An Analytical Method for Approximate Performance Evaluation of Binary Linear Block Codes

Ali Abedi, *Student Member, IEEE,* and Amir K. Khandani, *Member, IEEE*

***Abstract*—An analytical method for approximate performance evaluation of binary linear block codes using an additive white Gaussian noise channel model with binary phase-shift keying modulation is presented. We focus on the probability density function 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. It is proved that the approximation can be as accurate as desired, if we use enough terms in the Gram–Charlier series expansion. Numerical results are provided for some examples, which demonstrate close agreement with simulation results.**

***Index Terms*—Additive white Gaussian noise (AWGN) channel, binary phase-shift keying (BPSK), bit decoding, bit-error probability (BEP), block codes, log-likelihood ratio (LLR), weight distribution.**

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 codeword with the ML value. The ML algorithms are known to minimize the frame-error rate (FER) under the mild condition that the codewords 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 codewords according to the value of their component in a given bit position of interest. Reference [2] provides an efficient method (known as BCJR) to compute the bit probabilities of a given code using its trellis diagram. There are some special methods for bit decoding based on the coset decomposition principle [3], sectionalized trellis diagrams [4], and using the dual code [5], [6].

ML 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] and [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 symmetric channels (BSCs) [10], there has been no method for performance evaluation of bit decoding in general. Some asymptotic expressions are derived in [11] for bit-error probability (BEP) of binary linear block codes in the additive white Gaussian noise (AWGN) channel with bit decoding. The BEPs of convolutional codes over BSCs 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 paper, we employ Gram–Charlier series expansion to find the probability density function (pdf) of the bit log-likelihood ratio (LLR). This method is used in some other communications applications, including calculation of pdf of the sum of log-normal variates [13], evaluation of the error probability in pulse amplitude modulation (PAM) digital data transmission systems with correlated symbols in the presence of intersymbol interference (ISI) and additive noise [14], computing nearly Gaussian distributions [15], and computation of the error probability of equal-gain combiners 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 the maximum entropy criterion [20] to find the corresponding 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 the pdf of bit LLRs using the 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,

Paper approved by V. K. Bhargava, the Editor for Coding and Communication Theory of the IEEE Communications Society. Manuscript received March 28, 2002; revised August 20, 2003. This work was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC) and in part by Communications and Information Technology Ontario (CITO). This paper was presented in part at the IEEE International Symposium on Information Theory (ISIT), Lausanne, Switzerland, July 2002.

The authors are with the Coding and Signal Transmission Laboratory, Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: ali@cst.uwaterloo.ca; khandani@cst.uwaterloo.ca).

Digital Object Identifier 10.1109/TCOMM.2003.822704

where it is shown that we can compute the coefficients of Taylor expansion of the bit LLR recursively. We also present a closed-form expression for computing the BEP in Section V. In Section VI, the convergence issue of this approximation is discussed. Numerical results are provided in Section VII, which demonstrate a close agreement between our analytical method and simulation. We conclude in Section VIII.

II. Modeling

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

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

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

We define the following operators on the code book:

$$\mathbf{c}^i \oplus \mathbf{c}^j = \text{Bitwise binary addition of two codewords.} \tag{3}$$

Note that the subcode 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 codeword as mentioned above results in a vector of constant square norm

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

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

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

Modulating a codeword $\tilde{\mathbf{c}} = (\tilde{c}_1, \tilde{c}_2, \ldots, \tilde{c}_N)$ using BPSK and sending it through an AWGN channel, we will receive $\mathbf{x} = \mathbf{m}(\tilde{\mathbf{c}}) + \mathbf{n}$, where $\mathbf{n} = (n_1, n_2, \ldots, n_N)$ is an independent, identically distributed (i.i.d.) 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 LLR. The LLR of the kth bit position is defined by the following equation:

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

where $\tilde{c}_k$ is the value of the kth bit in the transmitted codeword and log stands for the natural logarithm. Assuming

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

and using (9), it follows:

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

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

Using (7), it follows:

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

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

Given a value of the bit LLR, a decision on the value of bit k is made by comparing $\mathrm{LLR}(k)$ with a threshold of zero. We are interested in studying the probabilistic behavior of $\mathrm{LLR}(k)$ 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 $\mathrm{LLR}(k)$ is not affected by the choice of transmitted codeword $\tilde{\mathbf{c}}$, as long as the value of the kth bit remains unchanged.

