Performance Analysis of Linear Block Codes Over the Queue-Based Channel

by

Haider Al-Lawati

A thesis submitted to the

Department of Mathematics and Statistics

in conformity with the requirements for

the degree of Master of Science (Engineering)

Queen's University

Kingston, Ontario, Canada

August 2007

Abstract

Most coding schemes used in today's communication systems are designed for memoryless channels. These codes break down when they are transmitted over channels with memory, which is in fact what real-world channels look like since errors often occur in bursts. Therefore, these systems employ interleaving to spread the errors so that the channel looks more or less memoryless (for the decoder) at the cost of added delay and complexity. In addition, they fail to exploit the memory of the channel which increases the capacity for a wide class of channels. On the other hand, most channels with memory do not have simple and mathematically tractable models, making the design of suitable channel codes more challenging and possibly not practical. Recently, a new model has been proposed known as the queue-based channel (QBC) which is simple enough for mathematical analysis and complex enough for modeling wireless fading channels.

In this work, we examine the performance of linear block codes when transmitted over this channel. We break down our focus into two parts. First, we investigate the maximum likelihood decoding of binary linear block codes over the QBC. Since it is well known that for binary symmetric memoryless channels, maximum likelihood decoding reduces to minimum Hamming distance decoding, our objective here is to explore whether there exists a similar relation between these two decoding schemes when the channel does have memory. We give a partial answer for the case of perfect and quasi perfect codes.

Next, we study Reed-Solomon (RS) codes and analyze their performance when transmitted over the QBC under the assumption of bounded distance decoding. In particular, we examine the two interleaving strategies encountered when dealing with non-binary codes over a binary input channel; namely, symbol interleaving and bit interleaving. We compare these two interleaving schemes analytically and show that symbol interleaving always outperforms bit interleaving. Non-interleaved Reed-Solomon codes are also covered. We derive some useful expressions pertaining to the calculation of the probability of codeword error. The performance of non-interleaved RS codes are compared to that of interleaved ones for the simplest scenario of the QBC which is the additive (first-order) Markov noise channel with non-negative noise correlation.

Acknowledgments

I would like to sincerely express my deep gratitude to my supervisor, Dr. Fady Alajaji, for accepting to supervise my research. I am greatly indebted to him for his invaluable guidance, continuous support and significant contribution throughout this work. I am filled with gratitude for his precious advice that has not only improved the quality of this work, but also proved to be of great value for my future endeavours. His warm-hearted coaching and dedicated work ethic will continue to be a source of inspiration for me.

I would also like to acknowledge Dr. Cecilio Pimentel for his aid when running some of the more difficult simulations needed in this work. His insightful comments were valuable and have enhanced parts of this thesis.

Special thanks goes to Queen's University at Kingston and the Ministry of Higher Education in Oman for their financial support during my studies.

I am also very grateful to my dear friends for their nonstop support and encouragement throughout my studies. I feel fortunate and extremely pleased to be surrounded by such loyal and trustful companions. They have made my experience at Queen's unforgettable.

Last, but not least, I would like to thank my parents as well as my wife for their everlasting love, care and patience.

Table of Contents

Chapter 5:

Chapter 6:

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Preliminaries

The art of designing good error-correcting codes for reliably transmitting data over a noisy environment has become an important component in designing any communication system ever since Shannon's landmark work in the past century [35]. Hamming and Slepian pioneered this field in the early 1950s by designing good codes along with efficient decoding techniques. Further research later on led to the construction of new codes that have rich algebraic structures with more efficient decoding techniques. Today's communication technologies utilize different coding schemes depending on their applications. The global system for mobile (GSM) communication which is one of the popular standards for mobile cellular phones, for instance, employs convolutional codes with Viterbi algorithm at the decoder. More recent standards that are extensions of the GSM such as the GPRS[1], EDGE[2] and UMTS[3] also implement different convolutional and turbo coding schemes [5]. Data storage applications such as compact discs (CD) and digital versatile discs (DVD), on the other hand, use cross-interleaved Reed-Solomon codes that offer a great error correction capability especially for bursty channels. Due to the high demand for wireless digital communication, the design of better coding schemes that are power efficient and can operate at high data rates while keeping the decoding error and decoding complexity as low as possible is an active area of research.

1.2 Problem Description

In order to design an efficient coding scheme, a comprehensive understanding of the nature of the medium through which the information is transmitted is necessary. Such a medium is called the communication channel. Various mathematical models have been developed to describe physical communication channels. A classical example of communication channel models is the additive white Gaussian noise (AWGN) channel, which is extensively studied in the literature. The information transmitted over such

[1] General packet radio service

[2] Enhanced data GSM environment

[3] Universal mobile telecommunications service

a channel is represented (via modulation) as continuous-time signals. Let $x(t)$ denote the information-bearing signal sent by the transmitter over the AWGN channel; then the decoder will receive a signal $y(t)$, given by

$$y(t) = x(t) + n(t)$$

where $n(t)$ is a Gaussian process corresponding to the distortion introduced by the channel. The channel noise is uncorrelated, which makes the AWGN model a simple one from an analysis point of view.

Wireless channels are not as simple as the AWGN channel. When signals travel over wireless channels, they may experience attenuation, superposition, delay, etc., due to the time varying nature of the wireless medium. Because of such distortions introduced by the channel, the received signal is said to be faded. As a result, better and more complex models are needed to account for this fading behavior. Examples of fading channels include the Rayleigh and Rician fading channels. Fading channel noise is multiplicative rather than additive. For instance, suppose that a signal $x(t)$ is transmitted over a fading channel. Then at the receiver the faded signal is given by $y(t) = h(t)x(t)$ where $h(t)$ is a multiplicative noise process. To better model the wireless channel, it is natural to assume the presence of the additive white Gaussian noise, due to electronic circuits noise and thermal noise. Therefore, $y(t)$ can be written

$$y(t) = h(t)x(t) + n(t).$$

The above channel models are known as continuous-time channels. The transmitted signal, the received signal and the noise process are all continuous functions of time. However, in digital communications, information signals produced from a source are usually quantized (i.e., transformed from continuous-time to discrete-time with finite alphabet) and then encoded via a channel encoder (see Chapter 3). The encoded information signals, known as codewords, are then modulated and transformed into continuous-time signals and then transmitted. At the decoder, the received signal is demodulated and transformed to a discrete-time signal and then decoded. The overall channel including the continuous-time channel along with the modulator and demodulator can be viewed as a discrete-time channel since its input and output are discrete (in time and amplitude). For example, the binary symmetric channel (BSC) (see Chapter 2) which is a discrete-time channel is equivalent to the AWGN channel with a binary phase shift keying modulation and a hard-decision demodulation at the decoder. The family of discrete channels is interesting for the purpose of mathematical and statistical analysis. More details on discrete channels is given in the next chapter. Given a channel model, the task of coding theorists becomes designing error-correcting codes that are capable of reliably communicating data in the sense of minimizing the probability of decoding error while operating at as high rate as possible. The existence of such codes is guaranteed by Shannon's channel coding theorem as long as the rate is kept below the channel capacity. Another fundamental element considered in such designs in today's communication systems is the issue of complexity and delay.

1.3 Literature Review

Conventional communication systems employ coding schemes that are designed for memoryless channels, such as the BSC. However, since most real world channels have statistical memory, interleaving is used in an attempt to spread the channel noise in a uniform fashion over the set of received words so that the channel appears memoryless to the decoder. This in fact adds more complexity and delay to the system, while failing to exploit the benefits of the channel memory [11].

Progress has been achieved on the statistical and information theoretic modeling of channels with memory (e.g., see [1], [23], [30], [33], [48]), as well as on the design of effective iterative decoders for such channels (e.g., see [12], [14], [24], [25]). Channel noise models based on Markov processes have been shown to reliably fit or approximate different types of fading channels. In [1], the authors describe a channel with memory in which the noise propagates in the channel in a similar fashion to the spread of a contagious disease through a population using the Polya urn model. A resultant channel model is called the finite-memory contagion channel (FMCC). The Gilbert-Elliott channel (GEC) [23] is a discrete channel model characterized by an underlying hidden Markov model (HMM). It is used in [33] to model the correlated Rician fading channel. Furthermore, Fritchman's Markov model is also used in [30] to describe channels of bursty nature. In [46], [49], the authors propose a queue-based channel (QBC) to model a discrete binary channel with an M^{th} order Markov noise. This is in fact a generalization of the FMCC. The advantages of the QBC are that it has a fixed number of parameters along with a closed form expression for the block transition probability, the channel capacity and the autocorrelation function. Moreover, it is shown that the QBC can be a better model to the Rician slow fading channel than the GEC [48], [50].

The power of binary linear codes with rich algebraic structures over memoryless channels has been demonstrated since the birth of algebraic coding. For instance, binary perfect and quasi-perfect codes have been shown to achieve the smallest probability of codeword error (PCE) amongst all codes with the same blocklength and rate. On the other hand, little is known about the performance of such codes over channels with memory. In [17], Hamada investigates whether algebraic codes can perform as well and thus give a good solution to the coding problem when the channel is not memoryless. He studies the optimality of Hamming codes and cyclic subcodes of the Hamming codes over the binary additive Markov noise channel (BAMNC) under maximum-likelihood decoding. He first defines the block transition probability in terms of a generalized weight function that reduces to the Hamming weight when the noise correlation coefficient is set to zero. The generalized weight is a decreasing function of the block transition probability under the assumption that the all-zero noise n-tuple is generated by the channel with a non-zero probability. Given an $[n, k]$ linear block code (where n is the code's blocklength and 2^k is the codes's size) and using this generalized weight, a set containing 2^{n-k} n-tuples is constructed such that any element in the set has a weight less than or equal to the weight of any element outside the set. Such a set is called an ideal decoding set (under maximum-likelihood decoding) which may not be unique. A decoding set refers to the set of all binary n-tuples the decoder may subtract from the received words when decoding them in its attempt to recover the original transmitted codewords. The set of all coset leaders is an example of a decoding set and is an ideal set when the channel is a BSC (with cross-over probability less than $\frac{1}{2}$). An optimal code then will have a decoding set identical to the ideal decoding set. The author showed that the family of binary Hamming codes are optimal over the BAMNC (with non-negative noise correlation) amongst all the codes of the same length and rate. He also showed near-optimality (ε-optimality) of even-weight subcodes of the cyclic Hamming codes. That is, the ratio of the probability of correct decoding for these subcodes to that of the optimal code is larger than or equal to $1 - \varepsilon$, for very low values of ε.

Contrary to the case of binary algebraic codes, non-binary codes have received more attention when dealing with channels with memory. A famous example of this family of codes is the class of Reed-Solomon (RS) codes which will be discussed in more details in Chapter 3. The performance of non-interleaved RS codes over correlated fading channels is analyzed in [7], [19], [20], [31] using a two step procedure. First, a channel model is introduced for the generation of the bit or symbol error process, and then a formula for the PCE under bounded distance decoding is derived for the proposed model. In [7] an L-state Markov chain is proposed to characterize the correlation of symbol errors. In [19], the channel is modeled via the GEC whose parameters are calculated using a simple threshold model; i.e., the channel is in the bad state whenever the instantaneous signal to noise ratio is below a given threshold. An approximation to the PCE is derived under the assumption that the channel state does not change during each symbol transmission. In [20], level crossing statistics are applied to characterize the fading arrival process and the fading durations, and the PCE is expressed in terms of the probability distribution of the fading durations. In [31], the bit error process resulting from the hard-decision demodulation of binary frequency-shift keying modulated signals over correlated Rician fading channels is modeled via a Fritchman channel. Furthermore, an analytical method based on the generating series approach for calculating the PCE of RS codes over finite state channels is presented. Imperfect (finite-length) symbol interleaving is also considered in [20], [29], [31].

1.4 Thesis Contribution

In this thesis, we study the performance of both binary and non-binary block codes over the QBC. First, we study the maximum-likelihood (ML) decoding of binary block codes when transmitted over the QBC. Since it is well known that over the memoryless

BSCs, ML decoding reduces to minimum distance (MD) decoding, our objective is to investigate whether a similar relation exists between these two decoding techniques when the channel is not memoryless. We first assume that the channel memory is longer than the blocklength of the codeword and establish a relation between ML and MD decoding. Next, we consider a special case of the QBC which is the BAMNC with non-negative noise correlation. We study both perfect and quasi-perfect codes when operated over this channel. Similar to the previous case, we show a relation between the ML and MD decoders for the case of perfect codes. For quasi-perfect codes, a new decoding technique is proposed based on MD decoding that is near-optimal for a range of channel parameters. The contributions of this part of the thesis (which in part appeared in [3]) are as follows.

- Proving that for binary block codes, ML decoding is equivalent to either MD decoding or maximum distance decoding when the channel memory is longer than the codeword blocklength.

- Proving that the all-zero n-tuple is the most likely noise output generated by the BAMNC.

- Proving some important properties of the block transition probability of the BAMNC.

- Deriving sufficient conditions under which ML decoding reduces to MD decoding for binary perfect codes sent over the BAMNC.

- Proposing a new decoding algorithm for binary quasi-perfect codes sent over the BAMNC that is nearly equivalent to ML decoding over a range of channel parameters.

Next, we investigate the performance of non-binary codes over the QBC. In particular, we focus on the performance of RS codes especially when the QBC is reduced to the BAMNC with non-negative noise correlation. Two important interleaving strategies are worth studying when dealing with such codes: interleaving the code (or channel) bits which reduces the channel to the memoryless BSC (under perfect or infinite interleaving depth) and interleaving the code symbols. In [41], the author observes experimentally that RS codes sent over Rayleigh fading channels perform better under symbol interleaving than under bit interleaving. In this work, we provide a proof that symbol interleaving is better than bit interleaving for all non-binary linear block codes when transmitted over the QBC. Moreover, we analyze the performance of RS codes when transmitted over the BAMNC with non-negative noise correlation under bounded distance decoding. We calculate the PCE after establishing a recursive expression for the probability of m symbol errors in a block of length n using the generating series method [15], [28] which provides a powerful combinatorial approach. Furthermore, we study four typical RS codes under both interleaving strategies as well as when they are not interleaved when sent over the BAMNC with non-negative correlation. The purpose of this study is to define the range of channel conditions under which interleaving either degrades the performance or does not provide a significant gain compared to the case when no interleaving is used. The contributions of this part of the thesis (which in part appeared in [4]) are as follows.

- Proving that symbol interleaving is better than bit interleaving for non-binary block codes over the QBC.
- Deriving a recursive expression for the probability of m symbol errors in a block of n symbols when transmitting over the BAMNC.
- Numerically studying the performance of four typical RS codes over the BAMNC and identifying the range of channel parameters for which interleaving can be avoided.

1.5 Thesis Overview

The organization of this thesis is as follows. In Chapter 2, a brief introduction to communication channel models is given. We begin this chapter by introducing Markov chains. Next we introduce some important definitions related to channel coding before stating Shannon's coding theorem. After that, we describe some discrete binary channel models along with their properties.

In Chapter 3, we introduce both binary and non-binary linear block codes. We start with some general notations and terminologies before introducing binary codes such as Hamming codes, extended Hamming codes, and the BCH codes. Next, syndrome decoding for these codes is explained. Finally, RS codes are introduced along with Berlekamp's iterative decoding algorithm and Forney's algorithm that are used to decode the RS codes.

In Chapter 4, we study the ML decoding of binary linear block codes over the QBC. First, the case of $M > n$ (where M is the length of the queue, while n is the length of the codeword) is studied. Next we consider the case $M \leq n$. For this case, we restrict our study to $M = 1$ resulting in the BAMNC with non-negative noise correlation. We derive some important and useful results pertaining to binary perfect and quasi-perfect codes. We also simulate the performance of another binary linear block code that is neither perfect nor quasi-perfect.

In Chapter 5, the performance analysis of non-binary block codes in general and RS codes in particular is investigated. Interleaved RS codes at both the symbol and bit levels as well as non-interleaved RS codes are analyzed. For the special case of BAMNC with non-negative noise correlation, we derive an expression to calculate the PCE using the generating series approach. Furthermore, four different high and low rate RS codes are studied and channel parameters are identified for which these codes can perform better without interleaving.

Finally, in Chapter 6, we summarize our work and discuss some possible future work that can be built upon our work presented in this thesis.

Chapter 2

Communication Channel Models

The birth of information theory and channel coding is linked to the work pioneered by C. Shannon in 1948. In his outstanding contribution [35], Shannon defined two important quantities in information theory. Namely, the source entropy, and the channel capacity. The former is a measure of the source's uncertainty, and the latter is the maximum rate at which information can be transmitted with arbitrary small probability of decoding error. As a result of Shannon's work, the communication system can be separated into two main portions: *source coding* and *channel coding* as shown in Fig. 2.1.

Definition 2.1 *Let X be a discrete random variable (representing a discrete memoryless or independent and identically distributed (i.i.d) source) with probability mass* *function* $p(x)$*. The entropy* $H(X)$ *is defined by*

$$H(X) = -\sum_{x\in\mathcal{X}} p(x)\log_2 p(x). \tag{2.1}$$

Shannon also defined another quantity called the *mutual information* between two random variables X and Y, denoted I(X;Y). The mutual information is a measure of the amount of information that one random variable contains about another one. In other words, it tells how much uncertainty about one random variable is reduced due to the knowledge of the other.

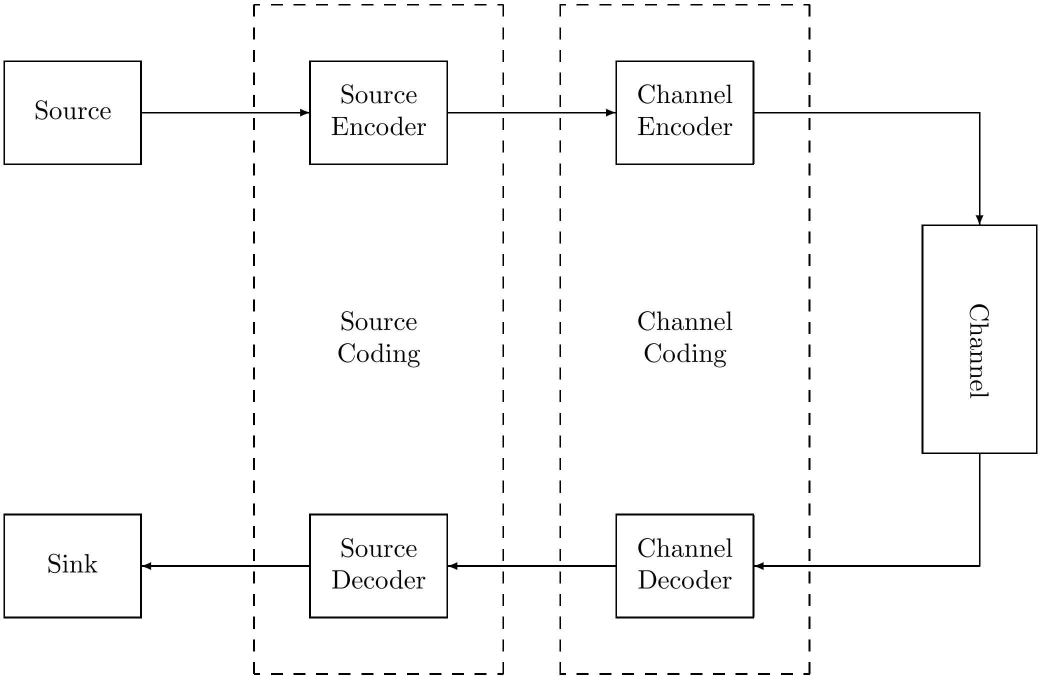


Figure 2.1: Communication system model.

Let X and Y be two random variables with joint probability mass function $p(x, y)$ and marginal probability mass functions $p(x)$ and $p(y)$, then $I(X; Y)$ is given by

$$I(X;Y) = \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(x,y) \log_2 \frac{p(x,y)}{p(x)p(y)}. \tag{2.2}$$

2.1 Discrete Markov Chains

Markov chains are involved in modeling wireless communication fading channels (e.g., see [6], [30], [33], [36] – [40], [43] – [48]). A brief introduction to *discrete Markov chains* is given in this section.

Definition 2.2 *A discrete process $\{Z_1, Z_2, \cdots\}$ with finite-alphabet $\mathcal{A}$ is said to be a* Markov chain *or* Markov process *if for $n = 1, 2, \cdots$*

$$\Pr(Z_n = z_n | Z_{n-1} = z_{n-1}, Z_{n-2} = z_{n-2}, \cdots, Z_1 = z_1) = \Pr(Z_n = z_n | Z_{n-1} = z_{n-1})$$

for all $z_1, \cdots, z_n \in \mathcal{A}$. In this case,

$$\Pr(Z^n = z^n) = \Pr(Z_1 = z_1) \prod_{i=2}^{n} \Pr(Z_i = z_i | Z_{i-1} = z_{i-1}),$$

where $z^n \overset{\triangle}{=} (z_1, z_2, \cdots, z_n)$.

Furthermore, a process is a Markov process of order *M (or memory M), where $M > 0$ is fixed, if*

$$\Pr(Z_n = z_n | Z_{n-1} = z_{n-1}, Z_{n-2} = z_{n-2}, \cdots, Z_1 = z_1)$$
$$= \Pr(Z_n = z_n | Z_{n-1} = z_{n-1}, Z_{n-2} = z_{n-2}, \cdots, Z_{n-M} = z_{n-M})$$

for $n > M$ and for all $z_1, \cdots, z_n \in \mathcal{A}$.

Proposition 2.1 *Define the process $\{\underline{S_n}\}_{n=1}^{\infty}$ by*

$$\underline{S}_n \triangleq (Z_n, Z_{n+1}, \cdots, Z_{n+M-1}).$$

Then $\{\underline{S}_n\}$ is a Markov process with $|\mathcal{A}|^M$ states.

Proof.

$$\begin{aligned} & \Pr(\underline{S}_n = \underline{s}_n | \underline{S}_{n-1} = \underline{s}_{n-1}, \cdots, \underline{S}_1 = \underline{s}_1) \\ = \; & \Pr(Z_{n+M-1} = z_{n+M-1} | Z_{n+M-2} = z_{n+M-2}, \cdots, Z_{n-1} = z_{n-1}) \\ = \; & \Pr(\underline{S}_n = \underline{s}_n | \underline{S}_{n-1} = \underline{s}_{n-1}). \end{aligned}$$

□

If the conditional probability of a Markov process, Z_n, does not depend on n, which means that the conditional probability satisfies

$$\Pr(Z_n = a | Z_{n-1} = b) = \Pr(Z_2 = a | Z_1 = b),$$

for $n > 1$ and for all $a, b \in \mathcal{A}$, then the process is said to be *time-invariant* or *homogeneous.*

Definition 2.3 *A stochastic process $\{Z_n\}_{n=1}^{\infty}$ with finite-alphabet $\mathcal{A}$ is said to be* stationary *if the joint distribution of any subset of the sequence of random variables is invariant with respect to time shifts; i.e.,*

$$\Pr(Z_{n_1} = z_{n_1}, Z_{n_2} = z_{n_2}, \cdots, Z_n = z_{n_k}) = \Pr(Z_{n_1+\tau} = z_1, Z_{n_2+\tau} = z_2, \cdots, Z_{n_k+\tau} = z_n)$$

for every time shift τ and for all $z_1, \cdots, z_n \in \mathcal{A}$ and all $k \geq 1$.

Definition 2.4 *Let $\{Z_n\}_{n=1}^{\infty}$ be a Markov process. The matrix $\boldsymbol{P}$ whose $(i,j)^{th}$ entry p_{ij} is the conditional probability of a transition from state i to state j,*

$$p_{ij} \stackrel{\triangle}{=} \Pr(Z_n = j | Z_{n-1} = i).$$

is called the probability transition matrix *of the process. A homogeneous Markov process is characterized by the distribution of Z_1 and its probability transition matrix, $\boldsymbol{P}$.*

A Markov process is said to be *irreducible* or *ergodic* if the process can reach any state from any other state in a finite number of steps with a non-zero probability [26].

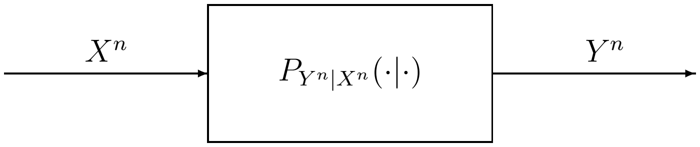
Definition 2.5 *For a Markov process $\{Z_n\}$, a distribution on the states such that the distribution at time $n+1$ is the same as the distribution at time n is called a* stationary distribution *and is denoted by $\boldsymbol{\pi} \triangleq (\pi_0; \pi_1; \cdots; \pi_{|\mathcal{A}|-1})$.*

Remark: The stationary distribution $\boldsymbol{\pi}$ always exists for any finite-alphabet homogeneous Markov process. In fact, if $\boldsymbol{P} \stackrel{\triangle}{=} [p_{ij}]$ is the probability transition matrix of the Markov process, then $\boldsymbol{\pi}$ can be obtained by solving $\boldsymbol{\pi} = \boldsymbol{\pi}\boldsymbol{P}$. In addition, if the initial state of a homogeneous Markov process is drawn according to the stationary distribution $\boldsymbol{\pi}$, then the Markov process is a stationary process.

2.2 The Channel Coding Theorem

The channel coding theorem is one of the main contributions Shannon has made to communication theory. Contrary to what was generally accepted before 1948, Shannon proved that increasing transmission rate does not increase the error rate as long as the transmission rate is kept below a certain threshold he called the capacity of the channel.

Definition 2.6 [10] *A discrete channel* $(\mathcal{X}^n, p(y^n \mid x^n), \mathcal{Y}^n)$ *consists of two finite sets: an input set* $\mathcal{X}$ *and an output set* $\mathcal{Y}$*, and a sequence of n-dimensional conditional distributions,* $\{p(y^n \mid x^n)\}_{n=1}^{\infty}$*, where* $p(y^n \mid x^n) \triangleq \Pr(Y^n = y^n \mid X^n = x^n) \geq 0$ $\forall x^n \triangleq (x_1, \cdots, x_n) \in \mathcal{X}^n$*,* $y^n \triangleq (y_1, \cdots, y_n) \in \mathcal{Y}^n$*, and* $\sum_{y^n \in \mathcal{Y}^n} p(y^n \mid x^n) = 1$ $\forall x^n \in \mathcal{X}^n$*.*



Definition 2.7 *Discrete Memoryless Channel* (DMC)*: The DMC is a channel with the property*

$$\Pr(Y^n = y^n | X^n = x^n) = \prod_{i=1}^{n} \Pr(Y_i = y_i | X_i = x_i).$$

A DMC is totally described by $p(y \mid x) \triangleq \Pr(Y = y \mid X = x)$, where $x \in \mathcal{X}$ and $y \in \mathcal{Y}$

Definition 2.8 *A discrete-time binary additive-noise channel is a communication channel with common input, noise and output alphabets* (i.e., $\mathcal{X} = \mathcal{Z} = \mathcal{Y} = \{0, 1\}$) *in which, the n^{th} channel output Y_n is given by $Y_n = X_n \oplus Z_n$, where $\oplus$ is modulo-2 addition, and X_n and Z_n are the channel input and noise, respectively. Furthermore, the input and noise processes — $\{X_n\}_{n=1}^{\infty}$ and $\{Z_n\}_{n=1}^{\infty}$, respectively — are independent from each other.*

The noise process $\{Z_i\}_{i=0}^{\infty}$ is generated according to a certain probabilistic model. An error occurs at time t if and only if $Z_t = 1$.

Definition 2.9 **[10]** *An (n, M) code for the channel $(\mathcal{X}^n, p(y^n \mid x^n), \mathcal{Y}^n)$ consists of:*

- *An index set $\{1, 2, \cdots, M\}$.*
- *An encoding (injective) function f*

  $$f : \{1, 2, \cdots, M\} \to \mathcal{X}^n$$

  resulting in a codebook, C, which is the set of codewords:

  $$C = \{f(1), f(2), \cdots, f(M)\}.$$

- *A decoding function g*

  $$g : \mathcal{Y}^n \to \{1, 2, \cdots, M\}$$

  which is a deterministic rule that assigns a guess to each possible received word.

An important measure for evaluating the performance of a given communication system (i.e., encoder and decoder) over a certain channel model is the probability of error. Obviously, a system achieving the smallest possible probability of error is always desirable.

Definition 2.10 (Probability of Error)*: Let*

$$\lambda_i = \Pr(g(Y^n) \neq i | X^n = f(i)) = \sum_{y^n \in \mathcal{Y}^n} p(y^n | f(i)) I(g(y^n) \neq i),$$

where $I(\cdot)$ is the indicator function. λ_i is called the conditional probability of error given that index i was sent.

Definition 2.11 *The maximal probability of error for an (n, M) code denoted by $\lambda^{(n)}$ is defined as*

$$\lambda^{(n)} = \max_{i \in \{1,2,\cdots,M\}} \lambda_i. \tag{2.3}$$

Definition 2.12 *The (arithmetic) average probability of error, denoted $P_e^{(n)}$, for an (n, M) code is defined as*

$$P_e^{(n)} = \frac{1}{M} \sum_{i=1}^{M} \lambda_i. \tag{2.4}$$

Definition 2.13 *The rate R of an (n, M) code is defined as*

$$R = \frac{\log_2 M}{n} \text{ bits per channel transmission or symbol.} \tag{2.5}$$

Definition 2.14 *A rate R is said to be achievable if there exists a sequence of $(n, \lceil 2^{nR} \rceil)$ codes such that $\lim_{n\to\infty} \lambda^{(n)} = 0$.*

Definition 2.15 *The capacity of a channel denoted by C, is defined as the supremum of all achievable rates. In other words, for every rate $R < C$, there exists a sequence of $(n, \lceil 2^{nR} \rceil)$ codes with maximum probability of error $\lambda^{(n)} \to 0$ as $n \to \infty$; also, any sequence of $(n, \lceil 2^{nR} \rceil)$ codes with $\lambda^{(n)} \to 0$ as $n \to \infty$ must have $R \leq C$.*

Theorem 2.1 [10] *For a DMC,*

$$C = \max_{p(x)} I(X^n; Y^n), \tag{2.6}$$

where the maximization is taken over all input distributions $p(x)$.

Theorem 2.2 [13, 16] *For a discrete-time binary channel with stationary ergodic noise,*

$$\begin{aligned} C &= \lim_{n\to\infty} \max_{p(x)} I(X^n; Y^n) \\ &= 1 - H(\mathcal{Z}) \end{aligned}$$

where $H(\mathcal{Z}) \triangleq \lim_{n\to\infty} \frac{1}{n} H(Z^n)$ is the noise entropy rate. In particular, if the noise process is homogeneous Markovian of order M, then its entropy rate reduces to the following conditional entropy

$$\begin{aligned} H(\mathcal{Z}) &= H(Z_{M+1} \mid Z_M, \cdots, Z_1) \\ &\triangleq -\sum_{z^{M+1}} p(z^{M+1}) \log_2 p(z_{M+1}|z^M) \end{aligned}$$

which can be readily calculated using the Markov chain's stationary and transition distributions.

2.3 The Binary Symmetric Channel (BSC)

The binary symmetric channel is an additive noise discrete channel. The noise process $\{Z_i\}_{i=1}^{\infty}$ is binary memoryless (i.i.d) and the channel's input/output transition distribution can be represented as shown in Fig. 2.2. The channel is completely determined by its crossover probability $p \triangleq P(Z_n = 1)$ and is hence denoted by *BSC(p)*.

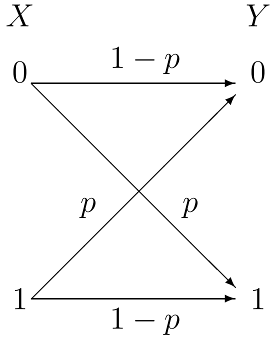


Figure 2.2: The binary symmetric channel with cross over probability p (BSC(p)).

The capacity of this channel is

$$\begin{aligned} C &= \max_{p(x)} I(X;Y) \\ &= 1 - h_b(p) \end{aligned} \tag{2.7}$$

where $h_b(x) \triangleq -x \log_2 x - (1-x)\log_2(1-x)$ is the binary entropy function.

