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
e-mail: {ali, khandani}@cst.uwaterloo.ca, Tel: 519-8848552, Fax: 519-8884338

Abstract— [1] **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 distribution 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 straight-forward recursive calculations involving certain weight enumerating functions of the code. It is shown that the estimate of the bit error probability provided by the proposed method will asymptotically converge to the true bit error performance. Numerical results are provided for the (15,11) Cyclic code which demonstrate close agreement with the simulation results.**

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. Note that unlike ML algorithms, in the case of the bit decoding algorithms the collection of decoded bits do not necessarily form a valid code-word. 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 position of interest.

Maximum Likelihood decoding algorithms have been the subject of numerous research activities, while bit decoding algorithms have received much less attention in the past. The reason being that the bit decoding algorithms are known to offer a BER performance very close to that of ML algorithms, while they have a substantially higher level of decoding complexity. More recently, bit decoding algorithms have received increasing attention, mainly due to the fact that they deliver reliability information.

In 1993, a new class of channel codes, called Turbo-codes, were announced by Berrou *et. al.* [1], 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.

Some asymptotic expressions are derived in [2] for bit error probability under optimum decoding for the AWGN channels. Reference [3] examines the performance of linear block codes when used on AWGN channel and computes approximations of the error probability for low values of signal to noise ratio. There have been also some works on bounds and approximation on the bit error probabilities of decoding convolutional codes [4] and trellis codes [5].

Another class of research works have addressed the problem of computing tight lower and upper bounds on the error probability of binary block codes. This includes the lower bounds given in [6, 7], and the upper bounds given in [8, 9]. More recently various lower bounds have been derived on the performance of Turbo-codes in [10–12]. All these bounding techniques are based on ML block decoding (versus bit decoding as used in the current article). Reference [13] presents lower and upper bounds on the block error probability including the effect of non-uniform source probability. Recently, we have investigated some properties of bit decoding algorithms over a general channel model [14]. Some of these properties are used in special case of AWGN channel in this article. Reference [15] provides an interesting discussion on certain properties of the bit *LLR*.

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. We present some useful theorems on bit decoding algorithms in section III. Computing *pdf* of bit *LLR* using Gram-Charlier expansion is presented in section IV. This is an orthogonal series expansion of a given *pdf* which requires knowledge of the moments of the corresponding random variable. After introducing Taylor expansion of the bit *LLR* in section V, an analytical method for computing the moments of the bit *LLR* using Taylor expansion is introduced in section VI. We also present a closed form expression for computing the bit error probability in section VII. In section VIII it is shown that the estimate of the bit error probability provided by the proposed method will asymptotically converge to the true bit error performance. Numerical results are provided in section IX which demonstrate a close agreement between our analytical method and simulation.

[1] This work is financially supported by Natural Sciences and Engineering Research Council of Canada (NSERC) and by Communications and Information Technology Ontario (CITO).

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 a 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 k^{th} bit position of its code-words $\mathbf{c}^i$.

$$\begin{aligned} \forall \mathbf{c}^i \in \mathcal{C}: \text{ if } c_k^i = 0 &\Longrightarrow \mathbf{c}^i \in \ C_k^0 \\ \text{if } c_k^i = 1 &\Longrightarrow \mathbf{c}^i \in \ C_k^1 \\ C_k^0 \cup C_k^1 = \mathcal{C}, \quad & C_k^0 \cap C_k^1 = \varnothing \end{aligned}$$

We use the following operators on our code book.

$$\mathbf{c}^i \oplus \mathbf{c}^j = \text{Bit wise binary addition of two code-words} \quad (1)$$

Note that the sub-code C_k^0 is closed under binary addition. The modulation scheme used here is BPSK which is defined as mapping M,

$$M : \mathbf{c} \longrightarrow \mathbf{m}(\mathbf{c}) \quad (2)$$

$$0 \longrightarrow m(0) = -1, \quad 1 \longrightarrow m(1) = 1 \quad (3)$$

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,

$$\mathbf{a}.\mathbf{b} = \sum_{i=1}^{N} a_i b_i \quad (4)$$

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

$$-\mathbf{1}.\mathbf{m}(\mathbf{c}) = N - 2\omega(\mathbf{c}) \quad (5)$$

If we modulate a code-word $\tilde{\mathbf{c}} = (\tilde{c}_1, \tilde{c}_2, ..., \tilde{c}_N)$ using BPSK modulation and send 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. 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,