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

Theorem 3: The probability distribution of $\mathrm{LLR}(k)$ is not affected by the choice of 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 codeword, denoted as $\tilde{\mathbf{c}} = (0, 0, \ldots, 0)$, is transmitted in all our following discussions. This means $\mathbf{m}(\tilde{\mathbf{c}}) = -\mathbf{1} = (-1, -1, \ldots, -1)$ is the transmitted modulated codeword.

In this case, (15) reduces to

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

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

Using (8), we obtain

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

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, respectively. We use the notation $H(\mathbf{n})$ to refer to the LLR expression given in (17), i.e.,

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

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 a bit LLR is approximately Gaussian [7], [22], [23], the appropriate basis can be a normal Gaussian pdf and its derivatives which form an orthogonal basis. There are a variety of equivalent formulations for this expansion [15], [24]–[26]. 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 [15] of order i, defined as

$$T_i(y)=(-1)^i e^{\frac{y^2}{2}}\frac{d^i}{dy^i}\left[e^{-\frac{y^2}{2}}\right] \tag{20}$$

and has the following closed form:

$$T_i(y)=\sum_{j=0}^{\lfloor\frac{i}{2}\rfloor}\frac{(-1)^j i!}{2^j(i-2j)!j!}y^{i-2j} \tag{21}$$

$$\alpha_i=\sum_{j=0}^{\lfloor\frac{i}{2}\rfloor}\frac{(-1)^j}{2^j(i-2j)!j!}\mu_{i-2j} \tag{22}$$

where

$$\mu_j=\int_{-\infty}^{+\infty}y^j f_Y(y)dy. \tag{23}$$

This is a commonly used method for approximating an unknown pdf. The only unknown components in (22) 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 Using Taylor Expansion of LLR

Applying the definition of the mth order $(m>2)$ moment to bit LLR results in

$$\mu_m=E\left[\left(\frac{H(\mathbf{n})-E\left[H(\mathbf{n})\right]}{\sqrt{\mathrm{var}\left[H(\mathbf{n})\right]}}\right)^m\right] \tag{24}$$

$$=\frac{1}{\mathrm{var}^{\frac{m}{2}}\left[H(\mathbf{n})\right]}\sum_{i=0}^{m}(-1)^i\binom{m}{i}\times E\left[H^{m-i}(\mathbf{n})\right]E^i\left[H(\mathbf{n})\right] \tag{25}$$

where $E[.]$ stands for expectation and $\mathrm{var}[.]$ denotes variance. Note that to compute (25), one needs $E[H^j(\mathbf{n})]$, $j=0,\ldots,m$.

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

$$H^j(\mathbf{n})=\left(\sum_{i=0}^{\infty}\frac{1}{i!}(\mathbf{n}.\nabla)^i H(\mathbf{0})\right)^j \tag{26}$$

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

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

$$H(\mathbf{n})=H(\mathbf{0})+\mathbf{n}.\nabla H(\mathbf{0})+\frac{1}{2!}(\mathbf{n}.\nabla)^2H(\mathbf{0})+\ldots \tag{27}$$

$$=H(\mathbf{0})+\sum_{q_1=1}^{N}\left.\frac{\partial H(\mathbf{n})}{\partial n_{q_1}}\right|_{\mathbf{n}=\mathbf{0}}n_{q_1}+\frac{1}{2}\sum_{q_1=1}^{N}\sum_{q_2=1}^{N}\left.\frac{\partial^2 H(\mathbf{n})}{\partial n_{q_1}\partial n_{q_2}}\right|_{\mathbf{n}=\mathbf{0}}n_{q_1}n_{q_2}+\ldots. \tag{28}$$

We can continue with calculation of different terms in the above equation. For simplicity, we define (18) as $H(\mathbf{n})=\log A^1(\mathbf{n})-\log A^0(\mathbf{n})$, where

$$A^1(\mathbf{n})=\sum_{\mathbf{c}\epsilon C^1}\exp\left[\frac{\mathbf{n}.\mathbf{m}(\mathbf{c})-2\omega(\mathbf{c})}{\sigma^2}\right] \tag{29}$$

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

$$\log A(\mathbf{n})=\log A(\mathbf{0})+\sum_{q_1=1}^{N}\left.\frac{\partial\log A(\mathbf{n})}{\partial n_{q_1}}\right|_{\mathbf{n}=\mathbf{0}}n_{q_1}+\frac{1}{2}\sum_{q_1=1}^{N}\sum_{q_2=1}^{N}\left.\frac{\partial^2\log A(\mathbf{0})}{\partial n_{q_1}\partial n_{q_2}}\right|_{\mathbf{n}=\mathbf{0}}n_{q_1}n_{q_2}+\ldots. \tag{30}$$