2.4 The Gilbert-Elliott Channel (GEC)

The Gilbert-Elliott channel (GEC) [23] is another binary additive noise channel. It is one of the simplest models widely used in the literature to model channels with

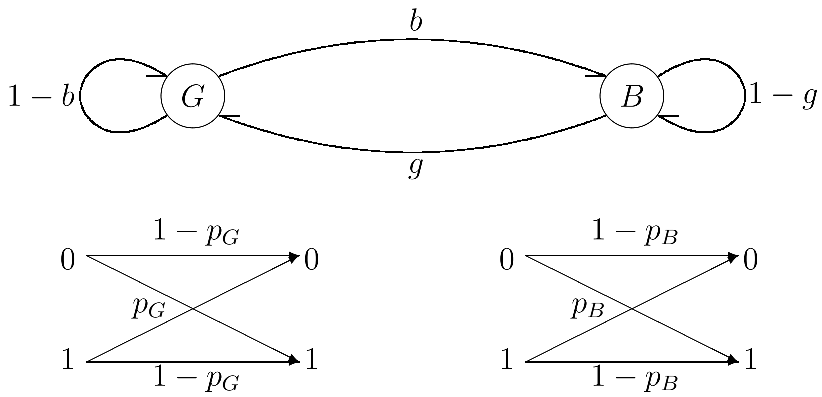


Figure 2.3: The Gilbert-Elliott channel model.

memory. The simplicity of this model lies in the fact that it is driven by a state process with two values: the *good* state, denoted by G or 0, and the *bad* state, denoted by B or 1, where each state corresponds to a BSC with a certain crossover probability, as shown in Fig. 2.3. The transition between the states takes place according to a Markov process. This type of channel model is known as a *hidden Markov model* (HMM) since its noise process is itself a *hidden* Markov process. The probability transition matrix for the Markov state process is

$$\boldsymbol{P} = \begin{bmatrix} 1-b & b \\ g & 1-g \end{bmatrix} \tag{2.8}$$

where $0 < b < 1$ and $0 < g < 1$.

Let S_k and Z_k be the state of the channel and the noise output, respectively, at time k. Define the matrix $\boldsymbol{P}(z_k)$, whose ij^{th} entry is given by $\Pr(Z_k = z_k, S_k = j | S_{k-1} = i)$, $i, j = 0, 1$. Then, $\boldsymbol{P}(0)$ and $\boldsymbol{P}(1)$ for the GEC are

$$\boldsymbol{P}(0) = \begin{bmatrix} (1-b)(1-p_G) & b(1-p_B) \\ g(1-p_G) & (1-g)(1-p_B) \end{bmatrix}, \tag{2.9}$$

$$\boldsymbol{P}(1) = \begin{bmatrix} (1-b)p_G & bp_B \\ gp_G & (1-g)p_B \end{bmatrix}. \tag{2.10}$$

Let $z^n = (z_1, z_2, \cdots, z_n)$ be an n-tuple noise vector; then

$$\Pr(Z^n = z^n) = \boldsymbol{\pi}^T \left(\prod_{i=1}^{n} \boldsymbol{P}(z_i) \right) \mathbf{1}, \tag{2.11}$$

where $\mathbf{1}$ is the all-ones column vector, and $\boldsymbol{\pi}$ is the state stationary distribution given by

$$\boldsymbol{\pi} = \begin{bmatrix} \pi_0 \\ \pi_1 \end{bmatrix} = \begin{bmatrix} \frac{g}{b+g} \\ \frac{b}{b+g} \end{bmatrix}. \tag{2.12}$$

2.5 Finite Memory Contagion Channel (FMCC)

The finite memory contagion channel (FMCC) is a binary additive M^{th} order Markov noise channel. This model was first presented in [1, Section VI] to describe a channel in which its errors spread in a similar fashion to the spread of disease through a population. The noise process $\{Z_i\}_{i=0}^{\infty}$ is generated according to the Polya contagion urn model with a slight modification in the following way. An urn contains originally R red and S black balls with a total of $T = R + S$ balls. Let $\rho = R/T$ and $\sigma =$

$1-\rho = S/T$. At the i^{th} draw ($i = 1, 2, \cdots$), a ball is drawn at a random from the urn and replaced with $1+\Delta$ balls of the same color, where $\Delta > 0$. At the $(i+M)^{th}$ draw, these Δ balls are removed from the urn. Then the i^{th} noise output Z_i is given by

$$Z_i = \begin{cases} 1 & \text{if the } i^{th} \text{ ball drawn is red,} \\ \\ 0 & \text{if the } i^{th} \text{ ball drawn is black.} \end{cases}$$

In such a scheme, the Δ balls added at each time instant affect only M future draws; this results in a finite-memory system. It can be shown [1, Section VI] that the resulting noise process $\{Z_i\}_{i=1}^{\infty}$ is a stationary ergodic M^{th} order Markov process.

2.5.1 Channel Properties

The block transition probability $\Pr(Y^n = y^n | X^n = x^n) = \Pr^{(M)}(Z^n = z^n)$, where $z_i = y_i \oplus x_i$, $i = 1, \cdots n$, is as follows.

- For blocklength $n \le M$,

  $$\Pr{}^{(M)}(Z^n = z^n) = \frac{\rho(\rho+\delta)\cdots[\rho+(d-1)\delta]\sigma(\sigma+\delta)\cdots[\sigma+(n-d-1)\delta]}{(1+\delta)(1+2\delta)\cdots[1+(n-1)\delta]},$$

  where $\delta \triangleq \frac{\Delta}{T}$, d is the Hamming distance between y^n and x^n (i.e., $d = d_H(y^n, x^n) = w_H(z^n = y^n \oplus x^n)$, and $w_H(a^n)$ denotes the Hamming weight of the tuple a^n (i.e., the number of "ones" in a^n)).

- For blocklength $n > M$,

$$\Pr^{(M)}(Z^n = z^n) = L \prod_{i=M+1}^{n} \left[\frac{\rho + \lambda_{i-1}\delta}{1+M\delta}\right]^{z_i} \left[\frac{\sigma + (M - \lambda_{i-1})\delta}{1+M\delta}\right]^{1-z_i},$$

where

$$L = \frac{\prod_{j=0}^{\lambda_M - 1}(\rho + j\delta) \prod_{j=0}^{M-1-\lambda_M}(\sigma + j\delta)}{\prod_{j=1}^{M-1}(1+j\delta)},$$

$\prod_{j=0}^{a}(\cdot) \overset{\triangle}{=} 1$, if $a < 0$, $z_i = x_i \oplus y_i$, and $\lambda_{i-1} = z_{i-1} + \cdots + z_{i-M}$ for $i = M+1, \cdots, n$.

Proposition 2.2 [**1**] *The capacity of the above channel is given by*

$$\begin{aligned} C_{FMCC}^{(M)} &= 1 - H(Z_{M+1} \mid Z_M, \cdots, Z_1) \\ &= 1 - \sum_{k=0}^{M} \binom{M}{k} L_k h_b\left(\frac{\rho + k\delta}{1+M\delta}\right), \end{aligned}$$

where

$$L_k = \frac{\prod_{j=0}^{k-1}(\rho + j\delta) \prod_{j=0}^{M-1-k}(\sigma + j\delta)}{\prod_{j=1}^{M-1}(1+j\delta)},$$

$\rho = R/T$ is the channel BER, and $\delta = \Delta/T$ is a correlation parameter. The correlation coefficient of the noise process is

$$Cor \triangleq \frac{E[Z_i Z_{i+1}] - E[Z_i]^2}{E[Z_i^2] - E[Z_i]^2} = \frac{\delta}{\delta + 1} \geq 0. \qquad (2.13)$$

2.6 Queue-Based Channel (QBC)

The queue-based channel (QBC) is a generalization of the FMCC described in the previous section. It was introduced by Zhong, Alajaji and Takahara in [46], [49] to model an M^{th}-order additive Markov noise channel using a finite queue. The noise process $\{Z_i\}_{i=1}^{\infty}$ is drawn according to the following scheme. Suppose that there are two parcels:

- **Parcel 1** is a queue of length M (see Fig. 2.4), that contains initially M balls, either red or black.

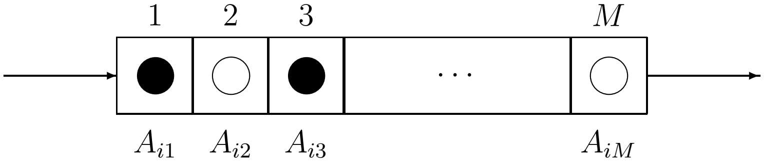


Figure 2.4: A queue of length M.

  Define the random variables A_{ik} where $i \geq 1$ is the i^{th} experiment and $1 \leq k \leq M$ as:

$$A_{ik} = \begin{cases} 1 & \text{if the } k\text{th cell contains a red ball} \\ 0 & \text{if the } k\text{th cell contains a black ball.} \end{cases}$$

- **Parcel 2** is an urn that contains a very large number of balls either red or black such that the proportion of the black balls is $1 - p$ and the proportion of red balls is p, where $p \in (0, 1)$, $p < 1/2$.

At the i^{th} experiment a biased coin with $\Pr(Head) = \varepsilon$, where $\varepsilon \in [0, 1)$, is tossed. If head occurs, we select the queue (Parcel 1), otherwise Parcel 2 is selected. If the queue is of a length $M \geq 2$ and is selected, then a pointer points at the k^{th} cell with a probability $\frac{1}{M-1+\alpha}$, for some constant $\alpha \geq 0$ if $1 \leq k \leq M - 1$, and with a probability

$\frac{\alpha}{M-1+\alpha}$ if $k = M$. Whereas, if the queue contains only one cell (i.e., $M = 1$), then the pointer selects it with probability 1. On the other hand, if Parcel 2 is selected, then a ball is drawn at random from the urn. Based on the color of the ball selected by the pointer or drawn from the urn, we insert a ball of the same color in cell 1 of the queue, shifting its content to the right and forcing the rightmost ball (in cell M) out of the queue. Hence, the queue, after the first M trials, will always contain the balls selected in the last M experiments. The i^{th} noise output is given by

$$Z_i = \begin{cases} 1 & \text{if the } i^{th} \text{ experiment selects a red ball} \\ 0 & \text{if the } i^{th} \text{ experiment selects a black ball.} \end{cases} \tag{2.14}$$

The state of the channel $\underline{\mathrm{S}}_i$ is defined to be the binary M-tuple in the queue, $\underline{\mathrm{S}}_i = (A_{i1}, A_{i2}, \cdots, A_{iM})$. In other words, the state of the channel at any time is the set of the last M noise outputs of the channel, $\underline{\mathrm{S}}_i = (Z_i, Z_{i-1}, \cdots, Z_{i-M+1})$ and the process $\{\underline{\mathrm{S}}_i\}_{i=1}^{\infty}$ is a first-order Markov process with an alphabet of size 2^M. We note that the effect of the memory is in the queue. Thus if $\varepsilon = 0$, then we will always select the urn, hence the channel becomes memoryless. Indeed, if $\varepsilon = 0$, then the QBC reduces to the BSC(p).

2.6.1 Channel Properties

Theorem 2.3 [49] *The noise process $\{Z_i\}_{i=0}^{\infty}$ generated by the QBC is a homogeneous M^{th}-order Markov process, with stationary distribution, $\boldsymbol{\pi}^{(M)}$ given by*

$$\pi_i^{(M)} = \frac{\prod_{j=0}^{M-1-\omega_i^{(M)}} \left[j\frac{\varepsilon}{M-1+\alpha} + (1-\varepsilon)(1-p)\right] \prod_{j=0}^{\omega_i^{(M)}-1} \left[j\frac{\varepsilon}{M-1+\alpha} + (1-\varepsilon)p\right]}{\prod_{j=0}^{M-1}\left[1-(\alpha+j)\frac{\varepsilon}{M-1+\alpha}\right]}, \quad (2.15)$$

for $i = 0, 1, 2, \cdots, 2^M - 1$, where $\prod_{j=0}^{a}(\cdot)\overset{\triangle}{=}1$ if $a < 0$, and $\omega_i^{(M)}$ is the number of "ones" in the M-bit binary representation of the decimal integer i.

The block transition probability for the QBC, $\Pr(Y^n = y^n | X^n = x^n) = \Pr^{(M)}(Z^n = z^n)$, where $z^n = y^n \oplus x^n$, is determined in [46], [49] as follows.

- For blocklength $n \leq M$,

  $$\begin{aligned}&\Pr{}^{(M)}(Z^n = z^n)\\ &= \frac{\prod_{j=0}^{n-d_1^n-1} \left[j\frac{\varepsilon}{M-1+\alpha} + (1-\varepsilon)(1-p)\right] \prod_{j=0}^{d_1^n-1} \left[j\frac{\varepsilon}{M-1+\alpha} + (1-\varepsilon)p\right]}{\prod_{j=M-n}^{M-1}\left[1-(\alpha+j)\frac{\varepsilon}{M-1+\alpha}\right]}, \end{aligned} \quad (2.16)$$

  where $d_a^b = z_b + z_{b-1} + \cdots + z_a$ ($d_a^b = 0$ if $a > b$), and $\prod_{j=0}^{a}(\cdot)\overset{\triangle}{=}1$ if $a < 0$.

- For blocklength $n \geq M + 1$,

  $$\begin{aligned}\Pr{}^{(M)}(Z^n = z^n) = L^{(M)} \prod_{i=M+1}^{n} &\left[\left(d_{i-M+1}^{i-1} + \alpha z_{i-M}\right)\frac{\varepsilon}{M-1+\alpha} + (1-\varepsilon)p\right]^{z_i}\\ &\left\{\left[\left(M-1-d_{i-M+1}^{i-1}\right) + \alpha(1-z_{i-M})\right]\frac{\varepsilon}{M-1+\alpha} + (1-\varepsilon)(1-p)\right\}^{1-z_i}, \end{aligned} \quad (2.17)$$

  where

  $$L^{(M)} = \frac{\prod_{j=0}^{M-1-d_1^M} \left[j\frac{\varepsilon}{M-1+\alpha} + (1-\varepsilon)(1-p)\right] \prod_{j=0}^{d_1^M-1} \left[j\frac{\varepsilon}{M-1+\alpha} + (1-\varepsilon)p\right]}{\prod_{j=0}^{M-1}\left[1-(\alpha+j)\frac{\varepsilon}{M-1+\alpha}\right]}.$$

The probability transition matrix for the process $\{\underline{S}_n\}_{n=1}^{\infty}$, denoted by $\boldsymbol{P}_{QBC}^{(M)} = [p_{ij}^{(M)}]$, where $p_{ij}^{(M)}$ is the probability that $\underline{S}_n$ goes from state i to state j, is given by

$$p_{ij}^{(M)} = \begin{cases} \left(M - \omega_i^{(M)} - 1 + \alpha\right) \frac{\varepsilon}{M-1+\alpha} + (1-\varepsilon)(1-p), & \text{if } j = \frac{i}{2}, \text{ and } i \text{ is even}, \\ \left(M - \omega_i^{(M)}\right) \frac{\varepsilon}{M-1+\alpha} + (1-\varepsilon)(1-p), & \text{if } j = \lfloor \frac{i}{2} \rfloor, \text{ and } i \text{ is odd}, \\ \omega_i^{(M)} \frac{\varepsilon}{M-1+\alpha} + (1-\varepsilon)p, & \text{if } j = \frac{i+2^M}{2}, \text{ and } i \text{ is even}, \\ \left(\omega_i^{(M)} - 1 + \alpha\right) \frac{\varepsilon}{M-1+\alpha} + (1-\varepsilon)p, & \text{if } j = \lfloor \frac{i+2^M}{2} \rfloor, \text{ and } i \text{ is odd}, \\ 0, & \text{otherwise}. \end{cases} \qquad (2.18)$$

Here states i and j are the integer representations (varying from 0 to $2^M - 1$) for states $\underline{s}_{n-1}$ and $\underline{s}_n$, respectively. For example, if the state at time $n-1$ is $\underline{s}_{n-1} = (z_{n-1}, z_{n-2}, \cdots, z_{n-M})$, then $i = \sum_{l=0}^{M-1} z_{n-1-l} \cdot 2^l$.

The channel bit error rate, *BER*, and the correlation coefficient *Cor*, for the QBC are given by

$$BER = \Pr{}^{(M)}(Z_i = 1) = \Pr{}^{(M)}(Z_1 = 1) = p, \qquad (2.19)$$

and

$$Cor = \frac{\frac{\varepsilon}{M-1+\alpha}}{1 - (M-2+\alpha)\frac{\varepsilon}{M-1+\alpha}} = \frac{\varepsilon}{(M-1+\alpha) - \varepsilon(M-2+\alpha)} \geq 0. \qquad (2.20)$$

It can be also shown [46], [49] that the capacity of the QBC, denoted $C_{QBC}^{(M)}$ is:

$$\begin{aligned} C_{\text{QBC}}^{(M)} &= 1 - \sum_{\omega=0}^{M-1} \binom{M-1}{\omega} L_\omega^{(M)} h_b \left[\omega \frac{\varepsilon}{M-1+\alpha} + (1-\varepsilon)p\right] \\ &\quad - \sum_{\omega=1}^{M} \binom{M-1}{\omega-1} L_\omega^{(M)} h_b \left[(\omega+\alpha-1)\frac{\varepsilon}{M-1+\alpha} + (1-\varepsilon)p\right] \qquad (2.21) \end{aligned}$$

where

$$L_{\omega}^{(M)} = \frac{\prod_{j=0}^{M-1-\omega}\left[j\frac{\varepsilon}{M-1+\alpha} + (1-\varepsilon)(1-p)\right]\prod_{j=0}^{\omega-1}\left[j\frac{\varepsilon}{M-1+\alpha} + (1-\varepsilon)p\right]}{\prod_{j=0}^{M-1}\left[1-(\alpha+j)\frac{\varepsilon}{M-1+\alpha}\right]},$$

$\prod_{j=0}^{a}(\cdot)\overset{\triangle}{=}1$, if $a < 0$, $\binom{a}{b}\overset{\triangle}{=}1$, if $a = 0$, and $h_b(\cdot)$ is the binary entropy function.

Theorem 2.4 [49] *The capacity $C_{QBC}^{(M)}$ of the QBC strictly increases with α for fixed $M \geq 2$, BER and Cor $\in (0,1)$.*

Remarks:

- $QBC^{(M)}$ has only 4 parameters: ε, p, α and M.

- if $\varepsilon = 0$, then the QBC$^{(M)}$ reduces to the BSC(p) .

- If $M = 1$, then we assume $\alpha = 1$, and in this case the QBC is a general first-order Markov noise channel (with non-negative noise correlation).

- If $\alpha = 1$, then the $QBC^{(M)}$ reduces to the $FMCC^{(M)}$.

In Chapter 4 we will further analyze this channel model and derive some interesting and important properties that will facilitate the performance analysis of linear block codes over this channel.

Chapter 3

Linear Block Codes

The channel coding theorem, as was stated in Chapter 2, proves the existence of good codes with asymptotically vanishing probability of decoding error provided that the rate is kept below the channel capacity. Ever since this remarkable discovery, error-control coding has gained lots of attention and constructing codes within the limits laid out by Shannon's theorem has become the prime task of coding theorists. Most of the codes that exist today belong to the class of *linear block codes.* Although binary codes are more popular due to their simplicity and their practical usefulness, non-binary codes were found to be as important, if not more, in some applications.

In this chapter, a glimpse of some famous binary and non-binary linear codes will be introduced. In particular, the Hamming codes, the binary BCH codes, and the Reed-Solomon (RS) codes will be discussed. The material of this chapter can be found in

[21]–[22].

3.1 Binary Linear Block Codes

Definition 3.1 *A binary linear code, $\mathcal{C}$ is a linear subspace of $\{0,1\}^n$. The dimension of the code, k is the size of the basis of $\mathcal{C}$, and given by $k = \log_2 |C|$. Such a code is denoted by $[n,k]$ and its elements are called codewords. The code's rate is thus given by $R(\mathcal{C}) = k/n$.*

Definition 3.2 *Let $\mathcal{C}$ be a binary linear code. Then, for any codeword $x^n \triangleq (x_1, \cdots, x_n) \in \mathcal{C}$, the Hamming weight of x^n, denoted by $w_H(x^n)$, is the number of ones in x^n. The Hamming distance between two codewords $x^n, y^n \in \mathcal{C}$, $d_H(x^n, y^n)$ is the Hamming weight of their difference, $d_H(x^n, y^n) = w_H(x^n - y^n) = w_H(x^n \oplus y^n)$, where the $-$ and $\oplus$ operations are performed component-wise on x^n and y^n.*

Definition 3.3 *The minimum distance, d, of the binary linear code $\mathcal{C}$, is the smallest non-zero Hamming weight of its codewords.*

From now on we shall denote a binary code of length n, dimension k and minimum distance d by $[n,k,d]$ or just simply $[n,k]$ if the minimum distance is not of interest.

Definition 3.4 *Any $k \times n$ matrix whose rows are a basis of $\mathcal{C}$ is called a generator matrix of $\mathcal{C}$ and denoted by $\boldsymbol{G}$.*

$\boldsymbol{G}$ is called systematic if it is of the form $\boldsymbol{G} = [\boldsymbol{I_k} \mid \boldsymbol{P}]$, where $\boldsymbol{I_k}$ is the $k \times k$ identity matrix.

Definition 3.5 *The parity-check matrix, $\boldsymbol{H}$ of the linear block code $\mathcal{C}$, is an $(n-k)\times n$ matrix of rank $n-k$ with the property that $\boldsymbol{G} \cdot \boldsymbol{H^T} = \mathbf{0}$.*

Remark: If $\boldsymbol{G}$ is systematic, then $\boldsymbol{H} = [\boldsymbol{P^T} \mid \boldsymbol{I_{n-k}}]$, where T denotes the transpose operation.

The generator matrix is used to encode the messages, while the parity-check matrix can be used at the decoder using *syndrome decoding* as will be described later.

Definition 3.6 *Let $x^n \in \{0,1\}^n$, then for any linear code $\mathcal{C}$, $x^n + \mathcal{C} \triangleq \{x^n \oplus c^n : c^n \in \mathcal{C}\}$ is called a coset of $\mathcal{C}$.*

Note that if $x^n \in \mathcal{C}$, then $x^n + \mathcal{C} = \mathcal{C}$. The element of the smallest weight in $x + \mathcal{C}$ is called the *coset leader*. For an $[n, k]$ linear code $\mathcal{C}$, the number of distinct cosets is equal to 2^{n-k}.

Definition 3.7 (Perfect Code) *A linear code $\mathcal{C}$ is said to be a* perfect code *if, for some non-negative integer t, it has all patterns (i.e., elements of $\{0,1\}^n$) of Hamming weight t or less and no others as coset leaders.*

Definition 3.8 (Quasi-Perfect Code) *A linear code $\mathcal{C}$ is said to be* quasi-perfect *if, for some non-negative integer t, it has all patterns of Hamming weight t or less, some of weight $t+1$, and none of greater weight as coset leaders.*

An equivalent definition for quasi-perfectness is that, for some non-negative integer t, $\mathcal{C}$ has a packing radius equal to t and a covering radius equal to $t+1$; i.e., the spheres with (Hamming) radius t around the codewords of $\mathcal{C}$ are disjoint, and the spheres with radius $t+1$ around the codewords cover $\{0,1\}^n$. On the other hand, perfectness means that both packing and covering radii are equal. For these two classes of codes, $t = \lfloor \frac{d-1}{2} \rfloor$ (with $d = 2t+1$ for perfect codes and $d = 2t+1$ or $d = 2t+2$ for quasi-perfect codes).

The $[2^m-1, 2^m-1-m, 3]$ Hamming codes, the $[n,1,n]$ repetition code with n odd and the $[23,12,7]$ Golay code are the only members of the family of the binary perfect codes. Examples of quasi-perfect binary linear codes include the $[n,1,n]$ repetition codes with n even, the $[2^m, 2^m-1-m, 4]$ extended Hamming codes as well as the $[2^m-2, 2^m-2-m, 3]$ shortened Hamming codes ($m \geq 2$), the $[2^m-1, 2^m-1-2m, 5]$ double-error correcting BCH codes ($m \geq 3$), and the $[24,12,8]$ extended Golay code.

Remark Perfect and quasi-perfect codes are optimal over the BSC [27]; i.e., they achieve the smallest probability of codeword error amongst all the codes of the same length and dimension when used over the BSC and decoded via a maximum likelihood decoder (which will be defined in the following subsection).

3.1.1 Hamming Codes

According to [9], the Hamming codes are the first class of error-correcting linear codes to be designed. They and their variations are used in both digital communications and data storage systems.

The family of binary Hamming codes are linear block codes with parameters $n = 2^m - 1, k = 2^m - m - 1$ and $d = 3$, where the integer $m \geq 2$.

Example: The $[7, 4, 3]$ Hamming code is a linear binary code with generator matrix $\boldsymbol{G}$ and $\boldsymbol{H}$ given by

$$\boldsymbol{G} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix}, \tag{3.1}$$

$$\boldsymbol{H} = \begin{bmatrix} 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}. \tag{3.2}$$

One can easily verify that $\boldsymbol{G} \cdot \boldsymbol{H^T} = \mathbf{0}$.

Encoding Hamming Codes

Every k message bits generated by the source are encoded by the channel encoder via a simple multiplication using the generator matrix to produce a codeword of length n. As was described earlier, the operation is one-to-one, thus making the reverse operation of recovering the original message from the decoded codeword easier. In the case of the systematic encoder, the first k bits of the codeword are nothing but the original message produced by the source.

Decoding Hamming Codes

The aim of the decoder is to make a guess of what the actual transmitted codeword was. The only pieces of information available to the decoder are the channel statistics and the received codeword. If y^n is received at the channel output, and is decoded into $c_0^n \in \mathcal{C}$, where c_0^n satisfies $\Pr(Y^n = y^n | X^n = c_0^n) \geq \Pr(Y^n = y^n | X^n = c^n)$ for all $c^n \in \mathcal{C}$, then the decoder performs *maximum likelihood (ML) decoding*. If there is more than one codeword for which the above condition holds, then the decoder picks one of such codewords at random. If the above inequality, however, is strict then the decoding is called *strict ML (SML) decoding*, in which a decoding failure is declared if more than one codeword have largest ML metric, $\Pr(Y^n = y^n | X^n = c^n)$. Another decoding strategy is based on minimizing the Hamming distance between the received word and the decoded codeword. In other words, the decoder chooses the codeword that is closest to the received word in terms of the Hamming distance. This is called *minimum distance (MD) decoding* defined below. When a code is used over a BSC(p) with crossover probability $p < 1/2$, then MD and ML decoding can be shown to be equivalent.

Definition 3.9 Minimum Distance (MD) Decoding*: y^n is decoded into codeword $c_0 \in \mathcal{C}$ if $w(c_0 \oplus \mathrm{y}) \leq w(c \oplus \mathrm{y})$ for all $c \in \mathcal{C}$. If there is more than one codeword for which the above condition holds, then the decoder picks one of such codewords at random.*

Definition 3.10 Strict Minimum Distance (SMD) Decoding*: It is identical to the MD rule with the exception of replacing the inequality with a strict inequality; if no codeword c_0 satisfies the strict inequality, the decoder declares a decoding failure.*

For both ML and MD decoding, exhaustive search over the codebook is not feasible for codes with large values of k, since the complexity increases exponentially. A better approach to decode linear block codes is to use *syndrome decoding* which is described next.

Syndrome Decoding

The syndrome of any binary $[n, k]$ code is an element of the vector space, $\{0, 1\}^{n-k}$, vector space obtained by multiplying all the n-tuple vectors $a^n \in \{0, 1\}^n$ by the transpose of the parity-check matrix $\boldsymbol{H^T}$. Let x^n be the transmitted codeword. Over the additive noise channel, the received word will be $y^n = x^n \oplus z^n$, where z^n is the additive noise vector generated according to a channel model described in the previous chapter. Now, the syndrome, denoted $s^{n-k} \triangleq (s_1, s_2, \cdots, s_{n-k})$ is given by

$$\begin{aligned} s^{n-k} &= y^n \cdot \boldsymbol{H^T} \\ &= (x^n \oplus z^n) \cdot \boldsymbol{H^T} \\ &= x^n \cdot \boldsymbol{H^T} \oplus z^n \cdot \boldsymbol{H^T} \\ &= z^n \cdot \boldsymbol{H^T}. \end{aligned} \tag{3.3}$$

Note that since $\boldsymbol{G} \cdot \boldsymbol{H^T} = \mathbf{0}$ and any codeword is spanned by the rows of $\boldsymbol{G}$, then $x^n \cdot \boldsymbol{H^T} = \mathbf{0}$ where x^n is a codeword. This means that the syndrome only depends on the error pattern z^n, and not on the transmitted codeword. Thus any element in $z^n + \mathcal{C}$ will have the same syndrome. In other words, elements of the same coset are mapped to the same syndrome. Hence, the number of distinct syndromes is equal to the number of cosets, which is 2^{n-k}.

Now assume that the syndrome of the received vector y^n is s^{n-k}. Then there are $|\mathcal{C}|$ possible noise outputs that could have been generated by the channel and produced y^n, where $|\mathcal{C}| = 2^k$ is the size of the codebook which also equals the size of any coset. For the BSC with crossover probability $p < \frac{1}{2}$, noise outputs with smaller Hamming weights are more likely to occur; therefore, the coset leader is the most likely noise pattern to be generated. Hence, given the syndrome, the decoder presumes that the corresponding coset leader, z^*, had been added to the actual transmitted codeword and thus subtracts (which is equivalent to the addition over the binary field) z^* from the received word to decode what it thinks is the most probable transmitted codeword $\hat{x}^n = y^n \oplus z^*$. Indeed in this case, syndrome decoding is identical to ML and MD decoding.

Example: For the $[7,4]$ Hamming code in the above example, assume that $y^n = (1,0,0,1,0,0,0)$, then the syndrome is given by $y^n \cdot H^T = (0,1,1)$. The coset leader for the syndrome can be shown to be $(0,1,0,0,0,0,0)$. Thus, the decoder adds this to the received word. So, the decoded codeword is $\hat{x}^n = (1,1,0,1,0,0,0)$.

Syndrome decoding requires the system to store all 2^{n-k} syndromes and their corresponding coset leaders. However, this storage requirement reduces the computation complexity incurred by the exhaustive search over the whole codebook. In many applications, more sophisticated decoding algorithms are used such as the *Viterbi* and *sum-product algorithms* (e.g., see [9]), which has less complexity than the syndrome decoding algorithm. However, for codes with small values of $(n-k)$, syndrome decoding can be used without a significant delay in the system.

Extended Hamming Codes

The extended Hamming codes is another family of the linear block codes. An $[n, k, 3]$ Hamming code can be easily transformed to an extended one by adding a bit to each codeword to make its Hamming weight even. As a result, the new code is an

$[n+1, k, 4]$ linear block code.

Example: The extended Hamming $[8, 4, 4]$ is obtained by adding a (parity) bit to the Hamming $[7, 4, 3]$ in the same way described above.

Remark: The extended Hamming codes are *quasi perfect*.

3.1.2 BCH Codes

A shortcoming of the Hamming and extended Hamming codes is their small Hamming distance. Codes with larger Hamming distances are desirable since they can correct more errors. BCH codes are a generalization of Hamming codes. Such codes have larger Hamming distances than the Hamming codes, in general. The BCH codes belong to the class of cyclic linear block codes, which have the property that any cyclic shift of any codeword yields another codeword.

For a binary $[n, k]$ code $\mathcal{C}$, a codeword $c^n \in \mathcal{C}$ can be represented in polynomial form. Let $c^n = (c_0, c_1, \cdots, c_{n-1})$, then $c(x) = c_0 + c_1 x + \cdots + c_{n-1} x^{n-1}$ is the polynomial representation for the codeword c^n. A cyclic code is completely specified by its nonzero codeword polynomial of minimum degree [22]. Such a polynomial is called the *generator polynomial* which is analogous to the generator matrix and is denoted by $g(x)$. The degree of this polynomial is $n - k$.