$$LLR(k) = \log \frac{P(\tilde{c}_k = 1|\mathbf{x})}{P(\tilde{c}_k = 0|\mathbf{x})} \quad (6)$$

where $\tilde{c}_k$ is the value of k^{th} bit in the transmitted code-word. In the case of transmitting equally likely code-words over AWGN channel the bit LLR can be calculated as follows,

$$LLR(k) = \log \frac{\sum_{\mathbf{c}^i \epsilon C_k^1} \exp\left[\frac{\mathbf{x}.\mathbf{m}(\mathbf{c}^i)}{\sigma^2}\right]}{\sum_{\mathbf{c}^i \epsilon C_k^0} \exp\left[\frac{\mathbf{x}.\mathbf{m}(\mathbf{c}^i)}{\sigma^2}\right]} \quad (7)$$

$$= \log \frac{\sum_{\mathbf{c}^i \epsilon C_k^1} \exp\left[\frac{\mathbf{n}.\mathbf{m}(\mathbf{c}^i) + \mathbf{m}(\tilde{\mathbf{c}}).\mathbf{m}(\mathbf{c}^i)}{\sigma^2}\right]}{\sum_{\mathbf{c}^i \epsilon C_k^0} \exp\left[\frac{\mathbf{n}.\mathbf{m}(\mathbf{c}^i) + \mathbf{m}(\tilde{\mathbf{c}}).\mathbf{m}(\mathbf{c}^i)}{\sigma^2}\right]} \quad (8)$$

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 $\mathbf{n}$.

III. Some Useful Theorems

Using the above definitions and notations, we have the following theorems which are proved in [16].

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 $k = 0 \ \ or \ \ 1$ are the reflections of one another through the origin (threshold point).

We will now concentrate on the conditions for two bit positions to have the same *pdf* for their bit LLR by examining the values of the LLRs in these positions. These conditions are presented in the following theorems. First we visit the definition of automorphism group which is used in the following theorems.

Let $\mathcal{C}$ be a binary linear code of length N. We define a permutation Π which simply permutes the elements of each code-word. The set of permutations which maps the code-book $\mathcal{C}$ onto itself, form a group and called Automorphism group of code $\mathcal{C}$.

Theorem 3: Consider two bit positions of a code-word, i, j such that $1 \leq i, j \leq N$, $i \neq j$. If there exists a permutation Π within Automorphism group of code $\mathcal{C}$ which transfers bit position i to j, the $LLR(i)$ and $LLR(j)$ possess the same probability distribution.

Note that set of permutations form a group, It is clear that inverse of Π exists and transfers bit position j to i. The existence of the permutation to yield two bit positions with the same probability distribution for their LLR is our next concern.

Theorem 4: The permutation mentioned in theorem (3) exists for the class of cyclic codes.

Using the above theorems, 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 (7) reduces to,

$$LLR(k) = \log \frac{\sum_{\mathbf{c}^i \epsilon C_k^1} \exp\left[\frac{\mathbf{n}.\mathbf{m}(\mathbf{c}^i) - \mathbf{1}.\mathbf{m}(\mathbf{c}^i)}{\sigma^2}\right]}{\sum_{\mathbf{c}^i \epsilon C_k^0} \exp\left[\frac{\mathbf{n}.\mathbf{m}(\mathbf{c}^i) - \mathbf{1}.\mathbf{m}(\mathbf{c}^i)}{\sigma^2}\right]} \quad (9)$$

Using (5) we obtain,

$$LLR(k) = \log \frac{\sum_{\mathbf{c}^i \epsilon C_k^1} \exp\left[\frac{\mathbf{n}.\mathbf{m}(\mathbf{c}^i) - 2\omega(\mathbf{c}^i)}{\sigma^2}\right]}{\sum_{\mathbf{c}^i \epsilon C_k^0} \exp\left[\frac{\mathbf{n}.\mathbf{m}(\mathbf{c}^i) - 2\omega(\mathbf{c}^i)}{\sigma^2}\right]} \quad (10)$$

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 (10).

