \frac{\sum_{\mathbf{c}^i \in C_k^1} p(\mathbf{y}_1 \dots \mathbf{y}_L | \mathbf{c}^i)}{\sum_{\mathbf{c}^i \in C_k^0} p(\mathbf{y}_1 \dots \mathbf{y}_L | \mathbf{c}^i)} = \log \frac{\sum_{\mathbf{c}^i \in C_k^1} \prod_{j=1}^L p(\mathbf{y}_j | \mathbf{I}_j^i)}{\sum_{\mathbf{c}^i \in C_k^0} \prod_{j=1}^L p(\mathbf{y}_j | \mathbf{I}_j^i)}. \quad (2.18)$$

Using condition (2.7), we conclude that (2.14) and (2.18) are equal to each other. This means that for each realization of the random variable  $LLR_{\tilde{\mathbf{c}}^1}(k)$ , there exists a realization of the random variable  $LLR_{\tilde{\mathbf{c}}^2}(k)$  with the same value and occurring with the same probability, i.e.,  $p(\mathbf{y}_1 \dots \mathbf{y}_L | \tilde{\mathbf{c}}^2) = p(\mathbf{x}_1 \dots \mathbf{x}_L | \tilde{\mathbf{c}}^1)$ . This completes the proof that the random variables  $LLR_{\tilde{\mathbf{c}}^1}(k)$  and  $LLR_{\tilde{\mathbf{c}}^2}(k)$  have the same *pdf*.  $\square$

**Theorem 2** *The pdf of  $LLR(k)$  for  $c_k = 0$  or 1 are the reflections of one another through the vertical axis, if the channel satisfies condition (2.7). i.e.,  $p(LLR(k = 1)) = p(-LLR(k = 0))$ .*

*Proof:* Consider two code-words,  $\tilde{\mathbf{c}}^1, \tilde{\mathbf{c}}^2$  which have different values in their  $k^{th}$  bit position. Let us assume that  $\tilde{\mathbf{c}}^1$  is transmitted through the channel and

$(\mathbf{x}_1 \dots \mathbf{x}_L)$  is received. This results in a realization of random variable  $LLR_{\tilde{\mathbf{c}}^1}(k)$  with a value of,

$$LLR_{\tilde{\mathbf{c}}^1}(k) = \log \frac{\sum_{\mathbf{c}^i \in C_k^1} p(\mathbf{x}_1 \dots \mathbf{x}_L | \mathbf{c}^i)}{\sum_{\mathbf{c}^i \in C_k^0} p(\mathbf{x}_1 \dots \mathbf{x}_L | \mathbf{c}^i)} = \log \frac{\sum_{\mathbf{c}^i \in C_k^1} \prod_{j=1}^L p(\mathbf{x}_j | \mathbf{I}_j^i)}{\sum_{\mathbf{c}^i \in C_k^0} \prod_{j=1}^L p(\mathbf{x}_j | \mathbf{I}_j^i)}, \quad (2.19)$$

that occurs with probability  $p(\mathbf{x}_1 \dots \mathbf{x}_L | \tilde{\mathbf{c}}^1)$ . Noting the  $\tilde{\mathbf{c}}^1 \oplus \tilde{\mathbf{c}}^2 \in C_k^1$ , it is easy to show that,

$$LLR_{\tilde{\mathbf{c}}^1}(k) = \log \frac{\sum_{\mathbf{c}^i \in C_k^0} \prod_{j=1}^L p(\mathbf{x}_j | \mathbf{I}_j^i \oplus \tilde{\mathbf{I}}_j^1 \oplus \tilde{\mathbf{I}}_j^2)}{\sum_{\mathbf{c}^i \in C_k^1} \prod_{j=1}^L p(\mathbf{x}_j | \mathbf{I}_j^i \oplus \tilde{\mathbf{I}}_j^1 \oplus \tilde{\mathbf{I}}_j^2)} = -\log \frac{\sum_{\mathbf{c}^i \in C_k^1} \prod_{j=1}^L p(\mathbf{x}_j | \mathbf{I}_j^i \oplus \tilde{\mathbf{I}}_j^1 \oplus \tilde{\mathbf{I}}_j^2)}{\sum_{\mathbf{c}^i \in C_k^0} \prod_{j=1}^L p(\mathbf{x}_j | \mathbf{I}_j^i \oplus \tilde{\mathbf{I}}_j^1 \oplus \tilde{\mathbf{I}}_j^2)}, \quad (2.20)$$

where  $\tilde{\mathbf{I}}_j^1, \tilde{\mathbf{I}}_j^2, \mathbf{I}_j^i$  are the  $j^{\text{th}}$   $m$ -blocks of the code-words  $\tilde{\mathbf{c}}^1, \tilde{\mathbf{c}}^2, \mathbf{c}^i$ , respectively.

Assuming condition (2.7) is satisfied and noting that the channel is memoryless, using the same approach as theorem 1, it is easy to show that if  $\tilde{\mathbf{c}}^2$  is transmitted, there exists a  $\mathbf{y} = (\mathbf{y}_1 \dots \mathbf{y}_L)$ ,  $\mathbf{y}_j \in \mathcal{O}$ ,  $j = 1, \dots, L$  occurring with probability  $p(\mathbf{y}_1 \dots \mathbf{y}_L | \tilde{\mathbf{c}}^2) = p(\mathbf{x}_1 \dots \mathbf{x}_L | \tilde{\mathbf{c}}^1)$ , and resulting in a realization of random variable  $LLR_{\tilde{\mathbf{c}}^2}(k)$  with a value of,

$$LLR_{\tilde{\mathbf{c}}^2}(k) = \log \frac{\sum_{\mathbf{c}^i \in C_k^1} p(\mathbf{y}_1 \dots \mathbf{y}_L | \mathbf{c}^i)}{\sum_{\mathbf{c}^i \in C_k^0} p(\mathbf{y}_1 \dots \mathbf{y}_L | \mathbf{c}^i)} = \log \frac{\sum_{\mathbf{c}^i \in C_k^1} \prod_{j=1}^L p(\mathbf{y}_j | \mathbf{I}_j^i)}{\sum_{\mathbf{c}^i \in C_k^0} \prod_{j=1}^L p(\mathbf{y}_j | \mathbf{I}_j^i)}. \quad (2.21)$$

Using condition (2.7), we conclude that (2.20) and (2.21) are only different in their signs. This means for each realization of the random variable  $LLR_{\tilde{\mathbf{c}}^1}(k)$ , there exists a realization of the random variable  $LLR_{\tilde{\mathbf{c}}^2}(k)$  with the same magnitude

and different sign which occurs with the same probability, i.e.,  $p(\mathbf{y}_1 \dots \mathbf{y}_L | \tilde{\mathbf{c}}^2) = p(\mathbf{x}_1 \dots \mathbf{x}_L | \tilde{\mathbf{c}}^1)$ . This completes the proof that the *pdf* of random variables  $LLR_{\tilde{\mathbf{c}}^1}(k)$  and  $LLR_{\tilde{\mathbf{c}}^2}(k)$  are the reflections of one another with respect to the vertical axis.

□

Note that for the above two theorems, it is not necessary to partition the code-words into blocks of equal length. In other words, channels with different number of inputs can be used in subsequent block transmissions. The only condition is that the channels in different transmissions should be independent of each other.

In the sequel, conditions for two bit positions to have the same *pdf* for their bit *LLR* are presented for a memoryless channel with binary input. Note that unlike in the previous two theorems, here, it is required that the channel remains the same in subsequent transmissions.

Let  $\mathcal{C}$  be a binary linear code of length  $N$ . Define a permutation  $\mathcal{P}$  which permutes the elements of each code-word. The set of permutations which map code  $\mathcal{C}$  onto itself forms a group called Auto-morphism group of code  $\mathcal{C}$ .

**Theorem 3** *Consider two bit positions of a code-word,  $a, b$  such that  $1 \leq a, b \leq N$ ,  $a \neq b$ . The channel model is assumed to be memoryless and time invariant with binary input. If there exists a permutation  $\mathcal{P}$  within Auto-morphism group of code  $\mathcal{C}$  which transfers bit position  $a$  to  $b$ ,*

$$f_{\tilde{c}_a}(y) = f_{\tilde{c}_b} [(-1)^{\tilde{c}_a \oplus \tilde{c}_b} y], \quad (2.22)$$

where  $f_{\tilde{c}_j}(y)$ ,  $j = 1, \dots, N$ , denotes the *pdf* of random variable  $Y$  corresponding to  $LLR_{\tilde{\mathbf{c}}}(j)$ , assuming code-word  $\tilde{\mathbf{c}}$  is transmitted.

*Proof:* From theorem 1, *pdf* of the bit *LLR* is independent of the transmitted code-word. For simplicity, let us consider the situation of sending the all-zero code-word bit by bit and receiving  $\mathbf{x}_j$  for bit  $\tilde{\mathbf{I}}_j$  in the  $j^{\text{th}}$  transmission. This results in a realization of random variable  $LLR_{\tilde{\mathbf{c}}=\mathbf{0}}(a)$  with a value of,

$$LLR_{\tilde{\mathbf{c}}=\mathbf{0}}(a) = \log \frac{\sum_{\mathbf{c}^i \in C_a^1} \prod_{j=1}^N p(\mathbf{x}_j | \mathbf{I}_j^i)}{\sum_{\mathbf{c}^i \in C_a^0} \prod_{j=1}^N p(\mathbf{x}_j | \mathbf{I}_j^i)}. \quad (2.23)$$

Note that in this theorem  $m = 1$ , which means  $I_j^i$  are scalars. Permutation  $\mathcal{P}$  acts on each code-word  $\mathbf{c}^i$  as follows,

$$\mathcal{P} : C_a^0 \longrightarrow C_b^0, \quad (2.24)$$

$$\mathcal{P} : C_a^1 \longrightarrow C_b^1. \quad (2.25)$$

In memoryless time invariant channels, for each  $(\mathbf{x}_1 \dots \mathbf{x}_L)$  there exists a  $(\mathbf{y}_1 \dots \mathbf{y}_L)$  with the conditional probability  $P(\mathbf{y}_1 \dots \mathbf{y}_L | \mathcal{P}(\tilde{\mathbf{c}})) = P(\mathbf{x}_1 \dots \mathbf{x}_L | \tilde{\mathbf{c}})$ , where  $\mathcal{P}(\tilde{\mathbf{c}})$  is the code-word obtained by applying permutation  $\mathcal{P}$  to  $\tilde{\mathbf{c}}$  and vector  $\mathbf{y}$  is obtained by applying permutation  $\mathcal{P}$  to elements of vector  $\mathbf{x}$ . Noting this fact and applying the permutation  $\mathcal{P}$  to the terms of summations in (2.23), reveals the one to one correspondence between terms within the summations in  $LLR_{\tilde{\mathbf{c}}=\mathbf{0}}(a)$  and  $LLR_{\tilde{\mathbf{c}}=\mathbf{0}}(b)$  as seen in (2.23) and (2.26),

$$LLR_{\tilde{\mathbf{c}}=\mathbf{0}}(b) = \log \frac{\sum_{\mathbf{c}^i \in C_b^1} \prod_{j=1}^N p(\mathbf{y}_j | \mathbf{I}_j^i)}{\sum_{\mathbf{c}^i \in C_b^0} \prod_{j=1}^N p(\mathbf{y}_j | \mathbf{I}_j^i)}. \quad (2.26)$$

The rest of the proof follows similar to the proof of theorem 1. This means for  $\tilde{c}_a = \tilde{c}_b$ , we have  $f_{\tilde{c}_a}(y) = f_{\tilde{c}_b}(y)$ . Using theorem 2, for the case of  $\tilde{c}_a \neq \tilde{c}_b$ , it easily follows that  $f_{\tilde{c}_a}(y) = f_{\tilde{c}_b}(-y)$ , which completes the proof.  $\square$

*Example: Cyclic Codes*

This result can be applied to the class of Cyclic codes as a good example for checking the existence of the desired permutation. Transferring bit position  $a$  to  $b$  ( $a \leq b$ ) in a Cyclic code is achievable by cyclic shifting elements of the code-words  $b - a$  times to the right. It is the property of Cyclic codes that any such shift results in another code-word. Hence, this permutation in Auto-morphism group of the code  $\mathcal{C}$  exists for the case of Cyclic codes.

## 2.4 Summary

In this chapter, the probabilistic behavior of the bit  $LLR$  has been investigated over a general channel model with discrete input and discrete or continuous output. It is proved that under certain symmetry conditions on the channel, the  $pdf$  of the bit  $LLR$  for a specific bit position is independent of the transmitted code-word, if the value of that bit position remains unchanged. It is also shown that a change in the value of a bit position makes the  $pdf$  of that bit  $LLR$  flip horizontally with respect to the vertical axis. Finally, a sufficient condition for two bit positions to have the same  $pdf$  for their bit  $LLR$  is presented.

# Chapter 3

## Performance Evaluation of Binary Linear Block Codes

### 3.1 Chapter Overview

This chapter is organized as follows. In Section 3.2, the model used to analyze the problem is presented. All notations and assumptions are in this section. Computing the *pdf* of bit *LLRs* using the Gram-Charlier expansion is presented in Section 3.3. This is an orthogonal series expansion of a given *pdf* which requires knowledge of the moments of the corresponding random variable. An analytical method for computing the moments of the bit *LLR* using Taylor expansion is proposed in Section 3.4, where it is shown that one can compute the coefficients of Taylor expansion of the bit *LLR* recursively. We also present a closed form expression for computing the bit error probability in Section 3.5. In Section 3.6, the convergence issue of this approximation is discussed. Numerical results are pro-

vided in Section 3.7 which demonstrate a close agreement between our analytical method and simulation. We summarize this chapter in Section 3.8.

## 3.2 Modeling

Using the theorems proved in Chapter 2 for the special case of AWGN channels with BPSK modulation, simplifies the analysis.

