An Analytical Method for Performance Evaluation of Binary Linear Block Codes

Ali Abedi and Amir K. Khandani

Coding & Signal Transmission Laboratory
Department of Electrical & Computer Engineering
University of Waterloo
Waterloo, Ontario, Canada, N2L 3G1

Technical Report UW-E&CE#2003-1

February 8, 2003

An Analytical Method for Performance Evaluation of Binary Linear Block Codes

Ali Abedi and Amir K. Khandani
Coding & Signal Transmission Laboratory(www.cst.uwaterloo.ca)
Dept. of Elec. and Comp. Eng., University of Waterloo
Waterloo, ON, Canada, N2L 3G1
Tel: 519-8848552, Fax: 519-8884338
e-mail: {ali, khandani}@cst.uwaterloo.ca

Abstract

An analytical method for performance evaluation of binary linear block codes using an Additive White Gaussian Noise (AWGN) channel model with Binary Phase Shift Keying (BPSK) modulation is presented. We focus on the Probability Density Function (pdf) of the bit Log-Likelihood Ratio (LLR) which is expressed in terms of the Gram-Charlier series expansion. This expansion requires knowledge of the statistical moments of the bit LLR. We introduce an analytical method for calculating these moments. This is based on some recursive calculations involving certain weight enumerating functions of the code. Numerical results are provided for some examples, which demonstrate close agreement with the simulation results.

Index Terms

Additive White Gaussian Noise Channel, Binary Phase Shift Keying, Bit Decoding, Bit Error Probability, Block Codes, Log-Likelihood Ratio, Weight Distribution.

This work is financially supported by Natural Sciences and Engineering Research Council of Canada (NSERC) and by Communications and Information Technology Ontario (CITO). Earlier version [1] of this work has been presented at ISIT 2002.

I. INTRODUCTION

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

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

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

In 1993, a new class of channel codes, called Turbo-codes, were announced [7], which have an astonishing performance and at the same time allow for a simple iterative decoding method using the reliability information produced by a bit decoding algorithm. Due to the importance of Turbo-codes, there has been a growing interest among communication researchers to work on the bit decoding algorithms.

The analytical performance evaluation of symbol by symbol decoders is considered a hard task in [8], [9]. Although there is a method for calculating exact performance (in the sense of expected hamming distortion) of Viterbi decoding of convolutional codes over Binary Symmetrical Channels [10], but there has been no method for performance evaluation of bit decoding in general. Some asymptotic expressions are derived in [11] for bit error probability of binary linear block codes in the Additive White Gaussian Noise (AWGN) channel with bit decoding. The bit error probabilities of convolutional codes over AWGN channel is considered in [9] with ML decoding. An upper bound is presented in [12] for the performance of finite-delay symbol-by-symbol decoding of trellis codes over discrete memoryless channels.

In this article, we employ Gram-Charlier series expansion to find the Probability Density Function (pdf) of the bit LLR. This method is used in some other communications applications including calculation of pdf of sum of Log-Normal variates [13], evaluation of the error probability in PAM (Pulse Amplitude Modulation) digital data transmission systems with correlated symbols in the presence of inter-symbol interference and additive noise [14], computing nearly Gaussian distributions [15], and computation of the error probability of equal-gain combiner with partially coherent fading signals [16]. Reference [17] presents a method for computing an unknown pdf using infinite series (also refer to [18]). Reference [19] computes moments of phase noise and uses maximum entropy criterion [20] to find pdf.

This paper is organized as follows. In section II, the model used to analyze the problem is presented. All notations and assumptions are in this section. Computing pdf of bit LLR using Gram-Charlier expansion is presented in section III. This is an orthogonal series expansion of a given pdf which requires knowledge of the moments of the corresponding random variable. An analytical method for computing the moments of the bit LLR using Taylor expansion is proposed in section IV. It is shown in section V that we can compute the coefficients of Taylor expansion of the bit LLR recursively. We also present a closed form expression for computing the bit error probability in section VI. In section VII, the convergence issue of this approximation is discussed. Numerical results are provided in section VIII which demonstrate a close agreement between our analytical method and simulation. We conclude in section IX.

II. Modeling

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

$$\forall \mathbf{c}^i \in \mathcal{C} : \begin{cases} c_k^i = 0 \Longrightarrow \mathbf{c}^i \in \; C_k^0, \\ c_k^i = 1 \Longrightarrow \mathbf{c}^i \in \; C_k^1, \end{cases} \tag{1}$$

$$C_k^0 \cup C_k^1 = \mathcal{C}, \qquad C_k^0 \cap C_k^1 = \varnothing. \tag{2}$$

We define the following operators on the code book.

$$\mathbf{c}^i \oplus \mathbf{c}^j = \text{Bit wise binary addition of two code-words.} \tag{3}$$

Note that the sub-code C_k^0 is closed under binary addition.

The dot product of two vectors $\mathbf{a} = (a_1, a_2, \ldots, a_N)$ and $\mathbf{b} = (b_1, b_2, \ldots, b_N)$ is defined as,

$$\mathbf{a}.\mathbf{b} = \sum_{l=1}^{N} a_l b_l. \tag{4}$$

The modulation scheme used here is Binary Phase Shift Keying (BPSK) which is defined as the mapping M,

$$M : \mathbf{c} \longrightarrow \mathbf{m}(\mathbf{c}), \tag{5}$$

$$0 \longrightarrow m(0) = -1, \quad 1 \longrightarrow m(1) = 1. \tag{6}$$

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

$$\forall \mathbf{c} \in \mathcal{C} : \quad \|\mathbf{m}(\mathbf{c})\|^2 = \mathbf{m}(\mathbf{c}).\mathbf{m}(\mathbf{c}) = \sum_{l=1}^{N} m^2(c_l) = N. \tag{7}$$

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

$$-\mathbf{1}.\mathbf{m}(\mathbf{c}) = N - 2\omega(\mathbf{c}). \tag{8}$$

Modulating a code-word $\tilde{\mathbf{c}} = (\tilde{c}_1, \tilde{c}_2, ..., \tilde{c}_N)$ using BPSK and sending it through an AWGN channel, we will receive $\mathbf{x} = \mathbf{m}(\tilde{\mathbf{c}}) + \mathbf{n}$, where $\mathbf{n} = (n_1, n_2, ..., n_N)$ is an independent, identically distributed Gaussian noise vector which has zero mean elements of variance σ^2. Note that for an AWGN channel, we have,

$$p(\mathbf{x}|\tilde{\mathbf{c}}) = \frac{1}{(\sqrt{2\pi}\sigma)^N} \exp\left[-\frac{\|\mathbf{x} - \mathbf{m}(\tilde{\mathbf{c}})\|^2}{2\sigma^2}\right]. \tag{9}$$

A common tool to express the bit probabilities in bit decoding algorithms is based on using the so-called Log-Likelihood-Ratio (LLR). The LLR of the k^{th} bit position is defined by the following equation,

$$LLR(k) = \log \frac{P(\tilde{c}_k = 1|\mathbf{x})}{P(\tilde{c}_k = 0|\mathbf{x})}, \tag{10}$$