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

$$F_{\{q_1,\ldots,q_j\}}(\mathbf{n}) = \frac{\partial^j A(\mathbf{n})}{\partial n_{q_1}\partial n_{q_2}\ldots\partial n_{q_j}} = \sigma^{-2j}\sum_{\mathbf{c}\epsilon C^1} M_{\{q_1,\ldots,q_j\}}\exp\left[\frac{\mathbf{n}.\mathbf{m}(\mathbf{c})-2\omega(\mathbf{c})}{\sigma^2}\right], \quad j\geq 1 \tag{31}$$

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

$$M_{\{q_1,\ldots,q_j\}} = \prod_{l=1}^{j} m(c_{q_l}), \quad j\geq 1 \tag{32}$$

where $m(c_{q_l}) = \pm 1$ is the modulated value for the q_lth, $q_l \in \{q_1,\ldots,q_j\}$, bit of codeword $\mathbf{c}$. It is clear that $M_{\{q_1,\ldots,q_j\}} = \pm 1$, as well.

To simplify (30), it easily follows that:

$$\frac{\partial \log A(\mathbf{n})}{\partial n_{q_1}} = A^{-1}(\mathbf{n})F_{\{q_1\}}(\mathbf{n}) = R_{\{q_1\}}(\mathbf{n}) \tag{33}$$

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

$$R_{\{q_1,\ldots,q_j\}}(\mathbf{n}) = A^{-1}(\mathbf{n})F_{\{q_1,\ldots,q_j\}}(\mathbf{n}), \quad j\geq 1 \tag{34}$$

where $A(\mathbf{n})$ and $F_{\{q_1,\ldots,q_j\}}(\mathbf{n})$ are given in (29) and (31), respectively.

The functions $A(\mathbf{n})$, $F_{\{q_1,\ldots,q_j\}}(\mathbf{n})$, and $R_{\{q_1,\ldots,q_j\}}(\mathbf{n})$, defined in (29), (31), and (34), respectively, reduce to special weight distribution functions when $\mathbf{n}=\mathbf{0}$

$$A(\mathbf{0}) = \mathcal{A}(Z) = \sum_{w=0}^{N} a(w)Z^w \tag{35}$$

where $Z=\exp(-2/\sigma^2)$, and $a(w)$ is the number of codewords with Hamming weight w in C^1

$$F_{\{q_1,\ldots,q_j\}}(\mathbf{0}) = \mathcal{F}_{\{q_1,\ldots,q_j\}}(Z) = \sigma^{-2j}\sum_{w=0}^{N}\left[f^{+}_{\{q_1,\ldots,q_j\}}(w) - f^{-}_{\{q_1,\ldots,q_j\}}(w)\right]Z^w, \quad j\geq 1 \tag{36}$$

where $f^{\pm}_{\{q_1,\ldots,q_j\}}(w)$ is the number of codewords $\mathbf{c}\in C^1$ with Hamming weight w, and $M_{\{q_1,\ldots,q_j\}} = \pm 1$

$$R_{\{q_1,\ldots,q_j\}}(\mathbf{0}) = \mathcal{R}_{\{q_1,\ldots,q_j\}}(Z) = \mathcal{A}^{-1}(Z)\mathcal{F}_{\{q_1,\ldots,q_j\}}(Z), \quad j\geq 1. \tag{37}$$

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

Using (33) and (37), we have

$$\left.\frac{\partial \log A(\mathbf{n})}{\partial n_{q_1}}\right|_{\mathbf{n}=\mathbf{0}} = \mathcal{R}_{\{q_1\}}(Z). \tag{38}$$

Replacing (33) and (38) in (30), we have

$$\log A(\mathbf{n}) = \log \mathcal{A}(Z) + \sum_{q_1=1}^{N}\mathcal{R}_{\{q_1\}}(Z)n_{q_1} + \frac{1}{2}\sum_{q_1=1}^{N}\sum_{q_2=1}^{N}\left.\frac{\partial R_{\{q_1\}}(\mathbf{n})}{\partial n_{q_2}}\right|_{\mathbf{n}=\mathbf{0}} n_{q_1}n_{q_2} + \ldots. \tag{39}$$