The modulation scheme used here is BPSK which is defined as the mapping  $M$ ,

$$M : \mathbf{c} \longrightarrow \mathbf{m}(\mathbf{c}), \quad (3.1)$$

$$0 \longrightarrow m(0) = -1, \quad 1 \longrightarrow m(1) = 1. \quad (3.2)$$

Note that modulating a code-word as mentioned above results in a vector of constant square norm,

$$\forall \mathbf{c} \in \mathcal{C} : \quad \|\mathbf{m}(\mathbf{c})\|^2 = \mathbf{m}(\mathbf{c}) \cdot \mathbf{m}(\mathbf{c}) = \sum_{l=1}^N m^2(c_l) = N. \quad (3.3)$$

Notation  $\omega(\mathbf{c})$  denotes the Hamming weight of a code-word  $\mathbf{c}$ , which is equal to the number of ones in  $\mathbf{c}$ . It follows,

$$-\mathbf{1} \cdot \mathbf{m}(\mathbf{c}) = N - 2\omega(\mathbf{c}). \quad (3.4)$$

Modulating a code-word  $\tilde{\mathbf{c}} = (\tilde{c}_1, \tilde{c}_2, \dots, \tilde{c}_N)$  using BPSK and sending it through an AWGN channel,  $\mathbf{x} = \mathbf{m}(\tilde{\mathbf{c}}) + \mathbf{n}$  is received, where  $\mathbf{n} = (n_1, n_2, \dots, n_N)$  is an independent, identically distributed Gaussian noise vector which has zero mean

elements of variance  $\sigma^2$ . Note that for an AWGN channel, we have,

$$p(\mathbf{x}|\tilde{\mathbf{c}}) = \frac{1}{(\sqrt{2\pi}\sigma)^N} \exp \left[ -\frac{\|\mathbf{x} - \mathbf{m}(\tilde{\mathbf{c}})\|^2}{2\sigma^2} \right]. \quad (3.5)$$

The *LLR* of the  $k^{\text{th}}$  bit position is defined by the following equation,

$$LLR(k) = \log \frac{P(\tilde{c}_k = 1|\mathbf{x})}{P(\tilde{c}_k = 0|\mathbf{x})}, \quad (3.6)$$

where  $\tilde{c}_k$  is the value of the  $k^{\text{th}}$  bit in the transmitted code-word and  $\log$  stands for the natural logarithm. Assuming,

$$P(\tilde{c}_k = 0) = P(\tilde{c}_k = 1) = \frac{1}{2}, \quad (3.7)$$

and using (3.5), it follows,

$$LLR(k) = \log \frac{p(\mathbf{x}|\tilde{c}_k = 1)}{p(\mathbf{x}|\tilde{c}_k = 0)} \quad (3.8)$$

$$= \log \frac{\sum_{\mathbf{c} \in C_k^1} \exp \left[ -\frac{\|\mathbf{x} - \mathbf{m}(\mathbf{c})\|^2}{2\sigma^2} \right]}{\sum_{\mathbf{c} \in C_k^0} \exp \left[ -\frac{\|\mathbf{x} - \mathbf{m}(\mathbf{c})\|^2}{2\sigma^2} \right]}. \quad (3.9)$$

Using (3.3), it follows,

$$LLR(k) = \log \frac{\sum_{\mathbf{c} \in C_k^1} \exp \left[ \frac{\mathbf{x} \cdot \mathbf{m}(\mathbf{c})}{\sigma^2} \right]}{\sum_{\mathbf{c} \in C_k^0} \exp \left[ \frac{\mathbf{x} \cdot \mathbf{m}(\mathbf{c})}{\sigma^2} \right]} \quad (3.10)$$

$$= \log \frac{\sum_{\mathbf{c} \in C_k^1} \exp \left[ \frac{\mathbf{n} \cdot \mathbf{m}(\mathbf{c}) + \mathbf{m}(\tilde{\mathbf{c}}) \cdot \mathbf{m}(\mathbf{c})}{\sigma^2} \right]}{\sum_{\mathbf{c} \in C_k^0} \exp \left[ \frac{\mathbf{n} \cdot \mathbf{m}(\mathbf{c}) + \mathbf{m}(\tilde{\mathbf{c}}) \cdot \mathbf{m}(\mathbf{c})}{\sigma^2} \right]}. \quad (3.11)$$

Given a value of the bit  $LLR$ , a decision on the value of bit  $k$  is made by comparing  $LLR(k)$  with a threshold of zero.

Using Theorems 1 and 2, without loss of generality, assume for convenience that the all-zero code-word, denoted as  $\tilde{\mathbf{c}} = (0, 0, \dots, 0)$ , is transmitted in all our following discussions. This means  $\mathbf{m}(\tilde{\mathbf{c}}) = -\mathbf{1} = (-1, -1, \dots, -1)$  is the transmitted modulated code-word.

In this case, (3.11) reduces to,

$$LLR(k) = \log \frac{\sum_{\mathbf{c} \in C_k^1} \exp \left[ \frac{\mathbf{n} \cdot \mathbf{m}(\mathbf{c}) - \mathbf{1} \cdot \mathbf{m}(\mathbf{c})}{\sigma^2} \right]}{\sum_{\mathbf{c} \in C_k^0} \exp \left[ \frac{\mathbf{n} \cdot \mathbf{m}(\mathbf{c}) - \mathbf{1} \cdot \mathbf{m}(\mathbf{c})}{\sigma^2} \right]}. \quad (3.12)$$

Using (3.4), we obtain,

$$LLR(k) = \log \frac{\sum_{\mathbf{c} \in C_k^1} \exp \left[ \frac{\mathbf{n} \cdot \mathbf{m}(\mathbf{c}) - 2\omega(\mathbf{c})}{\sigma^2} \right]}{\sum_{\mathbf{c} \in C_k^0} \exp \left[ \frac{\mathbf{n} \cdot \mathbf{m}(\mathbf{c}) - 2\omega(\mathbf{c})}{\sigma^2} \right]}. \quad (3.13)$$

In the following, for convenience of notation, the index  $k$  indicating bit position is dropped. This means the sets  $C^1$  and  $C^0$  are indeed  $C_k^1$  and  $C_k^0$ , respectively. We use the notation  $H(\mathbf{n})$  to refer to the  $LLR$  expression given in (3.13), i.e.,

$$H(\mathbf{n}) = \log \frac{\sum_{\mathbf{c} \in C^1} \exp \left[ \frac{\mathbf{n} \cdot \mathbf{m}(\mathbf{c}) - 2\omega(\mathbf{c})}{\sigma^2} \right]}{\sum_{\mathbf{c} \in C^0} \exp \left[ \frac{\mathbf{n} \cdot \mathbf{m}(\mathbf{c}) - 2\omega(\mathbf{c})}{\sigma^2} \right]}. \quad (3.14)$$

### 3.3 Gram-Charlier Expansion of *pdf*

One common method for representing a function is to use an expansion on an orthogonal basis which is suitable for that function. As the *pdf* of a bit *LLR* is approximately Gaussian [7,42,26], the appropriate basis can be a normal Gaussian *pdf* and its derivatives which form an orthogonal basis. There are a variety of equivalent formulations for this expansion [18,48,49,50]. We follow the notation used in [18].

Consider a random variable  $Y$ , which is normalized to have zero mean and unit variance. One can expand the *pdf* of  $Y$ , namely  $f_Y(y)$ , using the following formula which is called the Gram-Charlier series expansion,

$$f_Y(y) \simeq \frac{1}{\sqrt{2\pi}} e^{-\frac{y^2}{2}} \sum_{i=0}^{\infty} \alpha_i T_i(y), \quad (3.15)$$

where,  $T_i(y)$  is the Hermite polynomial [18] of order  $i$ , defined as,

$$T_i(y) = (-1)^i e^{\frac{y^2}{2}} \frac{d^i}{dy^i} [e^{-\frac{y^2}{2}}], \quad (3.16)$$

and has the following closed form,

$$T_i(y) = \sum_{j=0}^{\lfloor i/2 \rfloor} \frac{(-1)^j i!}{2^j (i-2j)! j!} y^{i-2j}, \quad (3.17)$$

and,

$$\alpha_i = \sum_{j=0}^{\lfloor i/2 \rfloor} \frac{(-1)^j}{2^j (i-2j)! j!} \mu_{i-2j}, \quad (3.18)$$

where,

$$\mu_j = \int_{-\infty}^{+\infty} y^j f_Y(y) dy. \quad (3.19)$$

This is a commonly used method for approximating an unknown *pdf*. The only unknown components in (3.18) are the moments,  $\mu_j$ . We propose an analytical method using Taylor series expansion to compute the moments of the bit *LLR* in the next section.

### 3.4 Computing Moments Using Taylor Expansion of *LLR*

Applying the definition of the  $m^{\text{th}}$  order ( $m > 2$ ) moment to bit *LLR* results in,

$$\mu_m = E \left[ \left( \frac{H(\mathbf{n}) - E[H(\mathbf{n})]}{\sqrt{\text{var}[H(\mathbf{n})]}} \right)^m \right] \quad (3.20)$$

$$= \frac{1}{\text{var}^{m/2}[H(\mathbf{n})]} \sum_{i=0}^m (-1)^i \binom{m}{i} E[H^{m-i}(\mathbf{n})] E^i[H(\mathbf{n})], \quad (3.21)$$

where  $E[\cdot]$  stands for expectation and  $\text{var}[\cdot]$  denotes variance. Note that to compute (3.21), one needs  $E[H^j(\mathbf{n})]$ ,  $j = 0, \dots, m$ .

$E[H^j(\mathbf{n})]$  can be computed using a method similar to the so called Delta method [51] and find the average of the Taylor series expansion of  $H^j(\mathbf{n})$ . We use the Taylor series expansion of  $H(\mathbf{n})$  in conjunction with the polynomial theorem [18] to find an expansion for  $H^j(\mathbf{n})$ ,

$$H^j(\mathbf{n}) = \left( \sum_{i=0}^{\infty} \frac{1}{i!} (\mathbf{n} \cdot \nabla)^i H(\mathbf{0}) \right)^j, \quad (3.22)$$

where  $\nabla H(\mathbf{0})$  is the gradient of  $H(\mathbf{n})$  at  $\mathbf{n} = \mathbf{0}$ . An alternative approach is to directly expand  $H^j(\mathbf{n})$ . Note that derivatives of  $H^j(\mathbf{n})$  are functions of derivatives of  $H(\mathbf{n})$ .

The Taylor series expansion of  $H(\mathbf{n})$  around vector zero,  $\mathbf{0} = (0, 0, \dots, 0)$ , is formulated using the expression below in terms of  $\mathbf{n}$ ,

$$H(\mathbf{n}) = H(\mathbf{0}) + \mathbf{n} \cdot \nabla H(\mathbf{0}) + \frac{1}{2!} (\mathbf{n} \cdot \nabla)^2 H(\mathbf{0}) + \dots \quad (3.23)$$

$$= H(\mathbf{0}) + \sum_{q_1=1}^N \left. \frac{\partial H(\mathbf{n})}{\partial n_{q_1}} \right|_{\mathbf{n}=\mathbf{0}} n_{q_1} + \frac{1}{2} \sum_{q_1=1}^N \sum_{q_2=1}^N \left. \frac{\partial^2 H(\mathbf{n})}{\partial n_{q_1} \partial n_{q_2}} \right|_{\mathbf{n}=\mathbf{0}} n_{q_1} n_{q_2} + \dots \quad (3.24)$$

We can continue with calculation of different terms in the above equation. For simplicity, define (3.14) as  $H(\mathbf{n}) = \log A^1(\mathbf{n}) - \log A^0(\mathbf{n})$ , where,

$$A^1(\mathbf{n}) = \sum_{\mathbf{c} \in C^1} \exp \left[ \frac{\mathbf{n} \cdot \mathbf{m}(\mathbf{c}) - 2\omega(\mathbf{c})}{\sigma^2} \right], \quad (3.25)$$

and  $A^0(\mathbf{n})$  has a similar formula. We only consider  $\log A^1(\mathbf{n})$  hereafter in this section. The same approach can be used for  $\log A^0(\mathbf{n})$ . For simplicity of notation, use  $A(\mathbf{n})$  instead of  $A^1(\mathbf{n})$ .

$$\log A(\mathbf{n}) = \log A(\mathbf{0}) + \sum_{q_1=1}^N \left. \frac{\partial \log A(\mathbf{n})}{\partial n_{q_1}} \right|_{\mathbf{n}=\mathbf{0}} n_{q_1} + \frac{1}{2} \sum_{q_1=1}^N \sum_{q_2=1}^N \left. \frac{\partial^2 \log A(\mathbf{0})}{\partial n_{q_1} \partial n_{q_2}} \right|_{\mathbf{n}=\mathbf{0}} n_{q_1} n_{q_2} + \dots \quad (3.26)$$

To simplify the subsequent derivations, the following functions are defined,

$$F_{\{q_1, \dots, q_j\}}(\mathbf{n}) = \frac{\partial^j A(\mathbf{n})}{\partial n_{q_1} \partial n_{q_2} \dots \partial n_{q_j}} = \sigma^{-2j} \sum_{\mathbf{c} \in C^1} M_{\{q_1, \dots, q_j\}} \exp \left[ \frac{\mathbf{n} \cdot \mathbf{m}(\mathbf{c}) - 2\omega(\mathbf{c})}{\sigma^2} \right], \quad j \geq 1, \quad (3.27)$$

where  $\{q_1, \dots, q_j\}$  is a set which contains  $j$  bit positions different from  $k$ , and,

$$M_{\{q_1, \dots, q_j\}} = \prod_{l=1}^j m(c_{q_l}), \quad j \geq 1, \quad (3.28)$$

where  $m(c_{q_l}) = \pm 1$  is the modulated value for the  $q_l^{\text{th}}$ ,  $q_l \in \{q_1, \dots, q_j\}$ , bit of code-word  $\mathbf{c}$ . It is clear that  $M_{\{q_1, \dots, q_j\}} = \pm 1$  as well.

To simplify (3.26), it easily follows that,

$$\frac{\partial \log A(\mathbf{n})}{\partial n_{q_1}} = A^{-1}(\mathbf{n}) F_{\{q_1\}}(\mathbf{n}) = R_{\{q_1\}}(\mathbf{n}), \quad (3.29)$$

where  $R_{\{q_1, \dots, q_j\}}(\mathbf{n})$  defined as,

$$R_{\{q_1, \dots, q_j\}}(\mathbf{n}) = A^{-1}(\mathbf{n}) F_{\{q_1, \dots, q_j\}}(\mathbf{n}), \quad j \geq 1, \quad (3.30)$$

where  $A(\mathbf{n})$  and  $F_{\{q_1, \dots, q_j\}}(\mathbf{n})$  are given in (3.25) and (3.27), respectively.

The functions  $A(\mathbf{n})$ ,  $F_{\{q_1, \dots, q_j\}}(\mathbf{n})$ , and  $R_{\{q_1, \dots, q_j\}}(\mathbf{n})$  defined in (3.25), (3.27), and (3.30) reduce to special weight distribution functions when  $\mathbf{n} = \mathbf{0}$ ,

$$A(\mathbf{0}) = \mathcal{A}(Z) = \sum_{w=0}^N a(w) Z^w, \quad (3.31)$$

where  $Z = \exp(-\frac{2}{\sigma^2})$  and  $a(w)$  is the number of code-words with Hamming weight  $w$  in  $C^1$ .

$$F_{\{q_1, \dots, q_j\}}(\mathbf{0}) = \mathcal{F}_{\{q_1, \dots, q_j\}}(Z) = \sigma^{-2j} \sum_{w=0}^N [f_{\{q_1, \dots, q_j\}}^+(w) - f_{\{q_1, \dots, q_j\}}^-(w)] Z^w, \quad j \geq 1, \quad (3.32)$$

where  $f_{\{q_1, \dots, q_j\}}^\pm(w)$ , is the number of code-words  $\mathbf{c} \in C^1$  with Hamming weight  $w$  and  $M_{\{q_1, \dots, q_j\}} = \pm 1$ .

$$R_{\{q_1, \dots, q_j\}}(\mathbf{0}) = \mathcal{R}_{\{q_1, \dots, q_j\}}(Z) = \mathcal{A}^{-1}(Z) \mathcal{F}_{\{q_1, \dots, q_j\}}(Z), \quad j \geq 1. \quad (3.33)$$

We can compute  $F_{\{q_1, \dots, q_j\}}(\mathbf{0})$ , using the trellis diagram of the code. This is achieved by constructing a new trellis diagram and augmenting each state into two states

according to the values of  $M_{\{q_1, \dots, q_{j_0}\}}$  where  $j_0 = 1, \dots, j$ . The complexity of computing  $F_{\{q_1, \dots, q_j\}}(\mathbf{0})$  exponentially grows with block length of the code, which is a limiting factor in applying this method to long codes. However, by partitioning the code into different shells according to weights of the codewords and considering the first few shells, this complexity can be significantly reduced, while at the same time it is accurate enough in high SNR regions. It can be shown that this is a better approximation than the famous union bound, which only considers the first shell. The other approach to solve the problem of long codes is to use the method described in Chapter 4.