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

$$P(\tilde{c}_k = 0) = P(\tilde{c}_k = 1) = \frac{1}{2}, \tag{11}$$

and using (9), it follows,

$$LLR(k) = \log \frac{p(\mathbf{x}|\tilde{c}_k = 1)}{p(\mathbf{x}|\tilde{c}_k = 0)} \tag{12}$$

$$= \log \frac{\sum\limits_{\mathbf{c}^i \epsilon C_k^1} \exp\left[-\frac{\|\mathbf{x}-\mathbf{m}(\mathbf{c}^i)\|^2}{2\sigma^2}\right]}{\sum\limits_{\mathbf{c}^i \epsilon C_k^0} \exp\left[-\frac{\|\mathbf{x}-\mathbf{m}(\mathbf{c}^i)\|^2}{2\sigma^2}\right]}. \tag{13}$$

Using (7), it follows,

$$LLR(k) = \log \frac{\sum\limits_{\mathbf{c}^i \epsilon C_k^1} \exp\left[\frac{\mathbf{x}.\mathbf{m}(\mathbf{c}^i)}{\sigma^2}\right]}{\sum\limits_{\mathbf{c}^i \epsilon C_k^0} \exp\left[\frac{\mathbf{x}.\mathbf{m}(\mathbf{c}^i)}{\sigma^2}\right]} \tag{14}$$

$$= \log \frac{\sum\limits_{\mathbf{c}^i \epsilon C_k^1} \exp\left[\frac{\mathbf{n}.\mathbf{m}(\mathbf{c}^i)+\mathbf{m}(\tilde{\mathbf{c}}).\mathbf{m}(\mathbf{c}^i)}{\sigma^2}\right]}{\sum\limits_{\mathbf{c}^i \epsilon C_k^0} \exp\left[\frac{\mathbf{n}.\mathbf{m}(\mathbf{c}^i)+\mathbf{m}(\tilde{\mathbf{c}}).\mathbf{m}(\mathbf{c}^i)}{\sigma^2}\right]}. \tag{15}$$

Given the value of the bit LLR, decision on the value of bit k is made by comparing the $LLR(k)$ with a threshold of zero. We are interested in studying the probabilistic behavior of the $LLR(k)$ as a function of the Gaussian random vector $\mathbf{n}$.

Using the following theorems from [21] [1], we can simplify our analysis.

Theorem 1: The probability distribution of $LLR(k)$ is not affected by the choice of transmitted code-word, $\tilde{\mathbf{c}}$ as long as the value of the k^{th} bit remains unchanged.

Theorem 2: The probability distribution of $LLR(k)$ for value of bit $k = 0 \quad or \quad 1$ are the reflections of one another through the origin.

Theorem 3: The probability distribution of $LLR(k)$ is not affected by the choice of the bit position, k, for the class of Cyclic codes.

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

In this case, equation (15) reduces to,

$$LLR(k) = \log \frac{\sum\limits_{\mathbf{c}^i \epsilon C_k^1} \exp\left[\frac{\mathbf{n}.\mathbf{m}(\mathbf{c}^i)-\mathbf{1}.\mathbf{m}(\mathbf{c}^i)}{\sigma^2}\right]}{\sum\limits_{\mathbf{c}^i \epsilon C_k^0} \exp\left[\frac{\mathbf{n}.\mathbf{m}(\mathbf{c}^i)-\mathbf{1}.\mathbf{m}(\mathbf{c}^i)}{\sigma^2}\right]}. \tag{16}$$

Using (8), we obtain,

[1] For the sake of brevity, the proofs are not given here. The reader is referred to [21] for proofs.

$$LLR(k) = \log \frac{\sum\limits_{\mathbf{c}^i \epsilon C_k^1} \exp\left[\frac{\mathbf{n}.\mathbf{m}(\mathbf{c}^i) - 2\omega(\mathbf{c}^i)}{\sigma^2}\right]}{\sum\limits_{\mathbf{c}^i \epsilon C_k^0} \exp\left[\frac{\mathbf{n}.\mathbf{m}(\mathbf{c}^i) - 2\omega(\mathbf{c}^i)}{\sigma^2}\right]}. \tag{17}$$

In the following, for convenience of notation, the index k indicating bit position is dropped. This means the sets C^1 and C^0 are indeed C_k^1 and C_k^0. We use the notation $H(\mathbf{n})$ to refer to the LLR expression given in (17). i.e.,

$$H(\mathbf{n}) = \log \frac{\sum\limits_{\mathbf{c}^i \epsilon C^1} \exp\left[\frac{\mathbf{n}.\mathbf{m}(\mathbf{c}^i) - 2\omega(\mathbf{c}^i)}{\sigma^2}\right]}{\sum\limits_{\mathbf{c}^i \epsilon C^0} \exp\left[\frac{\mathbf{n}.\mathbf{m}(\mathbf{c}^i) - 2\omega(\mathbf{c}^i)}{\sigma^2}\right]}. \tag{18}$$

III. Gram-Charlier Expansion of pdf

One common method for representing a function is to use an expansion on an orthogonal basis which is suitable for that function. As the pdf of bit LLR is approximately Gaussian [7], [22], [23], the appropriate basis can be normal Gaussian pdf and its derivatives which form an orthogonal basis. There are a variety of equivalent formulations for this expansion [15], [24]–[26], while we follow the notation used in [15].

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

$$f_Y(y) \simeq \frac{1}{\sqrt{2\pi}} e^{-\frac{y^2}{2}} \sum_{i=0}^{\infty} \alpha_i T_i(y), \tag{19}$$

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

$$T_i(y) = \sum_{j=0}^{\lfloor i/2 \rfloor} \frac{(-1)^j i!}{2^j (i-2j)! j!} y^{i-2j}, \tag{20}$$

and,

$$\alpha_i = \sum_{j=0}^{\lfloor i/2 \rfloor} \frac{(-1)^j}{2^j (i-2j)! j!} \mu_{i-2j}, \tag{21}$$

where,

$$\mu_j = \int_{-\infty}^{+\infty} y^j f_Y(y) dy. \tag{22}$$

This is a commonly used method for approximating an unknown pdf. The only unknown components in (21) are the moments, μ_j. We propose an analytical method using Taylor series expansion to compute the moments of the bit LLR in the next section.

IV. Computing moments

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

$$\mu_m = E\left[\left(\frac{H(\mathbf{n}) - E[H(\mathbf{n})]}{\sqrt{\text{var}[H(\mathbf{n})]}}\right)^m\right] \tag{23}$$

$$= \frac{1}{\text{var}^{m/2}[H(\mathbf{n})]} \sum_{i=0}^{m} (-1)^i \binom{m}{i} E[H^{m-i}(\mathbf{n})] E^i[H(\mathbf{n})], \tag{24}$$

where $E[.]$ stands for expectation and var$[.]$ denotes variance. Note that to compute (24), one needs $E[H^j(\mathbf{n})], \quad j = 1, .., m$.

To compute $E[H^j(\mathbf{n})]$, we take advantage of a method similar to the so called Delta method [27] and find average of the Taylor series expansion of $H^j(\mathbf{n})$. We use the Taylor series expansion of $H(\mathbf{n})$ in conjunction with polynomial theorem [15] to find an expansion for $H^j(\mathbf{n})$.

$$H^j(\mathbf{n}) = \left(\sum_{i=0}^{\infty} \frac{1}{i!} (\mathbf{n}.\nabla)^i H(\mathbf{0})\right)^j \tag{25}$$

An alternative approach is to directly expand $H^j(\mathbf{n})$. Note that derivatives of $H^j(\mathbf{n})$ are functions of derivatives of $H(\mathbf{n})$.