To compute (39), 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

$$\frac{\partial R_{\{q_1,\ldots,q_j\}}(\mathbf{n})}{\partial n_{q_i}} = \begin{cases} \sigma^{-4}R_{\{q_1,\ldots,q_{i-1},q_{i+1},\ldots,q_j\}}(\mathbf{n}) \\ \quad -R_{\{q_1,\ldots,q_j\}}(\mathbf{n})R_{\{q_i\}}(\mathbf{n}), & \text{if } q_i\in\{q_1,\ldots,q_j\} \\ R_{\{q_1,\ldots,q_j,q_i\}}(\mathbf{n}) \\ \quad -R_{\{q_1,\ldots,q_j\}}(\mathbf{n})R_{\{q_i\}}(\mathbf{n}), & \text{otherwise.} \end{cases} \tag{40}$$

Proof: For proof, refer to Appendix A. ∎

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

Theorem 5: We have

$$\frac{\partial^2 R_{\{q_1,\ldots,q_j\}}(\mathbf{n})}{\partial n_{q_i}^2} = -2R_{\{q_i\}}(\mathbf{n})\frac{\partial R_{\{q_1,\ldots,q_j\}}(\mathbf{n})}{\partial n_{q_i}}. \tag{41}$$

Proof: For proof, refer to Appendix A. ∎

Referring to (40), one can easily see that the coefficients of the expansion (39) are polynomials of $R_{\{q_1,\ldots,q_j\}}(\mathbf{0})$ for different values of j. It is noteworthy that these coefficients are polynomials of special weight distribution functions defined in (37). The above theorems and results enable us to compute all the derivatives required in the Taylor series expansion of $H(\mathbf{n}) = \log A^1(\mathbf{n}) - \log A^0(\mathbf{n})$.

V. Computing Probability of Error

The BEP 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

$$P(e|\tilde{c}_k = 0) = P(e|\tilde{c}_k = 1) \tag{42}$$

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

$$P(e) = P(e|\tilde{c}_k=0)P(\tilde{c}_k=0) + P(e|\tilde{c}_k=1)P(\tilde{c}_k=1) = P(e|\tilde{c}_k=0). \tag{43}$$

Hence, computation of the BEP involves calculating an integral of the following form:

$$P(e) = \int_a^{\infty} f_Y(y)dy \tag{44}$$

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

$$P(e) \simeq \int_a^{\infty}\frac{1}{\sqrt{2\pi}}e^{-\frac{y^2}{2}}\sum_{i=0}^{\infty}\alpha_i T_i(y)dy. \tag{45}$$

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

$$P(e) \simeq \int_a^\infty \frac{1}{\sqrt{2\pi}} e^{-\frac{y^2}{2}} dy + \int_a^\infty \frac{1}{\sqrt{2\pi}} e^{-\frac{y^2}{2}} \sum_{i=1}^\infty \alpha_i T_i(y) dy \quad (46)$$

$$= Q(a) + \int_a^\infty \frac{1}{\sqrt{2\pi}} e^{-\frac{y^2}{2}} \sum_{i=1}^\infty \alpha_i T_i(y) dy. \quad (47)$$

Changing the order of integration and summation and using the following property:[2]

$$e^{-\frac{y^2}{2}} T_i(y) = -\frac{d}{dy}\left[e^{-\frac{y^2}{2}} T_{i-1}(y)\right], \quad i \geq 1 \quad (48)$$

we can write

$$P(e) \simeq Q(a) - \frac{1}{\sqrt{2\pi}} \sum_{i=1}^\infty \alpha_i \int_a^\infty d\left[e^{-\frac{y^2}{2}} T_{i-1}(y)\right] \quad (49)$$

$$= Q(a) - \frac{1}{\sqrt{2\pi}} \sum_{i=1}^\infty \alpha_i \left[e^{-\frac{y^2}{2}} T_{i-1}(y)\right]_a^\infty \quad (50)$$

$$= Q(a) + \frac{1}{\sqrt{2\pi}} e^{-\frac{a^2}{2}} \sum_{i=1}^\infty \alpha_i T_{i-1}(a). \quad (51)$$

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

VI. Convergence Properties

Convergence properties of the Gram–Charlier expansion is investigated in [24], [28], and [29]. It is proved in [30] that the expansion is convergent if the expanded function satisfies the following condition:

$$\int_{-\infty}^{+\infty} f_Y(y) e^{\frac{y^2}{4}} dy < \infty. \quad (52)$$

Reference [13] mentions that this expansion has good asymptotic behavior as defined in [31]. In other words, a few terms will give a close approximation.