Using (3.29) and (3.33), we have,

$$\left. \frac{\partial \log A(\mathbf{n})}{\partial n_{q_1}} \right|_{\mathbf{n}=\mathbf{0}} = \mathcal{R}_{\{q_1\}}(Z). \quad (3.34)$$

Replacing (3.29) and (3.34) in (3.26), we have,

$$\log A(\mathbf{n}) = \log \mathcal{A}(Z) + \sum_{q_1=1}^N \mathcal{R}_{\{q_1\}}(Z) n_{q_1} + \frac{1}{2} \sum_{q_1=1}^N \sum_{q_2=1}^N \left. \frac{\partial R_{\{q_1\}}(\mathbf{n})}{\partial n_{q_2}} \right|_{\mathbf{n}=\mathbf{0}} n_{q_1} n_{q_2} + \dots \quad (3.35)$$

To compute (3.35), one needs derivatives of  $R_{\{q_1\}}(\mathbf{n})$ , which can be calculated using the following theorem.

**Theorem 4** *For any  $q_i$  representing a bit position other than  $k$ , we have,*

$$\frac{\partial R_{\{q_1, \dots, q_j\}}(\mathbf{n})}{\partial n_{q_i}} = \begin{cases} \sigma^{-4} R_{\{q_1, \dots, q_{i-1}, q_{i+1}, \dots, q_j\}}(\mathbf{n}) - R_{\{q_1, \dots, q_j\}}(\mathbf{n}) R_{\{q_i\}}(\mathbf{n}), & \text{If } q_i \in \{q_1, \dots, q_j\}, \\ R_{\{q_1, \dots, q_j, q_i\}}(\mathbf{n}) - R_{\{q_1, \dots, q_j\}}(\mathbf{n}) R_{\{q_i\}}(\mathbf{n}), & \text{Otherwise.} \end{cases} \quad (3.36)$$

*Proof:* For proof refer to Appendix A.  $\square$

Another theorem which simplifies the calculation of even order derivatives, is presented next.

**Theorem 5** *We have,*

$$\frac{\partial^2 R_{\{q_1, \dots, q_j\}}(\mathbf{n})}{\partial n_{q_i}^2} = -2R_{\{q_i\}}(\mathbf{n}) \frac{\partial R_{\{q_1, \dots, q_j\}}(\mathbf{n})}{\partial n_{q_i}}. \quad (3.37)$$

*Proof:* For proof refer to Appendix A.  $\square$

Referring to (3.36), one can easily see that the coefficients of the expansion (3.35) are polynomials of  $R_{\{q_1, \dots, q_j\}}(\mathbf{0})$  for different values of  $j$ . It is noteworthy that these coefficients are polynomials of special weight distribution functions defined in (3.33). The above theorems and results enable us to compute all the derivatives required in the Taylor series expansion of  $H(\mathbf{n}) = \log A^1(\mathbf{n}) - \log A^0(\mathbf{n})$ .

### 3.5 Computing Probability of Error

The bit error performance follows by a simple integration of the resulting *pdf*. We present a closed form formula for computing this integral in this section.

Using Theorem 2, we have,

$$P(e|\tilde{c}_k = 0) = P(e|\tilde{c}_k = 1), \quad (3.38)$$

where event  $e$  corresponds to bit  $k$  being in error. Using assumption (2.3), one can write,

$$P(e) = P(e|\tilde{c}_k = 0)P(\tilde{c}_k = 0) + P(e|\tilde{c}_k = 1)P(\tilde{c}_k = 1) = P(e|\tilde{c}_k = 0). \quad (3.39)$$

Hence, computation of the bit error probability involves calculating an integral of the following form,

$$P(e) = \int_a^{\infty} f_Y(y) dy, \quad (3.40)$$

where  $y$  is the bit *LLR* normalized to have zero mean and unit variance and  $a = -E[y]/\sigma_y$ . Substituting  $f_Y(y)$  with its Gram-Charlier expansion results in,

$$P(e) \simeq \int_a^{\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{y^2}{2}} \sum_{i=0}^{\infty} \alpha_i T_i(y) dy. \quad (3.41)$$

Noting that  $\alpha_0 = 1$ ,  $T_0(y) = 1$ , we have,

$$P(e) \simeq \int_a^{\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{y^2}{2}} dy + \int_a^{\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{y^2}{2}} \sum_{i=1}^{\infty} \alpha_i T_i(y) dy \quad (3.42)$$

$$= Q(a) + \int_a^{\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{y^2}{2}} \sum_{i=1}^{\infty} \alpha_i T_i(y) dy. \quad (3.43)$$

Changing the order of integration and summation and using the following property <sup>1</sup>,

$$e^{-\frac{y^2}{2}} T_i(y) = -\frac{d}{dy} \left[ e^{-\frac{y^2}{2}} T_{i-1}(y) \right], \quad i \geq 1, \quad (3.44)$$

one can write,

$$P(e) \simeq Q(a) - \frac{1}{\sqrt{2\pi}} \sum_{i=1}^{\infty} \alpha_i \int_a^{\infty} d \left[ e^{-\frac{y^2}{2}} T_{i-1}(y) \right] \quad (3.45)$$

$$= Q(a) - \frac{1}{\sqrt{2\pi}} \sum_{i=1}^{\infty} \alpha_i \left[ e^{-\frac{y^2}{2}} T_{i-1}(y) \right]_a^{\infty} \quad (3.46)$$

$$= Q(a) + \frac{1}{\sqrt{2\pi}} e^{-\frac{a^2}{2}} \sum_{i=1}^{\infty} \alpha_i T_{i-1}(a). \quad (3.47)$$

This results in a closed form expression for computing probability of error.

---

<sup>1</sup>Proof is in Appendix A.3.

### 3.6 Convergence properties

Convergence properties of the Gram-Charlier expansion is investigated in [48, 52, 53]. It is proved in [54], that the expansion is convergent if the expanded function satisfies the following condition,

$$\int_{-\infty}^{+\infty} f_Y(y) e^{y^2/3} dy < \infty. \quad (3.48)$$

Reference [16], mentions that this expansion has good asymptotic behavior as defined in [55]. In other words, a few terms will give a close approximation.

General properties of Hermite polynomials are discussed in [56], where it is shown that this class of polynomials form an orthogonal basis which span the interval  $(-\infty, +\infty)$ . Therefore, the *pdf* of the bit *LLR* can be expanded arbitrarily closely, in mean square sense, using the given set of orthogonal basis. i.e.,

$$\lim_{l \rightarrow \infty} \int_{-\infty}^{+\infty} \epsilon_l^2(y) dy \rightarrow 0, \quad (3.49)$$

where  $\epsilon_l(y)$  is truncation error defined as,

$$\epsilon_l(y) = f_Y(y) - \frac{1}{\sqrt{2\pi}} e^{-y^2/2} \sum_{i=0}^l \alpha_i T_i(y). \quad (3.50)$$

If  $f_Y(y)$  is piecewise continuous in the interval  $(-\infty, +\infty)$ , the result of this expansion converges to  $f_Y(y)$  at each point of  $(-\infty, +\infty)$  at which  $f_Y(y)$  is continuous. At points where  $f_Y(y)$  has a jump discontinuity, this series converges to  $(f_Y(y^+) + f_Y(y^-))/2$  [57]. In the following, it is shown that the error in the computation of bit error probability converges to zero.

In practice, computation of error probability is performed by integrating  $f_Y(y)$  from  $a$  to  $b$  instead of  $a$  to  $\infty$ , where  $a = -E[y]/\sigma_y$  and  $b$  is a large finite value.

Using the Cauchy-Schwartz inequality [58],

$$\left| \int_{-\infty}^{+\infty} f(y)g(y)dy \right|^2 < \int_{-\infty}^{+\infty} |f(y)|^2 dy \int_{-\infty}^{+\infty} |g(y)|^2 dy, \quad (3.51)$$

for the case of  $f(y) = \epsilon_l(y)$  and,

$$g(y) = \begin{cases} 1, & a < y < b, \\ 0, & \text{Otherwise,} \end{cases} \quad (3.52)$$

we have,

$$\left| \int_a^b \epsilon_l(y)dy \right|^2 < (b-a) \int_{-\infty}^{+\infty} \epsilon_l^2(y)dy \quad (3.53)$$

Applying (3.49) to (3.53), results in,

$$\lim_{l \rightarrow \infty} \int_a^b \epsilon_l(y)dy \rightarrow 0. \quad (3.54)$$

In this case, one can get as small as the desired error,  $\epsilon_l(y)$ , in computation of the error probability by increasing the number of terms,  $l$ .

### 3.7 Numerical Results

In this section, some examples are provided which show a close agreement between the analytical method and simulation results.

As an example, a (15,11,3) Cyclic code is used and evaluated its performance using the proposed method. The order of the Gram-Charlier expansion is 10. The comparison between the analytically calculated BER and the one obtained from simulation is shown in Figure 3.1. It is shown that in the case of Cyclic codes, the computed *pdf* is not affected by the choice of the bit position.

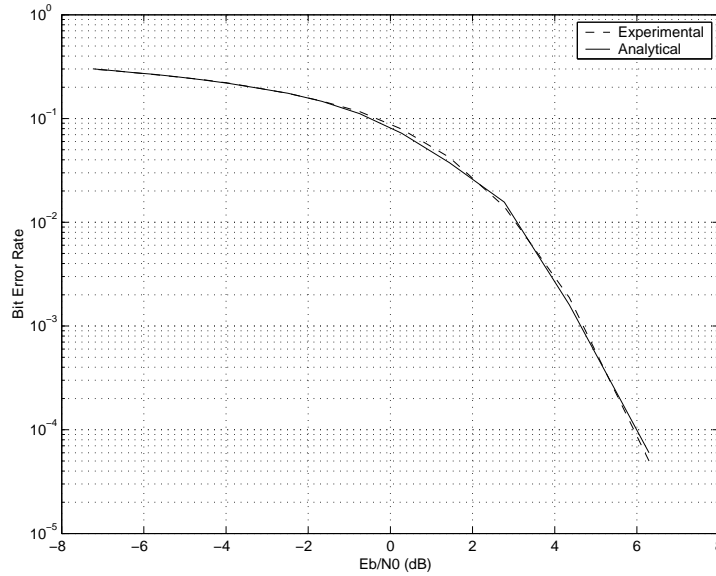


Figure 3.1: BER for (15,11,3) Cyclic code.

Another example is a (12,11,2) single parity check code. The order of the Gram-Charlier expansion is 10. The comparison between the analytically calculated BER and the one obtained from simulation is shown in Figure 3.2.

The last example is the binary extended (24,12,8) Golay code. Its performance is shown in Figure 3.3. The bit error rate is calculated using Gram-Charlier series with 14 terms.

There is not any known method in the literature to calculate the truncation error of the Gram-Charlier series. It is an open problem to determine where to truncate the series to get a good approximation.

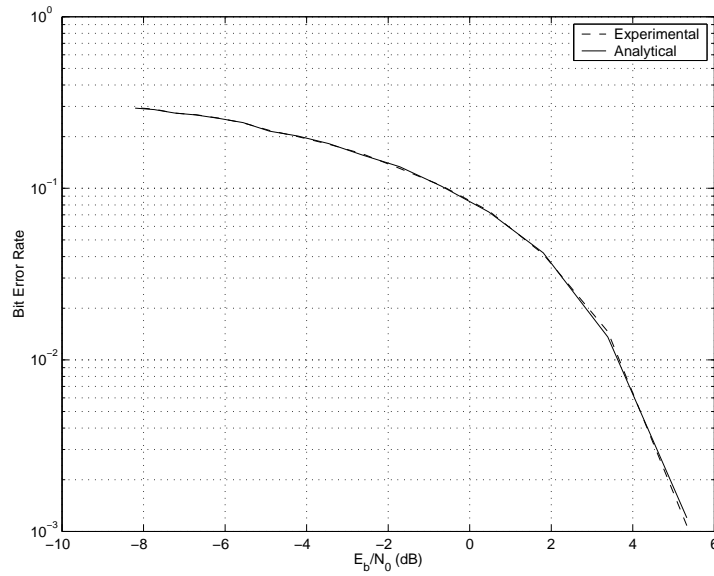


Figure 3.2: BER for (12,11,2) single parity check code.

### 3.8 Summary

A method is presented for calculating bit error probability of binary linear block codes over AWGN channel, using special weight enumerating functions of the code. A summary of the proposed method is presented here. Starting with calculation of special weight distribution functions defined in (3.33), proceed with a Taylor series of the  $LLR$  as indicated in (3.23). Averaging this expansion will give us moments of the  $pdf$  of the bit  $LLR$ , which can be used to compute the coefficients of the Gram-Charlier series using (3.18). A closed form expression (3.47) can be used to find the bit error probability. All these steps can be seen in Figure 3.4.

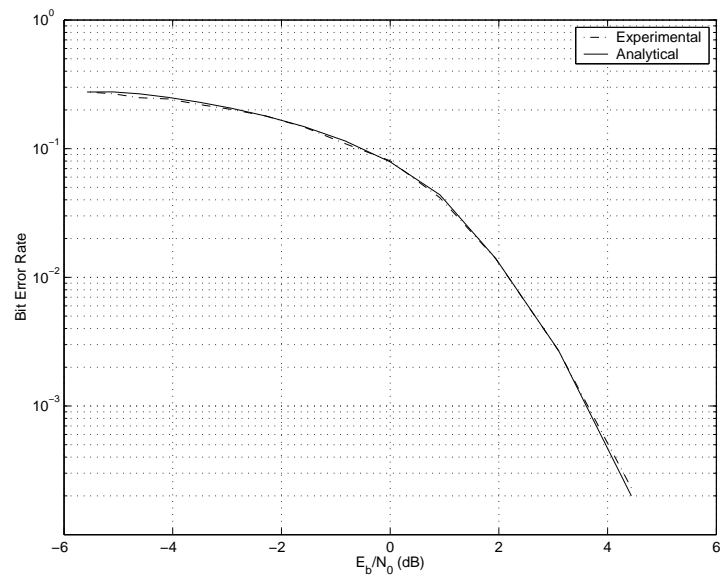


Figure 3.3: BER for binary extended (24,12,8) Golay code.