It easily follows that calculating $E[H^j(\mathbf{n})]$, using Taylor series expansion, involves computing the following terms,

$$\frac{\partial^L H(\mathbf{0})}{\partial n_1^{l_1} \partial n_2^{l_2} ... \partial n_N^{l_N}} E[n_1^{l_1}] E[n_2^{l_2}] ... E[n_N^{l_N}], \tag{26}$$

where L, l_i's, $i = 1, 2, ..., N$ are even and satisfy,

$$l_1 + l_2 + ... + l_N = L. \tag{27}$$

Note that for a Gaussian random variable n and an integer l, we have,

$$E[n^l] = \begin{cases} \frac{(l)!\sigma^l}{2^{l/2}(l/2)!}, & l \text{ even}, \\ 0, & l \text{ odd}. \end{cases} \quad (28)$$

The number of solutions to equation (27) can be obtained using the method described in [28] for partitioning an integer. Each solution corresponds to one partial derivative.

V. Taylor Expansion of LLR

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

$$H(\mathbf{n}) = H(\mathbf{0}) + \mathbf{n}.\nabla H(\mathbf{0}) + \frac{1}{2!}(\mathbf{n}.\nabla)^2 H(\mathbf{0}) + ... \quad (29)$$

$$= H(\mathbf{0}) + \sum_{q_1=1}^{N} \frac{\partial H(\mathbf{0})}{\partial n_{q_1}} n_{q_1} + \frac{1}{2} \sum_{q_1=1}^{N} \sum_{q_2=1}^{N} \frac{\partial^2 H(\mathbf{0})}{\partial n_{q_1} \partial n_{q_2}} n_{q_1} n_{q_2} + ... \quad (30)$$

We continue with calculation of different terms in the above equation. For simplicity, we define (18) as $H(\mathbf{n}) = \log A(\mathbf{n}) - \log B(\mathbf{n})$, where,

$$A(\mathbf{n}) = \sum_{\mathbf{c}^i \epsilon C^1} \exp\left[\frac{\mathbf{n}.\mathbf{m}(\mathbf{c}^i) - 2\omega(\mathbf{c}^i)}{\sigma^2}\right], \quad (31)$$

and $B(\mathbf{n})$ has a similar formula. We only consider $\log A(\mathbf{n})$ hereafter in this section. The same approach can be used for $\log B(\mathbf{n})$.

$$\log A(\mathbf{n}) = \log A(\mathbf{0}) + \sum_{q_1=1}^{N} \frac{\partial \log A(\mathbf{0})}{\partial n_{q_1}} n_{q_1} + \frac{1}{2} \sum_{q_1=1}^{N} \sum_{q_2=1}^{N} \frac{\partial^2 \log A(\mathbf{0})}{\partial n_{q_1} \partial n_{q_2}} n_{q_1} n_{q_2} + \ldots \quad (32)$$

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

$$F_{\{q_1,..,q_j\}}(\mathbf{n}) = \frac{\partial^j A(\mathbf{n})}{\partial n_{q_1} \partial n_{q_2} ... \partial n_{q_j}} = \sigma^{-2j} \sum_{\mathbf{c}^i \epsilon C^1} M^i_{\{q_1,..,q_j\}} \exp\left[\frac{\mathbf{n}.\mathbf{m}(\mathbf{c}^i) - 2\omega(\mathbf{c}^i)}{\sigma^2}\right], \quad j \geq 1, \quad (33)$$

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

$$M^i_{\{q_1,..,q_j\}} = \prod_{l=1}^{j} m(c^i_{q_l}), \quad j \geq 1, \tag{34}$$

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

$$R_{\{q_1,..,q_j\}}(\mathbf{n}) = A^{-1}(\mathbf{n})F_{\{q_1,..,q_j\}}(\mathbf{n}), \quad j \geq 1, \tag{35}$$

where $A(\mathbf{n})$ and $F_{\{q_1,..,q_j\}}(\mathbf{n})$ are given in (31) and (33), respectively.

The functions $A(\mathbf{n})$, $F_{\{q_1,..,q_j\}}(\mathbf{n})$, and $R_{\{q_1,..,q_j\}}(\mathbf{n})$ defined in (31), (33), and (35) reduce to special weight distribution functions when $\mathbf{n} = \mathbf{0}$,

$$A(\mathbf{0}) = \mathcal{A}(Z) = \sum_{w=0}^{N} a(w)Z^w, \tag{36}$$

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

$$F_{\{q_1,..,q_j\}}(\mathbf{0}) = \mathcal{F}_{\{q_1,..,q_j\}}(Z) = \sigma^{-2j} \sum_{w=0}^{N} [f^+_{\{q_1,..,q_j\}}(w) - f^-_{\{q_1,..,q_j\}}(w)]Z^w, \quad j \geq 1, \tag{37}$$

where $f^{\pm}_{\{q_1,..,q_j\}}(w)$, is the number of code-words $\mathbf{c}^i \in C^1$ with Hamming weight w and $M^i_{\{q_1,..,q_j\}} = \pm 1$.

$$R_{\{q_1,..,q_j\}}(\mathbf{0}) = \mathcal{R}_{\{q_1,..,q_j\}}(Z) = \mathcal{A}^{-1}(Z)\mathcal{F}_{\{q_1,..,q_j\}}(Z), \quad j \geq 1. \tag{38}$$

We can compute $F_{\{q_1,..,q_j\}}(\mathbf{0})$, using the trellis diagram of the code. This is achieved by constructing a new trellis diagram and augmenting each state into two states according to the values of $M^i_{\{q_1,..,q_{j_0}\}}$ where $j_0 = 1,..,j$.

To simplify (32), it easily follows that,

$$\frac{\partial \log A(\mathbf{n})}{\partial n_{q_1}} = A^{-1}(\mathbf{n})F_{\{q_1\}}(\mathbf{n}) = R_{\{q_1\}}(\mathbf{n}), \tag{39}$$

$$\frac{\partial \log A(\mathbf{0})}{\partial n_{q_1}} = \mathcal{R}_{\{q_1\}}(Z). \tag{40}$$

Replacing (39) and (40) in (32), we have,

$$\log A(\mathbf{n}) = \log \mathcal{A}(Z) + \sum_{q_1=1}^{N} \mathcal{R}_{\{q_1\}}(Z) n_{q_1} + \frac{1}{2} \sum_{q_1=1}^{N} \sum_{q_2=1}^{N} \frac{\partial R_{\{q_1\}}(\mathbf{0})}{\partial n_{q_2}} n_{q_1} n_{q_2} + \dots \tag{41}$$

To compute (41), one needs derivatives of $R_{\{q_1\}}(\mathbf{n})$, which can be calculated using the following theorem.