It simplifies the following derivations if we rewrite (10) as $H(\mathbf{n}) = F(\mathbf{n}) - G(\mathbf{n})$, where,

$$F(\mathbf{n}) = \log \sum_{\mathbf{c}^i \epsilon C^1} \exp\left[\frac{\mathbf{n}.\mathbf{m}(\mathbf{c}^i) - 2\omega(\mathbf{c}^i)}{\sigma^2}\right] \tag{11}$$

$$G(\mathbf{n}) = \log \sum_{\mathbf{c}^i \epsilon C^0} \exp\left[\frac{\mathbf{n}.\mathbf{m}(\mathbf{c}^i) - 2\omega(\mathbf{c}^i)}{\sigma^2}\right] \tag{12}$$

IV. Gram-Charlier Expansion of *pdf*

The best way for estimating a function using a series expansion is to choose an orthogonal basis which is suitable for that function. As the *pdf* of bit LLR is approximately Gaussian [1, 17, 18], the appropriate basis can be normal Gaussian *pdf* and its derivatives. These functions form an orthogonal basis and we will show that they are suitable basis in the sense that we can expand bit error probability as closely as desired to the real one.

Consider a random variable Y which is normalized to have zero mean and unit variance. One can expand the *pdf* of Y, $f_Y(y)$ using the following formula which is called the Gram-Charlier series expansion [19],

$$f_Y(y) = \frac{1}{\sqrt{2\pi}} e^{-\frac{y^2}{2}} \sum_{i=0}^{\infty} C_i T_i(y) \tag{13}$$

where, $T_i(y)$ which is called Tchebychev-Hermite polynomial, defined as,

$$T_i(y) = \sum_{j=0}^{\lfloor i/2 \rfloor} (-1)^{i+j} \frac{i!}{2^j (i-2j)! j!} y^{i-2j} \tag{14}$$

$$C_i = \sum_{j=0}^{\lfloor i/2 \rfloor} (-1)^{i+j} \frac{i!}{2^j (i-2j)! j!} \mu_{i-2j} \tag{15}$$

where,

$$\mu_j = \int_{-\infty}^{+\infty} y^j f_Y(y) dy, \quad \mu_1 = 0, \quad \mu_2 = 1 \tag{16}$$

It is interesting to note that $\frac{1}{\sqrt{2\pi}} e^{-\frac{y^2}{2}} T_i(y)$, is the i^{th} order derivative of the normalized Gaussian *pdf*. These functions have the following property,

$$e^{-\frac{y^2}{2}} T_i(y) = -\frac{d}{dy}\left[e^{-\frac{y^2}{2}} T_{i-1}(y)\right], \quad i \geq 1 \tag{17}$$

The only unknown components in (15) are the moments, μ_j. We propose an analytical method using Taylor series expansion to compute the statistical moments of the bit LLR.

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 $\mathbf{n}$,

$$\begin{aligned} H(\mathbf{n}) &= H(\mathbf{0}) + \mathbf{n}.\nabla H(\mathbf{0}) + \frac{1}{2}(\mathbf{n}.\nabla)^2 H(\mathbf{0}) + ... \qquad (18) \\ &= H(\mathbf{0}) + \sum_{p=1}^{N} \frac{\partial H(\mathbf{0})}{\partial n_p} n_p + \frac{1}{2} \sum_{p=1}^{N} \sum_{q=1}^{N} \frac{\partial^2 H(\mathbf{0})}{\partial n_p \partial n_q} n_p n_q + ... \end{aligned}$$

We continue with calculation of different terms in the above equation. Noting to the similarity of (11) and (12) we only compute the derivatives of $F(\mathbf{n})$ hereafter. The same approach can be used for $G(\mathbf{n})$ and using, $H(\mathbf{n}) = F(\mathbf{n}) - G(\mathbf{n})$ the derivatives of $H(\mathbf{n})$ can be calculated. To simplify the expressions, the following functions are defined,

$$A(\mathbf{n}) = \sum_{\mathbf{c}^i \epsilon C^1} \exp\left[\frac{\mathbf{n}.\mathbf{m}(\mathbf{c}^i) - 2\omega(\mathbf{c}^i)}{\sigma^2}\right] \tag{19}$$

$$A_{\mathbf{Q}_j}(\mathbf{n}) = \sigma^{-2j} \sum_{\mathbf{c}^i \epsilon C^1} M^i_{\mathbf{Q}_j} \exp\left[\frac{\mathbf{n}.\mathbf{m}(\mathbf{c}^i) - 2\omega(\mathbf{c}^i)}{\sigma^2}\right], \quad j \geq 1 \tag{20}$$

where $\mathbf{Q}_j = (q_1, q_2, ..., q_j)$ is a vector of j bit positions different from k and,

$$M^i_{\mathbf{Q}_j} = \prod_{l=1}^{j} m(c^i_{q_l}) \tag{21}$$

where $m(c^i_{q_l}) = \pm 1$ is the modulated value for the q_l^{th}, $q_l \in \mathbf{Q}_j$, bit of code-word $\mathbf{c}^i$. It is clear that $M^i_{\mathbf{Q}_j} = \pm 1$ as well. Mixing (19) and (20) we define,

$$R_{\mathbf{Q}_j}(\mathbf{n}) = A^{-1}(\mathbf{n}) A_{\mathbf{Q}_j}(\mathbf{n}), \quad j \geq 1 \tag{22}$$