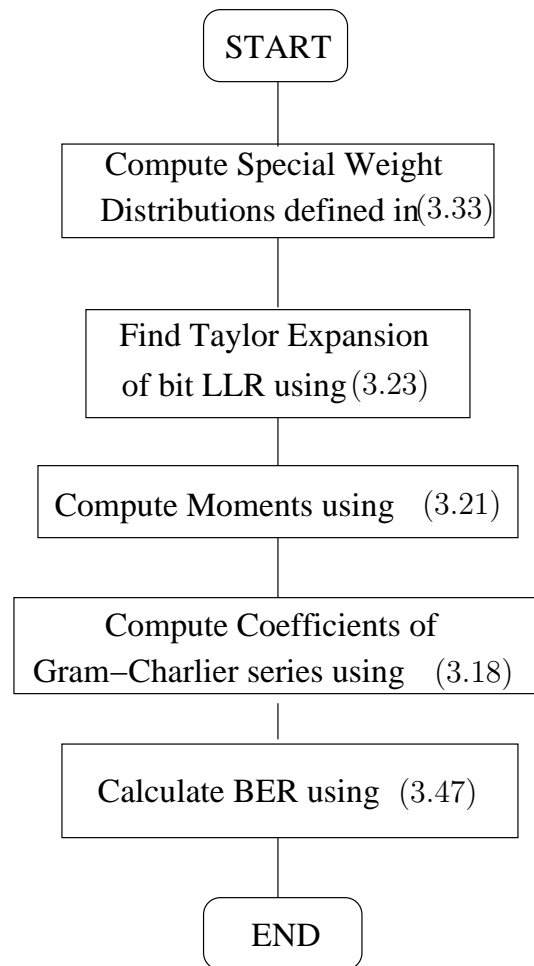


Figure 3.4: Flow chart of the proposed method.

# Chapter 4

## Performance Evaluation of Turbo-Like Codes

### 4.1 Chapter Overview

This chapter is organized as follows. We model the *pdf* of the bit *LLR* by using its symmetry properties in Section 4.2. In Section 4.3, the maximum entropy method to find the parameters of the proposed model is described. A method to compute the Probabilities of the Point Estimates (PPE) for the estimated parameters, as well as the estimated BER, is detailed in Section 4.4. The numerical results and summary are presented in Section 4.5 and Section 4.6, respectively.

## 4.2 Modeling the *pdf* of the Bit *LLR*

The *LLR* of the  $k^{th}$  bit position is defined in (3.6). Let us define the random variable  $Y = LLR(k)$  with its *pdf* denoted as  $f(y)$ . It is proved in [59] that the *pdf* of the bit *LLR* is independent of the transmitted code-word, as long as the value of the bit position under consideration remains unchanged. By using this result and without the loss of generality, consider the case of sending the all-zero code-word. It is proved in [9] that the *pdf* of the bit *LLR* has the following symmetry property:

$$f(y) = e^{-y} f(-y). \quad (4.1)$$

Taking the logarithm from both sides of (4.1), one can write the following:

$$\log f(y) - \log f(-y) = -y. \quad (4.2)$$

Utilizing the power series, it easily follows that

$$\log f(y) = -y/2 + \sum_{i=0}^{\infty} a_i y^{2i}. \quad (4.3)$$

The previous analysis suggests that the following model can be used for the *pdf* of the bit *LLR*:

$$f(y) \simeq \exp(-y/2 + \sum_{i=0}^N a_i y^{2i}). \quad (4.4)$$

The received bit is decoded to 0 (or 1), if the corresponding *LLR* is negative (or positive). Therefore, the following integral simplifies the remaining BER calculations:

$$P_e \simeq \int_0^{\infty} f(y) dy. \quad (4.5)$$

In the next section, the maximum entropy principle is used to find the parameters of the proposed model by using the moments of the bit *LLR*.

### 4.3 Moment Matching Using the Maximum Entropy Principle

There are various methods for parameter estimation. Typically, the unknown parameters of a *pdf* can be found by adopting moment matching, entropy matching, or ML. We use the moment matching method with the maximum entropy principle simply because it is mathematically tractable, and has been successfully implemented in a variety of applications [60]. An attractive feature of the class of the distributions with the maximum entropy is that a simple iterative maximization technique can be employed to compute their parameters. The maximum entropy principle was first introduced by Jaynes [60] in 1982. Since then, it has been widely used in various applications. In this method, the search, while satisfying the constraints on the moments, is limited to the *pdf* with the maximum entropy. For more recent discussions on this method, refer to [61, 62]. We follow an approach that is similar to the one introduced in [23]. The maximum entropy density can be found by maximizing the following with respect to  $\hat{f}(y)$ :

$$\text{Maximize } - \int_{-\infty}^{+\infty} \hat{f}(y) \log[\hat{f}(y)] dy, \quad (4.6)$$

$$\text{Subject to: } \hat{\mu}_i = \mu_i, \quad i = 1, 2, \dots, M, \quad (4.7)$$

with

$$\mu_i = \int_{-\infty}^{+\infty} y^i f(y) dy \quad (4.8)$$

and

$$\hat{\mu}_i = \int_{-\infty}^{+\infty} y^i \hat{f}(y) dy, \quad (4.9)$$

where  $M$  is the number of moments used in the parameter estimation. This maximization problem can be solved with the Lagrange multipliers  $\lambda_k$ ,  $k = 0, 1, \dots, M$  by following the methods of the calculus of variations [63]. Let us define the Lagrangian as

$$\int_{-\infty}^{+\infty} \hat{f}(y) \log[\hat{f}(y)] dy + c \int_{-\infty}^{+\infty} \hat{f}(y) dy + \sum_{k=1}^M \lambda_k \int_{-\infty}^{+\infty} y^k \hat{f}(y) dy. \quad (4.10)$$

Setting the variations of (4.10) with respect to  $\hat{f}(y)$  to zero, we have

$$\log[\hat{f}(y)] + \lambda_0 + \sum_{k=1}^M \lambda_k y^k = 0, \quad (4.11)$$

where  $\lambda_0 = c + 1$ . Solving for  $\hat{f}(y)$  results in

$$\hat{f}(y) = \exp\left(-\sum_{k=0}^M \lambda_k y^k\right). \quad (4.12)$$

From (4.4), it is clear that all of the odd coefficients, except  $\lambda_1$ , are zero. Hence, (4.12) can be reformulated with the new coefficients  $a_k = \lambda_{2k}$ ,  $k = 0, 1, \dots, N = \lfloor \frac{M}{2} \rfloor$ , as follows:

$$\hat{f}(y) = \exp\left(-y/2 - \sum_{k=0}^N a_k y^{2k}\right). \quad (4.13)$$

Normalizing the area under  $\hat{f}(y)$  to one, we write,

$$e^{a_0} = \int_{-\infty}^{+\infty} \exp\left(-y/2 - \sum_{k=1}^N a_k y^{2k}\right) dy. \quad (4.14)$$

If (4.14) is substituted for  $e^{a_0}$  in (4.13), then,

$$\hat{f}(y) = \exp\left\{-y/2 - \sum_{k=1}^N a_k y^{2k} - \log\left[\int_{-\infty}^{+\infty} \exp\left(-z/2 - \sum_{k=1}^N a_k z^{2k}\right) dz\right]\right\}. \quad (4.15)$$

The objective is to estimate the parameters  $a_k$ ,  $k = 0, 1, \dots, N$ , where  $a_0$  can be computed using (4.14). As it is seen later, one can estimate the parameters  $a_k$ ,  $k = 1, \dots, N$  using the first  $N$  moments of the bit *LLR*. In practice, the statistical estimates of the moments are used instead of the true moments<sup>1</sup>. Using (4.9), we have

$$\hat{\mu}_i = \int_{-\infty}^{+\infty} y^i \exp \left\{ -y/2 - \sum_{k=1}^N a_k y^{2k} - \log \left[ \int_{-\infty}^{+\infty} \exp(-z/2 - \sum_{k=1}^N a_k z^{2k}) dz \right] \right\} dy, \quad (4.16)$$

$$i = 1, 2, \dots, N.$$

Setting  $\hat{\mu}_i$  equal to the statistical estimates of the moments, one can find the unknown parameters. Since there is no closed form solution for this problem, let us continue with the numerical methods. The Newton-Raphson method is employed to iteratively solve the following problem:

$$\begin{aligned} G_i(a_1, a_2, \dots, a_N) &= \hat{\mu}_i - \mu_i \\ &= \int_{-\infty}^{+\infty} (y^i - \mu_i) \exp \left\{ -y/2 - \sum_{k=1}^N a_k y^{2k} - \log \left[ \int_{-\infty}^{+\infty} \exp(-z/2 - \sum_{k=1}^N a_k z^{2k}) dz \right] \right\} dy \\ &= 0, \end{aligned} \quad (4.17)$$

$$i = 1, 2, \dots, N.$$

Notation  $\mathbf{a}^{(r)} = \{a_1^{(r)}, a_2^{(r)}, \dots, a_N^{(r)}\}$  is used to denote the answer after  $r$  iterations. In this method, assume that for the small changes  $\Delta \mathbf{a}^{(r)}$  in the  $\mathbf{a}^{(r)}$ , we can

---

<sup>1</sup>The effect of an small error in the moment estimation on the parameter estimates is investigated in Appendix B.1.

write,

$$\mathbf{a}^{(r+1)} = \mathbf{a}^{(r)} + \Delta\mathbf{a}^{(r)}. \quad (4.18)$$

This signifies that

$$G_i(\mathbf{a}^{(r)} + \Delta\mathbf{a}^{(r)}) \simeq G_i(\mathbf{a}^{(r)}) + \sum_{k=1}^N \frac{\partial G_i(\mathbf{a}^{(r)})}{\partial a_k^{(r)}} \Delta a_k^{(r)}, \quad i = 1, 2, \dots, N. \quad (4.19)$$

Therefore,  $\Delta\mathbf{a}^{(r)}$  is a solution of the linear equation,

$$G_i(\mathbf{a}^{(r)}) = \hat{\mu}_i^{(r)} - \mu_i = \sum_{k=1}^N \left[ -\frac{\partial G_i(\mathbf{a}^{(r)})}{\partial a_k^{(r)}} \right] \Delta a_k^{(r)}, \quad i = 1, 2, \dots, N, \quad (4.20)$$

where notation  $\hat{\mu}_i^{(r)}$  is employed to point out that the estimated moments are updated by replacing  $\mathbf{a}^{(r)}$  in (4.16) after the  $r^{th}$  iteration. Differentiating (4.17) with respect to  $a_k$  yields

$$\frac{\partial G_i(\mathbf{a}^{(r)})}{\partial a_k^{(r)}} = \frac{\partial G_i(\mathbf{a})}{\partial a_k} \Big|_{\mathbf{a}=\mathbf{a}^{(r)}} = \hat{\mu}_{2k}^{(r)} \cdot \hat{\mu}_i^{(r)} - \hat{\mu}_{2k+i}^{(r)}. \quad (4.21)$$

The algorithm<sup>2</sup> is summarized in the following steps:

- *Step1*: Start with an initial value  $\mathbf{a}^{(0)} = \{a_1^{(0)}, a_2^{(0)}, \dots, a_N^{(0)}\}$ .
- *Step2*: Compute the estimated moments by replacing  $\mathbf{a}^{(r)}$  into (4.16).
- *Step3*: Plug the estimated moments into (4.20) to find  $\Delta\mathbf{a}^{(r)}$ .
- *Step4*: Compute the new parameters  $\mathbf{a}^{(r+1)} = \mathbf{a}^{(r)} + \Delta\mathbf{a}^{(r)}$ .
- *Step5*: Go to *Step 2*, if  $\|\Delta\mathbf{a}^{(r)}\| > \epsilon$ , where  $\epsilon$  is the desired precision and  $\|\cdot\|$  denotes the norm of a vector.

---

<sup>2</sup>This algorithm is adopted from [23].

The convexity of this maximization problem guarantees that if a stationary point is found for some finite values of  $a_1, \dots, a_N$ , it must be a unique absolute minimum [64]. However, the convexity alone does not imply that such a minimum should exist. More discussions on the convexity of the problem and existence of the solution can be found in [64, 65].

## 4.4 Probabilities of the Point Estimates (PPE)

In the following, a method to compute the PPE for the estimated parameters of the model in terms of the covariance matrix of the estimated moments is proposed. Subsequently, a relationship between the PPE for the BER and the PPE for the parameters is derived.

### 4.4.1 PPE for the Estimated Parameters

If the moment estimator satisfies a set of mild conditions, it follows that the estimated parameters are asymptotically normal with a derivable covariance matrix [66]. This allows for the PPE statements to be made concerning  $\hat{f}(y)$ . In the following, a method to compute the covariance matrix of the estimated parameters in terms of the covariance matrix of the moments is presented. The covariance matrix of the moments can be easily computed by using the method described in Appendix B.2.