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

$$\frac{\partial R_{\{q_1,..,q_j\}}(\mathbf{n})}{\partial n_{q_i}} = \begin{cases} \sigma^{-4} R_{\{q_1,..,q_{i-1},q_{i+1},..,q_j\}}(\mathbf{n}) - R_{\{q_1,..,q_j\}}(\mathbf{n}) R_{\{q_i\}}(\mathbf{n}), & \text{If } q_i \in \{q_1,..,q_j\}, \\ R_{\{q_1,..,q_j,q_i\}}(\mathbf{n}) - R_{\{q_1,..,q_j\}}(\mathbf{n}) R_{\{q_i\}}(\mathbf{n}), & \text{Otherwise.} \end{cases} \tag{42}$$

Proof: For proof refer to Appendix A. ■

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

Theorem 5: We have,

$$\frac{\partial^2 R_{\{q_1,..,q_j\}}(\mathbf{n})}{\partial n_{q_i}^2} = -2 R_{\{q_i\}}(\mathbf{n}) \frac{\partial R_{\{q_1,..,q_j\}}(\mathbf{n})}{\partial n_{q_i}}. \tag{43}$$

Proof: For proof refer to Appendix A. ■

Referring to (42), one can easily see that the coefficients of the expansion (41) are polynomials of $R_{\{q_1,..,q_j\}}(\mathbf{0})$ for different values of j. It is noteworthy that these coefficients are polynomials of special weight distribution functions defined in (38). The above theorems and results enable us to compute all the derivatives required in the Taylor series expansion of $H(\mathbf{n}) = \log A(\mathbf{n}) - \log B(\mathbf{n})$.

VI. Computing Probability of Error

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

Using theorem 2, we have,

$$P(e|\tilde{c}_k = 0) = P(e|\tilde{c}_k = 1), \tag{44}$$

where event e corresponds to bit k being in error. Using assumption (11), we can write,

$$P(e) = P(e|\tilde{c}_k = 0)P(\tilde{c}_k = 0) + P(e|\tilde{c}_k = 1)P(\tilde{c}_k = 1) = P(e|\tilde{c}_k = 0). \tag{45}$$

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

$$P(e) = \int_a^{\infty} f_Y(y)dy, \tag{46}$$

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

$$P(e) \simeq \int_a^{\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{y^2}{2}} \sum_{i=0}^{\infty} \alpha_i T_i(y)dy. \tag{47}$$

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

$$P(e) \simeq \int_a^{\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{y^2}{2}} dy + \int_a^{\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{y^2}{2}} \sum_{i=1}^{\infty} \alpha_i T_i(y)dy \tag{48}$$

$$= Q(a) + \int_a^{\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{y^2}{2}} \sum_{i=1}^{\infty} \alpha_i T_i(y)dy. \tag{49}$$

Changing the order of integration and summation and using the following property,

$$e^{-\frac{y^2}{2}} T_i(y) = -\frac{d}{dy}\left[e^{-\frac{y^2}{2}} T_{i-1}(y)\right], \quad i \geq 1, \tag{50}$$

we can write,

$$P(e) \simeq Q(a) - \frac{1}{\sqrt{2\pi}} \sum_{i=1}^{\infty} \alpha_i \int_a^{\infty} d\left[e^{-\frac{y^2}{2}} T_{i-1}(y)\right] \tag{51}$$

$$= Q(a) - \frac{1}{\sqrt{2\pi}} \sum_{i=1}^{\infty} \alpha_i \left[e^{-\frac{y^2}{2}} T_{i-1}(y)\right]_a^{\infty} \tag{52}$$

$$= Q(a) + \frac{1}{\sqrt{2\pi}} e^{-\frac{a^2}{2}} \sum_{i=1}^{\infty} \alpha_i T_{i-1}(a). \tag{53}$$

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

VII. Convergence properties

Convergence properties of Gram-Charlier expansion is investigated in [24], [29], [30]. It is proved in [31], that the expansion is convergent if the expanded function satisfies the following condition,

$$\int_{-\infty}^{+\infty} f_Y(y)e^{y^2/4}dy < \infty. \tag{54}$$

Reference [13], mentions that this expansion has good asymptotic behavior as defined in [32]. In other words, a few terms will give a close approximation.

General properties of Hermite polynomials are discussed in [33], where it is shown that this class of polynomials form an orthogonal basis which span the interval $(-\infty,+\infty)$. Therefore, pdf of the bit LLR can be expanded arbitrarily closely, in mean square sense, using the given set of orthogonal basis. i.e.,

$$\lim_{l\longrightarrow\infty}\int_{-\infty}^{+\infty} \epsilon_l^2(y)dy \longrightarrow 0, \tag{55}$$

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

$$\epsilon_l(y) = f_Y(y) - \frac{1}{\sqrt{2\pi}}e^{-y^2/2}\sum_{i=0}^{l}\alpha_i T_i(y). \tag{56}$$

If $f_Y(y)$ is piecewise continuous in the interval $(-\infty,+\infty)$, the result of this expansion converges to $f_Y(y)$ at each point of $(-\infty,+\infty)$ at which $f_Y(y)$ is continuous. At points where $f_Y(y)$ has a jump discontinuity, this series converges to $(f_Y(y^+) + f_Y(y^-))/2$ [34]. It seems that the pdf of the bit LLR is a continuous function, although there is no straight forward proof for it. In the following, we show that the error in the computation of bit error probability converges to zero, no matter $f_Y(y)$ is continuous or not.

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

Using Cauchy-Schwartz inequality [35],

$$\left|\int_{-\infty}^{+\infty} f(y)g(y)dy\right|^2 < \int_{-\infty}^{+\infty} |f(y)|^2dy \int_{-\infty}^{+\infty} |g(y)|^2dy, \tag{57}$$

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

$$g(y) = \begin{cases} 1, & a < y < b, \\ 0, & \text{Otherwise,} \end{cases} \tag{58}$$

we have,

$$\left| \int_a^b \epsilon_l(y) dy \right|^2 < (b-a) \int_{-\infty}^{+\infty} \epsilon_l^2(y) dy \tag{59}$$

Applying (55) to (59), results in,

$$\lim_{l \longrightarrow \infty} \int_a^b \epsilon_l(y) dy \longrightarrow 0. \tag{60}$$

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

VIII. Numerical Results

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

As an example, we used a (15,11,3) Cyclic code and evaluated its performance using the proposed method. The order of the Gram-Charlier expansion is 10. The comparison between the analytically calculated BER and the one obtained from simulation is shown in Figure 1. From theorem 3, we know that in the case of Cyclic codes, the computed pdf is not affected by the choice of the bit position.

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

The last example is the binary extended (24,12,8) Golay code. Its performance is shown in Figure 3. The bit error rate is calculated using Gram-Charlier series with 14 term.

IX. Concluding Remarks

A method is presented for calculating bit error probability of binary linear block codes over AWGN channel, using special weight enumerating functions of the code. Summary of the proposed method is presented here. Starting with calculation of special weight distribution functions defined in (38), proceed with Taylor series of LLR as indicated in (29). Averaging this expansion will give us moments of the pdf of bit LLR, which can be used to compute coefficients of Gram-Charlier series using (21). A closed form expression (53) can be used to find the bit error probability. All these steps can be seen in Figure 4.

We are currently working on extending this method for performance evaluation of Turbo-codes. Some existing approaches are bounds on the performance of Turbo-codes like the upper bound derived in [36].

Acknowledgments

The authors would like to thank M. H. Baligh, A. Heunis, and M. Thompson for their helpful discussions and comments.