Definition 3.11 *If $a \in GF(2^n)$, the Galois field of size 2^n, with the property that the smallest integer l satisfying $a^l = 1$ is $l = 2^n - 1$, then a is called primitive. The* *integer l is called the order of a.*

Definition 3.12 *Let $a \in GF(2^n)$. Let $\phi(x)$ be a (non-zero) polynomial over $GF(2) = \{0, 1\}$ with the smallest degree such that $\phi(a) = 0$. Then $\phi(x)$ is called the minimal polynomial of a.*

For any given integer $m \geq 3$ and $t < 2^{m-1}$ a binary BCH code with the properties $n = 2^m - 1, k \geq n - mt$ and $d \geq 2t + 1$ can be designed. The generator polynomial for such code is given by

$$\begin{aligned} g(x) &= LCM\{\phi_1(x), \phi_2(x), \cdots, \phi_{2t}(x)\} \\ &= LCM\{\phi_1(x), \phi_3(x), \cdots, \phi_{2t-1}(x)\} \end{aligned}$$

where LCM stands for the least common multiple, $\phi_i(x)$ is the minimal polynomial of a^i and a is a primitive element in GF(2^n).

Remark: Under MD decoding, the above BCH code can correct up to t errors per codeword (actually this applies to all block codes with $d \geq 2t + 1$).

3.2 Non-Binary Linear Block Codes

Similar to the case of binary linear block codes, one can define non-binary linear block codes or *q-ary linear block codes.* In this case, a codeword of a q-ary code will consist of symbols from the Galois field of size q, GF(q) [22] such that $q = p^m$, where p is prime and m is a positive integer.

Definition 3.13 *Let $\mathbb{F} = GF(q)$. A q-ary $[n, k]$ linear code, is a k-dimensional subspace of the vector space $\mathbb{F}^n$.*

Likewise, one can also define a q-ary linear cyclic code, with a generator polynomial of degree $n - k$ with coefficients from GF(q).

Example: The q-ary BCH code is cyclic. The design of a BCH code that is capable of correcting t errors under MD decoding is done by finding the corresponding generator polynomial, $g(x)$ which is in turn given by

$$g(x) = LCM\{\phi_1(x), \phi_2(x), \cdots, \phi_{2t}\}$$

where $\phi_i(x)$ is the minimal polynomial of a^i over GF(q) and a is a primitive element in GF(q^s), and s is a positive integer.

Remark: Each $\phi_i(x)$ is a monic polynomial, which means that the coefficient of the highest degree term is 1. Hence, $g(x)$ itself is monic.

Note that if $q = 2$, then we are reduced to the binary BCH codes discussed earlier.

If we let $s = 1$, then a special type of BCH codes is formed known as the *Reed-Solomon* (RS) codes that are widely used in data storage applications, such as CD's and DVD's due to their ability to correct bursts of errors. More details on this subclass of the BCH codes are given in the next section.

3.3 Reed-Solomon Codes

Reed-Solomon (RS) codes were first introduced by I. Reed and G. Solomon in 1960. These codes are of considerable interest both theoretically and practically. As explained in the previous section, RS codes are a subset of the non-binary BCH codes. In this work we will deal with RS codes with symbols from GF(2^m).

The generator polynomial, $g(x)$ of a t-error correcting Reed-Solomon code (under bounded distance decoding[1]) is constructed as follows. Let $\gamma \in$ GF(2^m) be primitive. Then $g(x)$ is given by

$$g(x) = (x + \gamma)(x + \gamma^2) \cdots (x + \gamma^{2t})$$

The resultant RS code is of length $n = 2^m - 1$, dimension $k = n - 2t$, and minimum distance $d = 2t + 1$, and is over the field GF(2^m) .

RS codes have the property of having the largest possible minimum distance amongst all $[n, k]$ codes. Such codes are called *maximum distance separable* (MDS). Throughout this work, we assume the transmission of RS codes over binary channels. In fact, each symbol in GF(2^b) can be mapped one-to-one to a binary b-tuple. As a result, the non-binary codewords are sent over a binary-input binary-output channel by transmitting the equivalent binary representation for each codeword. A transmitted symbol is received correctly if the noise corrupting it is a sequence of zeros of

[1] If an MD decoder can correct up to t errors within a codeword for some positive integer t, but declares a decoding failure if more than t errors are detected, then it is said to perform bounded distance decoding.

length b, denoted as 0^b. Otherwise, the transmitted symbol is received incorrectly and a symbol error occurs. A t-error correcting code is capable of correcting bt bits in the best case (i.e., if all the bits of the erroneous symbol are in error). This indeed illustrates the power of RS codes in correcting bursts of errors, which in turn makes this family of non-binary codes one of the favorites in many applications where errors occur in bursts, such as data storage systems.

The generator polynomial is used in order to encode the incoming messages from the source. Similar to the binary case, this is done by a simple polynomial multiplication of every k message symbols by the $g(x)$ to produce a codeword of length n symbols that is eventually transmitted. The reader is referred to [22] and [9] for further details on implementing $g(x)$ at the encoder.

3.3.1 Decoding RS Codes

First we define the parity-check matrix $\boldsymbol{H}$ for the RS code in (3.4) below, with γ being primitive in GF(2^m). Assume $r^n \in (\mathrm{GF}(2^m))^n$ is received at the channel output, with polynomial form $r(x)$. The decoding proceeds as follows.

$$\boldsymbol{H} = \begin{bmatrix} 1 & \gamma & \gamma^2 & \cdots & \gamma^{n-1} \\ 1 & \gamma^2 & (\gamma^2)^2 & \cdots & (\gamma^2)^{n-1} \\ 1 & \gamma^3 & (\gamma^3)^2 & \cdots & (\gamma^3)^{n-1} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ 1 & \gamma^{2t} & (\gamma^{2t})^2 & \cdots & (\gamma^{2t})^{n-1} \end{bmatrix}. \tag{3.4}$$

1. First, the *syndrome*, $s^{2t} = (s_1, s_2, \cdots, s_{2t})$ is computed as $s^{2t} = r^n \cdot \boldsymbol{H^T}$.

   Assume that the symbols in error are located at $x^{j_1}, x^{j_2}, \cdots, x^{j_\nu}$, then the error polynomial $e(x)$ is given by

$$e(x) = x^{j_1} + x^{j_2} + \cdots + x^{j_\nu} \tag{3.5}$$

   where $0 \leq j_1 < j_2 < \cdots < j_\nu < n$.

   Then we have

$$\begin{aligned} s_i &= (\gamma^{j_1})^i + (\gamma^{j_2})^i + \cdots + (\gamma^{j_\nu})^i \\ &= (\beta_1)^i + (\beta_2)^i + \cdots + (\beta_\nu)^i \qquad \text{where } 1 \leq i \leq 2t \end{aligned}$$

   where we define $\beta_l = \gamma^{j_l}$, for $l = 1, 2, \cdots \nu$, as the *error location numbers.*

2. Define the *error locator polynomial* $\sigma(x)$ by

$$\begin{aligned} \sigma(x) &\triangleq (1 + \beta_1 x)(1 + \beta_2 x) \cdots (1 + \beta_\nu x) \\ &= \sigma_0 + \sigma_1 x + \cdots + \sigma_\nu x^\nu \end{aligned} \tag{3.6}$$

   where

$$\begin{aligned} \sigma_0 &= 1 \\ \sigma_1 &= \beta_1 + \beta_2 + \cdots + \beta_\nu \\ \sigma_2 &= \beta_1\beta_2 + \beta_2\beta_3 + \cdots + \beta_{\nu-1}\beta_\nu \\ &\vdots \\ \sigma_\nu &= \beta_1\beta_2 \cdots \beta_\nu. \end{aligned}$$

The elements of the syndrome s_i's are related to the σ_i's by the following *Newton's identities*

$$\begin{aligned} s_1 + \sigma_1 &= 0 \\ s_2 + \sigma_1 s_1 + 2\sigma_2 &= 0 \\ s_3 + \sigma_1 s_2 + \sigma_2 s_1 + 3\sigma_3 &= 0 \\ &\vdots \\ s_\nu + \sigma_1 s_{\nu-1} + \cdots + \sigma_{\nu-1} s_1 + \nu\sigma_\nu &= 0. \end{aligned} \tag{3.7}$$

3. After determining $\sigma(x)$ from $s^{2t} = (s_1, s_2, \cdots, s_{2t})$, the error locator numbers $\beta_1, \beta_2, \cdots, \beta_\nu$ must be determined by finding the roots of $\sigma(x)$.

Berlekamp's iterative algorithm is used to find the error locator polynomial $\sigma(x)$. The magnitude of the error is then determined by the *Forney algorithm* [8].

Berlekamp's Iterative Algorithm

An important step in decoding RS codes is to find the error locator polynomial $\sigma(x)$. Berlekamp's iterative algorithm is one of many algorithms that are designed for this purpose. The number of iterations in this algorithm is $2t$. Therefore, it is linear in the number of correctable errors. In the i^{th} step, the task is to find a minimum degree polynomial satisfying the first i Newton's identities given in (3.7). A summary of the process of finding $\sigma(x)$ is given below.

- At the μ^{th} step , let the minimum-degree polynomial coefficients satisfying the first μ Newton's identities be

$$\sigma^{\mu}(x) = 1 + \sigma_1^{(\mu)} x + \sigma_2^{(\mu)} x^2 + \cdots + \sigma_{l_\mu}^{(\mu)} x^{l_\mu}. \tag{3.8}$$

- To find $\sigma^{(\mu+1)}(x)$, the μ^{th} *discrepancy*, d_μ is calculated by

$$d_\mu = s_{\mu+1} + \sigma_1^{(\mu)} s_\mu + \sigma_2^{(\mu)} s_{\mu-1} + \cdots + \sigma_{l_\mu}^{(\mu)} s_{\mu+1-l_\mu}. \tag{3.9}$$

- If $d_\mu = 0$, then $\sigma^{(\mu+1)}(x) = \sigma^{(\mu)}(x)$. Otherwise,

$$\sigma^{(\mu+1)}(x) = \sigma^{(\mu)}(x) - d_\mu d_\rho^{-1} x^{(\mu-\rho)} \sigma^{(\rho)}(x) \tag{3.10}$$

  where ρ is one of the steps prior to μ with $d_\rho \neq 0$ and $\rho - l_\rho$ has the largest value (l_ρ is the degree of $\sigma^{(\rho)}(x)$).

The proof of this algorithm is given in [22].

The Forney Algorithm

After finding the error locator polynomial $\sigma(x)$ and getting all of its roots, the error locations are determined. The next step in decoding the received vector is to find the magnitude of this error. Note that for the binary BCH codes this step is not required because once the error locations are known, bits in those locations are flipped. However this is not the case with the non-binary BCH codes and, in particular, for RS codes. Forney algorithm is a way to find the error magnitude. The steps of this algorithm are given next.

- Let $S(x)$ be the polynomial representation of the syndrome of the received word. Define the *error magnitude polynomial* $\Omega(x)$ by

$$\Omega(x) = \sigma(x)(1 + S(x)) \qquad \bmod (x^{2t+1}). \tag{3.11}$$

- Let the error location numbers be $\beta_1, \beta_2, \cdots, \beta_\nu$. These are already determined from the previous steps.

  Calculate the error magnitude by

$$e_{j_i} = \frac{-\beta_i \Omega(\beta_i^{-1})}{\sigma'(\beta_i^{-1})}. \tag{3.12}$$

  Note that $\beta_i = \gamma^{j_i}$ and $1 \leq i \leq \nu$.

Remark: If the number of errors $\nu > t$, then the decoder fails to decode the received word. Hence, the decoding algorithm is not complete. In other words, the decoder will flag an error signal if the degree of $\sigma(x)$ is greater than t. This is why such decoding method is called bounded distance decoding.

Chapter 4

Maximum Likelihood Decoding of Binary Linear Block Codes Over the QBC

In this chapter, ML decoding for binary linear block codes over the QBC with memory M is studied. As introduced earlier in Chapter 2, the n-block transition probability over the QBC has two different expressions depending on whether $M > n$ or $M \leq n$. Therefore, the two cases must be studied separately. For the case of $M \leq n$, we show that ML decoding is equivalent to either minimum distance decoding or maximum distance decoding; this result is similar to the one obtained for the infinite-memory Polya contagion channel in [1]. For the other case, however, a different approach is needed and the study is restricted to the case $M = 1$, i.e., the binary additive Markov noise channel (BAMNC).

4.1 Maximum Likelihood (ML) Decoding

Suppose that a channel encoder maps a binary k-bit message tuple u^k (which is the output of a uniformly distributed binary memoryless source) to a length n binary codeword $x^n \in \mathcal{C}$. When transmitted over a binary additive noise channel, x^n will be received as $y^n = x^n \oplus z^n$ (mod 2), where z^n is the noise output. The task of the decoder on the other hand is to estimate the transmitted message k-tuple by observing y^n. Denote this estimate by $\hat{u}^k$. Since the encoder mapping is one-to-one, this reduces the task of the decoder to making the best guess about what codeword was actually transmitted. Let $\hat{x}^n$ be the estimate of x^n. Obviously, an error occurs if the decoder outputs $\hat{x}^n \neq x^n$ given that x^n is transmitted. At the decoder, the only information available besides the received word y^n is the channel statistics. Hence, the probability of codeword decoding error, denoted $\Pr(E)$ can be computed mathematically as

$$\begin{aligned}\Pr(E) &= \sum_{x^n \in \mathcal{C}} \Pr(E|X^n = x^n)\Pr(X^n = X^n) \\ &= \sum_{x^n \in \mathcal{C}} \Pr(\hat{X}^n \neq X^n|X^n = x^n)\Pr(X^n = x^n) \qquad (4.1)\end{aligned}$$

where $\Pr(E|X^n = x^n)$ is the conditional probability of decoding error given that codeword x^n was sent.

The best or optimal decoder is the one that minimizes $\Pr(E)$. It can be shown (e.g., [13]) that optimal decoder is the so-called maximum a-posteriori (MAP) decoder, which selects the codeword x^n that maximizes $\Pr(X^n = x^n | Y^n = y^n)$.

On the other hand, the conditional probability $\Pr(X^n = x^n | Y^n = y^n)$ can be expressed as

$$\Pr(X^n = x^n | Y^n = y^n) = \frac{\Pr(Y^n = y^n | X^n = x^n) \Pr(X^n = x^n)}{\Pr(Y^n = y^n)}.$$

Hence, maximizing $\Pr(X^n = x^n | Y^n = y^n)$ with respect to $x^n \in \mathcal{C}$ is the same as maximizing $\Pr(Y^n = y^n | X^n = x^n) \Pr(X^n = x^n)$. However, since the source message tuples are equally likely, all the codewords have equal likelihoods to be transmitted. As a consequence, maximizing $\Pr(X^n = x^n | Y^n = y^n)$ is reduced to maximizing $\Pr(Y^n = y^n | X^n = x^n) = \Pr(Z^n = z^n)$, where $z^n = y^n \oplus x^n$. In other words, MAP decoding reduces to ML decoding.

4.2 ML Decoding over the QBC

4.2.1 Case I: $n \leq M$

First we consider the case of $n \leq M$. The QBC block transition probability $\Pr(Y^n = y^n | X^n = x^n) = \Pr^{(M)}(Z^n = z^n)$, with $z_i = y_i \oplus x_i$, $i = 1, \cdots, n$, for this case, is given by (2.16) as

$$
\begin{aligned}
&\Pr^{(M)}(Z^n = z^n) \\
&= \frac{\prod_{j=0}^{n-d_1^n-1}\left[j\frac{\varepsilon}{M-1+\alpha} + (1-\varepsilon)(1-p)\right]\prod_{j=0}^{d_1^n-1}\left[j\frac{\varepsilon}{M-1+\alpha} + (1-\varepsilon)p\right]}{\prod_{j=M-n}^{M-1}\left[1-(\alpha+j)\frac{\varepsilon}{M-1+\alpha}\right]},
\end{aligned}
$$

where $d_a^b = z_b + z_{b-1} + \cdots + z_a$ ($d_a^b = 0$ if $a > b$), and $\prod_{j=0}^{a}(\cdot) \overset{\triangle}{=} 1$ if $a < 0$. Since d_1^n is nothing but the Hamming weight of the noise output z^n, we wish to express the above expression of the block transition probability in terms of d_1^n for a given blocklength n and a channel condition p, M, ε and α. For simplicity, write d_1^n as d. Then, similar to the strategy used in [1], we use the fact that

$$
\prod_{j=0}^{n-1}(u + jv) = v^n \frac{\Gamma(u/v + n)}{\Gamma(u/v)}, \text{ for } u, v > 0
$$

where $\Gamma(\cdot)$ is the well-known gamma function, $\Gamma(x) = \int_0^\infty t^{x-1}e^{-t}dt$ for $x > 0$, and note that $\Pr^{(M)}(Z^n = z^n)$ can be expressed as

$$
\begin{aligned}
\Pr^{(M)}(Z^n = z^n) &= C \cdot \Gamma\left(\frac{1}{\varepsilon}(1-\varepsilon)(1-p)(M-1+\alpha) + n - d\right) \\
&\quad \times \Gamma\left(\frac{p}{\varepsilon}(1-\varepsilon)(M-1+\alpha) + d\right)
\end{aligned}
$$

where C is a constant that depends on n, p, ε and α given by

$$
C \triangleq \frac{\Gamma(\frac{M-1+(1-\varepsilon)\alpha}{\varepsilon} + M - n)}{\Gamma(\frac{M-1+(1-\varepsilon)\alpha}{\varepsilon} + M) \cdot \Gamma(\frac{(1-\varepsilon)(1-p)(M-1+\alpha)}{\varepsilon}) \cdot \Gamma(\frac{p(1-\varepsilon)(M-1+\alpha)}{\varepsilon})}.
$$

Now define, $f(d) \triangleq \Pr^{(M)}(Z^n = z^n)$. Since $f(d)$ is strictly log-convex[1] due to the fact that the gamma function is strictly log-convex and thus the product of two or more

[1] A positive-valued function f is said to be log-convex if $\log f$ is convex. Log-convex functions are convex.

gamma functions is also strictly log-convex, it has a unique minimum value. Let

$$d_0 = \frac{n}{2} + \frac{1-\varepsilon}{2\varepsilon}(M-1+\alpha)(1-2p).$$

Observe that for any $t > 0$, we have

$$\begin{aligned} f(d_0+t) &= C(M,n,\alpha,\varepsilon)\Gamma\left(\frac{n}{2} + \frac{1-\varepsilon}{2\varepsilon}(M-1+\alpha) - t\right) \\ &\quad \times\Gamma\left(\frac{n}{2} + \frac{1-\varepsilon}{2\varepsilon}(M-1+\alpha) + t\right) \\ &= f(d_0 - t). \end{aligned}$$

Therefore, $f(d)$ is symmetric about d_0, and therefore, the strict log-convexity of $f(d)$ implies that the minimum is attained at d_0.

When $\alpha = 1$, QBC $\Leftrightarrow$ FMCC; then the parameters of FMCC can be expressed in terms of the parameters of QBC as follows:

$$\begin{aligned} \delta &= \frac{\varepsilon}{(1-\varepsilon)(M-1+\alpha)} = \frac{\varepsilon}{(1-\varepsilon)M} \qquad \text{since } \alpha = 1, \\ \rho &= p. \end{aligned}$$

Thus

$$d_0 = \frac{n}{2} + \frac{1}{2\delta}(1-2\rho)$$

which is exactly the result derived in [1]. As a result, the decoding algorithm proposed in [1, Section III] applies to the case of the QBC as well. An optimal decoder will decode the received codeword y^n as following, assuming a codebook $\mathcal{C}$.

1. Compute $d_i \triangleq d(y^n, x_i^n)$ for $i = 1, \cdots, |\mathcal{C}|$ and $x_i^n \in \mathcal{C}$.

2. Compute $d_{max} \triangleq \max\{d_i\}$ and $d_{min} \triangleq \min\{d_i\}$.

3. If $|d_{max} - d_0| \leq |d_{min} - d_0|$, decode x_j^n for which $d_j = d_{min}$. Hence, ML decoding $\Leftrightarrow$ minimum distance (MD) decoding.

4. Otherwise, decode x_j^n for which $d_j = d_{max}$. Thus, ML decoding $\Leftrightarrow$ maximum distance decoding.

The above decoding method applies to the case where the blocklength n is smaller than the channel noise memory M. It can thus be useful for application involving delay-constrained wireless channels under deep or very slow fading (where the fading memory is larger than the coding blocklength).

4.2.2 Case II: $n > M$

In many communication systems codewords with long blocklength are desirable since they offer strong performance in noisy environments with the possibility of reaching Shannon's capacity limit (e.g., LDPC codes). Thus, the memory of the wireless fading channels is often smaller than the blocklength of the transmitted codeword. As a result, studying the case in which $n > M$ is important. However, such analysis is not straightforward for channels with memory in general and the QBC in particular. In this section, the block transition probability for the QBC under the assumption that $n > M$ is revisited. We derive a different expression for this probability that will facilitate the analysis of the BAMNC with non-negative noise correlation (i.e., the QBC with $M = 1$). Later, the performance of binary perfect and quasi-perfect linear block codes is analyzed and a new decoding algorithm is proposed that is nearly equivalent to ML decoding for a range of channel conditions.

The block transition probability for the QBC for the case $n > M$ is given by (2.17):

$$\Pr{}^{(M)}(Z^n = z^n) = L^{(M)} \prod_{i=M+1}^{n} \left[\left(d_{i-M+1}^{i-1} + \alpha z_{i-M} \right) \frac{\varepsilon}{M-1+\alpha} + (1-\varepsilon)p \right]^{z_i}$$
$$\left\{ \left[\left(M - 1 - d_{i-M+1}^{i-1} \right) + \alpha(1 - z_{i-M}) \right] \frac{\varepsilon}{M-1+\alpha} + (1-\varepsilon)(1-p) \right\}^{1-z_i},$$

where

$$L^{(M)} = \frac{\prod_{j=0}^{M-1-d_1^M} \left[j \frac{\varepsilon}{M-1+\alpha} + (1-\varepsilon)(1-p) \right] \prod_{j=0}^{d_1^M - 1} \left[j \frac{\varepsilon}{M-1+\alpha} + (1-\varepsilon)p \right]}{\prod_{j=0}^{M-1} \left[1 - (\alpha + j) \frac{\varepsilon}{M-1+\alpha} \right]}.$$

Assume $\alpha = 1$, then the QBC reduces to the FMCC and the above expression gets simplified to

$$\Pr{}^{(M)}(Z^n = z^n) = L^{(M)} \prod_{i=M+1}^{n} \left[d_{i-M}^{i-1} \frac{\varepsilon}{M} + (1-\varepsilon)p \right]^{z_i} \left[(M - d_{i-M}^{i-1}) \frac{\varepsilon}{M} + (1-\varepsilon)(1-p) \right]^{1-z_i} \tag{4.2}$$

where

$$L^{(M)} = \frac{\prod_{j=0}^{M-1-d_1^M} \left[j \frac{\varepsilon}{M} + (1-\varepsilon)(1-p) \right] \prod_{j=0}^{d_1^M - 1} \left[j \frac{\varepsilon}{M} + (1-\varepsilon)p \right]}{\prod_{j=0}^{M-1} \left[1 - (1 + j) \frac{\varepsilon}{M} \right]}.$$

For any given i, to evaluate the i^{th} term in the product in (4.2), we only need the Hamming weight of the last M noise bits as well as the value of the current i^{th} bit. In other words, if $z_i = z_j$ and $\sum_{k=1}^{M} z_{i-k} = \sum_{k=1}^{M} z_{j-k}$ then the i^{th} and the j^{th} terms are the same for $M + 1 \leq i, j \leq n$.

Define $t_{i,j}(z^n)$ as follows:

$$t_{i,j}(z^n) = \sum_{k=M+1}^{n} \delta\left(\sum_{r=k-M}^{k-1} z_r, i\right) \delta(z_k, j), \tag{4.3}$$

where $i = 0, 1, \cdots, M$, $j = 0, 1$ and $\delta(x, y)$ is the Kronecker-delta function

$$\delta(x, y) = \begin{cases} 1 & \text{if } x = y, \\ 0 & \text{if } x \neq y. \end{cases}$$

Note that $t_{i,j}(z^n)$ is the number of occurrences of a binary $M+1$-tuple whose first M bits (counting from left) are of Hamming weight i and the last bit (i.e., the $(M+1)^{st}$ bit) is j in the binary n-tuple z^n.

Example: Consider the case where $M = 1$, $n = 5$ and $z^n = (z_1, z_2, z_3, z_4, z_5) = (1, 0, 1, 1, 0)$. Then

$$\begin{aligned} t_{i,j}(z^n) &= \sum_{k=2}^{5} \delta(z_{k-1}, i)\delta(z_k, j) \\ &= \delta(z_1, i)\delta(z_2, j) + \delta(z_2, i)\delta(z_3, j) + \delta(z_3, i)\delta(z_4, j) + \delta(z_4, i)\delta(z_5, j), \\ \Rightarrow t_{1,0}(z^n) &= 1 + 0 + 0 + 1 = 2, \\ t_{0,1}(z^n) &= 0 + 1 + 0 + 0 = 1. \end{aligned}$$

Now we can re-write (4.2) as

$$\Pr(Z^n = z^n) = L^{(M)} \cdot g(\mathbf{t}(z^n)) \tag{4.4}$$

where

$$\begin{aligned}\mathbf{t}(z^n) &= (t_{0,0}(z^n), \cdots, t_{M,0}(z^n), t_{0,1}(z^n), \cdots, t_{M,1}(z^n)),\\ g(\mathbf{t}(z^n)) &= \prod_{i=0}^{M}\left(\left[\frac{M-i}{M}\varepsilon + (1-\varepsilon)(1-p)\right]^{t_{i,0}(z^n)}\left[\frac{i}{M}\varepsilon + (1-\varepsilon)p\right]^{t_{i,1}(z^n)}\right)\end{aligned}$$

$$\text{and} \sum_{j=0}^{M} (t_{j,0}(z^n) + t_{j,1}(z^n)) = n - M.$$

Now we extend our discussion to the general QBC with $\alpha \geq 1$.

In this case to evaluate the i^{th} term in $\Pr^{(M)}(Z^n = z^n)$, we need 3 pieces of information: the i^{th} noise bit, the $(i-M)^{th}$ noise bit, and the Hamming weight of the noise bits in between. Let $b_{i,j,k}$ represent the sequence of $M+1$ noise bits whose leftmost and rightmost bits are i and k, respectively, and whose bits in between have a Hamming weight equal to j. Let $t_{i,j,k}(z^n)$ be the number of times $b_{i,j,k}$ appears in z^n. Then, $t_{i,j,k}(z^n)$ is given by

$$t_{i,j,k}(z^n) = \sum_{s=M+1}^{n} \delta(z_{s-M}, i)\delta\left(\sum_{r=s-M+1}^{s-1} z_r, j\right)\delta(z_s, k).$$

It follows that the block transition probability can be re-written as:

$$\Pr(Z^n = z^n) = L^{(M)} \cdot g_\alpha(\mathbf{t}_\alpha(z^n))$$

where

$$\mathbf{t}_\alpha(z^n) = (t_{0,0,0}, \cdots, t_{0,M-1,0}, t_{0,0,1}, \cdots, t_{0,M-1,1}, t_{1,0,0}, \cdots, t_{1,M-1,0}, t_{1,0,1}, \cdots, t_{1,M-1,1})$$

and

$$\begin{aligned}
&g_\alpha(\mathbf{t}_\alpha(z^n)) \\
&= \prod_{j=0}^{M-1}\left[\frac{(M-1+\alpha-i)\varepsilon}{M-1+\alpha}+(1-\varepsilon)(1-p)\right]^{t_{0,j,0}(z^n)}\prod_{j=0}^{M-1}\left[\frac{i\varepsilon}{M-1+\alpha}+(1-\varepsilon)p\right]^{t_{0,j,1}(z^n)} \\
&\quad \prod_{j=0}^{M-1}\left[\frac{(M-1-i)\varepsilon}{M-1+\alpha}+(1-\varepsilon)(1-p)\right]^{t_{1,j,0}(z^n)}\prod_{j=0}^{M-1}\left[\frac{(i+\alpha)\varepsilon}{M-1+\alpha}+(1-\varepsilon)p\right]^{t_{1,j,1}(z^n)}.
\end{aligned}$$

4.3 Block Transition Probability for the Binary Additive Markov Noise Channel

As noted earlier, the BAMNC with non-negative noise correlation is a special case of the QBC obtained by setting $M = 1$ (α is forced to be 1 in this case). The probability of block transition is given then by (4.2) by substituting $M = \alpha = 1$. For this particular case the $t_{ij}(z^n)$'s are as given by (4.3) after replacing $M = 1$ (for the sake of simplicity, $t_{i,j}(\cdot)$ is written as $t_{ij}(\cdot)$). Then

$$\begin{aligned}
t_{ij}(z^n) &= \sum_{k=2}^{n}\delta\left(z_{k-1},i\right)\delta(z_k,j) \\
&= \sum_{k=1}^{n-1}\delta\left(z_k,i\right)\delta(z_{k+1},j).
\end{aligned}$$

Observe that

$$\delta(z_k,i)=\begin{cases}(1-z_k) & \text{if } i=0,\\ z_k & \text{if } i=1.\end{cases}$$

Hence,

$$
\begin{aligned}
t_{00}(z^n) &= \sum_{k=1}^{n-1} (1-z_k)(1-z_{k+1}), \\
t_{11}(z^n) &= \sum_{k=1}^{n-1} z_k z_{k+1}, \\
t_{10}(z^n) &= \sum_{k=1}^{n-1} z_k(1-z_{k+1}), \\
t_{01}(z^n) &= \sum_{k=1}^{n-1} (1-z_k) z_{k+1}.
\end{aligned}
$$

Also, $L^{(1)}$ is

$$
L^{(1)} = \begin{cases} 1-p & \text{if } z_1 = 0, \\ \\ p & \text{if } z_1 = 1, \end{cases}
$$

which can be rewritten as

$$
L^{(1)} = p^{z_1}(1-p)^{(1-z_1)}.
$$

Therefore, in terms of the $t_{ij}(z^n)$'s $\Pr(Z^n = z^n)$ can be written as

$$
\begin{aligned}
\Pr(Z^n = z^n) &= L^{(1)} \left[\varepsilon + (1-\varepsilon)(1-p)\right]^{t_{00}(z^n)} \left[(1-\varepsilon)p\right]^{t_{01}(z^n)} \\
&\quad \times \left[(1-\varepsilon)(1-p)\right]^{t_{10}(z^n)} \left[\varepsilon + (1-\varepsilon)p\right]^{t_{11}(z^n)}. \qquad (4.5)
\end{aligned}
$$

But from the definition of the $t_{ij}(z^n)$'s, we have the following.

$$
t_{10}(z^n) = n - 1 - w(z^n) - t_{00}(z^n) + z_1 \qquad (4.6)
$$

$$
t_{01}(z^n) = w(z^n) - z_1 - t_{11}(z^n), \qquad (4.7)
$$

where $w(z^n) = \sum_{k=1}^{n} z_k$ is the Hamming weight of z^n. Substituting (4.6) and (4.7) into (4.5) yields the following expression for the noise block distribution, which will be instrumental in our analysis.

$$\Pr(Z^n = z^n) = (1-\varepsilon)^{(n-1)}(1-p)^n A^{t_{00}(z^n)} B^{t_{11}(z^n)} \left[\frac{p}{1-p}\right]^{w(z^n)} \tag{4.8}$$

where

$$A = \left[\frac{\varepsilon + (1-\varepsilon)(1-p)}{(1-\varepsilon)(1-p)}\right] \quad \text{and} \quad B = \left[\frac{\varepsilon + (1-\varepsilon)p}{(1-\varepsilon)p}\right].$$

Proposition 4.1 *The properties of $t_{00}(\mathrm{z}^n)$ and $t_{11}(\mathrm{z}^n)$ in terms of only n and $w(\mathrm{z}^n)$ are as follows.*

1. *If $w(z^n) = 0$, then $t_{00}(z^n) = n-1$ and $t_{11}(z^n) = 0$.*

2. *If $0 < w(z^n) = l \leq n-1$, then*

$$t_{00}(z^n) \leq n - l - 1$$

with equality if and only if all the 0's in z^n occur consecutively. Also

$$t_{11}(z^n) \leq l - 1$$

with equality if and only if all the 1's in z^n occur consecutively.