The above functions reduce to special weight distribution functions when $\mathbf{n} = \mathbf{0}$,

$$A(\mathbf{0}) = \mathcal{A}(Z) = \sum_{w=0}^{N} A_w Z^w \tag{23}$$

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

$$A_{\mathbf{Q}_j}(\mathbf{0}) = \mathcal{A}_{\mathbf{Q}_j}(Z) = \sigma^{-2j} \sum_{w=0}^{N} A_w(\mathbf{Q}_j) Z^w, \quad j \geq 1 \tag{24}$$

$$A_w(\mathbf{Q}_j) = A_w^+(\mathbf{Q}_j) - A_w^-(\mathbf{Q}_j) \tag{25}$$

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

$$R_{\mathbf{Q}_j}(\mathbf{0}) = \mathcal{R}_{\mathbf{Q}_j}(Z) = \mathcal{A}^{-1}(Z) \mathcal{A}_{\mathbf{Q}_j}(Z), \quad j \geq 1 \tag{26}$$

Using (23), we can simplify $F(\mathbf{0})$ as follows,

$$F(\mathbf{0}) = \log \mathcal{A}(Z) \tag{27}$$

Using (19) it is easy to show that,

$$\frac{\partial^j A(\mathbf{n})}{\partial n_{q_1}\partial n_{q_2}...\partial n_{q_j}} = A_{\mathbf{Q}_j}(\mathbf{n}) \quad (28)$$

where $\mathbf{Q}_j = (q_1, q_2, ..., q_j)$.

In the special case of first order derivative ($j = 1$) we have,

$$\frac{\partial F(\mathbf{n})}{\partial n_{q_1}} = A^{-1}(\mathbf{n})A_{\mathbf{Q}_1}(\mathbf{n}) = R_{\mathbf{Q}_1}(\mathbf{n}) \quad (29)$$

$$\frac{\partial F(\mathbf{0})}{\partial n_{q_1}} = \mathcal{R}_{\mathbf{Q}_1}(Z) \quad (30)$$

where $\mathbf{Q}_1$ is a one dimensional vector of bit position q_1. Derivatives of higher orders can be calculated using the following property which uses the fact that $m^2(c_p^i) = 1$.

Property 1 : For any bit position $i \neq k$ we have,

$$\frac{\partial R_{\mathbf{Q}_j}(\mathbf{n})}{\partial n_{q_i}} = \begin{cases} \sigma^{-4}R_{\mathbf{Q}_{j-1}}(\mathbf{n}) - R_{\mathbf{Q}_j}(\mathbf{n})R_{\mathbf{Q}_1}(\mathbf{n}), & 1 \le i \le j \\ R_{\mathbf{Q}_{j+1}}(\mathbf{n}) - R_{\mathbf{Q}_j}(\mathbf{n})R_{\mathbf{Q}_1}(\mathbf{n}), & j < i \end{cases} \quad (31)$$

where,

$$\begin{aligned} \mathbf{Q}_{j+1} &= (q_1, q_2, ..., q_j, q_i) \quad (32)\\ \mathbf{Q}_{j-1} &= (q_1, q_2, ..., q_{i-1}, q_{i+1}, ..., q_j)\\ \mathbf{Q}_1 &= (q_i) \end{aligned}$$

This property results in the following theorem.

Theorem 5: The j^{th} order derivative of $F(\mathbf{n})$ can be stated in terms of $R_{\mathbf{Q}_l}(\mathbf{n})$, $l = 1, 2, ..., j$.

Proof: From (29) first order derivative of $F(\mathbf{n})$ is a function of $R_{\mathbf{Q}_1}(\mathbf{n})$. Using the *Property* 1 it is easy to see that higher order derivatives are functions of $R_{\mathbf{Q}_l}(\mathbf{n})$ for different l's. ■

The above theorems and results enable us to compute all the derivatives required in the Taylor series expansion of $H(\mathbf{n})$.