We can continue computing the covariance matrix of the estimated parameters

in terms of the covariance matrix of the moments. We can rewrite (4.20) as follows:

$$\hat{\mu}_i^{(r)} - \mu_i = \sum_{k=1}^N h_{ik}^{(r)} \Delta a_k^{(r)}, \quad (4.22)$$

where  $h_{ik}^{(r)} = -\frac{\partial G_i(\mathbf{a}^{(r)})}{\partial a_k^{(r)}}$ . Let us define  $u_{ij}^{(r)}$  as the covariance of the moments  $\hat{\mu}_i^{(r)}, \hat{\mu}_j^{(r)}$ ; namely,

$$u_{ij}^{(r)} = \text{cov}(\hat{\mu}_i^{(r)}, \hat{\mu}_j^{(r)}) = E[(\hat{\mu}_i^{(r)} - \mu_i)(\hat{\mu}_j^{(r)} - \mu_j)], \quad (4.23)$$

where it is assumed<sup>3</sup> that  $\mu_i = E[\hat{\mu}_i^{(r)}]$  and  $\mu_j = E[\hat{\mu}_j^{(r)}]$ . Using (4.22), we have

$$u_{ij}^{(r)} = E \left[ \left( \sum_{k=1}^N h_{ik}^{(r)} \Delta a_k^{(r)} \right) \left( \sum_{m=1}^N h_{jm}^{(r)} \Delta a_m^{(r)} \right) \right] \quad (4.24)$$

$$= \sum_{k=1}^N \sum_{m=1}^N h_{ik}^{(r)} h_{jm}^{(r)} E[\Delta a_k^{(r)} \Delta a_m^{(r)}] \quad (4.25)$$

$$= \sum_{k=1}^N \sum_{m=1}^N h_{ik}^{(r)} h_{jm}^{(r)} \text{cov}(\Delta a_k^{(r)}, \Delta a_m^{(r)}) \quad (4.26)$$

$$= \sum_{k=1}^N h_{ik}^{(r)} \sum_{m=1}^N h_{jm}^{(r)} c_{mk}^{(r)} \quad (4.27)$$

$$= \sum_{k=1}^N h_{ik}^{(r)} d_{jk}^{(r)}. \quad (4.28)$$

In the matrix notation, the following is defined:

$$D = \{d_{ij}^{(r)}\}, \quad \text{where } d_{ij}^{(r)} = \sum_{m=1}^N h_{im}^{(r)} c_{mj}^{(r)}, \quad (4.29)$$

$$H = \{h_{ij}^{(r)}\}, \quad \text{where } h_{ij}^{(r)} = -\frac{\partial G_i(\mathbf{a}^{(r)})}{\partial a_j^{(r)}}, \quad (4.30)$$

$$U = \{u_{ij}^{(r)}\}, \quad \text{where } u_{ij}^{(r)} = \text{cov}(\hat{\mu}_i^{(r)}, \hat{\mu}_j^{(r)}), \quad (4.31)$$

---

<sup>3</sup>It can be shown that this assumption is equivalent to  $E[\Delta a_k^{(r)}] = 0$ ,  $k = 0, 1, \dots, N$ .

and

$$C^{(r)} = \{c_{ij}^{(r)}\}, \quad \text{where } c_{ij}^{(r)} = \text{cov}(\Delta a_i^{(r)}, \Delta a_j^{(r)}). \quad (4.32)$$

For simplicity of notation, the dependency of the matrices to  $r$  is not shown explicitly except for the matrix  $C^{(r)}$  which will be used later. The following equation relates these matrices, where the superscript  $T$  denotes the transpose of a matrix:

$$U = H.(H.C^{(r)})^T = H.(C^{(r)})^T.H^T. \quad (4.33)$$

This indicates that after each iteration, one can compute the covariance matrix of the  $\Delta \mathbf{a}$  in terms of the covariance matrix of the moments by the following:

$$C^{(r)} = (H^{-1}.U.(H^T)^{-1})^T = H^{-1}.U^T.(H^T)^{-1}. \quad (4.34)$$

We assume that the  $\Delta \mathbf{a}^{(r)}$ 's for the different iterations are uncorrelated. In this case, the covariance matrix of the parameters, denoted as  $A$ , is expressed as

$$A = \sum_{r=1}^R C^{(r)}, \quad (4.35)$$

where  $R$  is the total number of iterations. With the covariance matrix of the parameters, the desired PPEs for the parameters can be easily computed. An alternate method to compute the covariance of the parameters is to employ the delta method and linearize  $G$ .

#### 4.4.2 PPE for the Estimated BER

In the following, the previous results are used to compute the PPE for the BER. Let us assume that the  $c\%$  PPE for each parameter  $a_i$  is equal to some positive

$\alpha_i$ , i.e.,

$$p(|\delta a_1| < \alpha_1, \dots, |\delta a_N| < \alpha_N) = \frac{c}{100}, \quad (4.36)$$

where  $\delta a_i$  represents the error in the computation of the parameters. Using this notation, one can rewrite the BER integral from (4.5) as follows:

$$P_e + \Delta P_e(\delta a_1, \dots, \delta a_N) = \int_0^\infty \exp[-y/2 + \sum_{i=0}^N (a_i + \delta a_i)y^{2i}] dy \quad (4.37)$$

$$= \int_0^\infty \exp(-y/2 + \sum_{i=0}^N a_i y^{2i}) \exp(\sum_{i=0}^N \delta a_i y^{2i}) dy \quad (4.38)$$

$$\simeq \int_0^\infty \exp(-y/2 + \sum_{i=0}^N a_i y^{2i}) (1 + \sum_{i=0}^N \delta a_i y^{2i}) dy \quad (4.39)$$

$$= \int_0^\infty \exp(-y/2 + \sum_{i=0}^N a_i y^{2i}) dy + \int_0^\infty \exp(-y/2 + \sum_{i=0}^N a_i y^{2i}) \sum_{i=0}^N \delta a_i y^{2i} dy \quad (4.40)$$

$$= P_e + \sum_{i=0}^N \delta a_i \int_0^\infty y^{2i} \exp(-y/2 + \sum_{i=0}^N a_i y^{2i}) dy \quad (4.41)$$

$$= P_e + \sum_{i=0}^N m_i \delta a_i, \quad (4.42)$$

where

$$m_i = \int_0^\infty y^{2i} \exp(-y/2 + \sum_{i=0}^N a_i y^{2i}) dy. \quad (4.43)$$

It can be seen that  $\Delta P_e$ , the error in the BER estimation, is a linear combination of  $m_i$ 's, which can be estimated during the procedure of the moment computation by considering the positive samples only. Recalling the PPE statement (4.36) for the parameters, and noting that  $\Delta P_e$  is a linear combination of  $\delta a_i$ 's, one can

$E_b/N_0(dB)$	BER	$v(\text{new})$	$n(\text{new})$	$v(\text{MC})$	$n(\text{MC})$	$G$
1	$3.81 \times 10^{-2}$	$6.78 \times 10^{-5}$	$10^4$	$9.51 \times 10^{-5}$	$10^4$	1.4
2	$4.95 \times 10^{-3}$	$1.46 \times 10^{-5}$	$10^4$	$9.90 \times 10^{-5}$	$10^4$	6.8
3	$1.76 \times 10^{-4}$	$4.95 \times 10^{-6}$	$10^5$	$9.99 \times 10^{-6}$	$10^6$	20.2
4	$3.51 \times 10^{-6}$	$2.30 \times 10^{-8}$	$10^6$	$1.00 \times 10^{-8}$	$10^8$	43.5

Table 4.1: Comparison of the proposed method and the MC simulation.

present a similar statement for the BER as follows:

$$p(|\Delta P_e(\delta a_1, \dots, \delta a_N)| < \Delta P_e(\alpha_1, \dots, \alpha_N)) = \frac{c}{100}. \quad (4.44)$$

This analysis enables us to make PPE statements on the estimated BER in terms of the PPEs for the model parameters.

## 4.5 Numerical Results

A Turbo-Code of the length 100 and rate 1/2 is employed to perform the simulations. In Table 4.1, variances of the BER estimations are computed for both methods.

The number of samples and the variance to the mean ratio of the BER are denoted as  $n(\cdot)$  and  $v(\cdot)$ , respectively. The variance of the MC method can be computed analytically (refer to Appendix C.1), although this analysis is very complex for the proposed method and one need to estimate the variances with numerical methods. The variance of the proposed method can be computed by repeating the experiment for  $J$  times (generating  $J$  independent sets of moments), and com-

puting the variance of the resulting sequence of the BER values, denoted as  $p_i$ ,  $i = 1, \dots, J$ , as follows:

$$E[\hat{P}_e] = \frac{1}{J} \sum_{i=1}^J p_i \quad (4.45)$$

and

$$\text{var}[\hat{P}_e] = -(E[\hat{P}_e])^2 + \frac{1}{J} \sum_{i=1}^J p_i^2. \quad (4.46)$$

In the computations of Table 4.1, set  $J = 1000$  to obtain a reasonable approximation, and at the same time, render the analysis feasible in the sense of the required time.

We use the relative gain  $G$  in Table 4.1 as a measure to compare the two methods. To incorporate both the variance reduction and the sample reduction advantage of the new method, and noting that  $v(\text{MC})$  is inversely proportional<sup>4</sup> to  $n(\text{MC})$ , define  $G$  as follows:

$$G = \frac{v(\text{MC})}{v(\text{our method})} \cdot \frac{n(\text{MC})}{n(\text{our method})}. \quad (4.47)$$

In addition the PPEs are computed by using the proposed method in Section 4.4 for this example. This PPE is closely related to  $n$ , the number of samples used to compute it. In Table 4.2, this relation is presented for three different values of  $n$  at  $E_b/N_0=2\text{dB}$ . We compute<sup>5</sup>  $p(|\Delta P_e| < \theta)$  for the different values of  $n$ , where  $\theta = \Delta P_e(\alpha|a_1|, \dots, \alpha|a_N|)$ . When the proposed method is compared with the MC simulation in Table 4.1 and Table 4.2, the number of samples required for the BER calculations indicate a significant reduction for our method. It can be seen

---

<sup>4</sup>Refer to Appendix C.1.

<sup>5</sup>Refer to Appendix C.2 for more details on the PPE for the MC simulation.

$n$	$\alpha$	$\theta = \Delta P_e(\alpha a_1 , \dots, \alpha a_N )$	$p( \Delta P_e  < \theta)$	$p( \Delta P_e  < \theta)$
			new method	MC
$10^4$	0.742	0.0058	0.95	0.67
$10^5$	0.742	0.0058	0.96	0.70
$10^6$	0.742	0.0058	0.97	0.96
$10^4$	0.251	0.0020	0.94	0.66
$10^5$	0.251	0.0020	0.95	0.70
$10^6$	0.251	0.0020	0.96	0.96
$10^4$	0.075	0.0005	0.93	0.33
$10^5$	0.075	0.0005	0.94	0.68
$10^6$	0.075	0.0005	0.95	0.95

Table 4.2: Relation between  $n$  and PPE at  $E_b/N_0=2\text{dB}$ .

that the proposed method is more accurate than the MC simulation even by using significantly fewer samples.

Simulation results are shown in Figure 4.1, where the same number of samples as indicated in Table 4.1 is used. It is evident that increasing the number of moments (the order of approximation) that are involved from two to five significantly improves the approximation.

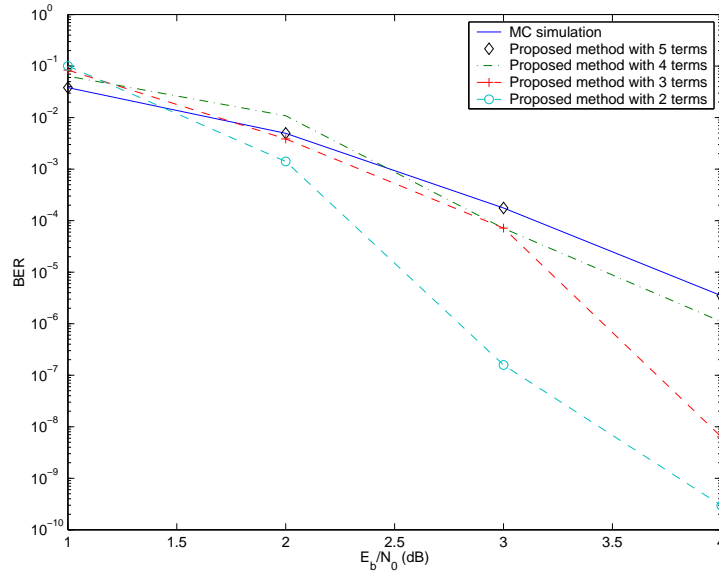


Figure 4.1: BER curves for Turbo-Code of the length 100 and rate 1/2.

## 4.6 Summary

In this chapter, a new method for the performance evaluation of Turbo-Like Codes is proposed. The problem of finding the BER in high signal to noise ratio regions can be solved with this method, since the MC simulation may not be feasible. We take advantage of the symmetry properties of the *pdf* of the bit *LLR* to propose a suitable model for this unknown density. The moment matching method is employed to find the density with the maximum entropy which satisfies the moment constraints. A simple method is introduced to make PPE statements both for the parameters of the model and the BER integral, which enables us to compute the BER values accurately. It is demonstrated that significantly fewer samples, compared to those required in the MC simulation, are necessary to compute the

statistical moments that are accurate enough. The complexity of the method scales with the block length of the code similar to the MC simulation with the difference that fewer samples are required to be generated in our case. In other words, the proposed method does not reduce the complexity of the iterative decoding algorithms, while it reduces the number of required samples. Note that solving the proposed maximization algorithm is not complicated at all.

# Chapter 5

## Concluding Remarks

### 5.1 Summary of the Contributions

The contributions of this thesis in the context of bit decoding algorithms are summarized as follows:

- **Invariance properties:**

- For a general channel model with discrete input and discrete or continuous output, it is shown that the *pdf* of the bit *LLR* is symmetric and independent of the transmitted code-word. It is proved that under certain symmetry conditions on the channel, the *pdf* of the bit *LLR* for a specific bit position is independent of the transmitted code-word if the value of that bit position remains unchanged.
- It is also shown that a change in the value of a bit position makes the *pdf* of that bit *LLR* reflect through the vertical axis.

- A sufficient condition for two bit positions to have the same *pdf* for their bit *LLR* is presented. This condition is satisfied if a permutation exists within the automorphism group of the code, which transfers one bit position to the other.
- It is shown that the class of Cyclic codes have this property.

These results are published in [43, 67, 59].

- **Performance analysis:**

- In the light of the above properties, a method is presented for calculating bit error probability of binary linear block codes over AWGN channel, using special weight enumerating functions of the code. These results are published in [68, 69].
- A new method for performance evaluation of Turbo-Like Codes is presented, which is significantly faster than MC simulations. This method is based on using an exponential model for the *pdf* of the bit *LLR*. The parameters of the model can be found using an iterative approach considering the maximum entropy principle. This method is published in [70].

## 5.2 Future Work

Some possible directions for future research are as follows:

- **Cumulant Matching:** The first direction is to develop an alternate method for estimating the parameters of the *pdf*. Some preliminary results are already established in Appendix D. This alternate method is based on using the cumulants versus the moments of the bit *LLR*. The first two cumulants of the normal density are its mean and variance and the higher order cumulants are zero. Since the *pdf* of the bit *LLR* is nearly normal, it is expected that its higher order cumulants are fairly small. This allows for easy truncation of the series expansion of the *pdf* in terms of the cumulants.
- **Average Performance:** In the current thesis, the performance of one specific bit position is evaluated. This is a good measure for codes with similar performance for their different bit positions, e.g. Cyclic codes. However, this is not true for Turbo-Like codes, since the average performance over a frame is required. It can be investigated then what type of modifications to the proposed analytical methods are necessary in order to get the average performance of the codes.
- **Code Design:** In chapter 3, the performance of linear codes is computed in terms of some special weight distribution functions of the code. This result can be used in code design.
- **Constellation Design:** It is widely known that mapping a bit to a multi-level constellation results in dependency between bit likelihood values. Noting this dependency between likelihood random variables, the performance of the iterative decoders can be improved [71]. A possible research direction toward using the results of this thesis in constellation labeling and constellation

design may be fruitful.

- **Space-Time codes:** Space-Time codes can be regarded as a multilevel signal constellation and, therefore, their properties can be investigated as a natural extension to the results of chapter 2.