3. *If $0 < w(\mathrm{z}^n) = l \leq \frac{n}{2}$, then*

$$t_{00}(z^n) \geq \max\{n - 2l - 1, 0\}$$

and

$$t_{11}(z^n) \geq 0.$$

4. *If* $\frac{n}{2} < w(z^n) = l \leq n-1$, *then*

$$t_{00}(z^n) \geq 0$$

and

$$t_{11}(z^n) \geq 2l - n - 1.$$

5. *If* $w(z^n) = n$, *then* $t_{11}(z^n) = n-1$ *and* $t_{00}(z^n) = 0$.

Proof.

(1) and (5) are trivial, while (4) can be proved from (3) by swapping the zeros and ones.

(2) follows directly by writing

$$\begin{aligned} t_{00}(z^n) &= \sum_{k=1}^{n-1}(1-z_k)(1-z_{k+1}) = \sum_{k=1}^{n-1} I(z_k = z_{k+1} = 0) \leq n - l - 1 \\ t_{11}(z^n) &= \sum_{k=1}^{n-1} z_k z_{k+1} = \sum_{k=1}^{n-1} I(z_k = z_{k+1} = 1) \leq l - 1 \end{aligned}$$

where $I(\cdot)$ is the indicator function.

To prove (3), we first note from (4.6) that

$$t_{00}(z^n) = n - 1 - w(z^n) - t_{10}(z^n) + z_1.$$

Note that if $w(z^n) = l \leq \frac{n}{2}$, then the maximum value for $t_{10}(z^n)$ is l which occurs if and only if every 1 in z^n is followed by a 0. Also, $z_1 \geq 0$. Thus,

$$t_{00}(z^n) \geq n - 1 - l - l + 0 = n - 2l - 1.$$

Finally, it is obvious that $t_{00}(z^n)$ and $t_{11}(z^n)$ are non-negative quantities. □

When there is no possibility for confusion, we will write $t_{00}(z^n)$ and $t_{11}(z^n)$ as t_{00} and t_{11}, respectively. We also assume throughout that the blocklength $n \geq 2$.

4.3.1 Analysis of the Noise Block Distribution

Lemma 4.1 *Let 0^n be the all-zero word (of length n) and let $z^n \neq 0^n$ be any non-zero binary word. Then*

$$\Pr(Z^n = z^n) < \Pr(Z^n = 0^n).$$

Proof. Using (4.5), we have

$$\begin{aligned}
\Pr(Z^n = z^n) &= L\left[\varepsilon + (1-\varepsilon)(1-p)\right]^{t_{00}} \left[(1-\varepsilon)p\right]^{t_{01}} \left[(1-\varepsilon)(1-p)\right]^{t_{10}} \left[\varepsilon + (1-\varepsilon)p\right]^{t_{11}} \\
&< (1-p)\left[\varepsilon + (1-\varepsilon)(1-p)\right]^{t_{00}} \left[\varepsilon + (1-\varepsilon)(1-p)\right]^{t_{01}} \\
&\qquad \cdot \left[\varepsilon + (1-\varepsilon)(1-p)\right]^{t_{10}} \left[\varepsilon + (1-\varepsilon)(1-p)\right]^{t_{11}} \\
&= (1-p)\left[\varepsilon + (1-\varepsilon)(1-p)\right]^{t_{00}+t_{01}+t_{10}+t_{11}} \\
&= (1-p)\left[\varepsilon + (1-\varepsilon)(1-p)\right]^{n-1} \\
&= \Pr(Z^n = 0^n)
\end{aligned}$$

where the strict inequality holds since $L = p < 1 - p$ if $z_1 = 1$, and since $p < 1 - p$ with $t_{01} > 0$ (since $z^n \neq 0^n$) if $z_1 = 0$. □

Lemma 4.2 *Let $z_1^n \neq 0^n$ be a non-zero noise word with Hamming weight $w(z_1^n) < n$, $t_{00} = n - w(z_1^n) - 1$ and $t_{11} = w(z_1^n) - 1$* (i.e., z_1^n is of the form $(11\cdots 100\cdots 0)$ or $(00\cdots 011\cdots 1)$)*. Let z_2^n be another non-zero noise word with $w(z_2^n) = w(z_1^n)$ but with different t_{00} and/or t_{11}. Then, if $\varepsilon > 0$,*

$$\Pr(Z^n = z_1^n) > \Pr(Z^n = z_2^n).$$

Proof. From (4.8), we note that $\Pr(Z^n = z^n)$ strictly increases with both t_{00} and t_{11} when the weight is kept constant and $\varepsilon > 0$. Since z_1^n has maximum values for both t_{00} and t_{11} amongst all noise words of weight $w(z_1^n)$ (but with different t_{00} and/or t_{11}), the strict inequality above follows. □

Note that when $\varepsilon = 0$, obviously all noise words with the same weight have identical distributions (since the channel reduces to the BSC(p)).

Lemma 4.3 *Suppose that*

$$u < u^* \triangleq \frac{\ln\left[\frac{\varepsilon+(1-\varepsilon)(1-p)}{(1-\varepsilon)(1-p)}\right] + \ln\left[\frac{1-p}{p}\right]}{\ln\left[\frac{\varepsilon+(1-\varepsilon)(1-p)}{(1-\varepsilon)(1-p)}\right] + \ln\left[\frac{\varepsilon+(1-\varepsilon)p}{(1-\varepsilon)p}\right]} - 1$$

and

$$0 < \varepsilon < \frac{1-2p}{2(1-p)}.$$

Let z^n be a noise word of weight $w(z^n) = m$ such that $0 \leq m \leq u + 1 \leq \frac{n}{2}$. Then $\Pr(Z^n = z^n) > \Pr(Z^n = \bar{z}^n)$ *where $\bar{z}^n$ is any noise word with weight $w(\bar{z}^n) = l > m$.*

Proof. First, note that the result directly holds if $m = 0$ by Lemma 4.1. Now let z^n be a noise word of nonzero weight $m \leq u + 1$, and let $\bar{z}^n$ be another noise word with $w(\bar{z}^n) > m$.

Case 1: Assume that $w(\bar{z}^n) = m + i$ where $i \in \{1, 2, ..., n - m - 1\}$. Then by (4.8), we have

$$\begin{aligned}
&\frac{\Pr(Z^n = \bar{z}^n)}{\Pr(Z^n = z^n)} \\
= \; & \left[\frac{\varepsilon + (1-\varepsilon)(1-p)}{(1-\varepsilon)(1-p)}\right]^{t_{00}(\bar{z}^n) - t_{00}(z^n)} \left[\frac{\varepsilon + (1-\varepsilon)p}{(1-\varepsilon)p}\right]^{t_{11}(\bar{z}^n) - t_{11}(z^n)} \left(\frac{p}{1-p}\right)^{w(\bar{z}^n) - w(z^n)} \\
\leq \; & \left[\frac{\varepsilon + (1-\varepsilon)(1-p)}{(1-\varepsilon)(1-p)}\right]^{m-i} \left[\frac{\varepsilon + (1-\varepsilon)p}{(1-\varepsilon)p}\right]^{m+i-1} \left(\frac{p}{1-p}\right)^{i} \\
\leq \; & \left[\frac{\varepsilon + (1-\varepsilon)(1-p)}{(1-\varepsilon)(1-p)}\right]^{m-1} \left[\frac{\varepsilon + (1-\varepsilon)p}{(1-\varepsilon)p}\right]^{m} \left(\frac{p}{1-p}\right) \\
\triangleq \; & f(m).
\end{aligned}$$

The first inequality above results directly by applying the bounds on t_{00} and t_{11} assuming $m \leq \frac{n}{2}$ (see Proposition 4.1), while the second inequality is obtained by observing that the right hand side of the first inequality is a decreasing function of i for a fixed m. Since $f(m)$ is strictly increasing in m (when $\varepsilon > 0$), and $m \leq u+1 < u^*+1$, we obtain that

$$f(m) < f(u^* + 1) = 1 \Rightarrow \frac{\Pr(Z^n = \bar{z}^n)}{\Pr(Z^n = z^n)} < 1.$$

Case 2: Assume that $w(\bar{z}^n) = n$. Let $\hat{z}^n$ be another noise word with $w(\hat{z}^n) = n - 1$, $t_{11}(\hat{z}^n) = n - 2$ and $t_{00}(\hat{z}^n) = 0$. Then

$$\frac{\Pr(Z^n = \bar{z}^n)}{\Pr(Z^n = z^n)} = \frac{\Pr(Z^n = \hat{z}^n)}{\Pr(Z^n = z^n)} \frac{\Pr(Z^n = \bar{z}^n)}{\Pr(Z^n = \hat{z}^n)}$$

$$\begin{aligned}
&< \frac{\Pr(Z^n = \bar{z}^n)}{\Pr(Z^n = \hat{z}^n)} \\
&= \left[\frac{\varepsilon + (1-\varepsilon)p}{(1-\varepsilon)p}\right]\left(\frac{p}{1-p}\right) \\
&= \left[\frac{\varepsilon + (1-\varepsilon)p}{(1-\varepsilon)(1-p)}\right] < 1
\end{aligned}$$

where the first strict inequality holds since $\Pr(Z^n = \hat{z}^n) < \Pr(Z^n = z^n)$ by Case 1, and the last strict inequality holds since $\varepsilon < \frac{1-2p}{2(1-p)}$.

□

4.4 Decoding Perfect and Quasi-Perfect Codes on the BAMNC

Next, the relationship between strict maximum likelihood (SML) decoding and strict minimum (Hamming) distance decoding for binary linear perfect and quasi-perfect codes sent over the BAMNC is studied. As noted in Section 4.1, strict maximum likelihood (SML) decoding is an optimal (incomplete) decoder in the sense of minimizing the probability of codeword error (PCE) when the codewords are operated on with equal probability (which we herein assume). Since it is well known that ML decoding of binary codes over the memoryless binary symmetric channel (with bit error rate less than 1/2) is equivalent to minimum Hamming distance decoding, it is natural to investigate whether a relation exists between these two decoding methods for the

Markov noise channel. We provide a partial answer to this problem by showing that for binary linear perfect codes ML decoding and MD decoding are equivalent provided that the channel correlation is kept below a particular threshold. We also show that the strict ML decoding of binary linear quasi-perfect codes can be nearly equivalent to strict minimum distance decoding. As a result we propose a (complete) decoder which is an improved version of the minimum distance decoder, and we illustrate its performance via simulation results.

In a related work [17], Hamada showed that for the Markov channel with a non-negative correlation coefficient (i.e., $\varepsilon \geq 0$) and bit error rate $p < 1/2$, the binary perfect Hamming codes (of minimum distance 3) are optimal in the sense of minimizing the probability of decoding error amongst all codes having the same blocklength and rate provided that $\varepsilon < (1-2p)/2(1-p)$. Thus for a communication system employing codes with short blocklength due to delay constraints, Hamming codes used with MD decoding will be optimal over the BAMNC amongst all codes of the same blocklength and rate.

Lemma 4.4 *Let $\mathcal{C}$ be an $[n,k]$ perfect code with a minimum distance d to be used over the BAMNC. Assume that*

$$\left\lfloor \frac{d-1}{2} \right\rfloor < \frac{\ln\left[\frac{\varepsilon+(1-\varepsilon)p}{(1-\varepsilon)p}\right] + \ln\left[\frac{1-p}{p}\right]}{\ln\left[\frac{\varepsilon+(1-\varepsilon)(1-p)}{(1-\varepsilon)(1-p)}\right] + \ln\left[\frac{\varepsilon+(1-\varepsilon)p}{(1-\varepsilon)p}\right]}$$

and

$$0 < \varepsilon < \frac{1-2p}{2(1-p)}.$$

Then SMD and SML decoding are equivalent.

Proof. First note that for perfect codes, the element within each coset of minimum weight (i.e., the coset leader) is unique. Also notice that the coset leader is of weight less than or equal to $\lfloor (d-1)/2 \rfloor \leq n/2$. Assume that y^n is received; then $\exists \hat{c} \in \mathcal{C}$ which is unique such that $w(\hat{c} \oplus y^n) < w(c \oplus y^n)\ \forall\ c \neq \hat{c} \in \mathcal{C}$. Using Lemma 4.3 with $u = \lfloor (d-1)/2 \rfloor - 1$, we conclude that $\forall\ c \neq \hat{c} \in \mathcal{C}$

$$\begin{aligned} & \Pr(Z^n = \hat{c} \oplus y^n) > \Pr(Z^n = c \oplus y^n) \\ \Leftrightarrow\ & \Pr(Y^n = y^n | X^n = \hat{c}) > \Pr(Y^n = y^n | X^n = c). \end{aligned}$$

Hence, given a received word y^n, the codeword with the smallest Hamming distance to y^n will be the most likely codeword that was sent over the channel amongst all the codewords in $\mathcal{C}$. Therefore, SMD and SML decoding are equivalent. □

Observations:

- The above lemma also proves that for perfect codes MD and ML decoding are equivalent under the same assumptions on d, ε and p. This is because for such codes SMD and MD are the same due to the uniqueness of their coset leaders which results in no ties in the MD decoder. Similarly, the uniqueness of coset leaders coupled with the proof of the above lemma also imply that SML and ML are equivalent for the perfect codes under the range of channel parameters given by the lemma.

- By closely examining the proof of the optimality of Hamming codes over the BAMNC in [17], one can deduce that ML and MD decoding are identical for the same range of p and ε given in Lemma 4.4. In this work, however, we derive sufficient conditions under which ML decoding of perfect codes reduces to MD decoding. The issue of the optimality of the two perfect codes (the repetition code with odd blocklength and the [23,12] Golay code) is not addressed here.

Lemma 4.5 *Let $\mathcal{C}$ be an $[n,k,d]$ binary linear quasi-perfect code to be used over the BAMNC. Assume that*

$$\left\lfloor \frac{d-1}{2} \right\rfloor < t^* \triangleq \frac{\ln\left[\frac{\varepsilon+(1-\varepsilon)p}{(1-\varepsilon)p}\right] + \ln\left[\frac{1-p}{p}\right]}{\ln\left[\frac{\varepsilon+(1-\varepsilon)(1-p)}{(1-\varepsilon)(1-p)}\right] + \ln\left[\frac{\varepsilon+(1-\varepsilon)p}{(1-\varepsilon)p}\right]} - 1$$

and

$$0 < \varepsilon < \frac{1-2p}{2(1-p)}.$$

Then, for a given word y^n received at the channel output, the following hold.

(a) If $\exists\ \hat{\mathrm{c}} \in \mathcal{C}$ such that $w(\hat{\mathrm{c}} \oplus \mathrm{y}^n) < w(c \oplus y^n)\ \forall\ \mathrm{c} \neq \hat{c} \in \mathcal{C}$, then $\Pr(Y^n = \mathrm{y}^n | X^n = \hat{\mathrm{c}}) > \Pr(Y^n = \mathrm{y}^n | X^n = \mathrm{c})\ \forall\ c \neq \hat{c} \in \mathcal{C}$.

(b) If $\exists\ \hat{\mathrm{c}} \in \mathcal{C}$ such that $\Pr(Y^n = \mathrm{y}^n | X^n = \hat{\mathrm{c}}) > \Pr(Y^n = \mathrm{y}^n | X^n = \mathrm{c})\ \forall\ c \neq \hat{c} \in \mathcal{C}$, then $w(\hat{\mathrm{c}} \oplus \mathrm{y}^n) \leq w(\mathrm{c} \oplus \mathrm{y}^n)\ \forall\ c \in \mathcal{C}$.

Proof.

(a) Let $\hat{c} \in \mathcal{C}$ such that $w(\hat{c} \oplus y^n) < w(c \oplus y^n)\ \forall\ c \neq \hat{c} \in \mathcal{C}$. Obviously, $\hat{c} \oplus y^n$ is a coset leader, thus $w(\hat{c} \oplus y^n) \leq \left\lfloor \frac{d-1}{2} \right\rfloor + 1 \leq \frac{n}{2}$ since $\mathcal{C}$ is quasi-perfect. By Lemma 4.3, $\Pr(Z^n = \hat{c} \oplus y^n) > \Pr(Z^n = c \oplus y^n)\ \forall c \in \mathcal{C} \iff \Pr(Y^n = y^n | X^n = \hat{c}) > \Pr(Y^n = y^n | X^n = c)\ \forall\ c \neq \hat{c} \in \mathcal{C}$.

(b) Let $\hat{c} \in \mathcal{C}$ such that $\Pr(Y^n = y^n | X^n = \hat{c}) > \Pr(Y^n = y^n | X^n = c)\ \forall\ c \neq \hat{c} \in \mathcal{C}$. Assume that $\exists \bar{c} \neq \hat{c} \in \mathcal{C}$ such that $w(\bar{c} \oplus y^n) < w(\hat{c} \oplus y^n)$; the existence of $\bar{c}$ is always guaranteed by choosing it such that $\bar{c} \oplus y^n$ is the coset leader of $\mathcal{C} \oplus y^n$. Thus, we can assume that $w(\bar{c} \oplus y^n) \leq \frac{n}{2}$ since the coset leader has weight less than or equal to $\frac{n}{2}$ (as $\mathcal{C}$ is quasi-perfect). Then by Lemma 4.3, $\Pr(Z^n = \hat{c} \oplus y^n) < \Pr(Z^n = \bar{c} \oplus y^n) \iff \Pr(Y^n = y^n | X^n = \hat{c}) < \Pr(Y^n = y^n | X^n = \bar{c})$ which contradicts our assumption that $\hat{c}$ maximizes $\Pr(y^n | c)$ over all codewords. Hence, $w(\hat{c} \oplus y^n) \leq w(c \oplus y^n)\ \forall\ c \in \mathcal{C}$. □

Observation: Note that Lemma 4.5 implies that if a quasi-perfect code has no decoding failures in its SMD decoder, then its SMD and SML decoders are equivalent under the stated conditions on the Markov channel parameters (p, ε) and the code's minimum distance.[2]

In light of the above result and Lemma 4.2, we next propose the following complete decoder that improves over MD decoding. It includes SMD decoding and exploits the

[2] In contrast, recall that for the BSC(p) with $p < 1/2$, SML and SMD decoding are equivalent for all binary codes (the same equivalence also holds between ML and MD decoding). Note also that when $\varepsilon \downarrow 0$, the conditions in the above lemma reduce to $\left\lfloor \frac{d-1}{2} \right\rfloor < \infty$, and $p < \frac{1}{2}$ (which is consistent with what was just mentioned).

knowledge of t_{00} and t_{11} to resolve ties (which occur when there are more than one codeword that are closest to the received word).

MD+ Decoding:

Assume that y^n is received at the channel output. Suppose the decoder outputs the codeword c_0 satisfying the MD decoding condition. If there is more than one such codeword, then the decoder chooses c_0 that maximizes $t_{00}(c_0 \oplus y^n) + t_{11}(c_0 \oplus y^n)$. If there is still a tie, then the decoder chooses c_0 that maximizes $t_{11}(c_0 \oplus y^n)$. Finally, if there is still a tie, then the codeword c_0 is picked at random.[3] The advantage of the MD+ decoding over the ML decoding is the complexity. Whereas ML decoding requires an exhaustive search, MD+ can be implemented using syndrome decoding where the coset leaders are chosen according to the MD+ criteria.

4.5 Simulation Results

4.5.1 Perfect Codes

We examine the [15,11,3] Hamming code under different channel conditions, and show that indeed MD decoding and ML decoding are equivalent for the channel conditions specified by Lemma 4.4, as illustrated in Table 4.1 with $\varepsilon \leq \varepsilon_{t-1}$ for $t = 1, 2, 3$. Typical

[3] Clearly, MD+ and MD decoding are equivalent for the BSC, since for this channel, it does not matter what codeword the decoder selects when there is a tie (as long as it is one of the codewords closest to the received word).

values are shown for $\varepsilon \in \{0.1, 0.5, 0.9\}$ in Figs. 4.1 and 4.2. Note that $\varepsilon = 0.1$ satisfies the conditions of Lemma 4.4 while $\varepsilon = 0.5$ and $\varepsilon = 0.9$ do not. The simulation results show that MD and ML are identical for the case $\varepsilon = 0.1$ and almost identical at $\varepsilon = 0.5$. Finally, for high correlations, we observe that MD decoding is slightly worse than ML. When MD+ decoding is implemented we see that it does not show any improvement over MD decoding (as expected, since there are no ties in MD decoding for a perfect code).

4.5.2 Quasi-Perfect Codes

Given an $[n, k, d]$ quasi-perfect code and a fixed CBER p, we let ε_t be the largest ε for which both conditions of Lemma 4.5 hold, where $t \triangleq \lfloor (d-1)/2 \rfloor$. In Table 4.1, we provide the values of ε_t for $t = 1, 2, 3$ and different values of p.

We herein present simulation results for decoding the binary $[8, 4, 4]$ extended Hamming code and the $[15, 7, 5]$ BCH code over the BAMNC. A large sequence of a uniformly distributed binary i.i.d. source was generated, encoded via one of these codes and sent over the channel. For the extended Hamming code, $t = 1$; thus the values for ε_1 in Table 4.1 provide the largest values of ε for which Lemma 4.5 holds for different CBERs p. As a result, we simulated the Hamming system for the 5 values of p listed in Table 4.1 and $\varepsilon \in \{0.05, 0.1, 0.2, 0.25\}$. Note that these values are less than ε_1 values in Table 4.1. We then ran simulations for ε values that are larger than ε_1 in the same table: $\varepsilon \in \{0.5, 0.9, 0.99\}$. Similarly, since $t = 2$ for the BCH code, the values for ε_2 apply, and the BCH system was simulated for $\varepsilon = 0.05$ and all values of p in Table 4.1 except $p = 10^{-3}$. Additional simulations were conducted for larger correlations: $\varepsilon \in \{0.1, 0.9\}$. The extended Hamming code simulation results are presented in Figs. 4.3 – 4.9 for the sets of ε indicated above, and the BCH code simulations is shown in Figs. 4.10 – 4.12 for different values of ε.

Discussion

First we consider the figures obtained for the values of ε for which Lemma 4.5 holds. Figs. 4.3 – 4.6 and 4.10 indicate that MD+ performs nearly identically to ML decoding and provides significant gain over MD decoding. We also note that the performance gap between MD and ML decoding decreases with ε (which is consistent with the fact that MD and ML decoding are equivalent when $\varepsilon = 0$). Next, we look at the figures obtained for the ε values that are larger than the previous ones. We observe that the performance of the two given codes when the channel noise is highly correlated depends on the code. For instance, the PCE performance of the different decoding schemes applied to the extended Hamming code are almost identical when the channel correlation is very high (e.g., for $\varepsilon = 0.99$), whereas the ML decoded BCH code at $\varepsilon = 0.9$ for example is much better than the other two decoding schemes which seem to have similar performance; see Figs. 4.9 and 4.12.

Finally, note that one limitation of Lemma 4.5 is that its conditions are too stringent to accommodate quasi-perfect codes with large minimum distance, unless if the channel correlation ε is substantially decreased towards 0, thus rendering the Markov channel nearly memoryless (e.g, see how ε_t decreases as t increases in Table 4.1). The determination of less stringent conditions is an interesting topic for future work.

4.5.3 Non-Perfect Non-Quasi-Perfect Codes

The intention of this part is to examine the performance of any binary code that is neither perfect nor quasi-perfect under MD, MD+ and ML decoding. For this purpose, we chose the BCH [15,5] code, which is a triple-error correcting code. Simulation results for this code are shown in Figs 4.13 – 4.15. We note, that for correlations $\varepsilon \leq 0.05$, the MD+ and ML decoders perform similarly. The performance of the MD+ decoder degrades as ε increases. On the other hand, the MD decoder always yields the worst performance. As ε increases, MD and MD+ performance become closer to each other (see Figs. 4.13 – 4.15).

p	ε_0	ε_1	ε_2	ε_3
1×10^{-3}	499/999	0.3172	0.02843	0.08801
5×10^{-3}	99/199	0.3152	0.05628	0.02277
1×10^{-2}	49/99	0.3126	0.07297	0.03308
5×10^{-2}	9/19	0.2918	0.11492	0.06644
1×10^{-1}	4/9	0.2645	0.12367	0.07995

Table 4.1: Values of ε_t for different p and t. Lemma 4.4 holds for all $\varepsilon \leq \varepsilon_{t-1}$ and Lemma 4.5 holds for all $\varepsilon \leq \varepsilon_t$.

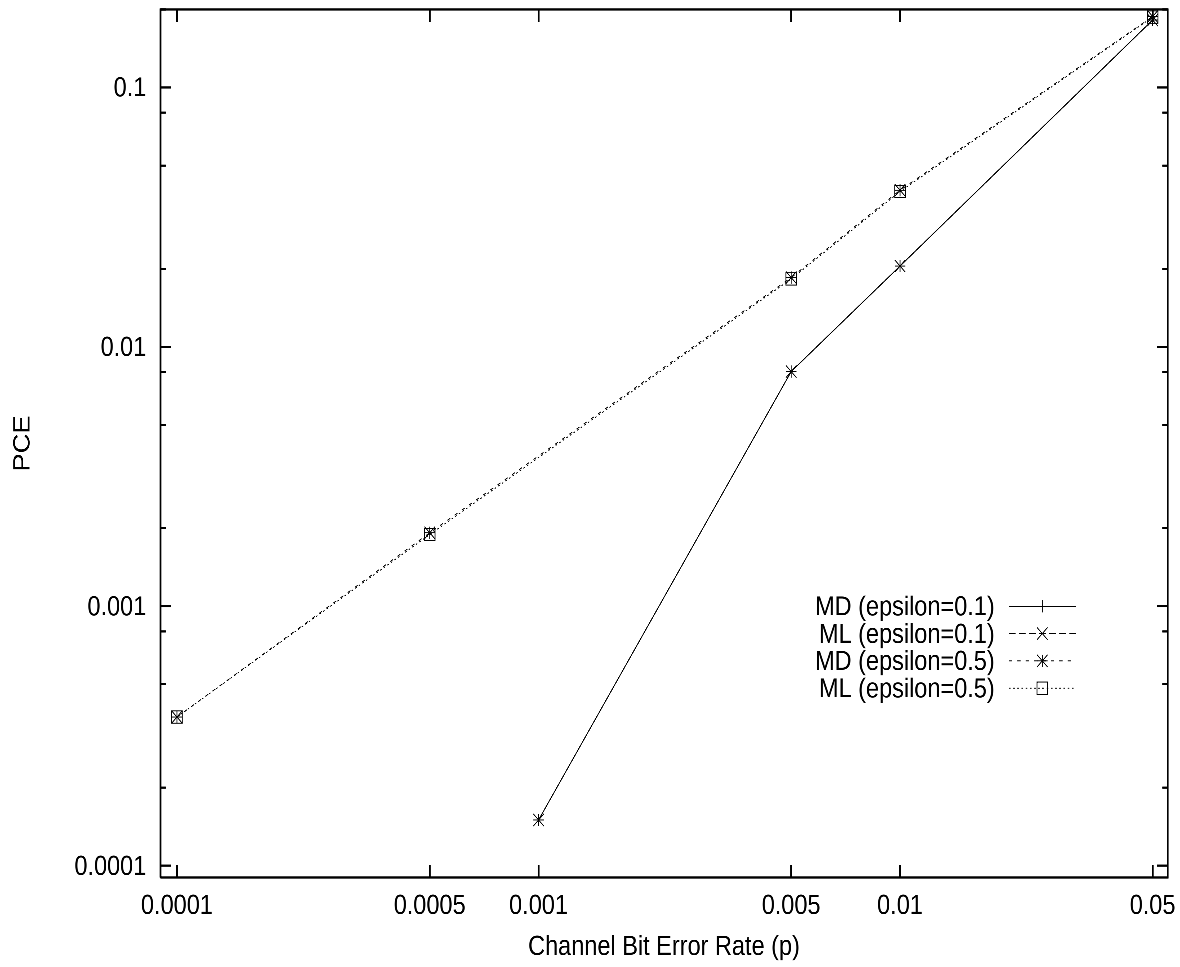


Figure 4.1: PCE vs CBER p under different decoding schemes for the Hamming $[15, 11, 3]$ code over the BAMNC with noise correlation $\varepsilon = 0.1, 0.5$.

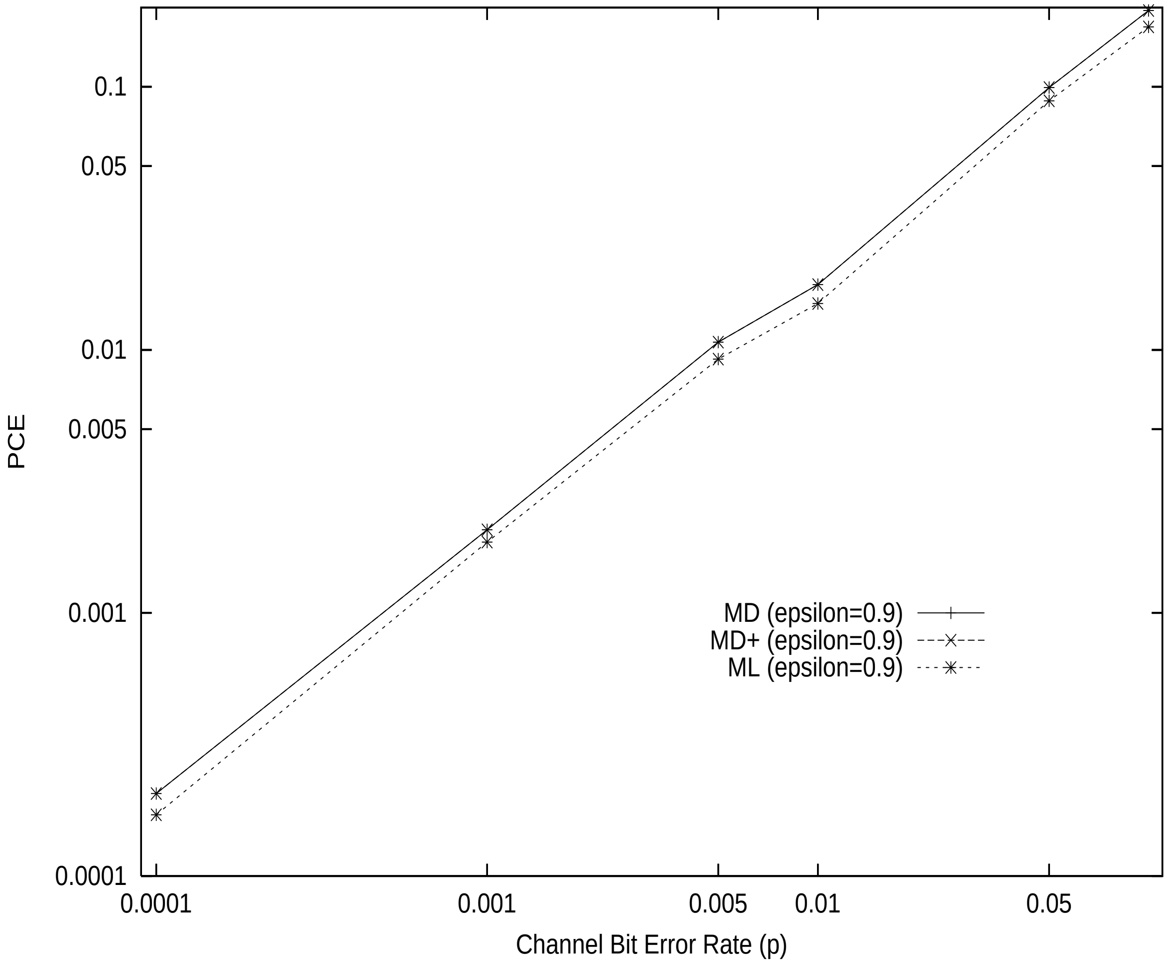


Figure 4.2: PCE vs CBER p under different decoding schemes for the Hamming $[15, 11, 3]$ code over the BAMNC with noise correlation $\varepsilon = 0.9$.