VI. Computing moments

The definition of m^{th} order moment is,

$$\mu_m = E[H^m(\mathbf{n})] \quad (33)$$

where $E[.]$ stands for expectation. To compute (33), we use the Taylor series expansion of $H^m(\mathbf{n})$ and average this expansion with respect to different components of $\mathbf{n}$. Calculating the coefficients of the Taylor expansion of $H^m(\mathbf{n})$ involves computing the following terms,

$$\frac{\partial^r H(\mathbf{0})}{\partial n_{p_1}^{l_1}\partial n_{p_2}^{l_2}...\partial n_{p_j}^{l_j}} E[n_{p_1}^{l_1}]E[n_{p_2}^{l_2}]...E[n_{p_j}^{l_j}] \quad (34)$$

where r, l_i's, $i = 1, 2, ..., j$ are even and satisfy

$$l_1 + l_2 + ... + l_j = r \quad (35)$$

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

$$E[n^l] = \begin{cases} \frac{(l)!\sigma^l}{2^{l/2}(l/2)!} & , \quad l \text{ even} \\ 0 & , \quad l \text{ odd} \end{cases} \quad (36)$$

Each solution to 35 corresponds to one partial derivative which can be computed precisely. Now we can compute moments analytically and use them in the Gram-Charlier expansion to estimate the *pdf* of bit LLR. The bit error performance follows by a simple integration of the resulting *pdf*. We present a closed form formula for computing this integral in the next section.

VII. Computing Probability of Error

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

$$P_\epsilon(a) = \int_a^\infty f_Y(y)dy = \frac{1}{2} - \int_0^a f_Y(y)dy \quad (37)$$

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,

$$\begin{aligned} P_\epsilon(a) &= \frac{1}{2} - \int_0^a \frac{1}{\sqrt{2\pi}}e^{-\frac{y^2}{2}}\sum_{i=0}^{\infty} C_iT_i(y)dy \quad (38)\\ &= \frac{1}{2} - \int_0^a \frac{1}{\sqrt{2\pi}}e^{-\frac{y^2}{2}}dy - \int_0^a \frac{1}{\sqrt{2\pi}}e^{-\frac{y^2}{2}}\sum_{i=1}^{\infty} C_iT_i(y)dy\\ &= Q(a) - \int_0^a \frac{1}{\sqrt{2\pi}}e^{-\frac{y^2}{2}}\sum_{i=1}^{\infty} C_iT_i(y)dy \end{aligned}$$

Changing the order of integral and summation and using (17) we can write,

$$P_\epsilon(a) = Q(a) + \frac{1}{\sqrt{2\pi}}\left[e^{-\frac{a^2}{2}}\sum_{i=1}^{\infty} C_iT_{i-1}(a) - \sum_{i=1}^{\infty} C_iT_{i-1}(0)\right] \quad (39)$$

To simplify the expression we define a new function,

$$\lambda(x) = \frac{1}{\sqrt{2\pi}}e^{-\frac{x^2}{2}}\sum_{i=1}^{\infty} C_iT_{i-1}(x) \quad (40)$$

Now we can write,

$$P_\epsilon(a) = Q(a) + \lambda(a) - \lambda(0) \quad (41)$$

VIII. Convergence properties

Let us define an error function for the Gram-Charlier series expansion,

$$\epsilon_l(y) = f_Y(y) - \frac{1}{\sqrt{2\pi}}e^{-\frac{y^2}{2}}\sum_{i=0}^{l} C_iT_i(y) \quad (42)$$

The *pdf* of the bit LLR can be expanded arbitrarily closely using the given set of orthogonal basis [19]. hence,

$$\lim_{l \longrightarrow \infty} \int_{-\infty}^{+\infty} \epsilon_l^2(y)dy \longrightarrow 0 \quad (43)$$

Using Cauchy-Schwartz inequality,

$$\left|\int_{-\infty}^{+\infty} f(y)g(y)dy\right|^2 < \int_{-\infty}^{+\infty} |f(y)|^2 dy \int_{-\infty}^{+\infty} |g(y)|^2 dy \tag{44}$$

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

$$g(y) = \begin{cases} 1, & 0 < y < a \\ 0, & O.W. \end{cases} \tag{45}$$

we have,

$$\left|\int_0^a \epsilon_l(y)dy\right|^2 < a \int_{-\infty}^{+\infty} \epsilon_l^2(y)dy \tag{46}$$

where a is a positive constant.

Applying (43) to (46) results in,

$$\lim_{l \longrightarrow \infty} \int_0^a \epsilon_l(y)dy \longrightarrow 0 \tag{47}$$

This means that the estimate of the bit error probability will asymptotically converge to the true bit error performance.