# Appendix A

## Proofs of Theorems

### A.1 Proof of Theorem 4

Theorem 4:

*Proof:* Using (3.30), one can write,

$$\frac{\partial R_{\{q_1, \dots, q_j\}}(\mathbf{n})}{\partial n_{q_i}} = \frac{\partial}{\partial n_{q_i}} [A^{-1}(\mathbf{n})F_{\{q_1, \dots, q_j\}}(\mathbf{n})] \quad (\text{A.1})$$

$$= A^{-1}(\mathbf{n}) \frac{\partial F_{\{q_1, \dots, q_j\}}(\mathbf{n})}{\partial n_{q_i}} + \frac{\partial A^{-1}(\mathbf{n})}{\partial n_{q_i}} F_{\{q_1, \dots, q_j\}}(\mathbf{n}) \quad (\text{A.2})$$

$$= A^{-1}(\mathbf{n}) \frac{\partial F_{\{q_1, \dots, q_j\}}(\mathbf{n})}{\partial n_{q_i}} - A^{-2}(\mathbf{n}) \frac{\partial A(\mathbf{n})}{\partial n_{q_i}} F_{\{q_1, \dots, q_j\}}(\mathbf{n}) \quad (\text{A.3})$$

$$= A^{-1}(\mathbf{n}) \frac{\partial F_{\{q_1, \dots, q_j\}}(\mathbf{n})}{\partial n_{q_i}} - A^{-2}(\mathbf{n}) F_{\{q_i\}}(\mathbf{n}) F_{\{q_1, \dots, q_j\}}(\mathbf{n}) \quad (\text{A.4})$$

$$= A^{-1}(\mathbf{n}) \frac{\partial F_{\{q_1, \dots, q_j\}}(\mathbf{n})}{\partial n_{q_i}} - [A^{-1}(\mathbf{n}) F_{\{q_1, \dots, q_j\}}(\mathbf{n})][A^{-1}(\mathbf{n}) F_{\{q_i\}}(\mathbf{n})] \quad (\text{A.5})$$

$$= A^{-1}(\mathbf{n}) \frac{\partial F_{\{q_1, \dots, q_j\}}(\mathbf{n})}{\partial n_{q_i}} - R_{\{q_1, \dots, q_j\}}(\mathbf{n}) R_{\{q_i\}}(\mathbf{n}). \quad (\text{A.6})$$

Using (3.27) and noting that  $m^2(c_{q_i}^l) = 1$ , we have,

$$\frac{\partial F_{\{q_1, \dots, q_j\}}(\mathbf{n})}{\partial n_{q_i}} = \begin{cases} \sigma^{-4} F_{\{q_1, \dots, q_{i-1}, q_{i+1}, \dots, q_j\}}(\mathbf{n}), & \text{If } q_i \in \{q_1, \dots, q_j\}, \\ F_{\{q_1, \dots, q_j, q_i\}}(\mathbf{n}), & \text{Otherwise.} \end{cases} \quad (\text{A.7})$$

Substituting (A.7) in (A.6), and using (3.30), completes the proof.  $\square$

## A.2 Proof of Theorem 5

Theorem 5:

*Proof:* Two different cases are considered. If  $q_i \in \{q_1, \dots, q_j\}$  using (3.36), one can write,

$$\frac{\partial^2 R_{\{q_1, \dots, q_j\}}(\mathbf{n})}{\partial n_{q_i}^2} = \frac{\partial}{\partial n_{q_i}} [\sigma^{-4} R_{\{q_1, \dots, q_{i-1}, q_{i+1}, \dots, q_j\}}(\mathbf{n}) - R_{\{q_1, \dots, q_j\}}(\mathbf{n}) R_{\{q_i\}}(\mathbf{n})] \quad (\text{A.8})$$

$$= \sigma^{-4} \frac{\partial}{\partial n_{q_i}} [R_{\{q_1, \dots, q_{i-1}, q_{i+1}, \dots, q_j\}}(\mathbf{n})] - \frac{\partial}{\partial n_{q_i}} [R_{\{q_1, \dots, q_j\}}(\mathbf{n}) R_{\{q_i\}}(\mathbf{n})] \quad (\text{A.9})$$

$$= \sigma^{-4} [R_{\{q_1, \dots, q_j\}}(\mathbf{n}) - R_{\{q_1, \dots, q_{i-1}, q_{i+1}, \dots, q_j\}}(\mathbf{n}) R_{\{q_i\}}(\mathbf{n})] - \frac{\partial}{\partial n_{q_i}} [R_{\{q_1, \dots, q_j\}}(\mathbf{n}) R_{\{q_i\}}(\mathbf{n})] \quad (\text{A.10})$$

$$= \sigma^{-4} [R_{\{q_1, \dots, q_j\}}(\mathbf{n}) - R_{\{q_1, \dots, q_{i-1}, q_{i+1}, \dots, q_j\}}(\mathbf{n}) R_{\{q_i\}}(\mathbf{n})] - [\sigma^{-4} R_{\{q_1, \dots, q_{i-1}, q_{i+1}, \dots, q_j\}}(\mathbf{n}) - R_{\{q_1, \dots, q_j\}}(\mathbf{n}) R_{\{q_i\}}(\mathbf{n})] R_{\{q_i\}}(\mathbf{n}) - R_{\{q_1, \dots, q_j\}}(\mathbf{n}) [\sigma^{-4} - R_{\{q_i\}}^2(\mathbf{n})] \quad (\text{A.11})$$

$$= -2\sigma^{-4} R_{\{q_1, \dots, q_{i-1}, q_{i+1}, \dots, q_j\}}(\mathbf{n}) R_{\{q_i\}}(\mathbf{n}) + 2R_{\{q_1, \dots, q_j\}}(\mathbf{n}) R_{\{q_i\}}^2(\mathbf{n}) \quad (\text{A.12})$$

$$= -2R_{\{q_i\}}(\mathbf{n}) [\sigma^{-4} R_{\{q_1, \dots, q_{i-1}, q_{i+1}, \dots, q_j\}}(\mathbf{n}) - R_{\{q_1, \dots, q_j\}}(\mathbf{n}) R_{\{q_i\}}(\mathbf{n})] \quad (\text{A.13})$$

$$= -2R_{\{q_i\}}(\mathbf{n}) \frac{\partial R_{\{q_1, \dots, q_j\}}(\mathbf{n})}{\partial n_{q_i}}. \quad (\text{A.14})$$

For the other case where  $q_i \notin \{q_1, \dots, q_j\}$ , we have,

$$\frac{\partial^2 R_{\{q_1, \dots, q_j\}}(\mathbf{n})}{\partial n_{q_i}^2} = \frac{\partial}{\partial n_{q_i}} [R_{\{q_1, \dots, q_j, q_i\}}(\mathbf{n}) - R_{\{q_1, \dots, q_j\}}(\mathbf{n}) R_{\{q_i\}}(\mathbf{n})] \quad (\text{A.15})$$

$$= \frac{\partial}{\partial n_{q_i}} [R_{\{q_1, \dots, q_j, q_i\}}(\mathbf{n})] - \frac{\partial}{\partial n_{q_i}} [R_{\{q_1, \dots, q_j\}}(\mathbf{n}) R_{\{q_i\}}(\mathbf{n})] \quad (\text{A.16})$$

$$\begin{aligned} &= \sigma^{-4} R_{\{q_1, \dots, q_j\}}(\mathbf{n}) - R_{\{q_1, \dots, q_j, q_i\}}(\mathbf{n}) R_{\{q_i\}}(\mathbf{n}) \\ &\quad - \frac{\partial}{\partial n_{q_i}} [R_{\{q_1, \dots, q_j\}}(\mathbf{n})] R_{\{q_i\}}(\mathbf{n}) - R_{\{q_1, \dots, q_j\}}(\mathbf{n}) \frac{\partial}{\partial n_{q_i}} [R_{\{q_i\}}(\mathbf{n})] \end{aligned} \quad (\text{A.17})$$

$$\begin{aligned} &= \sigma^{-4} R_{\{q_1, \dots, q_j\}}(\mathbf{n}) - R_{\{q_1, \dots, q_j, q_i\}}(\mathbf{n}) R_{\{q_i\}}(\mathbf{n}) \\ &\quad - [R_{\{q_1, \dots, q_j, q_i\}}(\mathbf{n}) - R_{\{q_1, \dots, q_j\}}(\mathbf{n}) R_{\{q_i\}}(\mathbf{n})] R_{\{q_i\}}(\mathbf{n}) \\ &\quad - R_{\{q_1, \dots, q_j\}}(\mathbf{n}) [\sigma^{-4} - R_{\{q_i\}}^2(\mathbf{n})] \end{aligned} \quad (\text{A.18})$$

$$= -2R_{\{q_1, \dots, q_j, q_i\}}(\mathbf{n}) R_{\{q_i\}}(\mathbf{n}) + 2R_{\{q_1, \dots, q_j\}}(\mathbf{n}) R_{\{q_i\}}^2(\mathbf{n}) \quad (\text{A.19})$$

$$= -2R_{\{q_i\}}(\mathbf{n}) [R_{\{q_1, \dots, q_j, q_i\}}(\mathbf{n}) - R_{\{q_1, \dots, q_j\}}(\mathbf{n}) R_{\{q_i\}}(\mathbf{n})] = -2R_{\{q_i\}}(\mathbf{n}) \frac{\partial R_{\{q_1, \dots, q_j\}}(\mathbf{n})}{\partial n_{q_i}}. \quad (\text{A.20})$$

It can be seen from (A.14) and (A.20) that both cases ended up to the same expression as the one in (3.37), which completes the proof.  $\square$

### A.3 Proof of Property (3.44)

The right hand side of (3.44) can be expanded as follows,

$$ye^{-\frac{y^2}{2}} T_{i-1}(y) - e^{-\frac{y^2}{2}} \frac{d}{dy} T_{i-1}(y) \quad (\text{A.21})$$

$$= e^{-\frac{y^2}{2}} \left[ y T_{i-1}(y) - \frac{d}{dy} T_{i-1}(y) \right] \quad (\text{A.22})$$

Using (3.16), we have,

$$T_{i-1}(y) = (-1)^{i-1} e^{\frac{y^2}{2}} \frac{d^{i-1}}{dy^{i-1}} [e^{-\frac{y^2}{2}}]. \quad (\text{A.23})$$

It follows that,

$$\frac{d}{dy} T_{i-1}(y) = (-1)^{i-1} \left( y e^{y^2/2} \frac{d^{i-1}}{dy^{i-1}} [e^{-y^2/2}] + e^{y^2/2} \frac{d^i}{dy^i} [e^{-y^2/2}] \right) \quad (\text{A.24})$$

$$= y T_{i-1}(y) - T_i(y) \quad (\text{A.25})$$

Substituting (A.25) in (A.22), we have,

$$e^{-\frac{y^2}{2}} T_i(y), \quad (\text{A.26})$$

which is the left hand side of (3.44).

# Appendix B

## Moment Estimation

### B.1 Error in Moment Estimation

The effect of a small error in the moment estimation on the parameter estimates is investigated in this appendix. Define the  $k^{th}$  moment of the random variable  $Y$  (which corresponds to the bit  $LLR$ ) as,

$$\mu_k = E[Y^k], \tag{B.1}$$

which can be estimated by statistical averaging as follows:

$$\tilde{\mu}_k = \frac{1}{n} \sum_{i=1}^n y_i^k, \tag{B.2}$$

where  $y_i$  is one instance of the random variable  $Y$ , and  $n$  is the number of samples. Let us define  $\Delta U = \{d_i\}$  as the  $N$ -dimensional error vector of statistical moment estimates, where  $\tilde{\mu}_i = d_i + \mu_i$ . Assuming that the iterative algorithm for parameter estimation has converged to some answer and the moments are matched, there will

be still a residual error,  $\Delta A$  in parameter estimates taking  $\Delta U$  into the account.

Using (4.22), we have,

$$\Delta U = \mathbf{H} \cdot \Delta A, \quad (\text{B.3})$$

where the Hessian matrix  $\mathbf{H}$ , has the following properties:

- $\mathbf{H}$  is positive definite, which means its singular values are positive and finite.
- $\mathbf{H}^{-1}$  is also positive definite.
- $\mathbf{H}$  is Hermitian, which means  $\mathbf{H} = \mathbf{H}^H$  where superscript  $H$  stands for hermitian.

One can partition  $\mathbf{H}^{-1}$  as  $\mathbf{V} \cdot \mathbf{\Lambda} \cdot \mathbf{V}^H$ , where  $\mathbf{\Lambda}$  is the diagonal matrix of singular values, and  $\mathbf{V}$  is a unitary matrix. This results in,

$$\Delta A = \mathbf{V} \cdot \mathbf{\Lambda} \cdot \mathbf{V}^H \cdot \Delta U, \quad (\text{B.4})$$

$$\mathbf{V}^{-1} \cdot \Delta A = \mathbf{\Lambda} \cdot \mathbf{V}^H \cdot \Delta U, \quad (\text{B.5})$$

$$\mathbf{A}' = \mathbf{\Lambda} \cdot \mathbf{U}'. \quad (\text{B.6})$$

If the elements of  $\Delta U$  tend to zero (by increasing the number of samples), then the elements of  $\Delta A$  will approach zero, since the singular values are positive and finite. The conclusion is that an arbitrary small error in moment estimation results in an arbitrary small error in parameter estimates.

## B.2 Covariance Matrix of the Moments

The covariance matrix of the moments can be computed as follows:

$$\text{cov}(\tilde{\mu}_k, \tilde{\mu}_m) = E[(\tilde{\mu}_k - \mu_k)(\tilde{\mu}_m - \mu_m)] \quad (\text{B.7})$$

$$= E\left[\left(\frac{1}{n} \sum_{i=1}^n y_i^k - \mu_k\right)\left(\frac{1}{n} \sum_{j=1}^n y_j^m - \mu_m\right)\right] \quad (\text{B.8})$$

$$= \frac{1}{n^2} \sum_{i=1}^n \sum_{j=1}^n E[y_i^k y_j^m] - \mu_k \mu_m \quad (\text{B.9})$$

$$= \frac{1}{n^2} [n\mu_{k+m} + (n^2 - n)\mu_k \mu_m] - \mu_k \mu_m \quad (\text{B.10})$$

$$= \frac{1}{n} (\mu_{k+m} - \mu_k \mu_m). \quad (\text{B.11})$$

# Appendix C

## Properties of the Monte-Carlo (MC) Simulation

### C.1 Variance of the MC Simulation

Consider the situation of transmitting a bit  $b_i$  and decoding  $\hat{b}_i$ , for  $i = 1, \dots, n$ , where  $n$  is the number of samples used for the MC simulation. Define the following random variable:

$$e_i = \begin{cases} 1, & b_i \neq \hat{b}_i, \\ 0, & \text{otherwise.} \end{cases} \quad (\text{C.1})$$

An error event is represented by  $e_i$ . One can find the BER by averaging the following random variable,  $\hat{P}_e$ . i.e.,  $P_e = E[\hat{P}_e]$ ,

$$\hat{P}_e = \frac{1}{n} \sum_{i=1}^n e_i. \quad (\text{C.2})$$

To compute the variance of  $\hat{P}_e$ , one need to use the definition of variance as follows:

$$\text{var}[\hat{P}_e] = E\left[\left(\frac{1}{n} \sum_{i=1}^n e_i\right)^2\right] - \left(E\left[\frac{1}{n} \sum_{i=1}^n e_i\right]\right)^2 \quad (\text{C.3})$$

$$= \frac{1}{n^2} \left( E\left[\sum_{i=1}^n e_i^2\right] + E\left[\sum_{i=1}^n \sum_{i \neq j=1}^n e_i e_j\right] \right) - P_e^2 \quad (\text{C.4})$$

$$= \frac{1}{n^2} \sum_{i=1}^n E[e_i^2] + \frac{1}{n^2} \sum_{i=1}^n \sum_{i \neq j=1}^n E[e_i]E[e_j] - P_e^2 \quad (\text{C.5})$$

$$= \frac{P_e}{n} + \frac{n(n-1)}{n^2} P_e^2 - P_e^2 \quad (\text{C.6})$$

$$= \frac{P_e}{n} (1 - P_e). \quad (\text{C.7})$$

In practice, an estimation of  $\text{var}[\hat{P}_e]$  is obtained by substituting  $P_e$  with  $\hat{P}_e$  in (C.7).