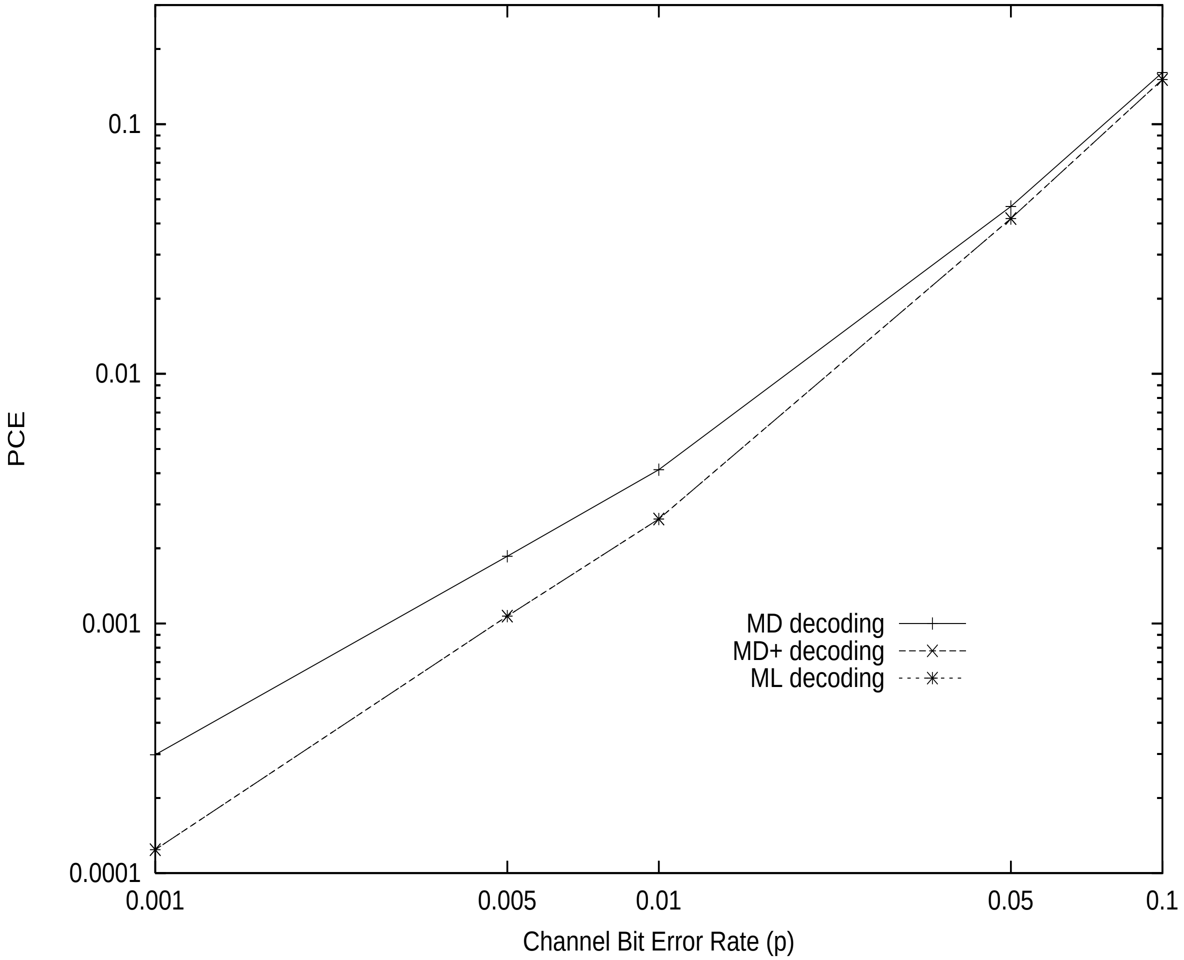


Figure 4.3: PCE vs CBER p under different decoding schemes for the extended Hamming $[8, 4, 4]$ code over the BAMNC with noise correlation $\varepsilon = 0.05$.

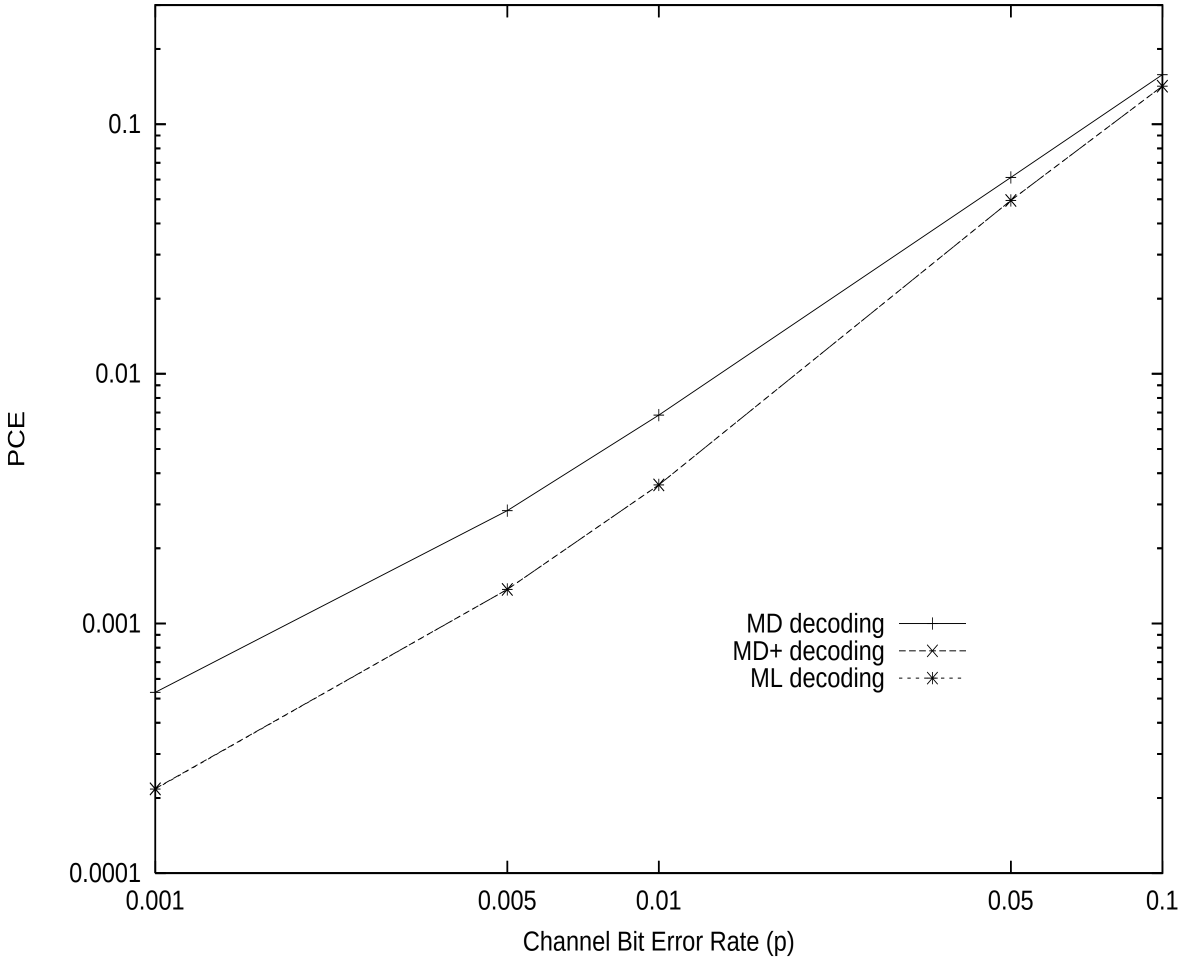


Figure 4.4: PCE vs CBER p under different decoding schemes for the extended Hamming $[8, 4, 4]$ code over the BAMNC with noise correlation $\varepsilon = 0.1$.

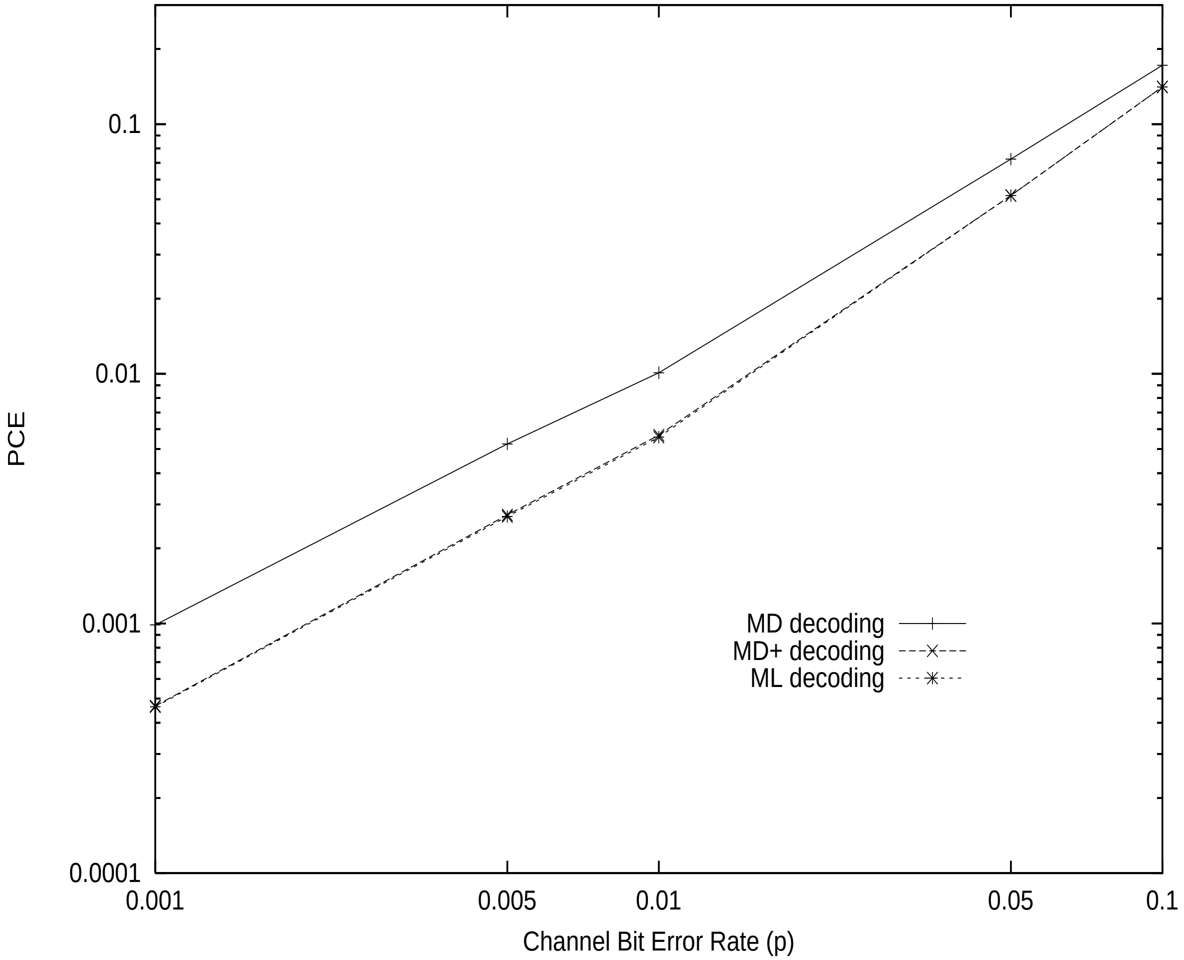


Figure 4.5: PCE vs CBER p under different decoding schemes for the extended Hamming $[8, 4, 4]$ code over the BAMNC with noise correlation $\varepsilon = 0.2$.

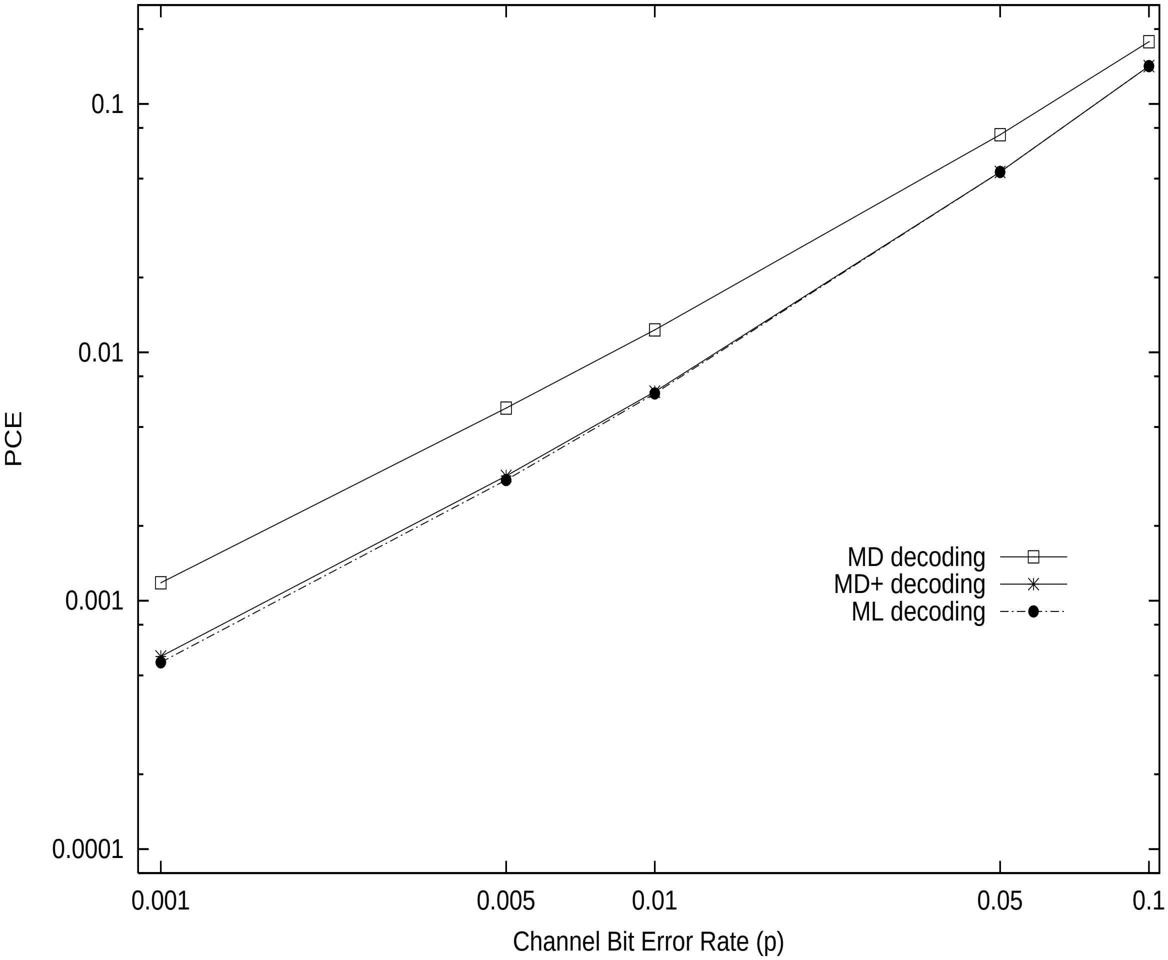


Figure 4.6: PCE vs CBER p under different decoding schemes for the extended Hamming $[8,4,4]$ code over the BAMNC with noise correlation $\varepsilon = 0.25$.

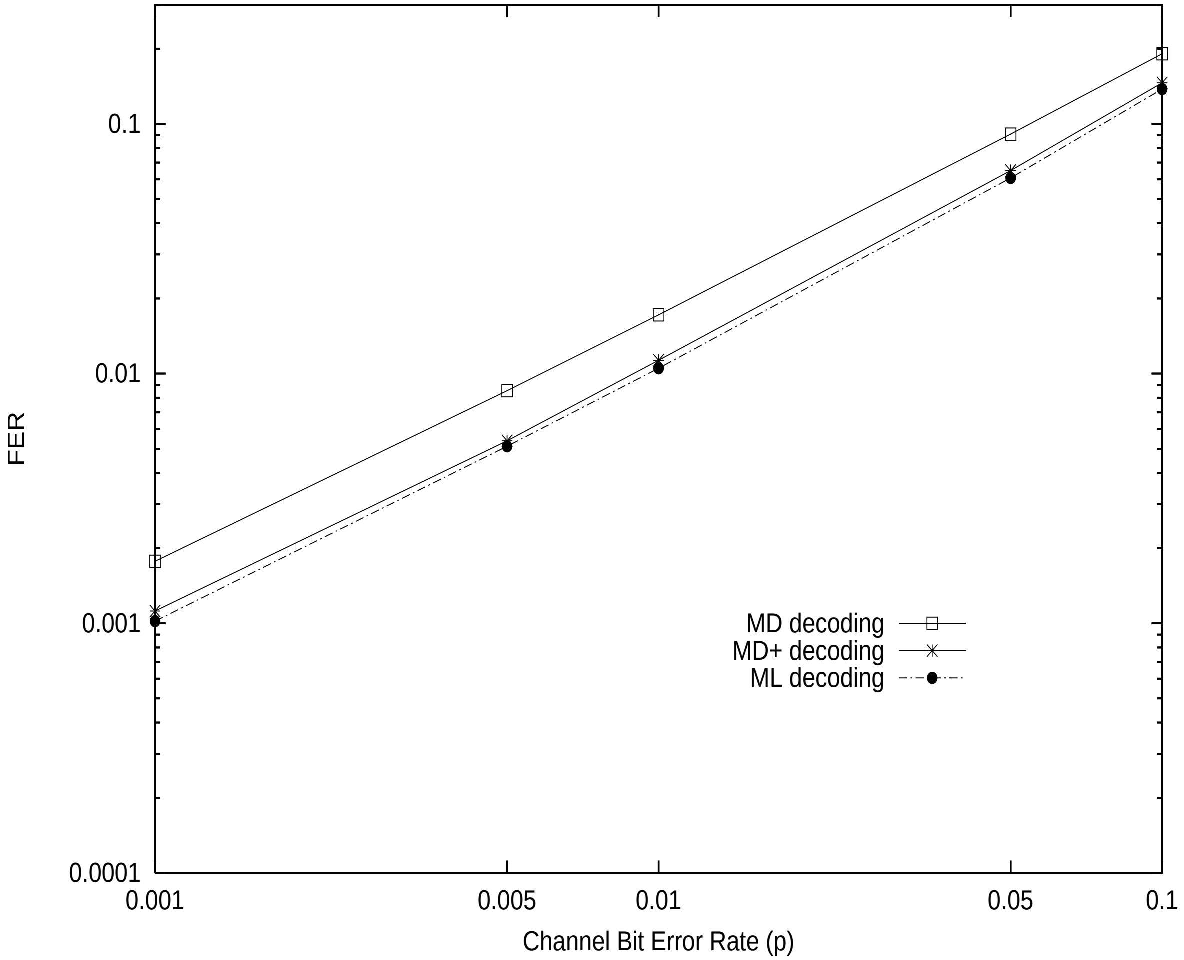


Figure 4.7: PCE vs CBER p under different decoding schemes for the extended Hamming $[8, 4, 4]$ code over the BAMNC with noise correlation $\varepsilon = 0.5$.

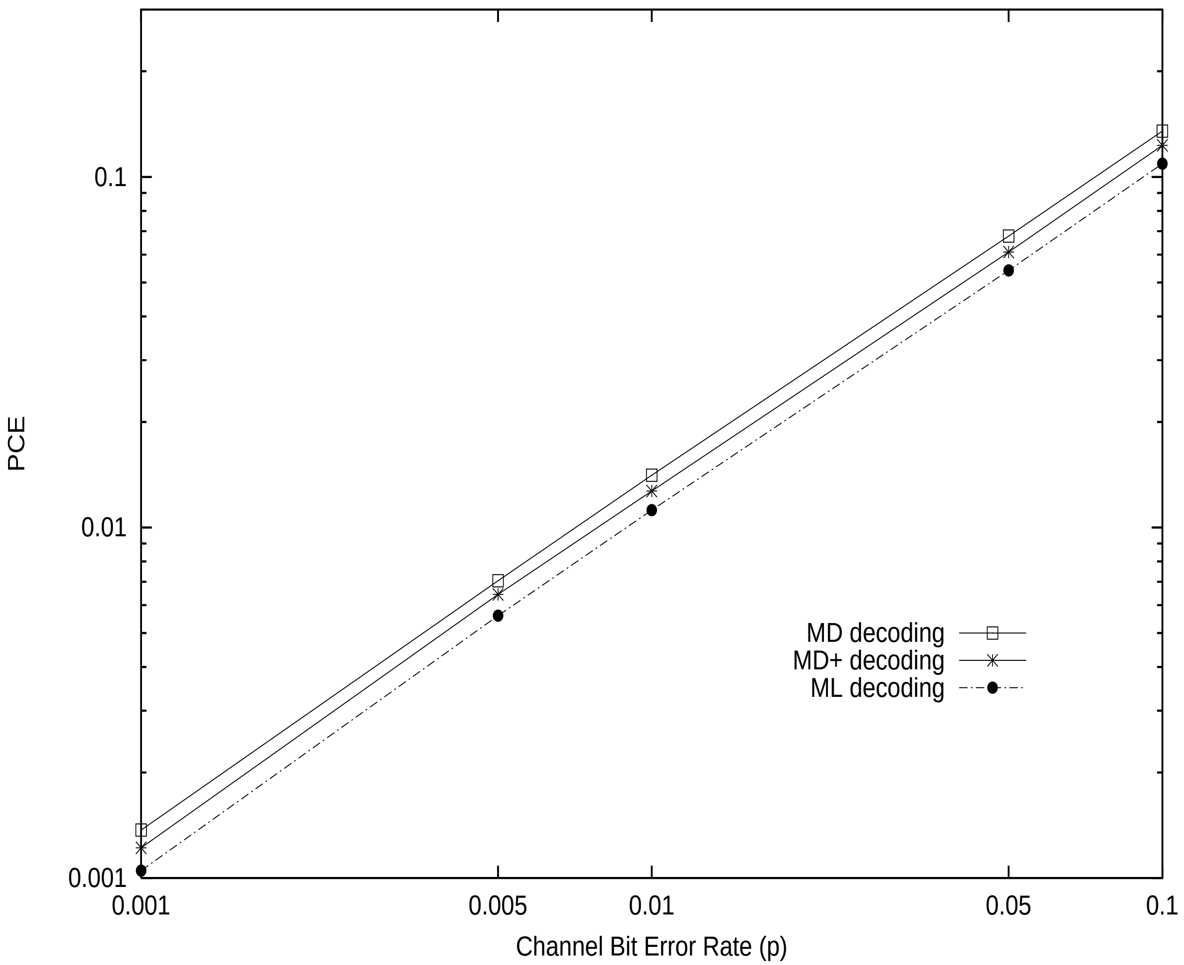


Figure 4.8: PCE vs CBER p under different decoding schemes for the extended Hamming $[8, 4, 4]$ code over the BAMNC with noise correlation $\varepsilon = 0.9$.

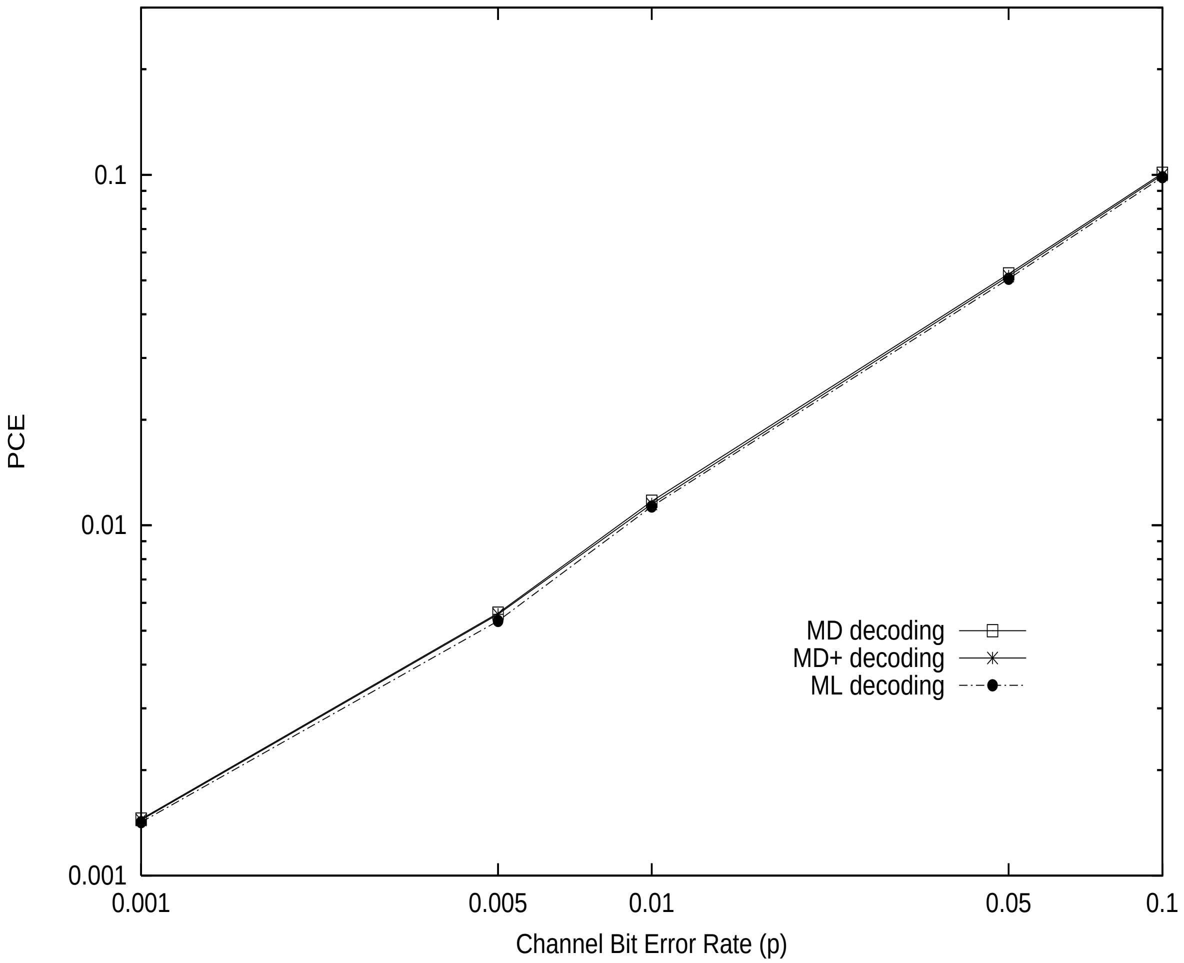


Figure 4.9: PCE vs CBER p under different decoding schemes for the extended Hamming $[8, 4, 4]$ code over the BAMNC with noise correlation $\varepsilon = 0.99$.

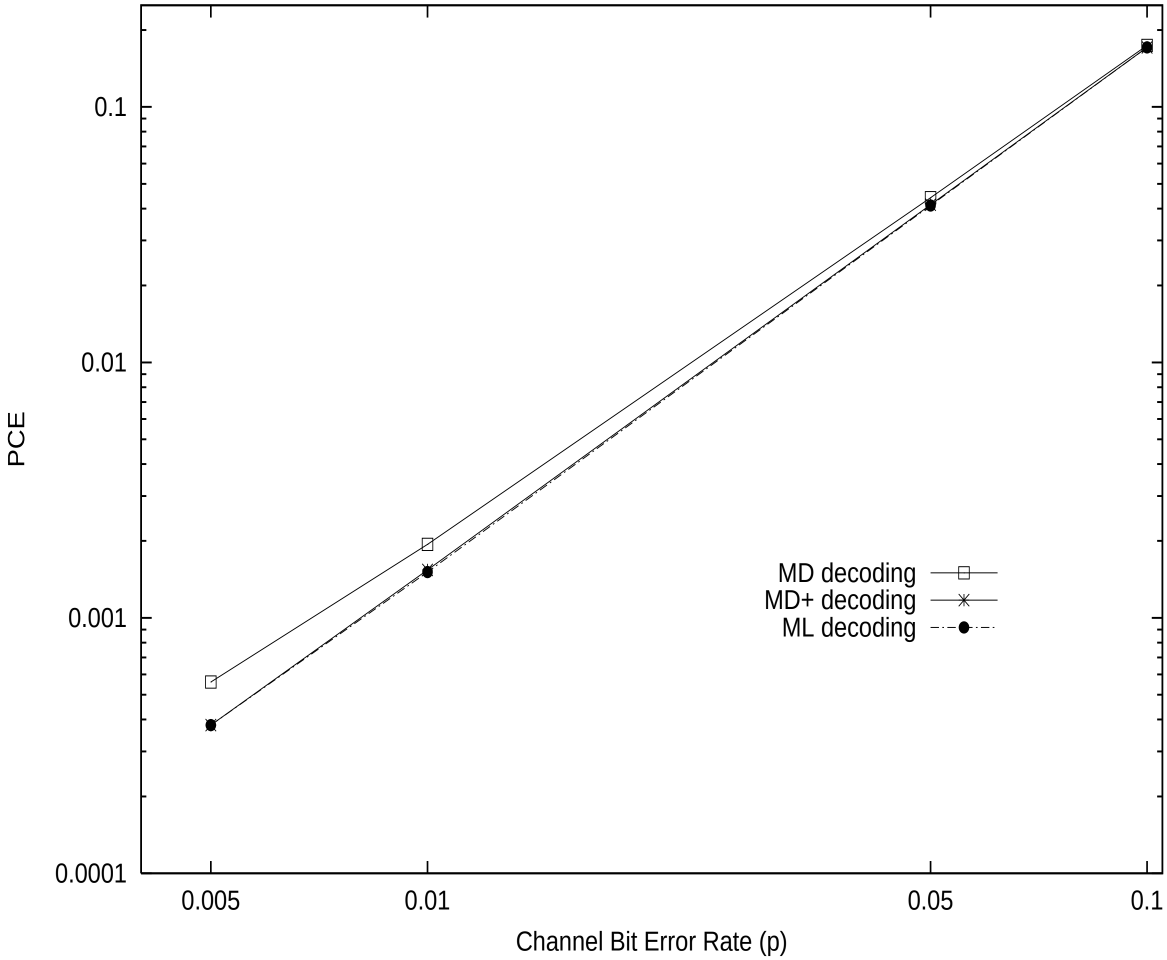


Figure 4.10: PCE vs CBER p under different decoding schemes for the BCH $[15, 7, 5]$ code over the BAMNC with noise correlation $\varepsilon = 0.05$.