APPENDIX

A. Proofs of theorems

Theorem 4:

Proof: Using (35), one can write,

$$\frac{\partial R_{\{q_1,..,q_j\}}(\mathbf{n})}{\partial n_{q_i}} = \frac{\partial}{\partial n_{q_i}}\left[A^{-1}(\mathbf{n})F_{\{q_1,..,q_j\}}(\mathbf{n})\right] \tag{61}$$

$$= A^{-1}(\mathbf{n})\frac{\partial F_{\{q_1,..,q_j\}}(\mathbf{n})}{\partial n_{q_i}} + \frac{\partial A^{-1}(\mathbf{n})}{\partial n_{q_i}}F_{\{q_1,..,q_j\}}(\mathbf{n}) \tag{62}$$

$$= A^{-1}(\mathbf{n})\frac{\partial F_{\{q_1,..,q_j\}}(\mathbf{n})}{\partial n_{q_i}} - A^{-2}(\mathbf{n})\frac{\partial A(\mathbf{n})}{\partial n_{q_i}}F_{\{q_1,..,q_j\}}(\mathbf{n}) \tag{63}$$

$$= A^{-1}(\mathbf{n})\frac{\partial F_{\{q_1,..,q_j\}}(\mathbf{n})}{\partial n_{q_i}} - A^{-2}(\mathbf{n})F_{\{q_i\}}(\mathbf{n})F_{\{q_1,..,q_j\}}(\mathbf{n}) \tag{64}$$

$$= A^{-1}(\mathbf{n})\frac{\partial F_{\{q_1,..,q_j\}}(\mathbf{n})}{\partial n_{q_i}} - [A^{-1}(\mathbf{n})F_{\{q_1,..,q_j\}}(\mathbf{n})][A^{-1}(\mathbf{n})F_{\{q_i\}}(\mathbf{n})] \tag{65}$$

$$= A^{-1}(\mathbf{n})\frac{\partial F_{\{q_1,..,q_j\}}(\mathbf{n})}{\partial n_{q_i}} - R_{\{q_1,..,q_j\}}(\mathbf{n})R_{\{q_i\}}(\mathbf{n}). \tag{66}$$

Using (33) and noting that $m^2(c^l_{q_i}) = 1$, we have,

$$\frac{\partial F_{\{q_1,..,q_j\}}(\mathbf{n})}{\partial n_{q_i}} = \begin{cases} \sigma^{-4}F_{\{q_1,..,q_{i-1},q_{i+1},...,q_j\}}(\mathbf{n}), & \text{If } q_i \in \{q_1,..,q_j\}, \\ F_{\{q_1,..,q_j,q_i\}}(\mathbf{n}), & \text{Otherwise.} \end{cases} \tag{67}$$

Substituting (67) in (66), and using (35), completes the proof. ∎

Theorem 5:

Proof: We consider two different cases. If $q_i \in \{q_1, .., q_j\}$ using (42), one can write,

$$\frac{\partial^2 R_{\{q_1,..,q_j\}}(\mathbf{n})}{\partial n_{q_i}^2} = \frac{\partial}{\partial n_{q_i}}[\sigma^{-4} R_{\{q_1,..,q_{i-1},q_{i+1},..,q_j\}}(\mathbf{n}) - R_{\{q_1,..,q_j\}}(\mathbf{n}) R_{\{q_i\}}(\mathbf{n})] \tag{68}$$

$$=\sigma^{-4}\frac{\partial}{\partial n_{q_i}}[R_{\{q_1,..,q_{i-1},q_{i+1},..,q_j\}}(\mathbf{n})] - \frac{\partial}{\partial n_{q_i}}[R_{\{q_1,..,q_j\}}(\mathbf{n}) R_{\{q_i\}}(\mathbf{n})] \tag{69}$$

$$\begin{aligned}=&\sigma^{-4}[R_{\{q_1,..,q_j\}}(\mathbf{n}) - R_{\{q_1,..,q_{i-1},q_{i+1},..,q_j\}}(\mathbf{n}) R_{\{q_i\}}(\mathbf{n})]\\ &- \frac{\partial}{\partial n_{q_i}}[R_{\{q_1,..,q_j\}}(\mathbf{n})] R_{\{q_i\}}(\mathbf{n}) - R_{\{q_1,..,q_j\}}(\mathbf{n}) \frac{\partial}{\partial n_{q_i}}[R_{\{q_i\}}(\mathbf{n})]\end{aligned} \tag{70}$$

$$\begin{aligned}=&\sigma^{-4}[R_{\{q_1,..,q_j\}}(\mathbf{n}) - R_{\{q_1,..,q_{i-1},q_{i+1},..,q_j\}}(\mathbf{n}) R_{\{q_i\}}(\mathbf{n})]\\ &- [\sigma^{-4} R_{\{q_1,..,q_{i-1},q_{i+1},..,q_j\}}(\mathbf{n}) - R_{\{q_1,..,q_j\}}(\mathbf{n}) R_{\{q_i\}}(\mathbf{n})] R_{\{q_i\}}(\mathbf{n})\\ &- R_{\{q_1,..,q_j\}}(\mathbf{n})[\sigma^{-4} - R^2_{\{q_i\}}(\mathbf{n})]\end{aligned} \tag{71}$$

$$= - 2\sigma^{-4} R_{\{q_1,..,q_{i-1},q_{i+1},..,q_j\}}(\mathbf{n}) R_{\{q_i\}}(\mathbf{n}) + 2R_{\{q_1,..,q_j\}}(\mathbf{n}) R^2_{\{q_i\}}(\mathbf{n}) \tag{72}$$

$$= - 2R_{\{q_i\}}(\mathbf{n})[\sigma^{-4} R_{\{q_1,..,q_{i-1},q_{i+1},..,q_j\}}(\mathbf{n}) - R_{\{q_1,..,q_j\}}(\mathbf{n}) R_{\{q_i\}}(\mathbf{n})] = -2R_{\{q_i\}}(\mathbf{n}) \frac{\partial R_{\{q_1,..,q_j\}}(\mathbf{n})}{\partial n_{q_i}}. \tag{73}$$

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

$$\frac{\partial^2 R_{\{q_1,..,q_j\}}(\mathbf{n})}{\partial n_{q_i}^2} = \frac{\partial}{\partial n_{q_i}}[R_{\{q_1,..,q_j,q_i\}}(\mathbf{n}) - R_{\{q_1,..,q_j\}}(\mathbf{n})R_{\{q_i\}}(\mathbf{n})] \tag{74}$$

$$= \frac{\partial}{\partial n_{q_i}}[R_{\{q_1,..,q_j,q_i\}}(\mathbf{n})] - \frac{\partial}{\partial n_{q_i}}[R_{\{q_1,..,q_j\}}(\mathbf{n})R_{\{q_i\}}(\mathbf{n})] \tag{75}$$

$$\begin{aligned} =&\sigma^{-4}R_{\{q_1,..,q_j\}}(\mathbf{n}) - R_{\{q_1,..,q_j,q_i\}}(\mathbf{n})R_{\{q_i\}}(\mathbf{n}) \\ &- \frac{\partial}{\partial n_{q_i}}[R_{\{q_1,..,q_j\}}(\mathbf{n})]R_{\{q_i\}}(\mathbf{n}) - R_{\{q_1,..,q_j\}}(\mathbf{n})\frac{\partial}{\partial n_{q_i}}[R_{\{q_i\}}(\mathbf{n})] \end{aligned} \tag{76}$$