## C.2 Computing PPEs for the MC Simulation

Define the  $c\%$  PPE for the MC, denoted as  $\alpha$ , as follows:

$$p(|P_e - \hat{P}_e| < \alpha) = \frac{c}{100}, \quad (\text{C.8})$$

where  $P_e, \hat{P}_e$  are the true and the estimated values of the BER. Following the same notation and definitions as Appendix C.1, for a large  $n$  and some integers  $m$  and  $a$ , one can represent  $P_e$  and  $\alpha$  as  $\frac{m}{n}$  and  $\frac{a}{n}$ , respectively. The PPE for the MC

simulation can be found as follows:

$$p(|P_e - \hat{P}_e| < \frac{a}{n}) = p\left(\frac{m-a}{n} < \frac{1}{n} \sum_{i=1}^n e_i < \frac{m+a}{n}\right) \quad (\text{C.9})$$

$$= p\left(\sum_{i=1}^n e_i < m+a\right) - p\left(\sum_{i=1}^n e_i \leq m-a\right) \quad (\text{C.10})$$

$$= \sum_{j=0}^{m+a-1} p(j \text{ errors among } n \text{ bits}) - \sum_{j=0}^{m-a} p(j \text{ errors among } n \text{ bits}) \quad (\text{C.11})$$

$$= \sum_{j=0}^{m+a-1} \binom{n}{j} P_e^j (1-P_e)^{n-j} - \sum_{j=0}^{m-a} \binom{n}{j} P_e^j (1-P_e)^{n-j} \quad (\text{C.12})$$

$$= \sum_{j=m-a+1}^{m+a-1} \binom{n}{j} P_e^j (1-P_e)^{n-j}. \quad (\text{C.13})$$

In practice, an estimation of (C.9) is obtained by substituting  $P_e$  with  $\hat{P}_e$  in (C.13).

# Appendix D

## Cumulant Method

The higher order statistics, i.e., the *cumulants* have been widely used in a variety of applications including analysis of digital communications systems. The problem of performance evaluation of coherent optical communication systems is considered in [72], where a solution based on estimating the cumulants of the noise process is presented. A condition is derived to quantify under what system conditions a Gaussian *pdf* is a good approximation. A discrete-time method is proposed in [73] for estimating the impulse response of a frequency selective digital modulated communication channel. This method is based on estimating the cumulants up to the fourth order. Parameters of a moving average model are estimated in [74], using second and third order cumulant matching. This estimation is further improved in [75] by employing only one of the third or fourth cumulant versus both of them. Cumulants of symmetric distributions like uniform, triangular, and Gaussian are estimated in [76] using a robust estimation technique. The application of Edgeworth series and higher-order statistics to the discrete-time detection of a known

constant signal in multivariate non-Gaussian noise are considered in [77]. A numerical algorithm based on knowledge of the noise cumulants is presented in order to analyze the finite-sample size performance of the sub-optimum detectors.

Non-Gaussian sources are modeled in [78] using Gaussian mixture densities. It is shown that in high SNR regions, this method outperforms the cumulant based algorithms for parameter estimation. The problem of blind equalization and estimation of digital communication finite impulse response channels is considered in [79]. The channel parameters are estimated by nonlinear optimization of a quadratic cumulant matching criterion involving second and fourth order cumulants. This problem is later considered in [80] for partial-response signals. A method for phase recovery in Quadrature Amplitude Modulation (QAM) communication systems based on higher order statistics is presented in [81]. A relation is derived between the phase error and the fourth order cumulant of the output.

Since the higher order cumulant-based criteria are multi-modal, conventional gradient search techniques require a good initial estimate to avoid converging to local minima. This problem is solved in [82], where a novel scheme based on genetic algorithms is employed to optimize the cumulant fitting cost function. A method based on higher order statistics is proposed in [83] to mitigate the performance degradation caused by multi-path propagation in a mobile radio communication system. It is shown that an over-determined system of linear equations (involving only cumulants of the received baseband signal) can be obtained to perform non-iterative deconvolution. The study of chaos communication systems with AWGN interference is considered in [84] by employing suitable cumulant analysis tools.

The first two cumulants of the normal density are its mean and variance and the

higher order cumulants are zero. Since the *pdf* of the bit *LLR* is nearly normal, it is expected that its higher order cumulants are fairly small. This allows for easy truncation of the series expansion of the *pdf* in terms of the cumulants. The characteristic function of a random variable  $Y$ , with its *pdf* denoted as  $f(y)$  is defined as

$$\Phi(t) = \int_{-\infty}^{+\infty} f(y)e^{ity} dy. \quad (\text{D.1})$$

The cumulants of the random variable  $Y$ , denoted as  $k_m$ , are the coefficients of the following series expansion:

$$\log \Phi(t) = \sum_{m=0}^{\infty} k_m \frac{(it)^m}{m!} \quad (\text{D.2})$$

Cumulants can be expressed in terms of the raw moments as follows [49]:

$$k_1 = \mu_1, \quad (\text{D.3})$$

$$k_2 = \mu_2 - \mu_1^2, \quad (\text{D.4})$$

$$k_3 = 2\mu_1^3 - 3\mu_1\mu_2 + \mu_3, \quad (\text{D.5})$$

$$\dots \quad (\text{D.6})$$

The  $K$ -statistics are the unique symmetric unbiased estimators of the cumulants [50].

$$E[K_m] = k_m \quad (\text{D.7})$$

where notation  $K_m$  is used for the  $m^{\text{th}}$   $K$ -statistics of a given density. In addition, the variance,

$$V[K_m] = E[(K_m - k_m)^2], \quad (\text{D.8})$$

is a minimum compared to all other unbiased estimators [85, 86]. The first few  $K$ -statistics are as follows:

$$K_1 = \frac{S_1}{n}, \quad (\text{D.9})$$

$$K_2 = \frac{nS_2 - S_1^2}{n(n-1)}, \quad (\text{D.10})$$

$$K_3 = \frac{2S_1^3 - 3nS_1S_2 + n}{n(n-1)(n-2)}, \quad (\text{D.11})$$

$$\dots \quad (\text{D.12})$$

where  $n$  is the number of samples used in the estimation, and

$$S_r = \sum_{i=1}^n X_i^r. \quad (\text{D.13})$$

Once the first few cumulants are estimated by using the  $K$ -statistics, the characteristic function of the bit  $LLR$  can be approximated by using (D.2). Following that, the  $pdf$  of the bit  $LLR$  can be computed by taking inverse Fourier transform from  $\Phi(t)$ .

$$f(y) = \int_{-\infty}^{+\infty} \Phi(t)e^{-ity} dt \quad (\text{D.14})$$

Cumulative distribution function ( $CDF$ ) of the bit  $LLR$  can be defined as,

$$F(T) = \int_{-\infty}^T f(y)dy \quad (\text{D.15})$$

We are interested in computing error probability,  $P_e$ ,

$$P_e = \int_0^{\infty} f(y)dy = 1 - \int_{-\infty}^0 f(y)dy = 1 - F(0) \quad (\text{D.16})$$

Taking Inverse Fourier Transform (IFT) of the characteristics function,  $f(y) = \text{IFT}\{\Phi(t)\}$ , and noting that

$$\log \Phi(t) = \sum_{m=0}^{\infty} k_m \frac{(it)^m}{m!}, \quad (\text{D.17})$$

we have,

$$F(T) = \text{IFT} \left\{ \frac{1}{t} \exp \left[ \sum_{m=0}^{\infty} k_m \frac{(it)^m}{m!} \right] \right\} \quad (\text{D.18})$$

Small error,  $\Delta k_m$  in estimating each cumulant, results in an error,  $\Delta F(T)$  in *CDF*,

$$F(T) + \Delta F(T) = \text{IFT} \left\{ \frac{1}{t} \exp \left[ \sum_{m=0}^{\infty} (k_m + \Delta k_m) \frac{(it)^m}{m!} \right] \right\} \quad (\text{D.19})$$

$$= \text{IFT} \left\{ \frac{1}{t} \exp \left[ \sum_{m=0}^{\infty} k_m \frac{(it)^m}{m!} \right] \exp \left[ \sum_{n=0}^{\infty} \Delta k_n \frac{(it)^n}{n!} \right] \right\} \quad (\text{D.20})$$

$$\simeq \text{IFT} \left\{ \frac{1}{t} \exp \left[ \sum_{m=0}^{\infty} k_m \frac{(it)^m}{m!} \right] \left( 1 + \sum_{n=0}^{\infty} \Delta k_n \frac{(it)^n}{n!} \right) \right\} \quad (\text{D.21})$$

$$= F(T) + \text{IFT} \left\{ \frac{1}{t} \exp \left[ \sum_{m=0}^{\infty} k_m \frac{(it)^m}{m!} \right] \sum_{n=0}^{\infty} \Delta k_n \frac{(it)^n}{n!} \right\} \quad (\text{D.22})$$

This means,

$$\Delta F(T) \simeq \text{IFT} \left\{ \frac{1}{t} \exp \left[ \sum_{m=0}^{\infty} k_m \frac{(it)^m}{m!} \right] \sum_{n=0}^{\infty} \Delta k_n \frac{(it)^n}{n!} \right\} \quad (\text{D.23})$$

$$= \sum_{n=0}^{\infty} \Delta k_n \frac{i^n}{n!} \text{IFT} \left\{ t^{n-1} \exp \left[ \sum_{m=0}^{\infty} k_m \frac{(it)^m}{m!} \right] \right\} \quad (\text{D.24})$$

$$= \sum_{n=0}^{\infty} \Delta k_n \frac{i^n}{n!} f^{(n-1)}(T), \quad (\text{D.25})$$

where  $f^{(n)}(T)$  is the  $n^{\text{th}}$  derivative of  $f(y)$  at point  $y = T$  for  $n > 0$ . To have a consistent notation, we define  $f^{(0)}(T) = f(T)$ , and  $f^{(-1)}(T) = F(T)$ . This results in the following interesting relationship between the error in computing  $P_e$  and the error in estimating cumulants:

$$\Delta P_e \simeq \sum_{n=0}^{\infty} \Delta k_n \frac{i^n}{n!} f^{(n-1)}(0) \quad (\text{D.26})$$

# Bibliography

- [1] C. E. Shannon, “A Mathematical Theory of Communication,” Bell System Technical Journal, vol. 27, pp. 379-423 and 623-656, July and October, 1948.
- [2] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv. , “Optimal Decoding of Linear Codes for Minimizing Symbol Error Rate,” *IEEE Transactions on Information Theory*, vol.20, pp. 284-287, March 1974.
- [3] L. Ping, K. L. Yeung , “Symbol-by-Symbol Decoding of the Golay Code and Iterative Decoding of Concatenated Golay Codes,” *IEEE Transactions on Information Theory*, vol.45, no.7, pp. 2558-2562, November 1999.
- [4] Y. Liu, S. Lin, M. P. C. Fossorier , “MAP Algorithms for Decoding Linear Block Codes Based on Sectionalized Trellis Diagrams,” *IEEE Transactions on Communications*, vol.48, no.4, pp. 577-586, April 2000.
- [5] S. Riedel , “Symbol-by-Symbol MAP Decoding Algorithm For High-Rate Convolutional Codes That Use Reciprocal Dual Codes,” *IEEE Journal on Selected Areas in Communications*, vol.16, no.2, pp. 175-185, February 1998.

- [6] C. R. P. Hartmann, L. D. Rudolph, "An Optimum Symbol-by-Symbol Decoding Rule for Linear Codes," *IEEE Transactions on Information Theory*, vol.22, no.5, pp. 514-517, September 1976.
- [7] C. Berrou, A. Glavieux, and P. Thitimajshima. "Near Shannon Limit Error-Correcting Coding and Decoding: Turbo-Codes (1)," *Proceedings IEEE International Conference on Communications*, Geneva, Switzerland, pp. 1064-1070, May 1993.
- [8] S. Y. Chung, T. J. Richardson, R. L. Urbanke, "Analysis of Sum-Product Decoding of Low-Density-Parity-Check Codes Using Gaussian Approximation," *IEEE Transactions on Information Theory*, vol.47, no.2, pp. 657-670, February 2001.
- [9] T. J. Richardson, M. A. Shokrollahi, R. L. Urbanke, "Design of Capacity Approaching Irregular Low-Density-Parity-Check Codes," *IEEE Transactions on Information Theory*, vol.47, no.2, pp. 619-637, February 2001.
- [10] T. J. Richardson, R. L. Urbanke, "The Capacity of Low-Density-Parity-Check Codes Under Message Passing Decoding," *IEEE Transactions on Information Theory*, vol.47, no.2, pp. 599-618, February 2001.
- [11] J. G. Proakis, *Digital Communication*, 2nd ed., New York: McGraw-Hill, 1989.
- [12] S. S. Pietrobon, "On the Probability of Error of Convolutional Codes," *IEEE Transactions on Information Theory*, vol.42, no.5, pp. 1562-1568, September 1996.

- [13] M. R. Best, M. V. Burnashev, Y. Levy, A. Rabinovich, P. C. Fishburn, A. R. Calderbank, and D. J. Costello, Jr., "On a technique to calculate the exact performance of a convolutional code," *IEEE Transactions on Information Theory*, vol. 41, no. 2, pp. 441-447, March 1995.
- [14] C. R. P. Hartmann, L. D. Rudolph, K. G. Mehrotra, "Asymptotic Performance of Optimum Bit-by-Bit Decoding for the White Gaussian Channel," *IEEE Transactions on Information Theory*, vol.23, no.4, pp. 520-522, July 1977.
- [15] E. Baccarelli, R. Cusani, G. D. Blasio, "Performance Bound and Trellis-Code Design Criterion for Discrete Memoryless Channels and Finite-Delay Symbol-by-Symbol Decoding," *IEEE Transactions on Communications*, vol.45, no.10, pp. 1192-1199, October 1997.
- [16] D. C. Schleher, "Generalized Gram-Charlier Series with Application to the Sum of Log-normal Variates," *IEEE Transactions on Information Theory*, vol. 23, no. 2, pp. 275-280, March 1977.
- [17] G. L. Cariolaro, S. G. Pupolin, "Considerations on Error Probability in Correlated-Symbol Systems," *IEEE Transactions on Communications*, vol. 25, no. 4, pp. 462-467, April 1977.
- [18] S. Blinnikov, R. Moessner, "Expansions for Nearly Gaussian Distributions," *Astronomy and Astrophysics Supplement Series*, vol. 130, no. 1, pp. 193-205, May 1998.