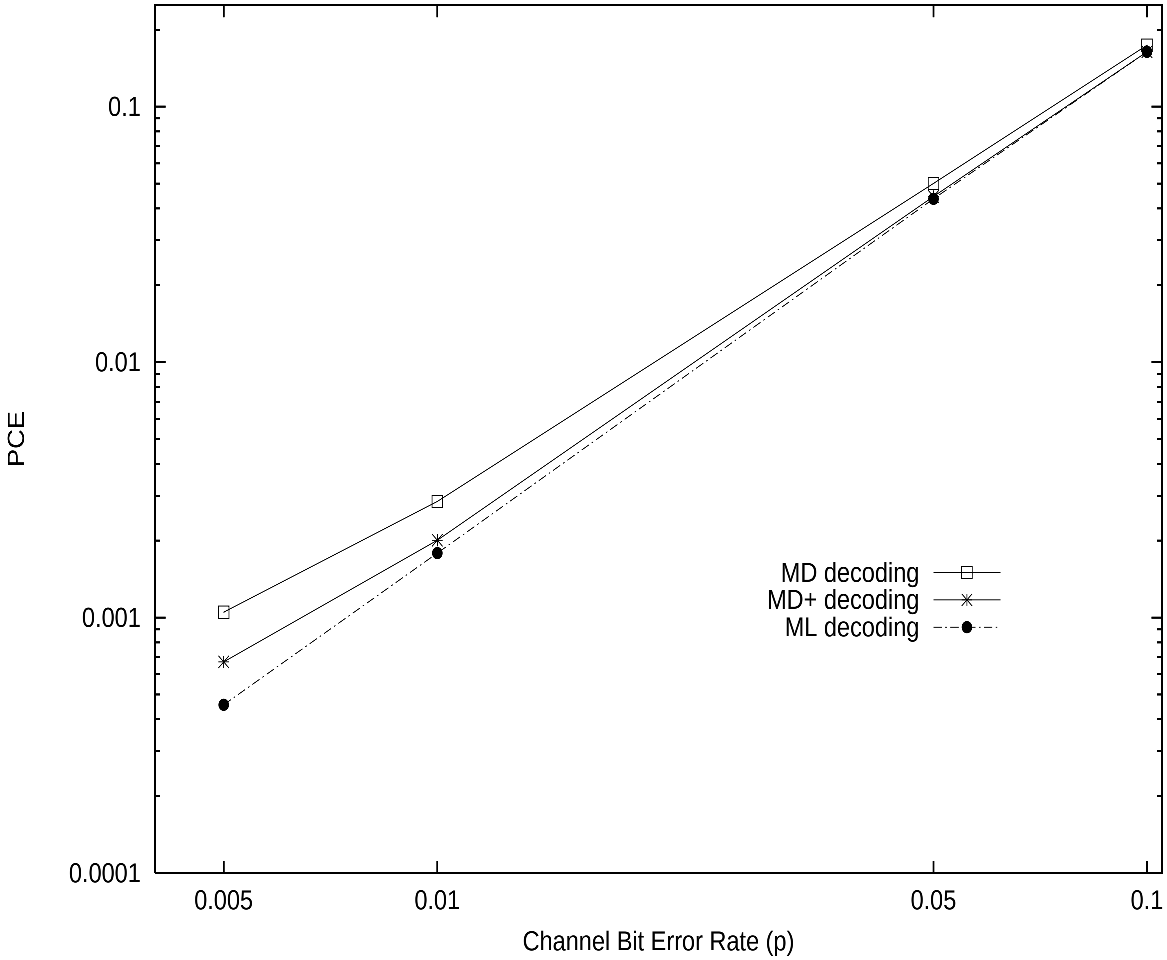


Figure 4.11: PCE vs CBER p under different decoding schemes for the BCH $[15, 7, 5]$ code over the BAMNC with noise correlation $\varepsilon = 0.1$.

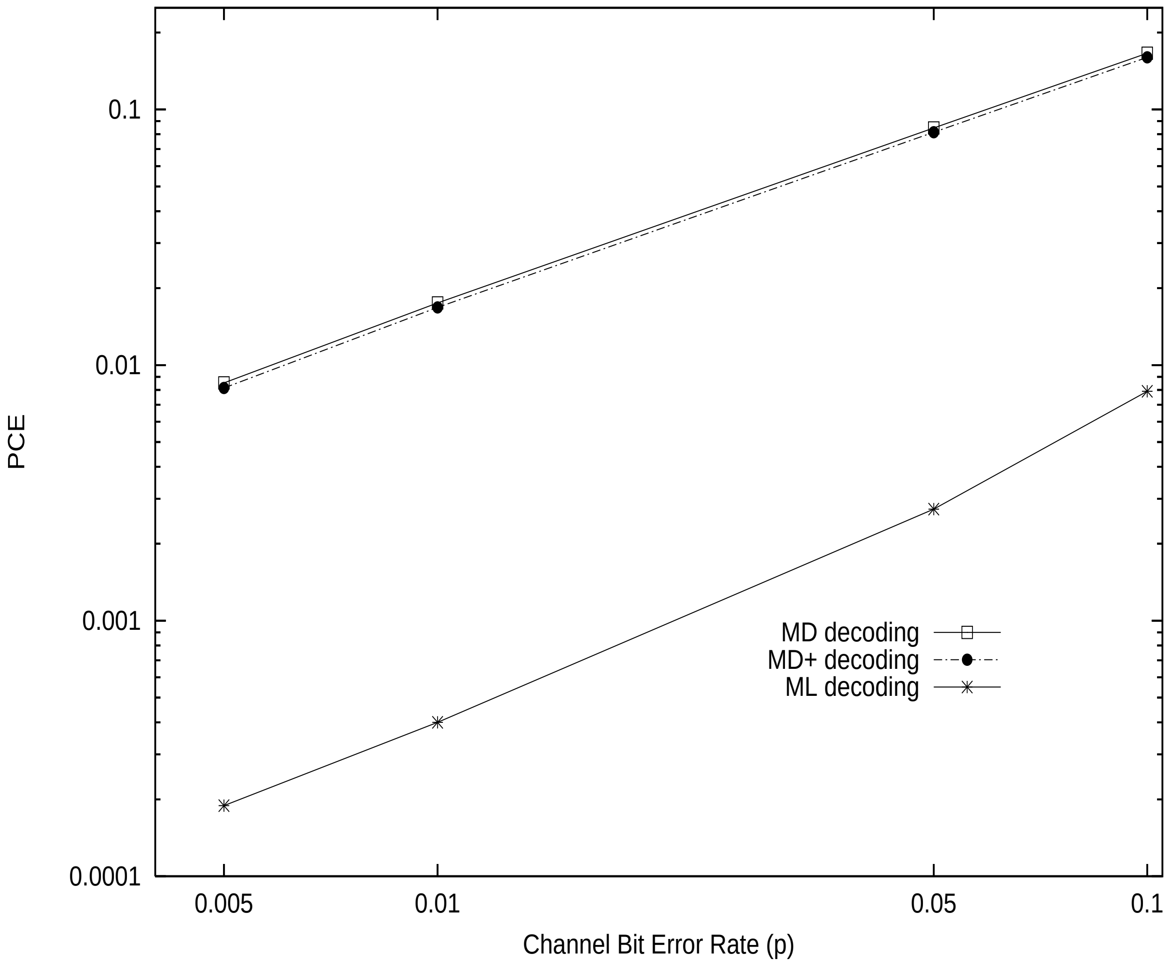


Figure 4.12: PCE vs CBER p under different decoding schemes for the BCH $[15, 7, 5]$ code over the BAMNC with noise correlation $\varepsilon = 0.9$.

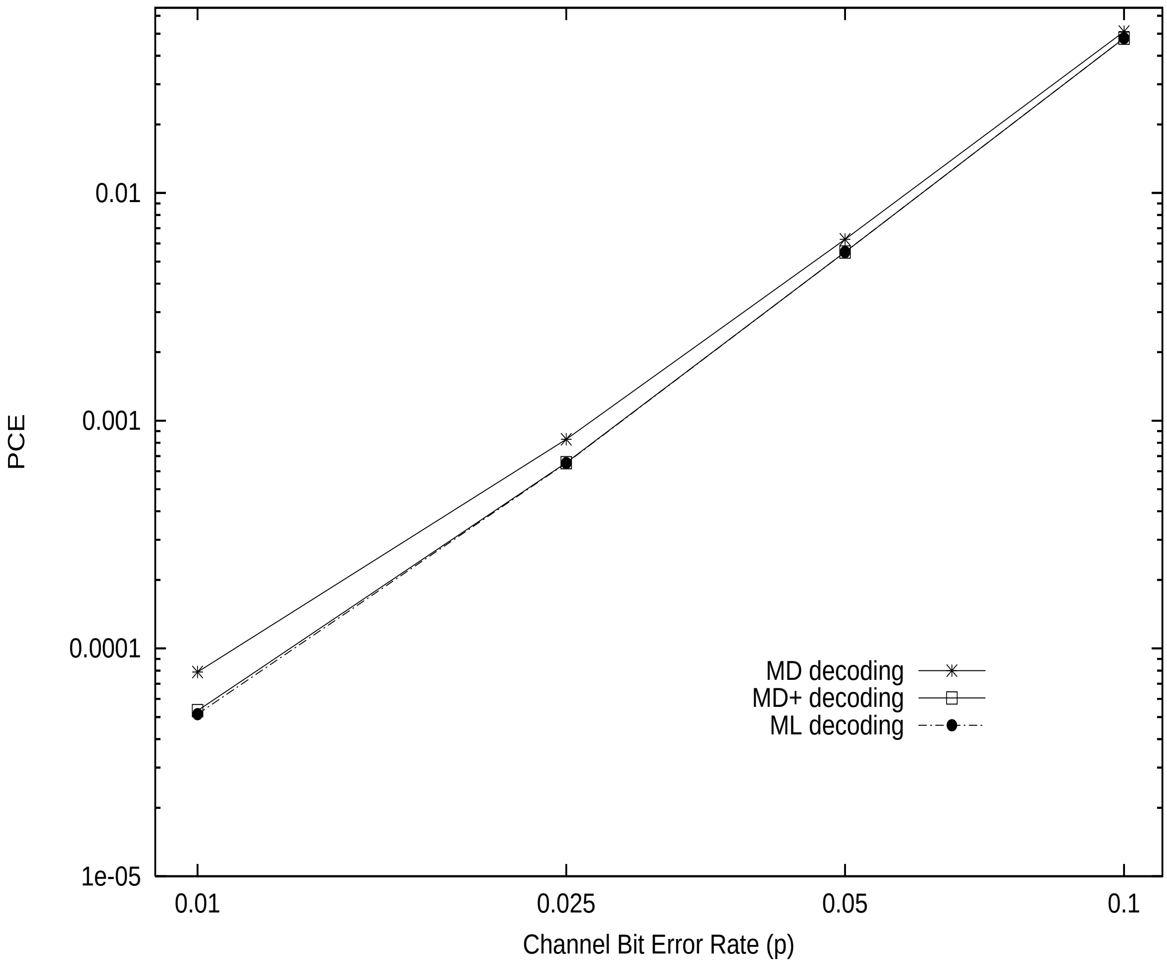


Figure 4.13: PCE vs CBER p under different decoding schemes for the BCH $[15, 5]$ code over the BAMNC with noise correlation $\varepsilon = 0.05$.

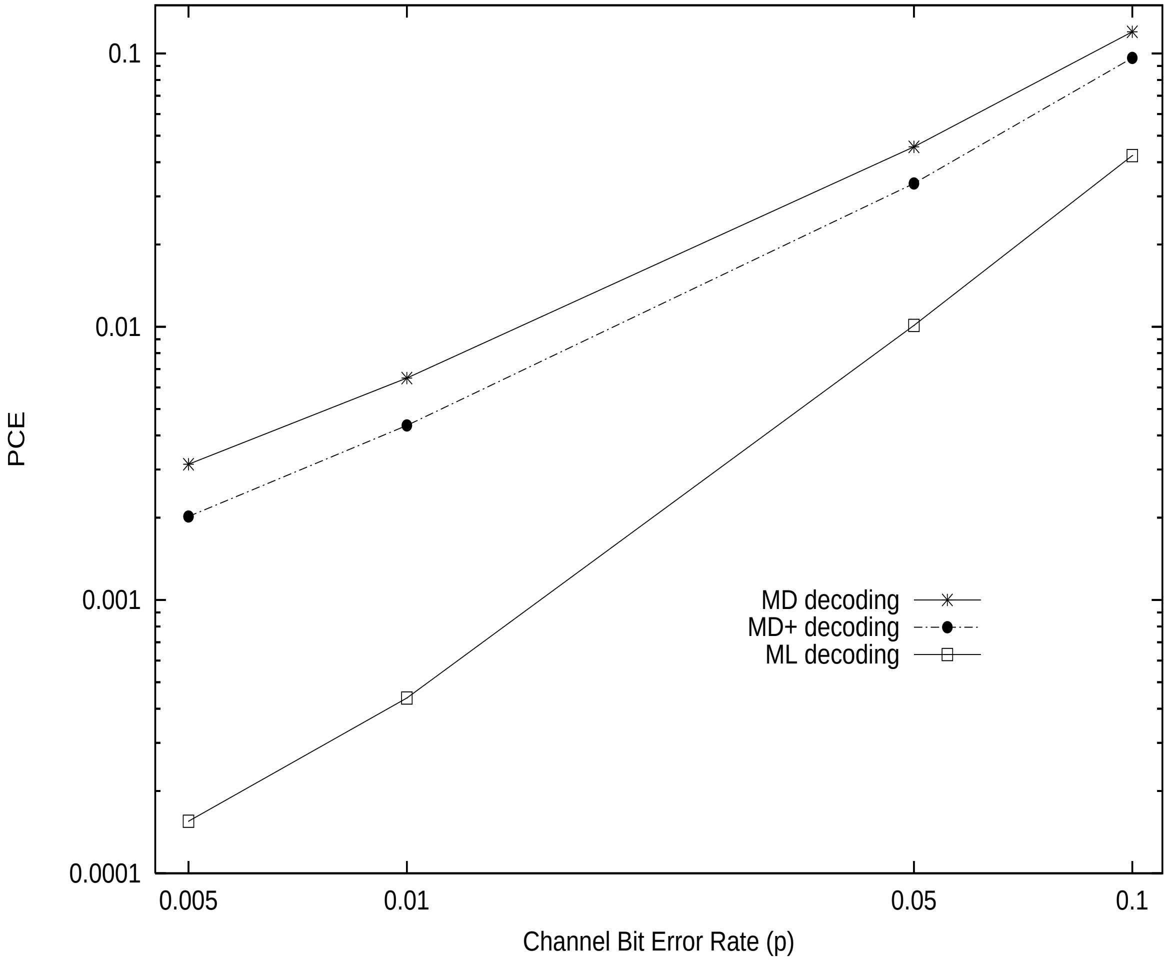


Figure 4.14: PCE vs CBER p under different decoding schemes for the BCH $[15,5]$ code over the BAMNC with noise correlation $\varepsilon = 0.5$.

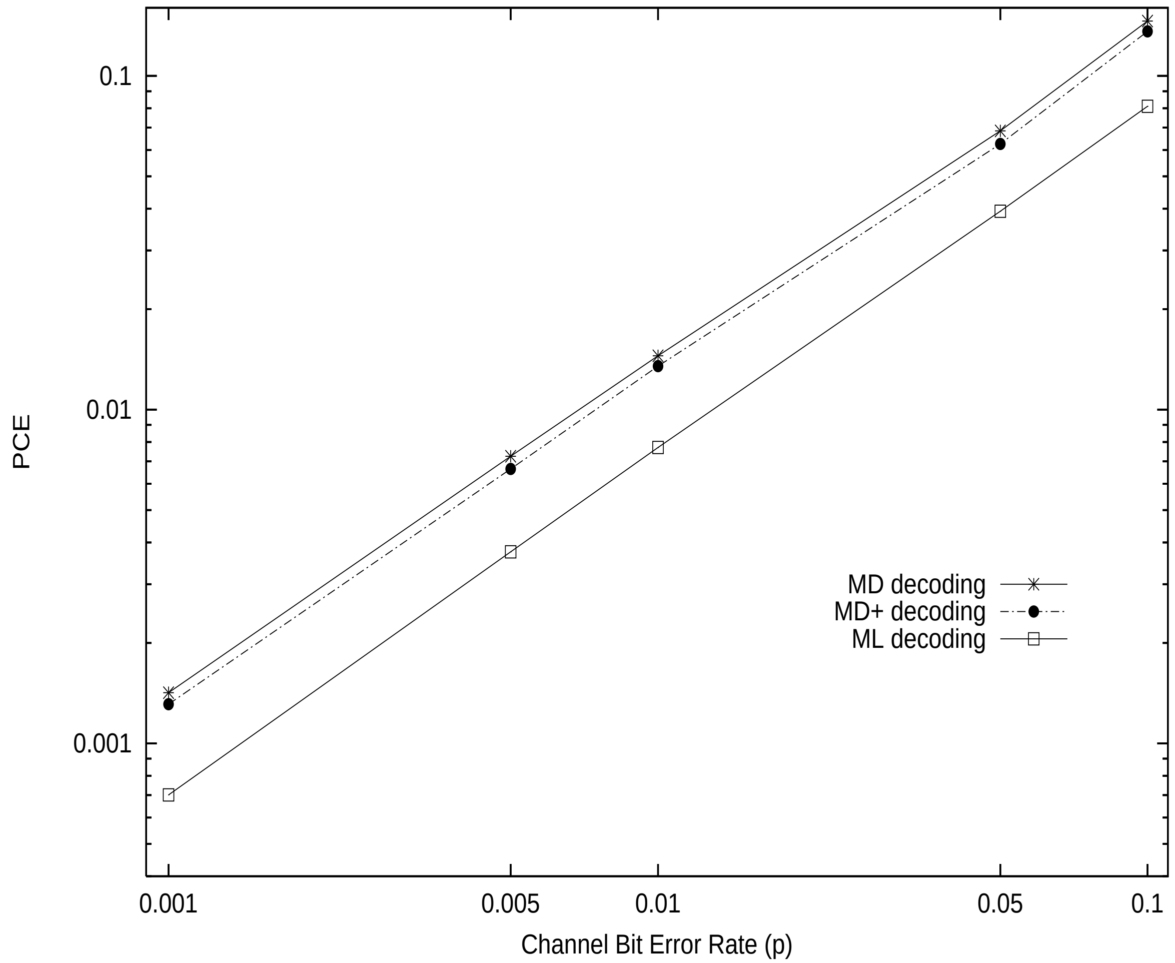


Figure 4.15: PCE vs CBER p under different decoding schemes for the BCH $[15, 5]$ code over the BAMNC with noise correlation $\varepsilon = 0.9$.

Chapter 5

Performance Evaluation of Reed-Solomon Codes over the QBC

Burst-error correcting codes are of prime theoretical and practical interest due to the bursty nature of real-world wireless digital communication channels. An important class of non-binary burst-error correcting codes used widely in data transmission and storage systems is the family of Reed-Solomon (RS) codes[8], [34], which was briefly described in Chapter 3. Conventional communication systems employing these codes are designed for memoryless channels, which is not an accurate model for wireless fading channels. As a consequence, interleaving is used to render the channel memoryless; this introduces additional delay and complexity to the system. Furthermore, such interleaved system fails to exploit the benefits of the statistical memory of the channel noise. When non-binary codes are sent over a stationary binary additive noise channel with memory, two interleaving strategies are worth considering: interleaving the code (or channel) bits which reduces the channel to the memoryless BSC (under perfect or infinite interleaving depth) and interleaving the code symbols.

This chapter is concerned about the performance of RS codes (under bounded-distance decoding) over the QBC. The QBC has the advantage of having a finite number of parameters and closed form expressions for the block transition probability, capacity and autocorrelation function as described in Chapter 2. Furthermore, it has been shown in [48], [49] that QBC can fit or approximate the discrete channel with Clarke's autocorrelation (DCCA) model that employs binary frequency-shift keying modulation, a Rician flat-fading channel, and a hard quantized non-coherent demodulation. We first prove that under bounded-distance decoding, symbol interleaving results in a better performance compared to bit interleaving for any non-binary block code over the QBC with any $M \geq 1$. We next restrict our analysis for the case $M = 1$ (i.e., the BAMNC with non-negative noise correlation) and derive a useful analytical expression for the probability of m symbol errors in a block of n symbols, $P(m, n)$, which we will eventually use to study the performance of four different RS codes over the BAMNC with non-negative noise correlation. In particular, for the given codes, we investigate whether we can avoid interleaving. A similar analysis can be carried out for the QBC with $M > 1$; although in this case, a closed-form expression for the key quantity $P(m, n)$ is tedious to explicitly derive. But it can be calculated numerically (for any $M > 1$) using (5.13); see Section 5.3.

5.1 Probability of Codeword Error for Non-Binary Block Codes over the QBC

Assume $\mathcal{C}$ is a non-binary block code over GF(2^b). In addition, assume that $\mathcal{C}$ can correct up to t symbol errors under bounded distance decoding. Moreover, we assume a binary-input binary-output channel. Recall from our discussion in Chapter 3 that the transmission of non-binary codes with symbols from GF(2^b) over such channels is possible by transmitting the binary b-tuple representation of each code symbol we want to transmit and that a transmitted symbol is received correctly if the noise corrupting it is a sequence of zeros of length b, denoted as 0^b. Otherwise, the transmitted symbol is received incorrectly and a symbol error occurs. Then the PCE for $\mathcal{C}$ is given by

$$PCE = 1 - \sum_{m=0}^{t} P(m, n)$$

where $P(m, n)$ is the probability of having m symbol errors in a block of n symbols.

5.1.1 Error Sequence Probability for an M^{th}-Order Markov Noise Channel

Suppose we are given a stationary binary M^{th}-order Markov noise channel model. Then the channel state process $\{\underline{\mathrm{S}}_k\}_{k=1}^{\infty}$, where $\underline{\mathrm{S}}_k = (Z_k, Z_{k-1}, \cdots, Z_{k-M+1})$, is a (first-order) Markov process with $N = 2^M$ states as explained in Section 2.6. The channel is completely characterized by the state probability transition matrix $\boldsymbol{P}$. Let Z_k be the noise output at time k. Then $\Pr(Z^n = z^n)$ can be evaluated as in Section 2.4. First, define the $N \times N$ matrices $\boldsymbol{P}(z_k)$ where $z_k \in \{0, 1\}$ whose $(i, j)^{th}$ entry is $\Pr(Z_k = z_k, S_k = j | S_{k-1} = i)$. By definition, $\boldsymbol{P} = \boldsymbol{P}(0) + \boldsymbol{P}(1)$. Let $\boldsymbol{\Pi}$ be the stationary distribution of the state process. Then $\Pr(Z^n = z^n)$ is given by

$$\Pr(Z^n = z^n) = \boldsymbol{\Pi}^T \left(\prod_{k=1}^{n} \boldsymbol{P}(z_k) \right) \mathbf{1} \tag{5.1}$$

where T denotes the transpose of the matrix and $\mathbf{1}$ is the all-one column matrix.

The QBC with memory M has $N = 2^M$ states. Thus the dimension of the matrices $\boldsymbol{P}, \boldsymbol{P}(0)$ and $\boldsymbol{P}(1)$ depends on the value of M. For example if $M = 1$ then these matrices are

$$\boldsymbol{P} = \begin{bmatrix} \varepsilon + (1-\varepsilon)(1-p) & (1-\varepsilon)p \\ (1-\varepsilon)(1-p) & \varepsilon + (1-\varepsilon)p \end{bmatrix},$$

$$\boldsymbol{P}(0) = \begin{bmatrix} \varepsilon + (1-\varepsilon)(1-p) & 0 \\ (1-\varepsilon)(1-p) & 0 \end{bmatrix} \qquad \boldsymbol{P}(1) = \begin{bmatrix} 0 & (1-\varepsilon)p \\ 0 & \varepsilon + (1-\varepsilon)p \end{bmatrix}.$$

If $M = 2$ then we get the following 4×4 matrices

$$\boldsymbol{P} = \begin{bmatrix} \varepsilon + (1-\varepsilon)(1-p) & 0 & (1-\varepsilon)p & 0 \\ \frac{\varepsilon}{1+\alpha} + (1-\varepsilon)(1-p) & 0 & \frac{\varepsilon\alpha}{1+\alpha} + (1-\varepsilon)p & 0 \\ 0 & \frac{\varepsilon\alpha}{1+\alpha} + (1-\varepsilon)(1-p) & 0 & \frac{\varepsilon\alpha}{1+\alpha} + (1-\varepsilon)(1-p) \\ 0 & (1-\varepsilon)(1-p) & 0 & \varepsilon + (1-\varepsilon)p \end{bmatrix},$$

$$\boldsymbol{P}(0) = \begin{bmatrix} \varepsilon + (1-\varepsilon)(1-p) & 0 & 0 & 0 \\ \frac{\varepsilon}{1+\alpha} + (1-\varepsilon)(1-p) & 0 & 0 & 0 \\ 0 & \frac{\varepsilon\alpha}{1+\alpha} + (1-\varepsilon)(1-p) & 0 & 0 \\ 0 & (1-\varepsilon)(1-p) & 0 & 0 \end{bmatrix},$$

$$\boldsymbol{P}(1) = \begin{bmatrix} 0 & 0 & (1-\varepsilon)p & 0 \\ 0 & 0 & \frac{\varepsilon\alpha}{1+\alpha} + (1-\varepsilon)p & 0 \\ 0 & 0 & 0 & \frac{\varepsilon}{1+\alpha} + (1-\varepsilon)p \\ 0 & 0 & 0 & \varepsilon + (1-\varepsilon)p \end{bmatrix}.$$

For the general QBC, $\boldsymbol{P}(0)$ and $\boldsymbol{P}(1)$ can be derived easily from $\boldsymbol{P}$. The first 2^{M-1} columns of $\boldsymbol{P}(0)$ is always the same as those of $\boldsymbol{P}$ while the remaining columns have all zero entries. The opposite is true for $\boldsymbol{P}(1)$ (i.e., it agrees with $\boldsymbol{P}$ in their last 2^{M-1} columns while the remaining columns contain all zeros).

5.2 Symbol Interleaving vs Bit Interleaving for the QBC

For any non-binary linear block code over GF(2^b) with length n and error correction capability t, the PCE under (ideal) bit interleaving when the code is transmitted over the QBC and under bounded distance decoding is given by

$$PCE = 1 - \sum_{i=0}^{t} \binom{n}{i} [1-(1-p)^b]^i [(1-p)^b]^{n-i}. \tag{5.2}$$

On the other hand, if the above code is transmitted with (ideal) symbol interleaving then the PCE is given by

$$PCE = 1 - \sum_{i=0}^{t} \binom{n}{i} [1-\Pr(Z^b = 0^b)]^{n-i} \cdot [\Pr(Z^b = 0^b)]^i \tag{5.3}$$

where

$$\Pr(Z^b = 0^b) = \begin{cases} \prod_{j=0}^{b-1} \frac{j\frac{\varepsilon}{M-1+\alpha}+(1-\varepsilon)(1-p)}{1-(\alpha+M-1-j)\frac{\varepsilon}{M-1+\alpha}} & \text{if } b \le M, \\ \prod_{j=0}^{M-1} \frac{j\frac{\varepsilon}{M-1+\alpha}+(1-\varepsilon)(1-p)}{1-(\alpha+M-1-j)\frac{\varepsilon}{M-1+\alpha}}(\varepsilon+(1-\varepsilon)(1-p))^{b-M} & \text{if } b > M. \end{cases}$$

Lemma 5.1 *Let $\mathcal{C}$ be any non-binary linear block code over GF(2^b) with length n and error correction capability t (e.g., a Reed-Solomon code). Under bounded distance decoding, symbol interleaving outperforms bit interleaving over the QBC with $p > 0$ and $\varepsilon > 0$.*

Proof. Let the parameter of the QBC, M, p, ε and α, be given and a non-binary code $\mathcal{C}$ over the Galois field GF(2^b) with length n that can correct up to t symbols.

Case 1: $b \leq M$

Define the following two functions $f(x)$ and $g(x)$

$$f(x) = \left(1 - \prod_{j=0}^{b-1} \frac{j\frac{\varepsilon}{M-1+\alpha} + (1-\varepsilon)(1-p)}{1-(\alpha+M-1-j)\frac{\varepsilon}{M-1+\alpha}}\right)^{x} \left(\prod_{j=0}^{b-1} \frac{j\frac{\varepsilon}{M-1+\alpha} + (1-\varepsilon)(1-p)}{1-(\alpha+M-1-j)\frac{\varepsilon}{M-1+\alpha}}\right)^{n-x} \tag{5.4}$$

$$g(x) = [1-(1-p)^b]^x[(1-p)^b]^{n-x}. \tag{5.5}$$

For each $j > 0$ we notice that

$$\begin{aligned}
& \frac{j\frac{\varepsilon}{M-1+\alpha} + (1-\varepsilon)(1-p)}{1-(\alpha+M-1-j)\frac{\varepsilon}{M-1+\alpha}} > (1-p) \\
\Leftrightarrow\ & j\frac{\varepsilon}{M-1+\alpha} + (1-\varepsilon)(1-p) > (1-p)\left(1-(\alpha+M-1-j)\frac{\varepsilon}{M-1+\alpha}\right) \\
\Leftrightarrow\ & j\frac{\varepsilon}{M-1+\alpha} > (1-p)\varepsilon\frac{j}{M-1+\alpha} \\
\Leftrightarrow\ & 1 > (1-p).
\end{aligned}$$

Because $b > 1$ (for non-binary codes), we get

$$\prod_{j=0}^{b-1} \frac{j\frac{\varepsilon}{M-1+\alpha} + (1-\varepsilon)(1-p)}{1-(\alpha+M-1-j)\frac{\varepsilon}{M-1+\alpha}} > (1-p)^b \Leftrightarrow f(0) > g(0) \Leftrightarrow f(n) < g(n). \tag{5.6}$$

Since $\log f(x)$ and $\log g(x)$ are both linear functions of x and due to (5.6), $f(x)$ and $g(x)$ have a unique point of intersection, denoted by x_0 obtained by solving the equation $f(x) = g(x)$. In particular, $x_0 = n \ \ln A / \ln B$

where

$$A \triangleq \frac{(1-p)^b}{\prod_{j=0}^{b-1} \frac{j\frac{\varepsilon}{M-1+\alpha}+(1-\varepsilon)(1-p)}{1-(\alpha+M-1-j)\frac{\varepsilon}{M-1+\alpha}}}$$

and

$$B \triangleq \frac{\left(1-\prod_{j=0}^{b-1} \frac{j\frac{\varepsilon}{M-1+\alpha}+(1-\varepsilon)(1-p)}{1-(\alpha+M-1-j)\frac{\varepsilon}{M-1+\alpha}}\right)(1-p)^b}{\left(\prod_{j=0}^{b-1} \frac{j\frac{\varepsilon}{M-1+\alpha}+(1-\varepsilon)(1-p)}{1-(\alpha+M-1-j)\frac{\varepsilon}{M-1+\alpha}}\right)(1-(1-p)^b)}.$$

Furthermore, for each $x < x_0$, $f(x) > g(x)$ and for each $x > x_0$, $f(x) < g(x)$.

First, assume that the code's error-correction capability (under bounded distance decoding), t satisfies $t \leq \lfloor x_0 \rfloor$, then

$$\sum_{i=0}^{t} \binom{n}{i} f(i) > \sum_{i=0}^{t} \binom{n}{i} g(i).$$

The left hand side in the above equation is the probability of correct decoding for the symbol interleaved system while the right hand side is the probability of correct decoding for the bit interleaved system.

Now assume $t \geq \lfloor x_0 \rfloor$, then

$$\sum_{i=t+1}^{n} \binom{n}{i} f(i) < \sum_{i=t+1}^{n} \binom{n}{i} g(i).$$

Now the left hand side of the last equation above is the PCE for the symbol interleaved system whereas the right hand is the PCE for the bit interleaved one, both under bounded distance decoding. Hence, we have proved that if $b \leq M$, then symbol interleaving outperforms bit interleaving.

Case 2: $b > M$

Define $h(x)$ as following

$$h(x) = \left[1 - \left(\prod_{j=0}^{M-1} \frac{j\frac{\varepsilon}{M-1+\alpha} + (1-\varepsilon)(1-p)}{1-(\alpha+M-1-j)\frac{\varepsilon}{M-1+\alpha}} (\varepsilon + (1-\varepsilon)(1-p))^{b-M}\right)\right]^x \left(\prod_{j=0}^{M-1} \frac{j\frac{\varepsilon}{M-1+\alpha} + (1-\varepsilon)(1-p)}{1-(\alpha+M-1-j)\frac{\varepsilon}{M-1+\alpha}} (\varepsilon + (1-\varepsilon)(1-p))^{b-M}\right)^{n-x}. \quad (5.7)$$

We have already proved that $\frac{j\frac{\varepsilon}{M-1+\alpha}+(1-\varepsilon)(1-p)}{1-(\alpha+M-1-j)\frac{\varepsilon}{M-1+\alpha}} > (1-p)$ for $j > 0$. We also notice that

$$\varepsilon + (1-\varepsilon)(1-p) = (1-p) + \varepsilon p > (1-p).$$

Therefore, we combine the above two inequalities to get the following

$$\left(\prod_{j=0}^{M-1} \frac{j\frac{\varepsilon}{M-1+\alpha} + (1-\varepsilon)(1-p)}{1-(\alpha+M-1-j)\frac{\varepsilon}{M-1+\alpha}} (\varepsilon + (1-\varepsilon)(1-p))^{b-M}\right) > (1-p)^M(1-p)^{b-M} = (1-p)^b.$$

Therefore $h(0) > g(0)$ and $h(n) < g(n)$ (the strict inequality is because we assume both p and $\varepsilon \neq 0$).