$$\begin{aligned} =&\sigma^{-4}R_{\{q_1,..,q_j\}}(\mathbf{n}) - R_{\{q_1,..,q_j,q_i\}}(\mathbf{n})R_{\{q_i\}}(\mathbf{n}) \\ &- [R_{\{q_1,..,q_j,q_i\}}(\mathbf{n}) - R_{\{q_1,..,q_j\}}(\mathbf{n})R_{\{q_i\}}(\mathbf{n})]R_{\{q_i\}}(\mathbf{n}) \\ &- R_{\{q_1,..,q_j\}}(\mathbf{n})[\sigma^{-4} - R^2_{\{q_i\}}(\mathbf{n})] \end{aligned} \tag{77}$$

$$= -2R_{\{q_1,..,q_j,q_i\}}(\mathbf{n})R_{\{q_i\}}(\mathbf{n}) + 2R_{\{q_1,..,q_j\}}(\mathbf{n})R^2_{\{q_i\}}(\mathbf{n}) \tag{78}$$

$$= -2R_{\{q_i\}}(\mathbf{n})[R_{\{q_1,..,q_j,q_i\}}(\mathbf{n}) - R_{\{q_1,..,q_j\}}(\mathbf{n})R_{\{q_i\}}(\mathbf{n})] = -2R_{\{q_i\}}(\mathbf{n})\frac{\partial R_{\{q_1,..,q_j\}}(\mathbf{n})}{\partial n_{q_i}}. \tag{79}$$

It can be seen from (73) and (79) that both cases ended up to the same expression as one in (43), which completes the proof. ■

REFERENCES

[1] A. Abedi, A. K. Khandani, "An Analytical Method for Performance Analysis of Binary Linear Block Codes," *Proceedings of the IEEE International Symposium on Information Theory (ISIT 2002)*, Lausanne, Switzerland, pp. 403, July 2002. (available from www.cst.uwaterloo.ca)

[2] L. R. Bahl, J. Cocke, F. Jelinek, J. Raviv, "Optimal Decoding of Linear Codes for Minimizing Symbol Error Rate," *IEEE Transactions on Information Theory*, vol. 20, pp. 284-287, March 1974.

[3] L. Ping, K. L. Yeung, "Symbol-by-Symbol Decoding of the Golay Code and Iterative Decoding of Concatenated Golay Codes," *IEEE Transactions on Information Theory*, vol. 45, no. 7, pp. 2558-2562, November 1999.

[4] Y. Liu, S. Lin, M. P. C. Fossorier, "MAP Algorithms for Decoding Linear Block Codes Based on Sectionalized Trellis Diagrams," *IEEE Transactions on Communications*, vol. 48, no. 4, pp. 577-586, April 2000.

[5] S. Riedel, "Symbol-by-Symbol MAP Decoding Algorithm For High-Rate Convolutional Codes That Use Reciprocal Dual Codes," *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 2, pp. 175-185, February 1998.

[6] C. R. P. Hartmann, L. D. Rudolph, "An Optimum Symbol-by-Symbol Decoding Rule for Linear Codes," *IEEE Transactions on Information Theory*, vol. 22, no. 5, pp. 514-517, September 1976.

[7] C. Berrou, A. Glavieux, P. Thitimajshima, "Near Shannon Limit Error-Correcting Coding and Decoding: Turbo-Codes (1)," *Proceedings of IEEE International Conference on Communications*, Geneva, Switzerland, pp. 1064-1070, May 1993.

[8] J. G. Proakis, *Digital Communication*, 2nd ed., New York: McGraw-Hill, 1989.

[9] S. S. Pietrobon, "On the Probability of Error of Convolutional Codes," *IEEE Transactions on Information Theory*, vol. 42, no. 5, pp. 1562-1568, September 1996.

[10] M. R. Best, M. V. Burnashev, Y. Levy, A. Rabinovich, P. C. Fishburn, A. R. Calderbank, and D. J. Costello, Jr., "On a technique to calculate the exact performance of a convolutional code," *IEEE Transactions on Information Theory*, vol. 41, no. 2, pp. 441-447, March 1995.

[11] C. R. P. Hartmann, L. D. Rudolph, K. G. Mehrotra, "Asymptotic Performance of Optimum Bit-by-Bit Decoding for the White Gaussian Channel," *IEEE Transactions on Information Theory*, vol. 23, no. 4, pp. 520-522, July 1977.

[12] E. Baccarelli, R. Cusani, G. D. Blasio, "Performance Bound and Trellis-Code Design Criterion for Discrete Memoryless Channels and Finite-Delay Symbol-by-Symbol Decoding," *IEEE Transactions on Communications*, vol. 45, no. 10, pp. 1192-1199, October 1997.

[13] D. C. Schleher, "Generalized Gram-Charlier Series with Application to the Sum of Log-Normal Variates," *IEEE Transactions on Information Theory*, vol. 23, no. 2, pp. 275-280, March 1977.

[14] G. L. Cariolaro, S. G. Pupolin, "Considerations on Error Probability in Correlated-Symbol Systems," *IEEE Transactions on Communications*, vol. 25, no. 4, pp. 462-467, April 1977.

[15] S. Blinnikov, R. Moessner, "Expansions for Nearly Gaussian Distributions," *Astronomy and Astrophysics Supplement Series*, vol. 130, no. 1, pp. 193-205, May 1998.

[16] M. A. Najib, V. K. Prabhu, "Analysis of Equal-Gain Diversity with Partially Coherent Fading Signals," *IEEE Transactions on Vehicular Technology*, vol. 49, no. 3, pp. 783-791, May 2000.

[17] N. C. Beaulieu, "An Infinite Series for the Computation of the Complementary Probability Distribution of a Sum of Independent Random Variables and Its Application to the Sum of Rayleigh Random Variables," *IEEE Transactions on Communications*, vol. 38, no. 9, pp. 1463-1474, September 1990.

[18] C. Tellambura, A. Annamalai, "Further Results on the Beaulieu Series," *IEEE Transactions on Communications*, vol. 48, no. 11, pp. 1774-1777, November 2000.

[19] I. T. Monroy, G. Hooghiemstra, "On a Recursive Formula for the Moments of Phase Noise," *IEEE Transactions on Communications*, vol. 48, no. 6, pp. 917-920, June 2000.

[20] M. Kavehrad, M. Joseph, "Maximum Entropy and the Method of moments in Performance Evaluation of Digital Communications Systems," *IEEE Transactions on Communications*, vol. 34, pp. 1183-1189, December 1986.

[21] A. Abedi, P. Chaudhari, A. K. Khandani, "On Some Properties of Bit Decoding Algorithms," *Proceedings of the Canadian Workshop on Information Theory (CWIT 2001)*, Vancouver, Canada, pp. 106-109, June 2001 (available from www.cst.uwaterloo.ca)

[22] H. El-Gamal, A. R. Hammons Jr., "Analyzing the Turbo Decoder Using the Gaussian Approximation", *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 671-686, February 2001.

[23] D. Divsalar, S. Dolinar, F. Pollara, “Iterative Turbo Decoder Analysis Based on Density Evolution,” *IEEE Journal on Selected Areas in Communications*, vol. 19, no. 5, pp. 891-907, May 2001.

[24] R.S. Freedman, “On Gram-Charlier Approximations,” *IEEE Transactions on Communications*, vol. 29, no. 2, pp. 122-125, February 1981.

[25] M. Abramowitz, *Handbook of Mathematical Functions*, Washington, U.S. Dept. of Commerce, 1958.

[26] J. F. Kenney, E. S. Keeping, *Mathematics of Statistics*, Part 2, Second edition Princeton, NJ: Van Nostrand, 1951.