- [19] M. A. Najib, V. K. Prabhu, "Analysis of Equal-Gain Diversity with Partially Coherent Fading Signals," *IEEE Transactions on Vehicular Technology*, vol. 49, no. 3, pp. 783-791, May 2000.
- [20] N. C. Beaulieu, "An Infinite Series for the Computation of the Complementary Probability Distribution of a Sum of Independent Random Variables and Its Application to the Sum of Rayleigh Random Variables," *IEEE Transactions on Communications*, vol. 38, no. 9, pp. 1463-1474, September 1990.
- [21] C. Tellambura, A. Annamalai, "Further Results on the Beaulieu Series," *IEEE Transactions on Communications*, vol. 48, no. 11, pp. 1774-1777, November 2000.
- [22] I. T. Monroy, G. Hooghiemstra, "On a Recursive Formula for the Moments of Phase noise," *IEEE Transactions on Communications*, vol. 48, no. 6, pp. 917-920, June 2000.
- [23] M. Kavehrad, M. Joseph, "Maximum Entropy and the Method of moments in Performance Evaluation of Digital Communications Systems," *IEEE Transactions on Communications*, vol. 34, pp. 1183-1189, December 1986.
- [24] T. M. Duman and M. Salehi, "New Performance Bounds for Turbo-Codes," *IEEE Transactions on Communications*, vol. 46, no. 6, pp. 717-723, June 1998.
- [25] I. Sason and S. Shamai, "Improved Upper Bounds on the ML Decoding Error Probability of Parallel and Serial Concatenated Turbo-Codes via their En-

- semble Distance Spectrum,” *IEEE Transactions on Information Theory*, vol. 46, no. 1, pp. 24-47, January 2000.
- [26] D. Divsalar, S. Dolinar, F. Pollara, “Iterative Turbo Decoder Analysis Based on Density Evolution,” *IEEE Journal on Selected Areas in Communications*, vol. 19, no. 5, pp. 891-907, May 2001.
- [27] D. J. Torrieri, “The Information-Bit Error Rate for Block Codes,” *IEEE Transactions on Communications*, vol. 32, no. 4, pp. 474-476, April 1984.
- [28] A. B. Kiely, J. T. Coffey, and M. R. Bell, “Optimal Information Bit Decoding of Linear Block Codes,” *IEEE Transactions on Information Theory*, vol. 41, no. 1, pp. 130-140, January 1995.
- [29] M. P. C. Fossorier, L. Shu, R. Dojun, “Bit Error Probability for Maximum-Likelihood Decoding of Linear Block Codes and Related Soft-Decision Decoding Methods,” *IEEE Transactions on Information Theory*, vol. 44, no. 7, pp. 3083-3090, November 1998.
- [30] H. Yoshikawa, I. Oka, C. Fujiwara, Y. Daido, “Exact Analysis of Bit Error Probability for 4-state Soft Decision Viterbi Decoding,” *IEICE Trans. Fundamentals*, vol. E85-A, pp. 2263-2266, October 2002.
- [31] H. Yoshikawa, “Theoretical Analysis of Bit Error Probability for Maximum a Posteriori Probability Decoding,” *Proceedings of the IEEE International Symposium on Information Theory (ISIT 2003)*, Yokohama, Japan, pp. 276, July 2003.

- [32] M. R. D. Rodrigues, J. E. Mitchell, I. Darwazeh and J. J. O'Reilly, "Error Probability Evaluation with a Limited Number of Moments," *Proceedings of the IEEE International Symposium on Information Theory (ISIT 2003)*, Yokohama, Japan, pp. 9, July 2003.
- [33] B. Aiazzi, L. Alparone and S. Baronti, "Estimation Based on Entropy Matching for Generalized Gaussian *pdf* Modeling," *IEEE Signal Processing Letters*, vol. 6, no. 6, pp. 138-140, June 1999.
- [34] N. C. Beaulieu, "An Investigation of Gaussian Tail and Rayleigh Tail Density Functions for Importance Sampling Digital Communication System Simulation," *IEEE Transactions on Communications*, vol. 38, no. 9, pp. 1288-1292, September 1990.
- [35] K. S. Shanmugam and P. Balaban, "A Modified Monte-Carlo Simulation Technique for the Evaluation of Error Rate in Digital Communication Systems," *IEEE Transactions on Communications*, vol. 28, no. 11, pp. 1916-1924, November 1980.
- [36] M. C. Jeruchim, "On the Application of Importance Sampling to the Simulation of Digital Satellite and Multi-hop Links," *IEEE Transactions on Communications*, vol. 32, no. 10, pp. 1088-1092, October 1984.
- [37] Q. Wang and V. K. Bhargava, "On the Application of Importance Sampling to BER Estimation in the Simulation of Digital Communication Systems," *IEEE Transactions on Communications*, vol. 35, no. 11, pp. 1231-1233, November 1987.

- [38] J. S. Sadowsky, "A New Method for Viterbi Decoder Simulation Using Importance Sampling," *IEEE Transactions on Communications*, vol. 38, no. 9, pp. 1341-1351, September 1990.
- [39] J. C. Chen, D. Lu, J. S. Sadowsky and K. Yao, "On Importance Sampling in Digital Communications. I. Fundamentals," *IEEE Journal on Selected Areas in Communications*, vol. 11, no. 3, pp. 289-299, April 1993.
- [40] J. A. Bucklew and R. Radeke, "On the Monte-Carlo Simulation of Digital Communication Systems in Gaussian noise," *IEEE Transactions on Communications*, vol. 51, no. 2, pp. 267-274, February 2003.
- [41] M. Ferrari and S. Bellini, "Importance Sampling Simulation of Turbo-Product Codes," *Proceedings of IEEE International Conference on Communications (ICC 2001)*, vol. 9, pp. 2773-2777, June 2001.
- [42] H. El-Gamal, A. R. Hammons Jr., "Analyzing the Turbo Decoder Using the Gaussian Approximation", *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 671-686, February 2001.
- [43] A. Abedi, P. Chaudhari, A. K. Khandani, "On Some Properties of Bit Decoding Algorithms," *Proceedings of the Canadian Workshop on Information Theory (CWIT 2001)*, Vancouver, Canada, pp. 106-109, June 2001.
- [44] G. D. Forney Jr., "Geometrically Uniform Codes," *IEEE Transactions on Information Theory*, vol.37, no.5, pp. 1241-1260, September 1991.

- [45] P. Delsarte, P. Piret, "Algebraic Constructions of Shannon Codes for Regular Channels," *IEEE Transactions on Information Theory*, vol.28, no.4, pp. 593-599, July 1982.
- [46] R. G. Gallager, *Information Theory and Reliable Communication*, New York: Wiley, 1968.
- [47] G. D. Forney Jr., M. D. Trott, S. Y. Chung, "Sphere-Bound-Achieving Coset Codes and Multilevel Coset Codes," *IEEE Transactions on Information Theory*, vol.46, no.2, pp. 820-850, May 2000.
- [48] R. S. Freedman, "On Gram-Charlier Approximations," *IEEE Transactions on Communications*, vol. 29, no. 2, pp. 122-125, February 1981.
- [49] M. Abramowitz, *Handbook of Mathematical Functions*, Washington, U.S. Dept. of Commerce, 1958.
- [50] J. F. Kenney, E. S. Keeping, *Mathematics of Statistics*, Part 2, Second edition Princeton, NJ: Van nostrand, 1951.
- [51] A. W. Van der vaart, *Asymptotic Statistics*, Cambridge University press, 1998.
- [52] H. Cramer, *Mathematical Methods of Statistics*, Princeton University press, 1957.
- [53] H. Cramer, *Random Variables and Probability Distributions*, 3rd edition, Cambridge University Press, London, 1970.

- [54] H. Cramer “On Some Classes of Series Used in Mathematical Statistics,” *Proceedings. of the 6th Scandinavian. Congress of Mathematicians*, Copenhagen. pp. 399-425, 1925.
- [55] E. T. Whittaker, G. N. Watson, *A Course of Modern Analysis*, 4th edition, Cambridge University Press, London, 1962.
- [56] G. Szego, *Orthogonal Polynomials*, Providence, R. I. American Mathematical Society, 1967.
- [57] P. V. Oneil, *Advanced Engineering Mathematics*, 4th edition, International Thomson Publishing, 1995.
- [58] I. S. Gradshteyn, I. M. Ryzhik, *Table of Integrals, Series, and Products*, 5th edition, Academic press, 1994.
- [59] A. Abedi, A. K. Khandani, “Some Properties of Bit Decoding Algorithms for Binary Linear Block Codes, ” *IEEE Transactions on Information Theory*, Revised, March 2004.
- [60] E. T. Jaynes, “On The Rationale of Maximum-Entropy Methods,” *Proceedings of the IEEE*, vol. 70, no. 9, pp. 939-952, September 1982 (Invited paper).
- [61] M. Grendar Jr. and M. Grendar, “Maximum Entropy: Clearing up Mysteries,” *Journal of Entropy*, vol. 3, pp. 58-63, 2001.
- [62] P. D. Grunwald and A. P. Dawid, “Game Theory, Maximum Generalized Entropy, Minimum Discrepancy, Robust Bayes and Pythagoras,” *Information Theory Workshop*, Bangalore, India, pp. 94-07, October 2002.

- [63] R. Weinstock, *Calculus of Variations with Applications to Physics and Engineering*, New York: Dover, 1974.
- [64] L. R. Mead and N. Papanicolaou, "Maximum entropy in the Problem of Moments," *Journal of Mathematical Physics*, vol. 25, pp. 2404-2417, August 1984.
- [65] N. Agmon, Y. Alhassid and R. D. Levine, "An Algorithm for Finding the Distribution of Maximal Entropy," *Journal of Comp. Physics*, vol. 30, pp. 250, 1979.
- [66] S. Kullback, *Information Theory and Statistics*, New York: Wiley, 1959.
- [67] A. Abedi, A. K. Khandani, "Some Properties of Bit Decoding Algorithms Over A Generalized Channel Model," *Proceedings of Conference on Information Sciences and Systems (CISS 2002)*, Princeton, USA, pp. 112-117, March 2002.
- [68] A. Abedi, A. K. Khandani, "An Analytical Method for Performance Analysis of Binary Linear Block Codes," *Proceedings of the IEEE International Symposium on Information Theory (ISIT 2002)*, Lausanne, Switzerland, pp. 403, July 2002.
- [69] A. Abedi, A. K. Khandani, "An Analytical Method for Approximate Performance Evaluation of Binary Linear Block Codes," *IEEE Transactions on Communications*, vol. 52, no. 2, pp. 228-235, February 2004.
- [70] A. Abedi, A. K. Khandani, "A New Method for Performance Evaluation of Turbo-Codes Using Statistics of Log Likelihood Ratio," *Submitted to the IEEE Transactions on Communications*, February 2004.

- [71] T. H. Liew, J. Pliquett, B. L. Yeap, L. L. Yang, and L. Hanzo, "Comparative study of space-time block codes and various concatenated turbo coding schemes," *Proceedings of IEEE PIMRC'00*, pp. 741-745, London, UK, September 2000.
- [72] K. Hinton, G. Nicholson, "Probability Density Function for the Phase and Frequency noise in a Semiconductor Laser," *IEEE Journal of Quantum Electronics*, vol. 22, no. 11, pp. 2107-2115, November 1986.
- [73] D. Hatzinakos, C. L. Nikias, "Estimation of Multipath Channel Response in Frequency Selective Channels," *IEEE Journal on selected areas in communications*, vol. 7, no. 1, pp. 12-19, January 1989.
- [74] L. X. Wang, J. M. Mendel, "Cumulant-Based Parameter Estimation Using Structured Networks," *IEEE Transactions on Neural Networks*, vol. 2, no. 1, pp. 73-83, January 1991.
- [75] A. G. Stogioglou, S. McLaughlin, "MA Parameter Estimation and Cumulant Enhancement," *IEEE Transactions on Signal Processing*, vol. 44, no. 7, pp. 1704-1718, July 1996.
- [76] A. K. Nandi, "Robust Estimation of Third-Order Cumulants in Applications of Higher-Order Statistics," *IEE Proceedings-F*, vol. 140, no. 6, pp. 380-389, December 1993.
- [77] B. C. Y. Wong, I. F. Blake, "Detection in Multivariate non-Gaussian noise," *IEEE Transactions on Communications*, vol. 42, no. 2/3/4, pp. 1672-1683, February/March/April 1994.

- [78] Y. Zhao, X. Zhuang, S. J. Ting, "Gaussian Mixture Modeling of non-Gaussian Source for Autoregressive Process," *IEEE Transactions on Signal Processing*, vol. 43, no. 4, pp. 894-903, April 1995.
- [79] J. K. Tugnait, "Blind Equalization and Estimation of Digital Communication FIR Channel Using Cumulant Matching," *IEEE Transactions on Communications*, vol. 43, no. 2/3/4, pp. 1240-1245, February/March/April 1995.
- [80] J. K. Tugnait, U. Gummadavelli, "Blind Equalization and Channel Estimation with Partial Response Input Signals," *IEEE Transactions on Communications*, vol. 45, no. 9, pp. 1025-1031, September 1997.
- [81] L. Chen, H. Kusaka, M. Kominami, "Blind Phase Recovery in QAM Communication System Using Higher Order Statistics," *IEEE Signal Processing Letters*, vol. 3, no. 5, pp. 147-149, May 1996.
- [82] S. Chen, Y. Wu, and S. Mclaughlin, "Genetic Algorithm Optimization for Blind Channel Identification with Higher Order Cumulant Fitting," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 4, pp. 259-265, November 1997.
- [83] M. Martone, "Cumulant-Based Adaptive Multichannel Filtering for Wireless Communication Systems with Multipath RF Propagation Using Antenna Arrays," *IEEE Transactions on Vehicular Technology*, vol. 47, no. 2, pp. 377-391, May 1998.

- [84] A. Abel, W. Schwarz, M. Gotz, “noise Performance of Chaotic Communication Systems,” *IEEE Transactions on Circuits and Systems*, vol. 47, no. 12, pp. 1726-1732, December 2000.
- [85] P. R. Halmos, “The Theory of Unbiased Estimation,” *Ann. Math. Stat.* 17, pp. 34-43, 1946.
- [86] C. Rose, and M. D. Smith, “k-Statistics: Unbiased Estimators of Cumulants,” 7.2C in *Mathematical Statistics with Mathematica*. New York: Springer-Verlag, pp. 256-259, 2002.