The rest of the proof is identical to the first case (i.e. $b \leq M$) after replacing $f(x)$ with $h(x)$ with the only difference being the point of intersection x_0. However it is obtained similarly by solving the equation $h(x) = g(x)$. □

5.3 Calculating $P(m,n)$ Using the Generating Series Method

We are interested in evaluating $P(m,n)$ in order to analyze the performance of non-binary block codes over the QBC. The generating series method is a powerful tool that offers a combinatorial approach to determine $P(m,n)$ for a Markov noise process with N states [29] [28]. More details on the generating series topic can be found in [15].

Let $\mathbb{R}\langle x_0, x_1\rangle$ and $\mathbb{R}\langle\langle x_0, x_1\rangle\rangle$ be the set of polynomials and the ring of all formal power series in the non-commuting indeterminates x_0 and x_1, respectively, where $\mathbb{R}$ is the field of real numbers. Note that any element in that any element in $\mathbb{R}\langle\langle x_0, x_1\rangle\rangle$ is nothing but an infinite sum of elements from $\mathbb{R}\langle x_0, x_1\rangle$. The choice of non-commuting indeterminates is due to the fact that the probability of a sequence depends on the positions of 0's and 1's. Let $\mathcal{E}_n$ be a generic notation for an error event of length n (i.e., $\mathcal{E}_n$ is any subset of $\{0,1\}^n$). For example, if $\mathcal{E}_3$ is a subset of $\{0,1\}^3$ such that $\mathcal{E}_3 = \{\text{one error in a block of length 3}\}$, then $\mathcal{E}_3 = \{100, 010, 001\}$.

The generating series for $\mathcal{E}_n$ can be expressed as

$$F_{\mathcal{E}_n} = \sum_{e^n \in \mathcal{E}_n} x_{e_1} x_{e_2} \cdots x_{e_n}, \qquad x_{e_i} \in \{x_0, x_1\}. \tag{5.8}$$

Note that $F_{\mathcal{E}_n} \in \mathbb{R}\langle x_0, x_1\rangle$ and that x_0 and x_1 mark the noise bits 0 and 1, respectively.

Example: The generating series for the set $\mathcal{E}_3$

$$F_{\mathcal{E}_3} = x_1 x_0^2 + x_0 x_1 x_0 + x_0^2 x_1.$$

Using (5.1), the probability of the set $\mathcal{E}_3$ is given by

$$\begin{aligned} \Pr(\mathcal{E}_3) &= \Pr(Z^3 = (1,0,0)) + \Pr(Z^3 = (0,1,0)) + \Pr(Z^3 = (0,0,1)) \\ &= \mathbf{\Pi}^T \left(\boldsymbol{P}(1)(\boldsymbol{P}(0))^2 + \boldsymbol{P}(0)\boldsymbol{P}(1)\boldsymbol{P}(1) + (\boldsymbol{P}(0))^2 \boldsymbol{P}(1) \right) \mathbf{1}. \qquad (5.9) \end{aligned}$$

Note that $\Pr(\mathcal{E}_3)$ can be calculated using the generating series $F_{\mathcal{E}_3}$, by replacing x_0 and x_1 with $\boldsymbol{P}(0)$ and $\boldsymbol{P}(1)$, respectively and multiplying by $\mathbf{\Pi}^T$ from left and by $\mathbf{1}$ from right.

Let $\boldsymbol{M}_N(\mathbb{R})$ be the ring of all $N \times N$ matrices with entries from $\mathbb{R}$. Define the mapping Δ as

$$\Delta : \mathbb{R}\langle x_0, x_1 \rangle \rightarrow \boldsymbol{M}_N(\mathbb{R}) : x_k \mapsto \boldsymbol{P}(k) \text{ and } a \mapsto a \cdot \boldsymbol{I} \text{ for } a \in \mathbb{R}$$

where $\boldsymbol{I}$ is the $N \times N$ identity matrix.

Then, we can express $\Pr(\mathcal{E}_3)$ in a compact form as

$$\Pr(\mathcal{E}_3) = \mathbf{\Pi}^T \left(\Delta(F_{\mathcal{E}_3}) \right) \mathbf{1}.$$

In general, given a set $\mathcal{E}_n$, $\Pr(\mathcal{E}_n)$ can be readily computed using its generating series $F_{\mathcal{E}_n}$ by

$$\Pr(\mathcal{E}_n) = \mathbf{\Pi}^T \left(\Delta(F_{\mathcal{E}_n}) \right) \mathbf{1}.$$

Let $\mathbb{R}[[s, z]]$ be the ring of all formal power series in the commuting indeterminates s

and z. Let $P(s,z)$ be the generating series defined as

$$P(s,z) = \sum_{n=0}^{\infty} \sum_{m=0}^{n} P(m,n) s^m z^n \ \in \mathbb{R}[[s,z]].$$

Let $[s^m z^n]P(s,z)$ be the coefficient of the term $s^m z^n$ in $P(s,z)$. Then

$$P(m,n) = [s^m z^n] P(s,z). \tag{5.10}$$

We are interested in calculating an expression for $P(s,z)$. Let $\mathcal{E}_n^*$ be the set of all binary sequences of length n. The generating series for this set is obtained via (5.8) as

$$\begin{aligned} F_{\mathcal{E}_n^*} &= x_0^n + x_0 x_1^{n-1} + x_1 x_0 x_1^{n-2} + \cdots + x_1^{n-1} x_0 + x_0^2 x_1^{n-2} + \cdots + x_1^n \\ &= (x_0 + x_1)^n \ \in \mathbb{R}\langle x_0, x_1 \rangle. \end{aligned}$$

Define $\mathcal{E}^* = \bigcup_{n=0}^{\infty} \mathcal{E}_n^*$. Then

$$F_{\mathcal{E}^*} = \sum_{i=0}^{\infty} (x_0 + x_1)^i = (1 - (x_0 + x_1))^{-1} \ \in \mathbb{R}\langle\langle x_0, x_1 \rangle\rangle.$$

We need to enumerate the Hamming weight and the length of each sequence in $F_{\mathcal{E}^*}$. In order to do this we define the generating series

$$F(x_0, x_1, s, z) = \sum_{i=0}^{\infty} z^i (x_0 + s x_1)^i = (1 - z(x_0 + s x_1))^{-1} \ \in \mathbb{R}\langle x_0, x_1 \rangle[[s,z]]$$

where $\mathbb{R}\langle x_0, x_1 \rangle[[s,z]]$ is the ring of all formal power series in the commuting indeterminates s and z with coefficients from the ring $\mathbb{R}\langle x_0, x_1 \rangle$. Note that z marks the length of a binary sequence while s marks the Hamming weight of the sequence.

Let $\mathcal{E}_n^m$ denote the set composed of all binary sequences of length n and Hamming weight m. It is now clear that

$$\begin{aligned} F_{\mathcal{E}_n^m} &= [s^m z^n] F(x_0, x_1, s, z) \\ &= [s^m z^n](1 - z(x_0 + s x_1))^{-1}. \end{aligned}$$

Then

$$\begin{aligned} \Pr(\mathcal{E}_n^m) &= \mathbf{\Pi}^T (\Delta(F_{\mathcal{E}_n^m}))\mathbf{1} \\ &= [s^m z^n]\mathbf{\Pi}^T \left[\Delta(F(x_0, x_1, s, z))\right] \mathbf{1} \\ &= [s^m z^n]\mathbf{\Pi}^T \left[\Delta(1 - z(x_0 + s x_1))^{-1}\right] \mathbf{1} \\ &= [s^m z^n]\mathbf{\Pi}^T \left[\boldsymbol{I} - z(\boldsymbol{P}(0) + s\boldsymbol{P}(1))^{-1}\right] \mathbf{1}, \end{aligned} \tag{5.11}$$

where the last equality follows from the definition of the mapping Δ.

Comparing (5.10) with (5.11), we get

$$P(s, z) = \mathbf{\Pi}^T \left[\boldsymbol{I} - z(\boldsymbol{P}(0) + s\boldsymbol{P}(1))^{-1}\right] \mathbf{1}.$$

Now define $\mathcal{S}^*$ to be the set that contains all strings of symbols from GF(2^b). We are interested in finding the generating series of $\mathcal{S}^*$. Note that elements of $\mathcal{S}^*$ can be mapped one-to-one to binary strings due to the existence of a bijection between GF(2^b) and the set of all binary b-tuples $\{0, 1\}^b$. Hence, the generating series of $\mathcal{S}^*$ can still be represented in terms of x_0 and x_1.

Let $\mathcal{F} = \{0, 1\}^b$. Define $\mathcal{F}_1$ and $\mathcal{F}_2 \subset \mathcal{F}$ such that $\mathcal{F}_1 = \{0^b\}$ (i.e., the singleton set containing the all-zero b-tuple) and $\mathcal{F}_2 = \mathcal{F} \backslash \mathcal{F}_1$. In other words, $\mathcal{F}_1$ and $\mathcal{F}_2$

partition the set $\mathcal{F}$. Now suppose that a non-binary symbol from GF(2^b) is sent over a binary-input binary-output channel. Let z^b be a b-tuple noise output produced by the channel. Obviously if $z^b \in \mathcal{F}_1$ then the transmitted symbol will be received correctly otherwise (i.e., $z^b \in \mathcal{F}_2$) the received symbol will be in error. Let F_c and F_e be the generating series for $\mathcal{F}_1$ and $\mathcal{F}_2$, respectively. Therefore, F_c and F_e are given by

$$\begin{aligned} F_c &= x_0^b & \in \mathbb{R}\langle\langle x_0, x_1 \rangle\rangle \\ F_e &= (x_0 + x_1)^b - x_0^b & \in \mathbb{R}\langle\langle x_0, x_1 \rangle\rangle. \end{aligned}$$

It can be shown that the generating series for $\mathcal{S}^*$ can be expressed as [29], [28]

$$F_{\mathcal{S}^*} = (1 - F_c - F_e)^{-1}.$$

Note that this is analogous to the binary case mentioned earlier in this section (i.e., $F_{\mathcal{E}^*}$) since both of F_c and x_0 enumerate an error-free transmission of a symbol and bit, respectively, whereas F_e and x_1 represent an erroneous symbol and bit, respectively. Let the indeterminates z mark the length of the channel noise output and s represent the number of erroneous symbols in a noise output. Clearly, $P(m, n)$ is the probability that within n consecutive b-ary symbols, there are exactly m symbols from the set enumerated by F_e. In this case, the formal power series $P(s, z)$ is given by [32] [28]

$$P(s, z) = \mathbf{\Pi}^T \left[\boldsymbol{I} - z\{\boldsymbol{P}(0)^b + s(\boldsymbol{P}^b - \boldsymbol{P}(0)^b)\} \right]^{-1} \mathbf{1}. \tag{5.12}$$

Thus $P(m, n)$ can be derived as the coefficient of $s^m z^n$ in $P(s, z)$ above [31]. This can be represented as

$$P(m,n) = [s^m z^n]\mathbf{\Pi}^T \left[\mathbf{I} - z\{\boldsymbol{P}(0)^b + s(\boldsymbol{P}^b - \boldsymbol{P}(0)^b)\}\right]^{-1} \mathbf{1}. \tag{5.13}$$

The above equation is valid for any integer $b \geq 1$ and when $b = 1$, (5.13) reduces to (5.11).

When $M = 1$, the QBC reduces to the BAMNC with non-negative noise correlation. The following two lemmas are helpful in order to evaluate (5.12) and (5.13).

Lemma 5.2 *Let $\boldsymbol{P}$ be the probability transition matrix for the BAMNC (i.e. the QBC with $M = 1$). Then for any integer n,*

$$\boldsymbol{P}^n = \begin{bmatrix} \varepsilon^n + (1-p)(1-\varepsilon^n) & p(1-\varepsilon^n) \\ (1-p)(1-\varepsilon^n) & \varepsilon^n + (1-\varepsilon^n)p \end{bmatrix}.$$

Proof. The proof follows by induction. The reader is referred to [42] for more details.

□

Lemma 5.3 *The 2×2 matrix $\boldsymbol{P}(0)$ as defined earlier for the BAMNC has the property that for any integer n,*

$$\boldsymbol{P}(0)^n = \begin{bmatrix} (\varepsilon + (1-p)(1-\varepsilon))^n & 0 \\ (1-\varepsilon)(1-p)(\varepsilon + (1-p)(1-\varepsilon))^{n-1} & 0 \end{bmatrix}.$$

Proof. The proof is done by induction, as well. For $n = 1$, we just get the same matrix $\boldsymbol{P}(0)$. So now assume that the statement is true for $n = k$. Then, for $n = k+1$, we have

$$\begin{aligned}
\boldsymbol{P}^{k+1}(0) &= \boldsymbol{P}^{k}(0) \cdot \boldsymbol{P}(0) \\
&= \begin{bmatrix} (\varepsilon + (1-p)(1-\varepsilon))^{k} & 0 \\ (1-\varepsilon)(1-p)(\varepsilon + (1-p)(1-\varepsilon))^{k-1} & 0 \end{bmatrix} \begin{bmatrix} (\varepsilon + (1-p)(1-\varepsilon)) & 0 \\ (1-\varepsilon)(1-p) & 0 \end{bmatrix} \\
&= \begin{bmatrix} (\varepsilon + (1-p)(1-\varepsilon))^{k+1} & 0 \\ (1-\varepsilon)(1-p)(\varepsilon + (1-p)(1-\varepsilon))^{k} & 0 \end{bmatrix}.
\end{aligned}$$

□

Define the matrix $\boldsymbol{K}$ as follows.

$$\boldsymbol{K} \overset{\triangle}{=} \left[\boldsymbol{I} - z\{\boldsymbol{P}(0)^{b} + s(\boldsymbol{P}^{b} - \boldsymbol{P}(0)^{b})\}\right].$$

We can express K as

$$\boldsymbol{K} = \begin{bmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{bmatrix}$$

where

$$\begin{aligned}
k_{11} &= 1 - z[(\varepsilon + (1-p)(1-\varepsilon))^{b} + s(\varepsilon^{b} + (1-p)(1-\varepsilon^{b}) - (\varepsilon + (1-p)(1-\varepsilon))^{b})] \\
k_{12} &= -zsp(1-\varepsilon^{b})
\end{aligned}$$

$$\begin{aligned}
k_{21} &= -z[(1-\varepsilon)(1-p)(\varepsilon+(1-p)(1-\varepsilon))^{b-1}+s((1-p)(1-\varepsilon^b)- \\
&\qquad (1-\varepsilon)(1-p)(\varepsilon+(1-p)(1-\varepsilon))^{b-1})] \\
k_{22} &= 1-zs[\varepsilon^b+p(1-\varepsilon^b)].
\end{aligned}$$

Thus, the formal power series $P(s,z)$ can be expressed as

$$\begin{aligned}
P(s,z) &= \mathbf{\Pi}^T\boldsymbol{K}^{-1}\mathbf{1} \\
&= \begin{bmatrix} 1-p & p \end{bmatrix} \begin{bmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{bmatrix}^{-1} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \\
&= \frac{p(k_{12}-k_{22})+(1-p)(k_{21}-k_{11})}{\det(K)}
\end{aligned}$$

where

$$\begin{aligned}
\det(K) &= k_{11}k_{22}-k_{12}k_{21} \\
&= 1-z[\varepsilon+(1-\varepsilon)(1-p)]^b+zs[(\varepsilon+(1-\varepsilon)(1-p))^b-(1+\varepsilon^b)] \\
&\quad +z^2s[(\varepsilon+(1-\varepsilon)(1-p))^{b-1}(\varepsilon^b(1-p)+\varepsilon p)]+ \\
&\quad z^2s^2[\varepsilon^b-(\varepsilon^b(1-p)+p\varepsilon)(\varepsilon+(1-\varepsilon)(1-p))^{b-1}].
\end{aligned}$$

Note that the denominator in the expression above is a tool to obtain a recursive expression for the coefficient of the term $s^m z^n$, while the numerator gives the initial conditions. Therefore, $P(m,n)$ can be recursively written as

$$\begin{aligned} P(m,n) &= [\varepsilon+(1-\varepsilon)(1-p)]^b P(m,n-1) - [(\varepsilon+(1-\varepsilon)(1-p))^b - (1+\varepsilon^b)] \\ &\times P(m-1,n-1) - [(\varepsilon+(1-\varepsilon)(1-p))^{b-1}(\varepsilon^b(1-p)+\varepsilon p)]P(m-1,n-2) \\ &- [\varepsilon^b - (\varepsilon^b(1-p)+p\varepsilon)(\varepsilon+(1-\varepsilon)(1-p))^{b-1}]P(m-2,n-2) \qquad (5.14) \end{aligned}$$

for $n \geq 2$, with initial conditions given by

$$\begin{aligned} P(m,n) &= 0 \qquad \text{if } m,n<0 \text{ or } m<n \\ P(0,0) &= 1 \\ P(0,1) &= (1-p)(\varepsilon+(1-\varepsilon)(1-p))^{b-1} \\ P(1,1) &= 1-(1-p)(\varepsilon+(1-\varepsilon)(1-p))^{b-1}. \end{aligned}$$

If $b=1$, then we have binary codes, and for this special case $P(m,n)$ reduces to

$$P(m,n) = (\varepsilon+(1-\varepsilon)(1-p))P(m,n-1)+(\varepsilon+(1-\varepsilon)p)P(m-1,n-1)-\varepsilon P(m-1,n-2).$$

This is a simpler expression than the one derived in [42] for the same binary system as it contains one less term.

We are now ready to calculate the probability of decoding error for any non-binary code over the BAMNC using bounded distance decoding. Figs. (5.1) – (5.4) show a perfect agreement between the simulation results of four typical Reed-Solomon (RS) codes described in Table 5.1 over the BAMNC and the analytical results using the recursion developed above. In fact, the recursive expression is a reliable instrument for calculating the PCE at a reduced complexity compared to the simulation especially for relatively large values of b and t.

5.4 Interleaving vs Non-Interleaving

Given a particular non-binary linear block code, we wish to examine the effect of interleaving (symbol or bit) on the performance of this code when transmitted over the BAMNC with non-negative correlation (i.e., the QBC with $M = 1$). We also use the four RS codes of Table 5.1, two of which are high-rate codes and the other two are low-rate ones. Using the PCE expression developed in the previous section, the performance can be readily evaluated.

5.4.1 When Can Symbol Interleaving Be Avoided ?

When a non-binary linear code over GF(2^b) is transmitted over the QBC with $M = 1$ (i.e., the BAMNC), the probability of m symbol errors in a block of n symbols under symbol interleaving is given by (5.15)

$$\begin{aligned} P'(m,n) &= \binom{n}{m} \Pr(1 \text{ symbol error})^m \Pr(\text{No symbol error})^{n-m} \\ &= \binom{n}{m} \left(1-(1-p)[\varepsilon+(1-\varepsilon)(1-p)]^{b-1}\right)^m \\ &\qquad \left((1-p)[\varepsilon+(1-\varepsilon)(1-p)]^{b-1}\right)^{(n-m)}. \end{aligned} \tag{5.15}$$

Symbol interleaving is equivalent to the assumption that the binary noise process is only Markovian within each symbol (of length b bits), and it is independent between symbols. As a result, the performance of the non-interleaved RS codes and the symbol interleaved one may be nearly the same at certain channel conditions. Table 5.2 lists the regions ε and p for which the performance of both schemes is similar over the BAMNC (within an absolute relative error less than or equal to 0.1) for the codes listed in Table 5.1.

5.4.2 When Can Bit Interleaving Be Avoided ?

We next evaluate the RS codes of Table 5.1 over the BAMNC using (5.14) to systematically identify the (p, ε) values for which the codes' performance without interleaving (with $\varepsilon > 0$) is superior to that with perfect bit interleaving (with $\varepsilon = 0$). The results, with ε shown in the form $\varepsilon_{min} \leq \varepsilon \leq \varepsilon_{max}$ for p given, are summarized in Tables 5.3 – 5.6 (the dash symbols in the tables indicate that perfect bit interleaving yields better performance for the specified p value). Thus for such channel conditions, not only can one forgo additional delay and complexity by avoiding bit interleaving, but improved performance can also be achieved as illustrated in Figs. 5.5 – 5.8.

code	n	k	t	R
C_1	255	221	17	0.867
C_2	255	129	63	0.506
C_3	127	111	8	0.874
C_4	127	65	31	0.511

Table 5.1: Parameters of the considered RS codes.

code	p	ε_{max}
C_1	7×10^{-3}	0.2
	5×10^{-3}	0.1
	4×10^{-3}	0.06
C_2	4×10^{-2}	0.3
	3×10^{-2}	0.13
	2.3×10^{-2}	0.06
C_3	1×10^{-2}	0.38
	5×10^{-3}	0.12
	4×10^{-3}	0.08
C_4	4×10^{-2}	0.26
	3×10^{-2}	0.13
	2×10^{-3}	0.04

Table 5.2: $[0, \varepsilon_{max}]$ ε-intervals for different values of p for which to avoid symbol interleaving over the BAMNC.

p	ε_{min}	ε_{max}
$\geq 3 \times 10^{-2}$	0	1
3×10^{-3}	0	0.9
2×10^{-3}	0	0.83
1×10^{-3}	0.14	0.62
$\leq 9 \times 10^{-4}$	-	-

Table 5.3: BAMNC (p, ε) values for which to avoid bit interleaving for C_1.

p	ε_{min}	ε_{max}
$\geq 5 \times 10^{-2}$	0	1
1×10^{-2}	0	0.87
5×10^{-3}	0	0.73
4×10^{-3}	0	0.64
3.8×10^{-3}	0.11	0.61
3.6×10^{-3}	0.33	0.54
$\leq 3.5 \times 10^{-3}$	-	-

Table 5.4: BAMNC (p, ε) values for which to avoid bit interleaving for C_2.

p	ε_{min}	ε_{max}
$\geq 5 \times 10^{-3}$	0	1
3×10^{-3}	0	0.87
2×10^{-3}	0	0.78
1.5×10^{-3}	0	0.69
1.2×10^{-3}	0.29	0.53
$\leq 1 \times 10^{-3}$	-	-

Table 5.5: BAMNC (p, ε) values for which to avoid bit interleaving for C_3.

p	ε_{min}	ε_{max}
$\geq 5 \times 10^{-2}$	0	1
1×10^{-2}	0	0.82
7×10^{-3}	0	0.73
5×10^{-3}	0.05	0.58
$\leq 4.5 \times 10^{-3}$	-	-

Table 5.6: BAMNC (p, ε) values for which to avoid bit interleaving for C_4.

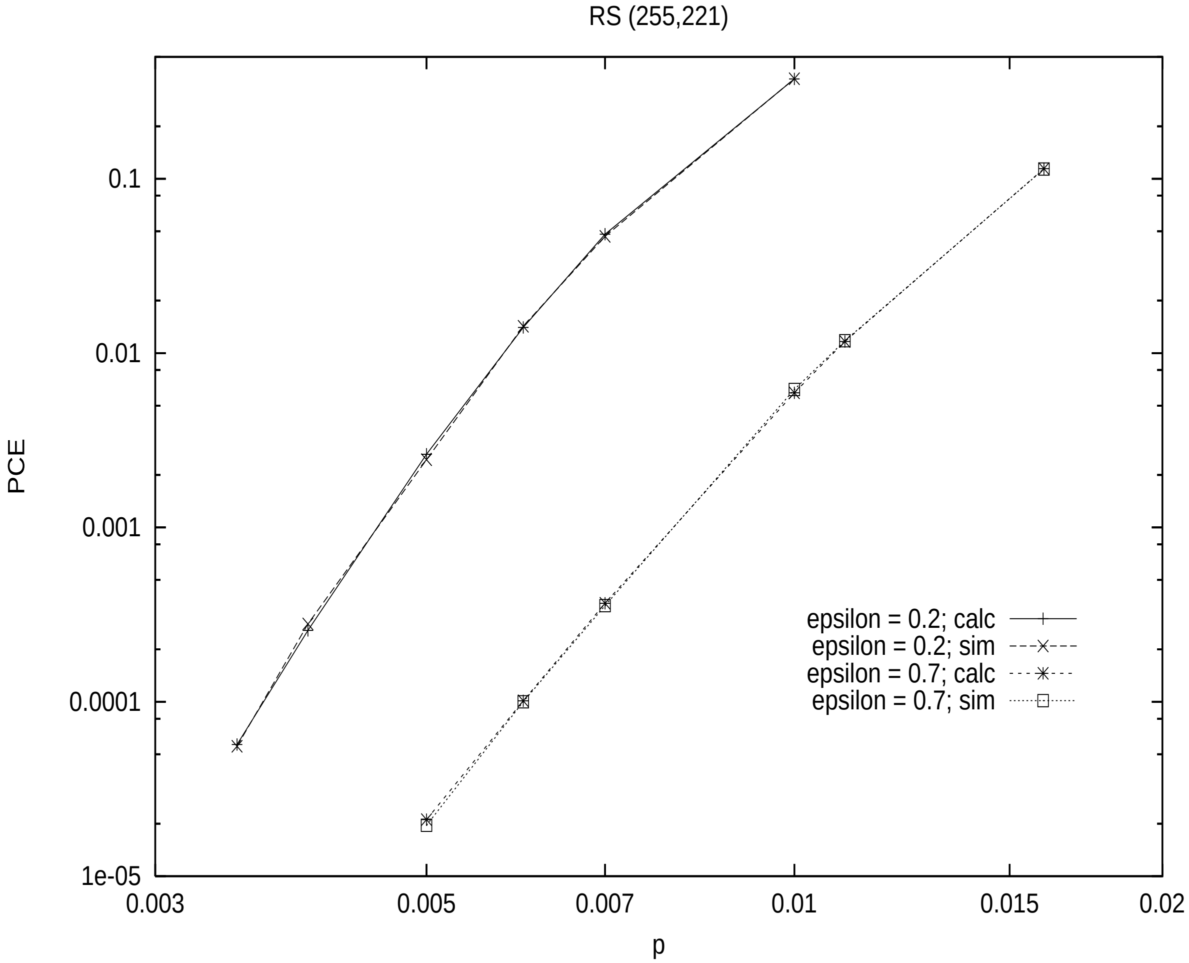


Figure 5.1: PCE for C_1 : simulation (sim.) vs analytical (calc.) results.

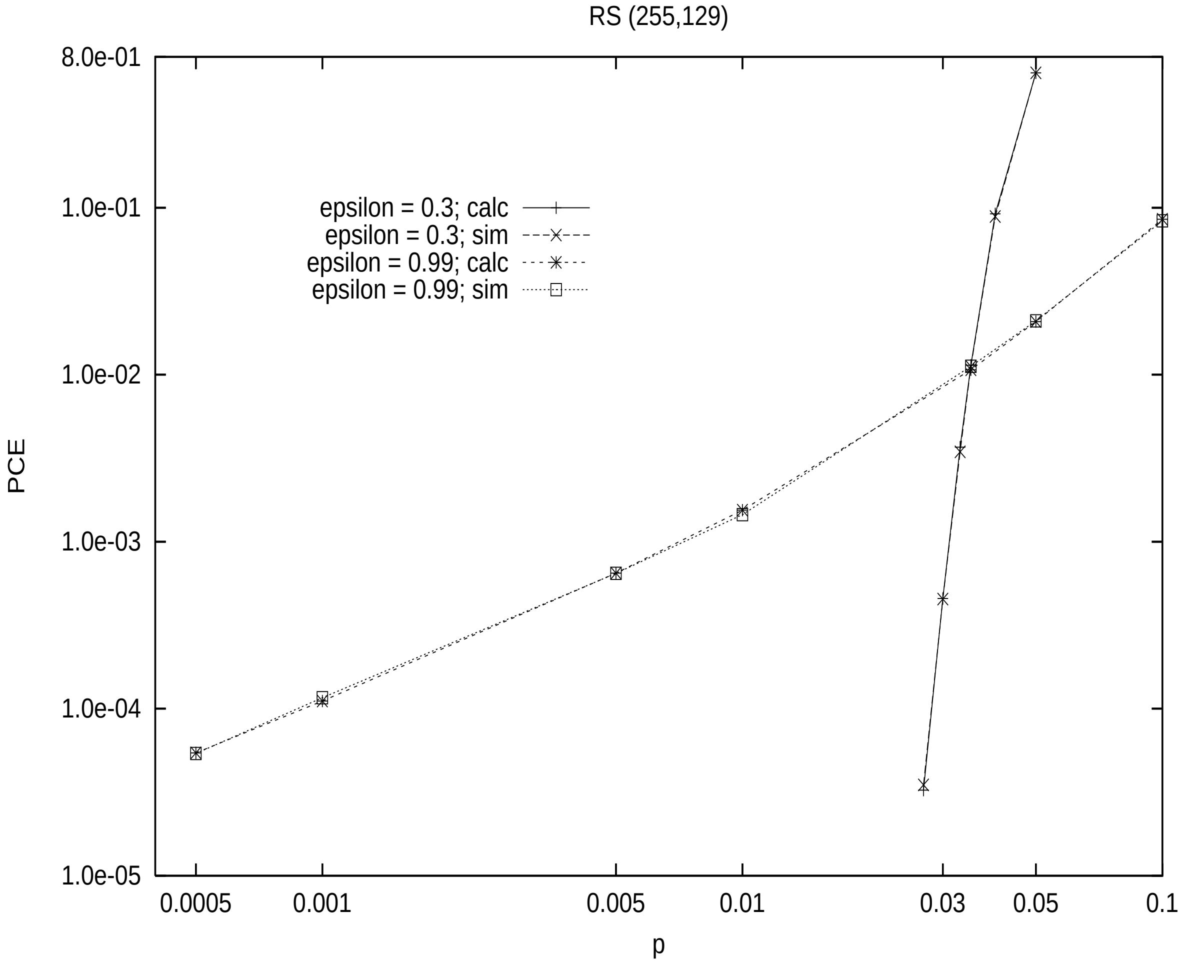


Figure 5.2: PCE for C_2 : simulation (sim.) vs analytical (calc.) results.

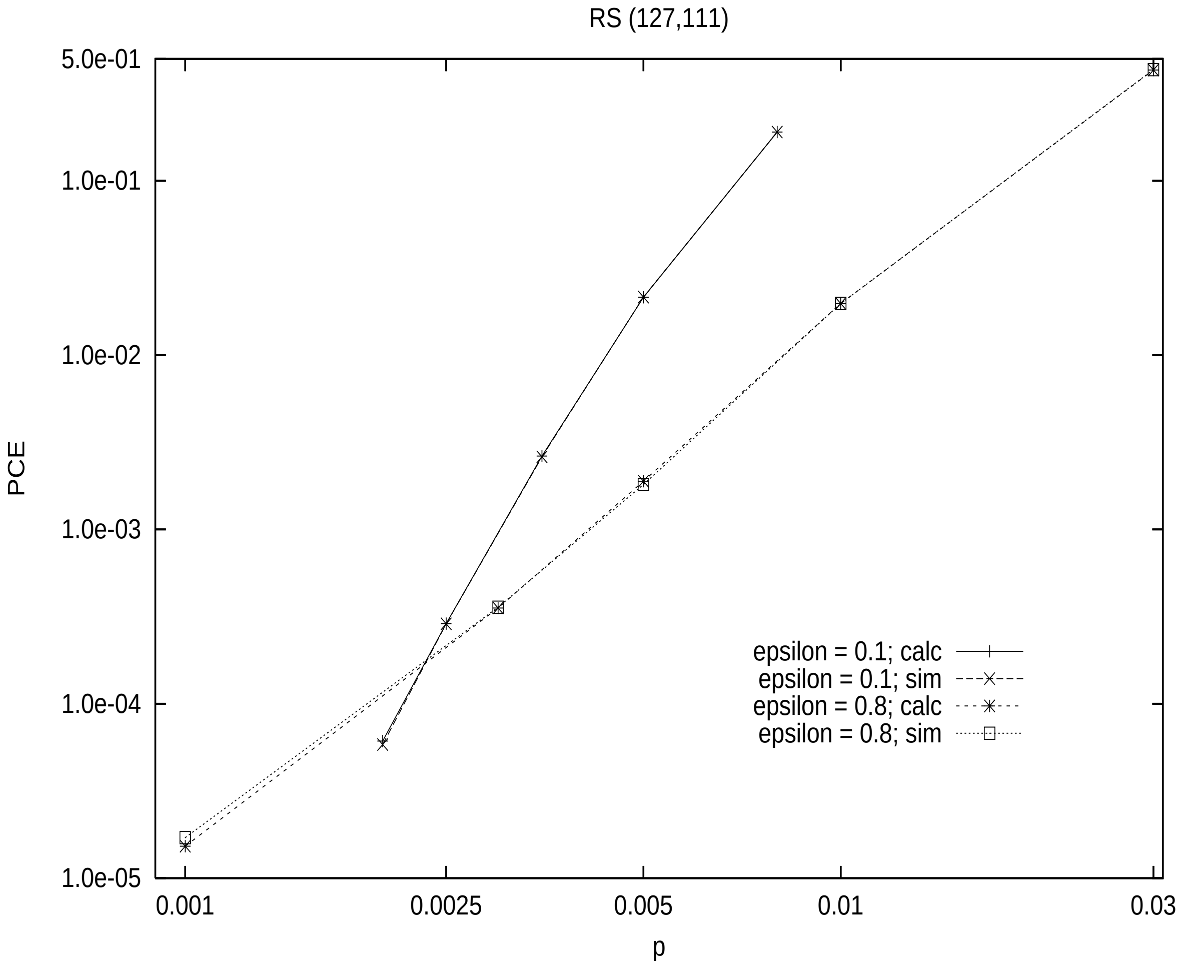


Figure 5.3: PCE for C_3 : simulation (sim.) vs analytical (calc.) results.