IX. Example for (15,11) Cyclic code

As an example we used a (15,11) Cyclic code and evaluated its performance using the proposed method. The order of the Gram-Charlier expansion is 14 and the order of the Taylor-expansion is 10. These values turned out to be sufficient for a close approximation of the true BER curve. The comparison between the analytically calculated BER and the one obtained from simulation is shown in Figure 1.

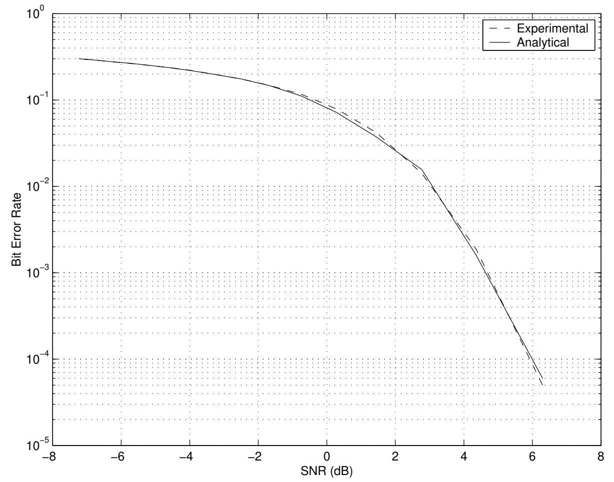

Fig. 1. Comparison between analytical and experimental BER

References

[1] 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.

[2] 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.

[3] Chi-Chao Chao, R.J. McEliece, L. Swanson, E.R. Rodemich, "Performance of Binary Block Codes at Low Signal-to-Noise Ratios," *IEEE Transactions on Information Theory*, Vol. 38, No.6, pp. 1677-1687, November 1992.

[4] 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.

[5] 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.

[6] P. Swaszek, "A lower bound on the error probability of signals in white Gaussian noise," *IEEE Transactions on Information Theory*, Vol.41, No.3, pp. 837-841, May 1995.

[7] G. E. Seguin, "A lower bound on the error probability of signals in white Gaussian noise," *IEEE Transactions on Information Theory*, Vol.44, No.7, pp. 3168-3175, November 1998.

[8] B. Hughes, "On the error probability of signals in additive white Gaussian noise," *IEEE Transactions on Information Theory*, vol.37, pp. 151-155, January 1991.

[9] G. Poltyrev, "Bounds on the decoding error probability of binary linear codes via their spectra," *IEEE Transactions on Information Theory*, Vol.40, No.4, pp. 1284-1292, July 1994.

[10] I. Sason, S. Shamai, "Improved Upper Bounds on the ML Decoding Error Probability of Parallel and Serial Concatenated Turbo Codes via Their Ensemble Distance Spectrum," *IEEE Transactions on Information Theory*, Vol.46, No.1, pp. 24-47, January 2000.

[11] T.M. Duman, M. Salehi, "New performance bounds for Turbo codes," *IEEE Transactions on Communications*, Vol.46, No.6, pp. 717-723, June 1998.

[12] D. Divsalar, "A simple tight bound on error probability of block codes with application to Turbo Codes," *TMO Progress Report 42-139*, NASA, JPL, Pasadena, CA, USA, 1999.

[13] H. Kuai,F. Alajaji, G. Takahara, "Tight error bounds for nonuniform signaling over AWGN channels," *IEEE Transactions on Information Theory*, Vol.46, No.7, pp. 2712 -2718, November 2000.

[14] Ali Abedi and Amir K. Khandani, "Some Properties of Bit Decoding Algorithms Over A Generalized Channel Model," Technical Report, UW-E&CE 2002-05 (available from www.cst.uwaterloo.ca)

[15] J. Hagenauer, "Log-likelihood ratios, mutual information and EXIT charts," *Proceedings of the 12th Joint Conference on Communications and Coding*, Saas Fe/Switzerland, 02-09.03.2002, Lehrstuhl fur Nachrichtentechnik, Technische Universitat Munchen

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

[17] H.El Gamal and A.R. Hammons Jr., "Analyzing the Turbo Decoder Using the Gaussian Approximation", *Proceedings of ISIT 2000*, Sorrento, Italy, pp. 319, June 2000.

[18] D. Divsalar, S. Dolinar, and 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.

[19] J.F. Kenney, E. S. Keeping, Mathematics of Statistics, Pt. 2, 2nd ed. Princeton, NJ: Van Nostrand, pp. 107-108, 1951.