[27] A. W. Van der vaart, *Asymptotic Statistics*, Cambridge University press, 1998.

[28] I. Niven, *Mathematics of Choice*, The mathematical association of America, 1965.

[29] H. Cramer, *Mathematical Methods of Statistics*, Princeton University press, 1957.

[30] H. Cramer, *Random Variables and Probability Distributions*, 3rd edition, Cambridge University Press, London, 1970.

[31] H. Cramer “On Some Classes of Series Used in Mathematical Statistics,” *Proc. of the 6th Scand. Congress of Mathematicians*, Copenhagen. pp. 399-425, 1925.

[32] E. T. Whittaker, G. N. Watson, *A Course of Modern Analysis*, 4th edition, Cambridge University Press, London, 1962.

[33] G. Szego, *Orthogonal Polynomials*, Providence, R. I. American Mathematical Society, 1967.

[34] P. V. Oneil, *Advanced Engineering Mathematics*, 4th edition, International Thomson Publishing, 1995.

[35] I. S. Gradshteyn, I. M. Ryzhik, *Table of Integrals, Series, and Products*, 5th edition, Academic press, 1994.

[36] T. M. Duman, M. Salehi, “New Performance Bounds for Turbo Codes,” *IEEE Transactions on Communications*, vol. 46, no. 6, pp. 717-723, June 1998.

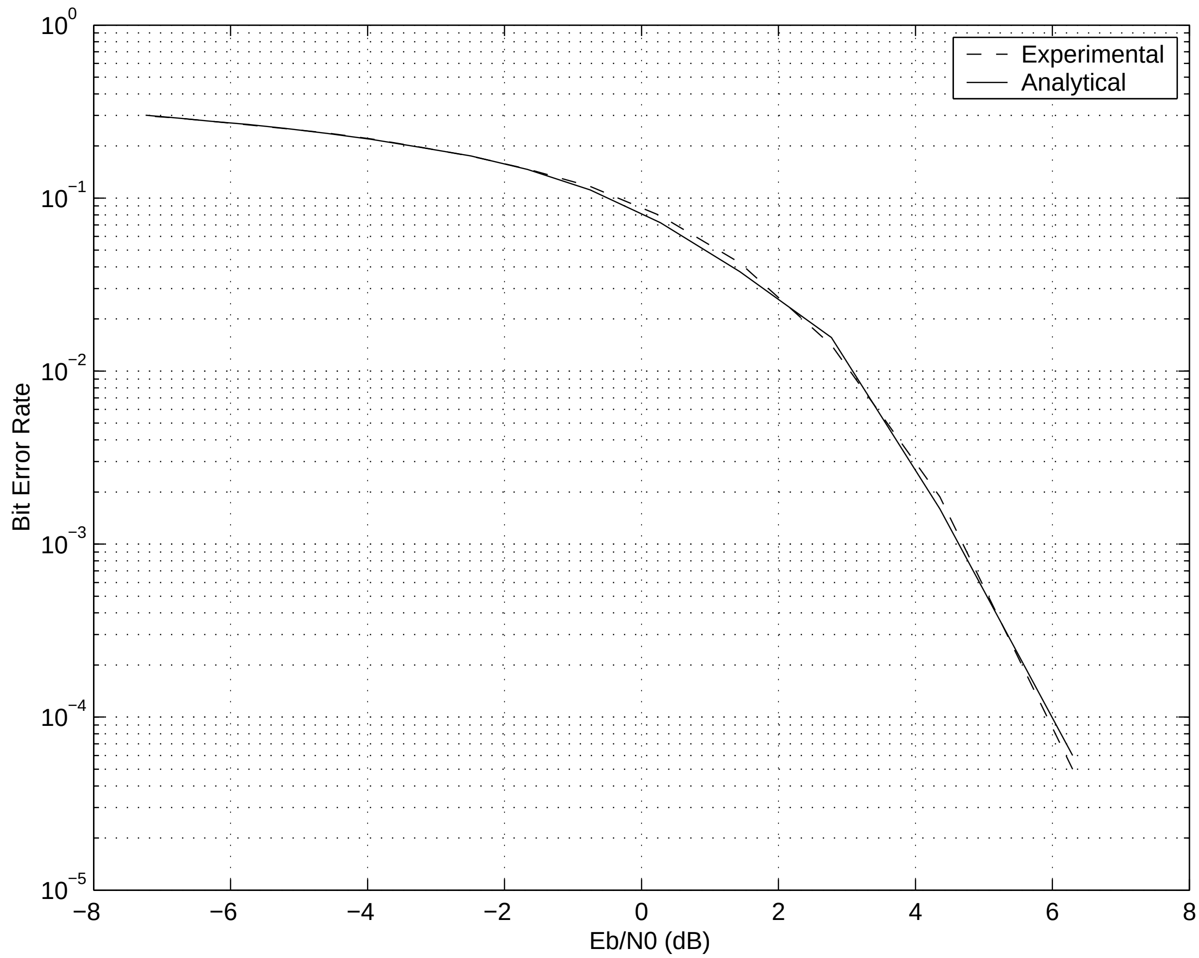

Fig. 1. Comparison between analytical and experimental BER for (15,11,3) Cyclic code.