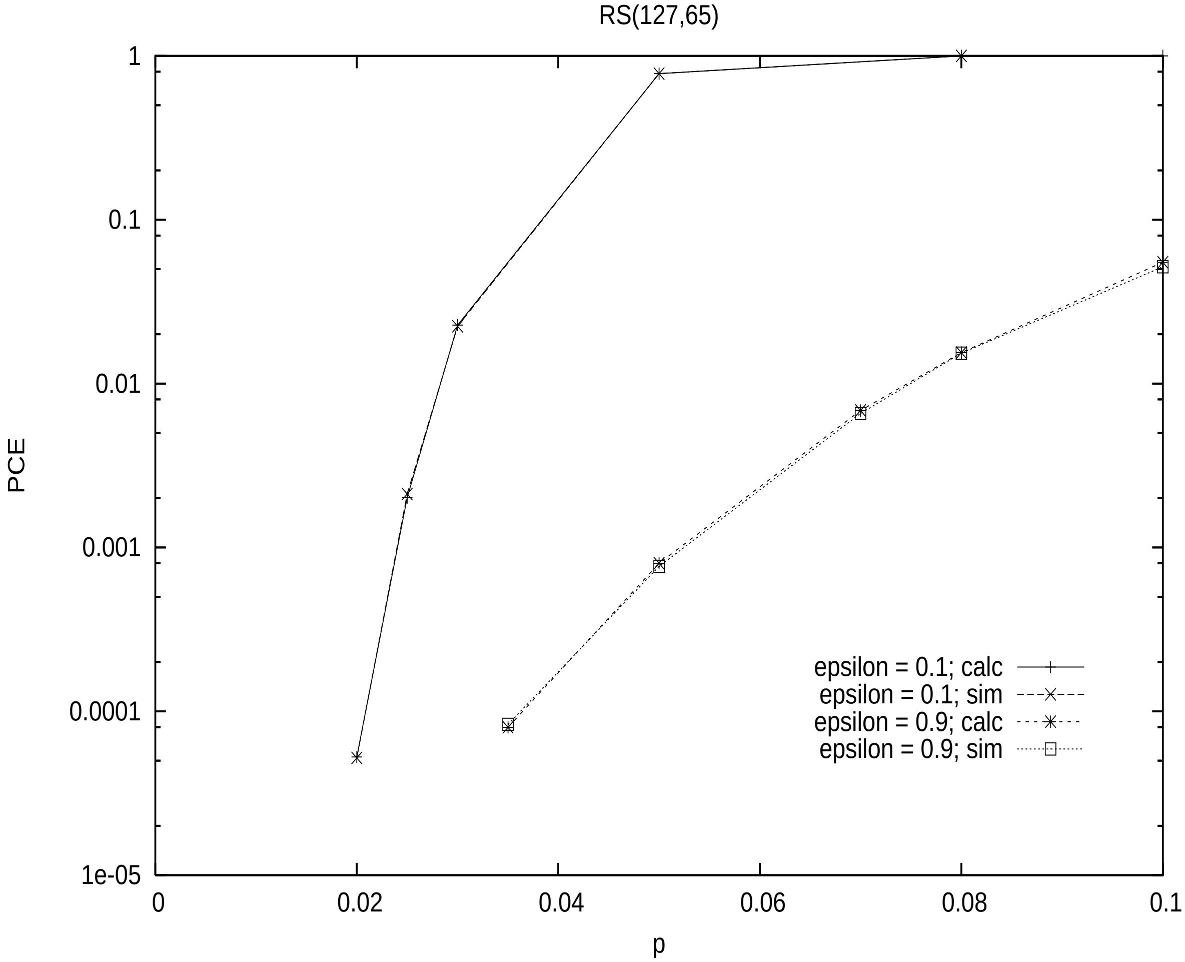


Figure 5.4: PCE for C_4 : simulation (sim.) vs analytical (calc.) results.

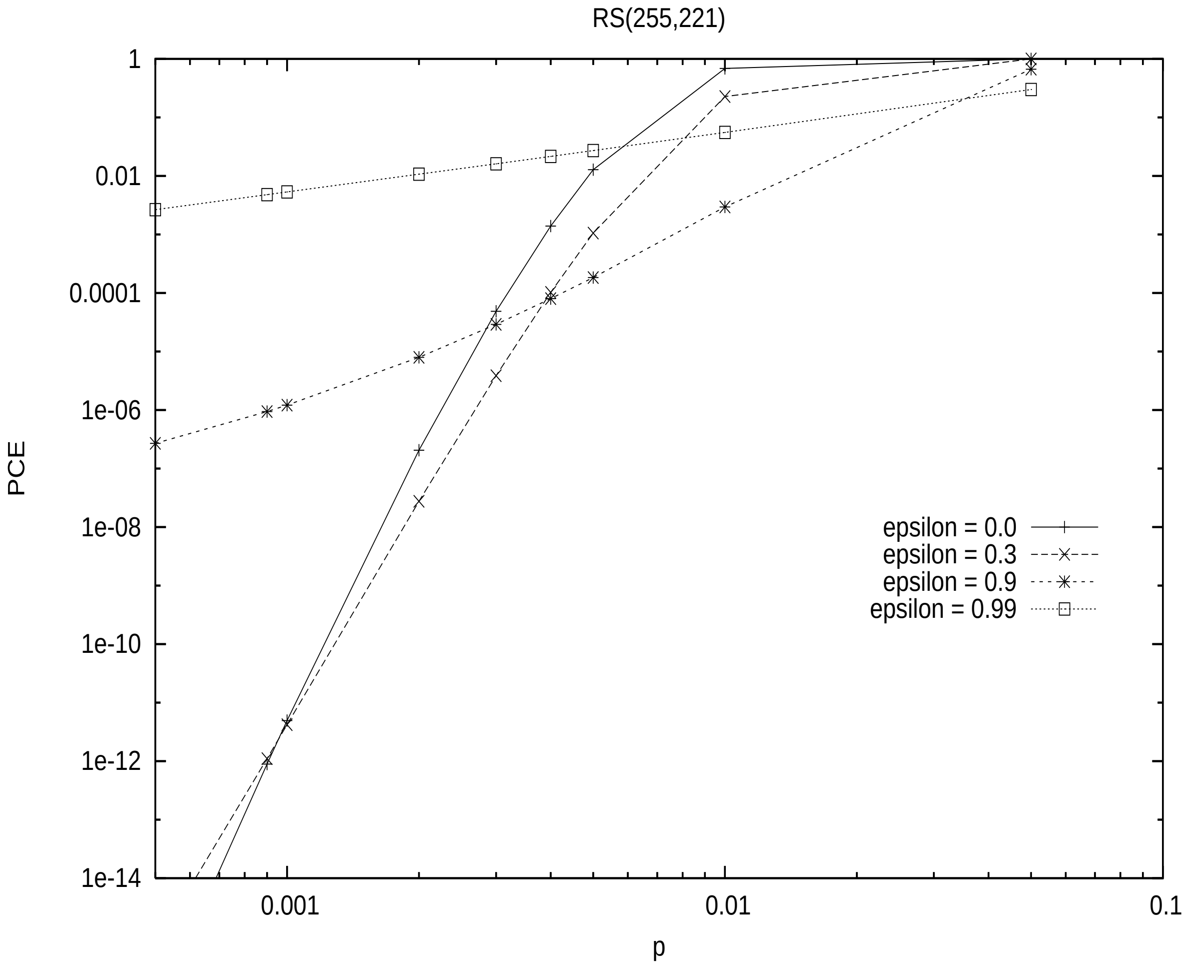


Figure 5.5: PCE for code C_1: BAMNC vs BSC.

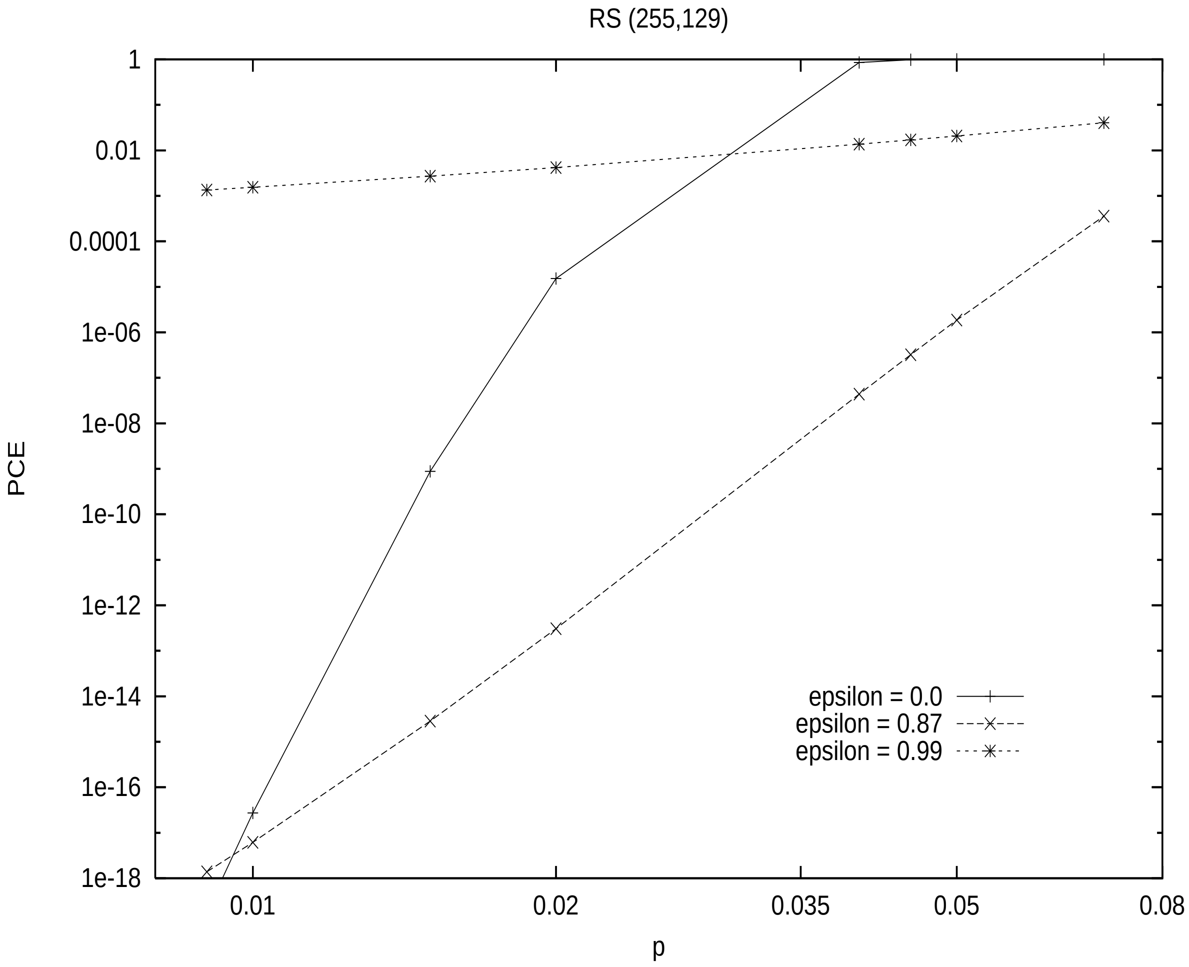


Figure 5.6: PCE for code C_2: BAMNC vs BSC.

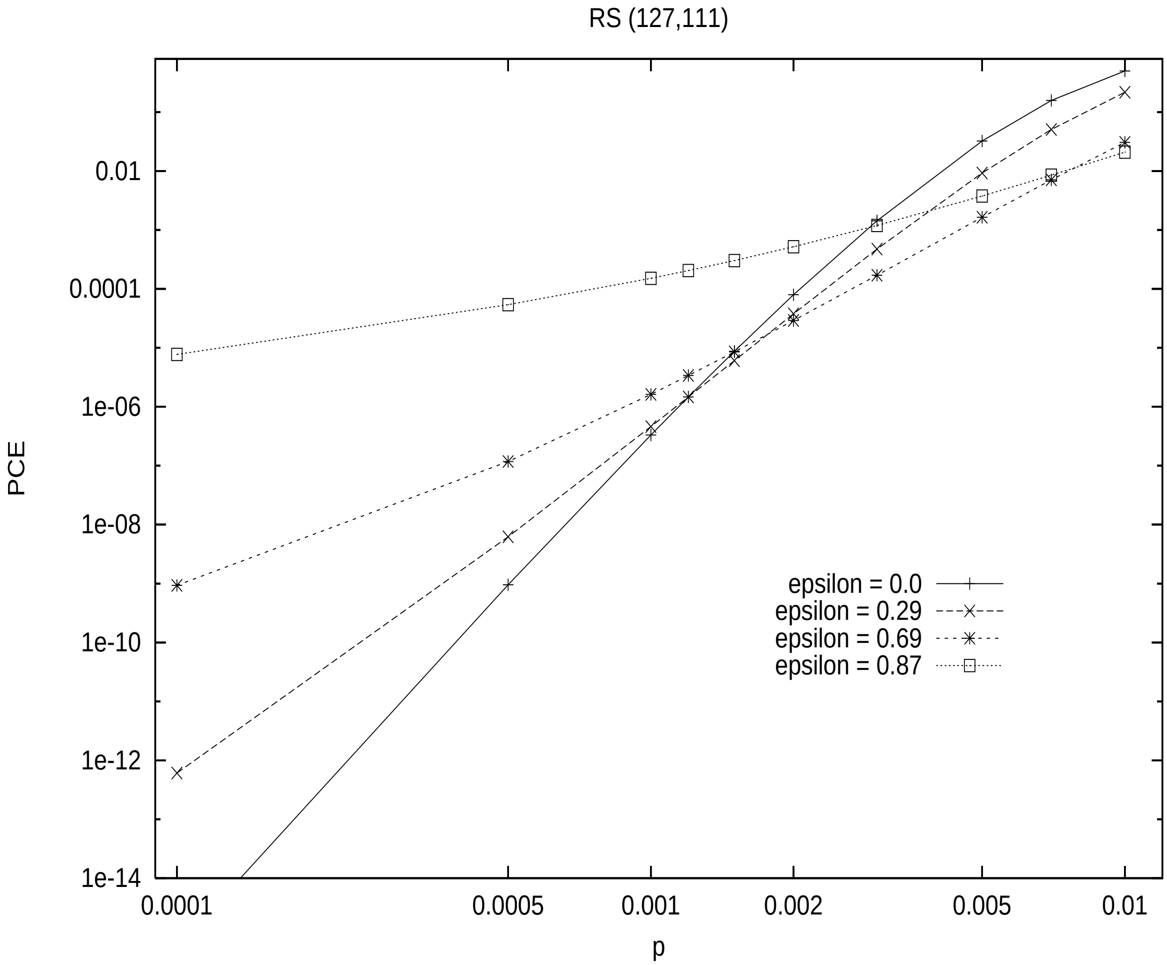


Figure 5.7: PCE for code C_3: BAMNC vs BSC.

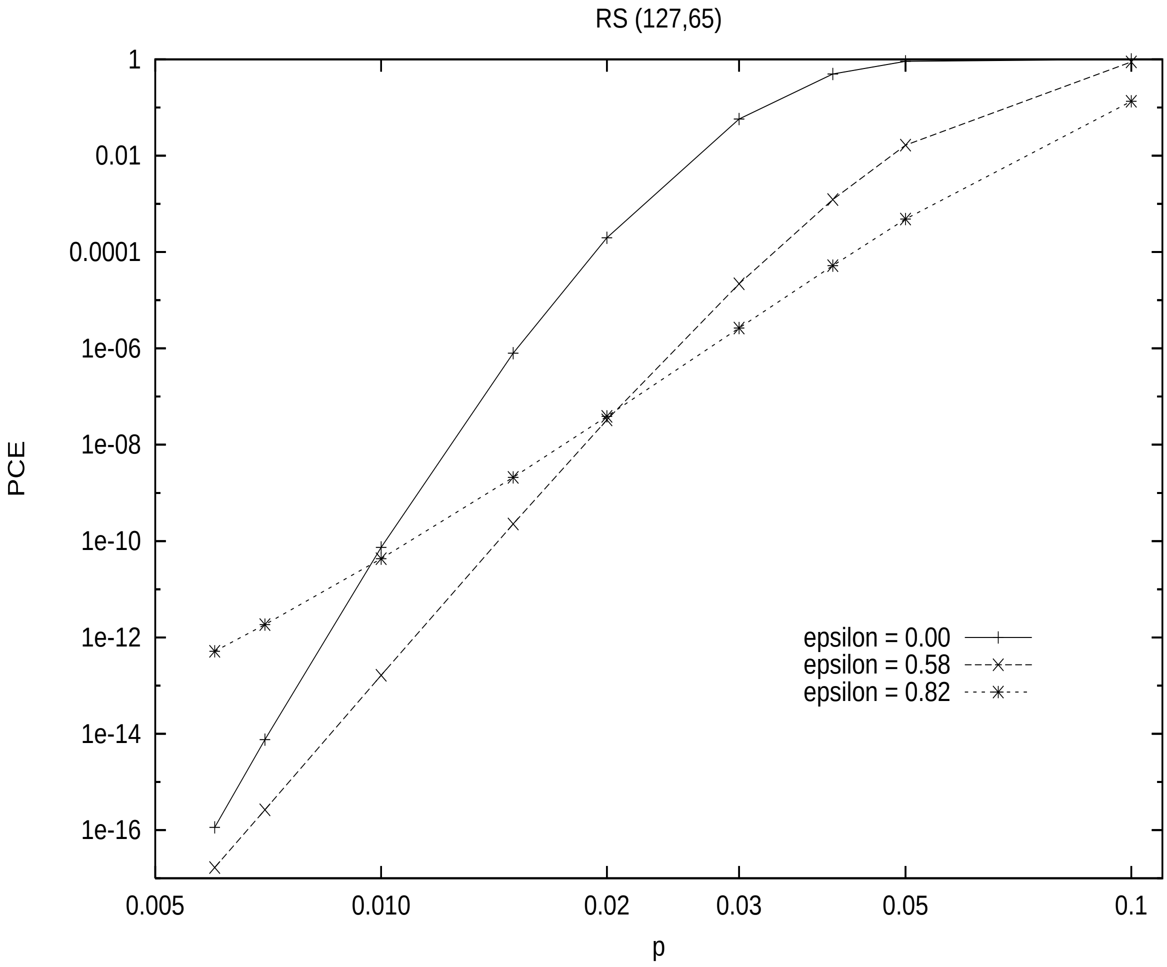


Figure 5.8: PCE for code C_4: BAMNC vs BSC.

Chapter 6

Conclusion and Future Work

The work presented in this thesis has two main themes. First of all, we study the issue of maximum-likelihood (ML) decoding of a binary linear block code on the QBC. For the case of $n \leq M$, we obtain that the ML decoding rule is identical to the case of infinite-memory Polya contagion channel studied in [1]. For the other case, $n > M$, we restrict our analysis to the basic scenario, i.e., $M = 1$, which reduces the channel to the binary additive Markov noise channel (with non-negative noise correlation). Perfect and quasi-perfect codes are examined and a relationship between ML decoding and MD decoding for these codes is established. In particular, for perfect codes, when operated over the BAMNC with positive correlation coefficient below a certain threshold, the ML and MD decoding schemes are found to be identical. This threshold is observed to decrease, as the error-correcting capability of the code increases. On the other hand, the ML decoding analysis of quasi-perfect codes leads us to propose a new decoding scheme, called MD+, that is nearly equivalent to the ML decoding under certain channel conditions. MD+, however, seems to provide negligible gain over the MD decoding when the channel correlation is high.

As far as future work is concerned, one may wish to investigate whether such relation exists for other types of linear block codes and probably suggest a different decoding algorithm for block codes that is optimal (i.e., equivalent to ML) or at least prove the optimality of MD+ under certain conditions. Another possible future direction is to design a decoder that exploits the memory between blocks by using estimates of the previous noise samples. This can result in an improved performance over the block-by-block ML and MD+ methods (studied in this thesis) at a cost of increased complexity.

We also study the performance of Reed-Solomon (RS) codes over the QBC. A close look at the performance of the RS codes when both (ideal) symbol interleaving and bit interleaving are used. We prove analytically that symbol-interleaved RS codes (as well as any other non-binary code) outperform the bit-interleaved ones over the QBC under bounded distance decoding. Four typical RS codes are also examined when transmitted over the BAMNC, and further analysis is carried out to derive a recursive expression for the probability of codeword error for the non-interleaved RS codes using a generating series approach. This is used to determine when interleaving can be avoided for those codes by numerically computing the probability of codeword error under different interleaving schemes and comparing it with the case when no interleaving is used. The advantage of using an RS code without interleaving is the reduction in system complexity and delay.

Designing a decoder for the RS-coded system that can exploit the memory in the channel is a worthy project for future work. In addition, deriving an exact or recursive expression for the probability of codeword error for non-binary block codes over the QBC with higher memory, i.e., $M > 1$, may be a possible extension to the work presented here.

Bibliography

[1] F. Alajaji and T. Fuja, “A communication channel modeled on contagion,” *IEEE Trans. Inform. Theory*, vol. 40, no. 6, pp. 2035–2041, Nov. 1994.

[2] F. Alajaji, N. Phamdo, N. Farvardin and T. Fuja, “Detection of binary Markov sources over channels with additive Markov noise,” *IEEE Trans. Inform. Theory,* vol. 42, pp. 230–239, Jan. 1996.

[3] H. Al-Lawati and F. Alajaji, “On decoding quasi-perfect codes over Markov noise channels,” *Proc. of the Tenth Canadian Workshop on Information Theory*, Edmonton, AB, Canada, June 2007.

[4] H. Al-Lawait, F. Alajaji and C. Pimentel, “When can interleaving be avoided for Reed-Solomon coded binary Markov channels ?” *Proc. of the IEEE Pacific Rim Conf. on Comm., Comp. and Sing. Proc.*, Victoria, BC, Canada, , August 2005.

[5] H. Arslan, J. F. Cheng and K. Balachandran, “Physical layer evolution for GSM/EDGE,” *Proc. of the IEEE Global Telecom. Conf.,*, San Antonio, TX, USA, Nov. 2001.

[6] F. Babich and G. Lombardi, “A Markov model for the mobile propagation channel,” *IEEE Trans. Veh. Technol.,* vol. 49, no. 1, pp. 63–73, Jan. 2000.

[7] T. Berman and J. Freedman “Non-interleaved Reed-Solomon coding over a bursty channel,” *Proc. IEEE Military Commun. Conf.,* 1992, pp. 580-583.

[8] H. Chang, C. Shung and C. Lee, “A Reed-Solomon product-code (RS-PC) decoder chip for DVD applications,” *IEEE Journal of Solid-State Circuits,* vol. 36, pp. 229-238, Feb. 2001.

[9] D. J. Costello, JR. and S. Lin, *Error Control Coding,* Second Edition, Prentice Hall, NJ, 2004.

[10] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, Second Edition, New York, Wiley, 2006.

[11] R. L. Dobrushin and M. S. Pinsker, “Memory increases transmission capacity”, *Probl. Pered. Inform.*, vol. 5, no. 1, pp. 94–95, 1969.

[12] A. Eckford, F. Kschischang and S. Pasupathy, “Analysis of low-density parity-check codes for the Gilbert-Elliott channel,” *IEEE Trans. Inform. Theory*, vol. 51, pp. 3872–3889, Nov. 2005.

[13] R.G. Gallager, *Information Theory and Reliable Communication,* Wiley, New York, 1968.

[14] J. Garcia-Frias,“Decoding of low-density parity-check codes over finite-state binary Markov channels,” *IEEE Trans. Commun.*, vol. 52, pp. 1840–1843, Nov. 2004.

[15] I. P. Goulden and D. M. Jackson, *Combinatorial Enumeration.* New York, 1983.

[16] R. M. Gray, *Entropy and Information Theory*, New York: Springer–Verlag, 1990.

[17] M. Hamada, “Near-optimality of subcodes of Hamming codes on the two-state Markovian additive channel,” *IEICE Trans. Fundamentals Electronics, Commun. and Comp. Sciences,* vol. E84-A, pp. 2383–2388, Oct. 2001.

[18] F. M. Ingels, *Information and Coding Theory,* Intext Educational Publisher, 1971.

[19] H. Labiod, “Performance of Reed Solomon error-correcting codes on fading channels,” *Proc. IEEE Int. Conf. Personal Wireless Commun.,* 1999, pp. 259-263.

[20] J. Lai and N. Mandayam, "Performance of Reed-Solomon codes for hybrid-ARQ over Rayleigh fading channels under imperfect interleaving," *IEEE Trans. Commun.,* vol. 48, pp. 1650-1659, Oct. 2000.

[21] J. H. van Lint, *Introduction to Coding Theory,* Springer-Verlag, New York, 1982.

[22] F. J. MacWilliams and N. J.A. Sloane, *The Theory of Error-Correcting Codes,* North-Holland, Amsterdam, 1977.

[23] M. Mushkin and I. Bar-David, "Capacity and coding for the Gilbert-Elliott channel," *IEEE Trans. Inform. Theory*, vol. 35, no. 6, pp. 1277–1290, Nov. 1989.

[24] V. Nagarajan and O. Milenkovic, "Performance analysis of structured LDPC codes in the Polya-urn channel with finite memory," *Proc. IEEE Canadian Conf. Elec. Comp. Eng.*, Niagara Falls, Canada, vol. 1, pp. 543–546, May 2004.

[25] C. Nicola, F. Alajaji and T. Linder, "Decoding LDPC codes over binary channels with additive Markov noise," *Proc. of the 2005 Canadian Workshop on Information Theory*, Montreal, QC, Canada, pp. 187–190, June 2005.

[26] K. Petersen, *Ergodic Theory,* Cambrige University Press, 1983.

[27] W. W. Peterson and E. J. Weldon Jr., *Error-Correcting Codes,* 2nd Ed., MIT Press, 1972.

[28] C. Pimentel, *Enumeration Techniques for Finite State Channels,* Ph.D. Dissertation, Dep. Elect. Eng., Univ. Waterloo, 1996.

[29] C. Pimentel and I. F. Blake, "Non-interleaved Reed-Solomon coding performance on finite state channels," *IEEE Int. Conf. Communication*, vol. 3, pp. 1493–1497, 1997.

[30] C. Pimentel and I. F. Blake, "Modeling burst channels using partitioned Fritchman's Markov models," *IEEE Trans. Veh. Technol.*, vol. 47, pp. 885–899, Aug. 1998.

[31] C. Pimentel and I. F. Blake, "Concatenated coding performance for FSK modulation on time-correlated Rician fading channels," *IEEE Trans. Commun.,* vol. 46, pp. 1610-1618, Dec. 1998.

[32] C. Pimentel and I. F. Blake, "Enumeration of Markov chains and burst error statistics for finite state channel models,"*IEEE Trans. Veh. Technol.,* vol. 48, no. 2, pp. 415–428, March 1999.

[33] C. Pimentel, T. H. Falk and L. Lisbôa, "Finite-state Markov modeling of correlated Rician fading channels," *IEEE Trans. Veh. Technol.,* vol. 53, no. 5, pp. 1491–1501, Sept. 2004.

[34] J. Roberts, A. Ryley, D. Jones and D. Burke, "Analysis of error-correction constraints in an optical disk," *Applied Optics,* vol. 35, pp. 3915-3924, July 1996.

[35] C. E. Shannon, “A mathematical theory of communication,” *Bell Sys. Tech. J.*, vol. 27, pp. 379–423, 623–656, 1948.

[36] C. C. Tan and N. C. Beaulieu, “First-order Markov modeling for the Rayleigh fading channel,” *Proc. IEEE GLOBECOM’98,* Sydney, Australia, vol. 6, pp. 3669–3674, Nov. 1998.

[37] —, “On first-order Markov modeling for the Rayleigh fading channel,” *IEEE Trans. Commun.,* vol. 48, no. 12, pp. 2032–2040, Dec. 2000.

[38] W. Turin, *Performance Analysis of Digital Transmission Systems.* New York: Computer Science, 1990.

[39] H. S. Wang and N. Moayeri, “Finite-state Markov channel – A useful model for radio communication channels,” *IEEE Trans. Veh. Technol.,* vol. 44, no. 1, pp. 163–171, Feb. 1995.

[40] H. S. Wang and P.-C. Chang, “On verifying the first-order Markovian assumption for a Rayleigh fading channel,” *IEEE Trans. Veh. Technol.,* vol. 45, no. 2, pp. 353–357, May 1996.

[41] S. B. Wicker, “Reed-Solomon error control coding for Rayleigh fading channels with feedback,” *IEEE Trans. Veh. Technol.,* vol. 41, no. 2, pp. 124–133, May 1992.

[42] J. Yee and E. Weldon, "Evaluation of the performance of error-correcting codes on a Gilbert channel," *IEEE Trans. Commun.*, vol. 43, pp. 2316-2323, Aug. 1995.

[43] Q. Zhang and S. Kassam, "Finite state Markov model for Rayleigh fading channels," *IEEE Trans. Commun.*, vol. 47, no. 11, pp. 1688–1692, Nov. 1999.

[44] L. Zhong, F. Alajaji and G. Takahara, "A model for a binary burst-noise channel based on a finite queue," *Proc. of the 2001 Canadian Workshop on Information Theory*, Vancouver, BC, Canada, pp. 60–63, June 2001.

[45] —, "A comparative study of burst-noise communication channel models," *Proc. of the 2002 Biennial Symposium on Communications*, Kingston, ON, Canada, pp. 437–441, June 2002.

[46] —, "A queue-based model for binary communication channels," *Proc. of Allerton Conference on Commun., Contr., and Comp.*, Monticello, IL, USA, pp. 130–139, Oct. 2003.

[47] —, "An approximation of the Gilbert-Elliott channel via a queue-based channel model," *Proc. of the 2004 IEEE International Symposium on Information Theory*, Chicago, IL, USA, pp. 63, June 2004.

[48] —, "A queue-based model for wireless Rayleigh fading channels with memory," *Proc. of IEEE 62nd Semiannual Vehicular Technology Conference*, Dallas, TX, USA, Sept. 2005.

[49] —, “A binary communication channel with memory based on a finite queue,” *IEEE Transaction on Information Theory*, vol. 53, no. 8, pp. 2815–2840, Aug. 2007.

[50] —, “A model for correlated Rician fading channels based on a finite queue,” *IEEE Trans. Veh. Technol.,* vol. 56, no. 6, Nov. 2007.