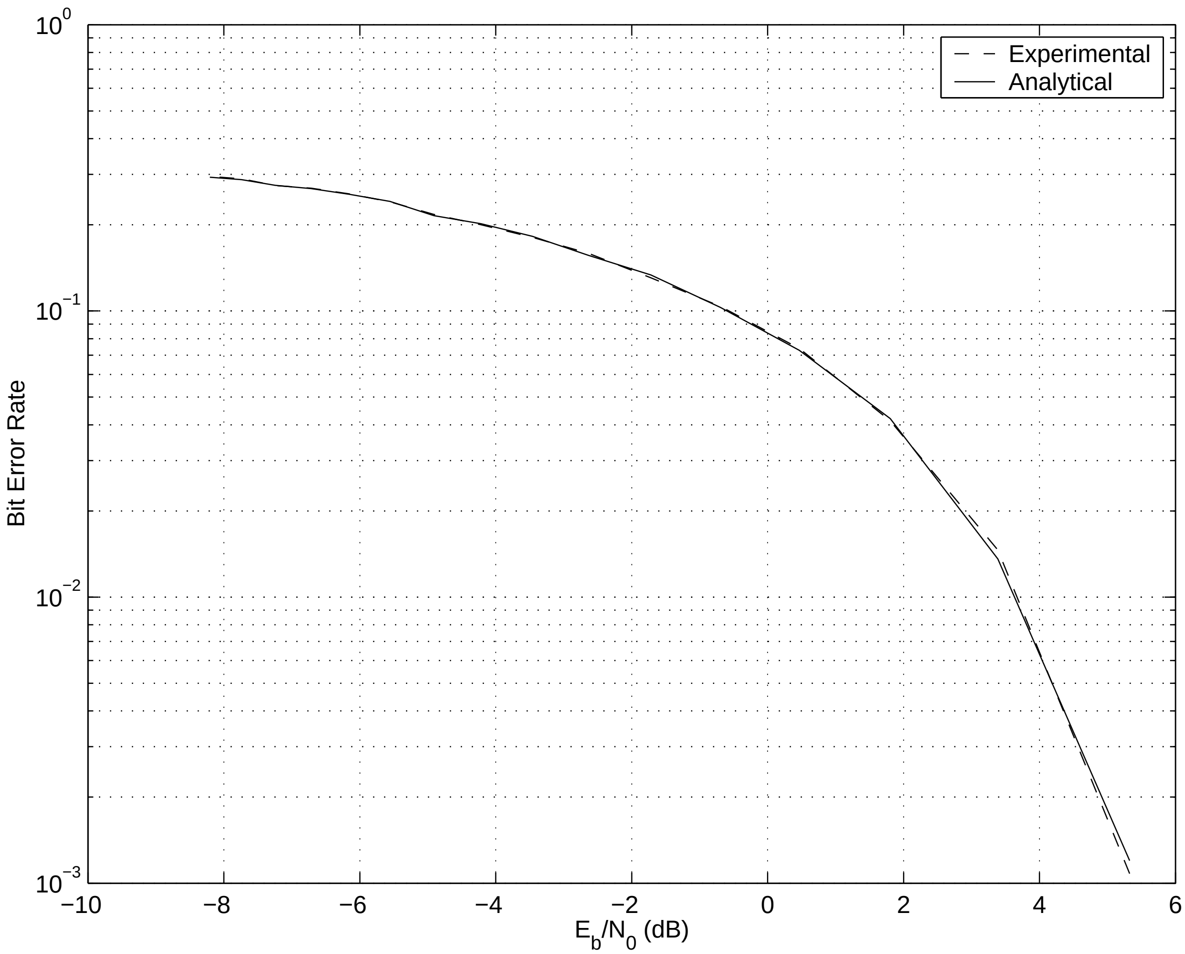

Fig. 2. Comparison between analytical and experimental BER for (12,11,2) single parity check code.

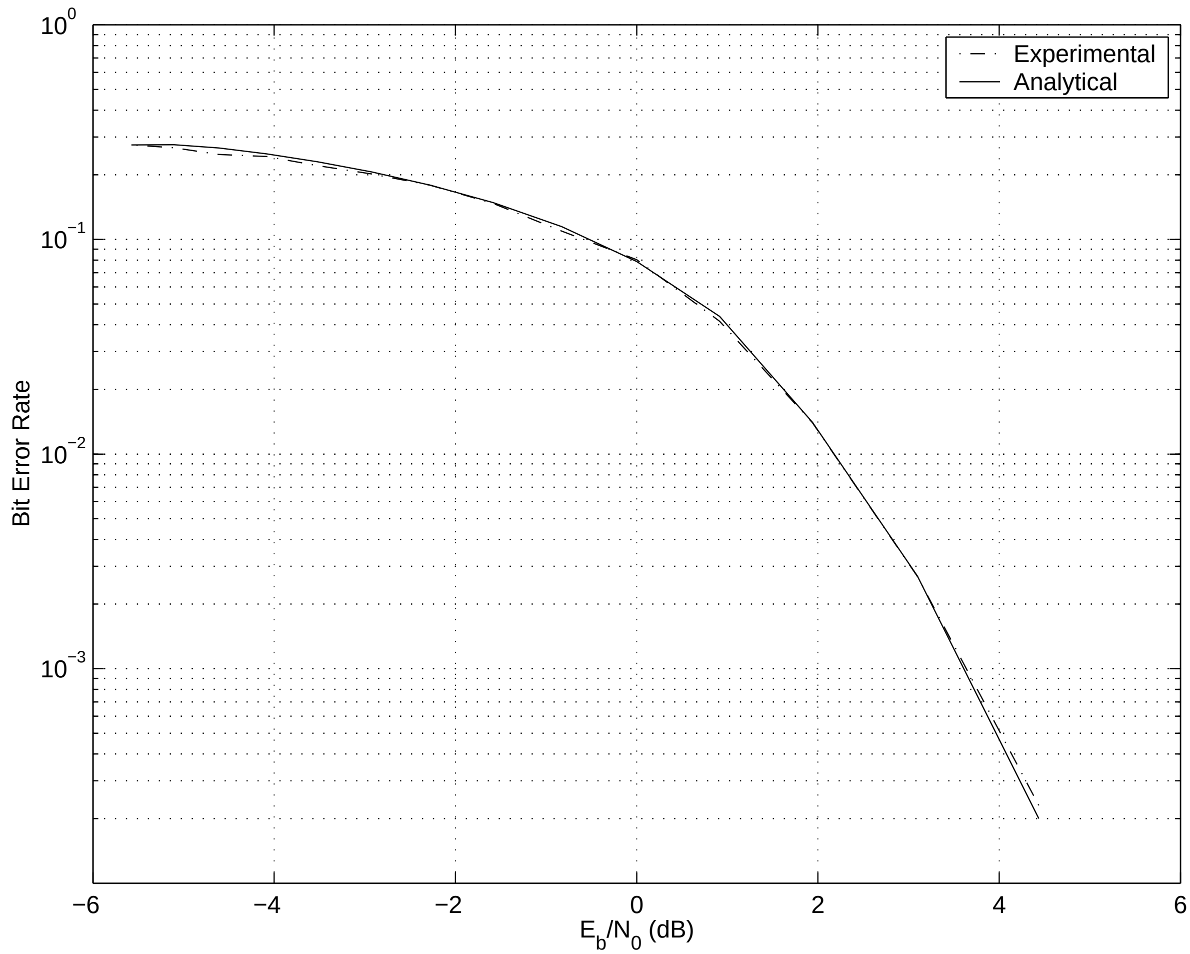

Fig. 3. Comparison between analytical and experimental BER for binary extended (24,12,8) Golay code.

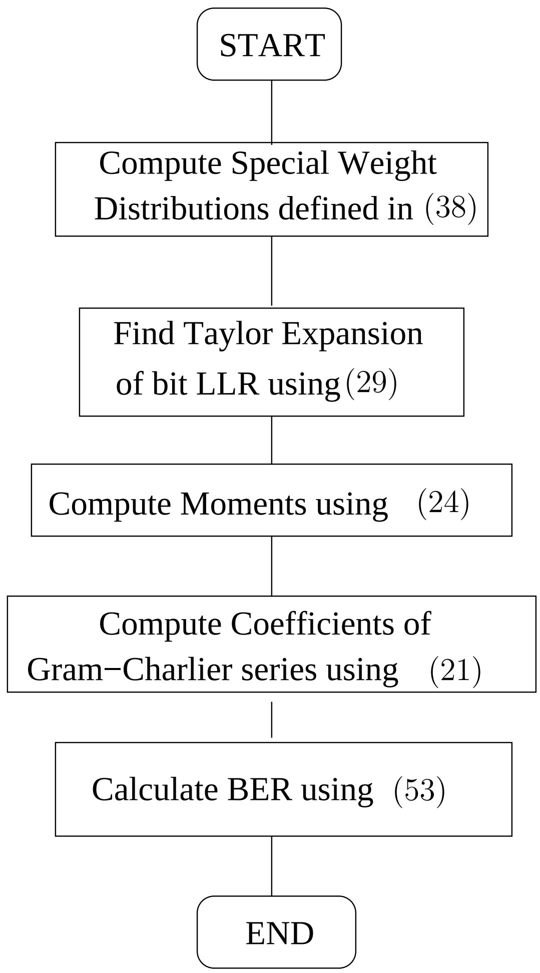

Fig. 4. Flow chart of the analytical method for performance evaluation of binary linear block codes.