General properties of Hermite polynomials are discussed in [32], where it is shown that this class of polynomials form an orthogonal basis which span the interval $(-\infty, +\infty)$. Therefore, the pdf of the bit LLR can be expanded arbitrarily closely, in a mean-square sense, using the given set of orthogonal basis, i.e.,

$$\lim_{l \longrightarrow \infty} \int_{-\infty}^{+\infty} \epsilon_l^2(y) dy \longrightarrow 0 \quad (53)$$

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

$$\epsilon_l(y) = f_Y(y) - \frac{1}{\sqrt{2\pi}} e^{-\frac{y^2}{2}} \sum_{i=0}^{l} \alpha_i T_i(y). \quad (54)$$

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 $(f_Y(y^+) + f_Y(y^-))/2$ [33]. In the following, we show that the error in the computation of BEP converges to zero.

[2]Proof is in Appendix B.

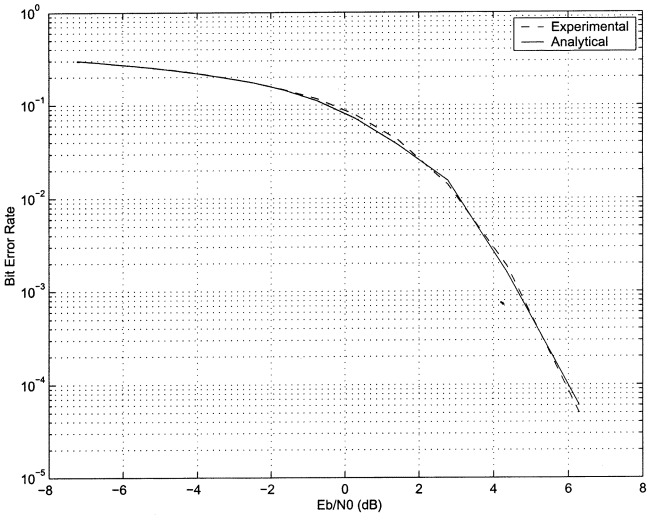


Fig. 1. Comparison between analytical and experimental BER for (15,11,3) Cyclic code.

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 the Cauchy–Schwartz inequality [34]

$$\left|\int_{-\infty}^{+\infty} f(y)g(y)dy\right|^2 < \int_{-\infty}^{+\infty} |f(y)|^2 dy \int_{-\infty}^{+\infty} |g(y)|^2 dy \quad (55)$$

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

$$g(y) = \begin{cases} 1, & a < y < b \\ 0, & \text{otherwise} \end{cases} \quad (56)$$

we have

$$\left|\int_a^b \epsilon_l(y) dy\right|^2 < (b-a) \int_{-\infty}^{+\infty} \epsilon_l^2(y) dy. \quad (57)$$

Applying (53)to (57) results in

$$\lim_{l \longrightarrow \infty} \int_a^b \epsilon_l(y) dy \longrightarrow 0. \quad (58)$$

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

VII. Numerical Results

In this section, some examples are 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 bit-error rate (BER) and the one obtained from simulation is shown in Fig. 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.

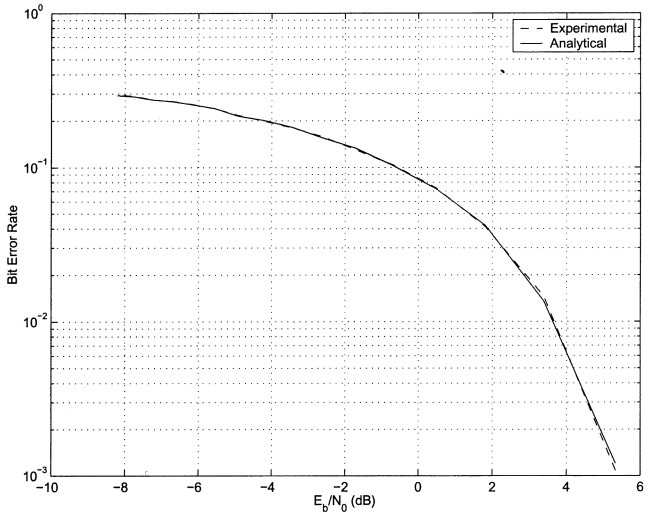


Fig. 2. Comparison between analytical and experimental BER for (12,11,2) single parity-check code.

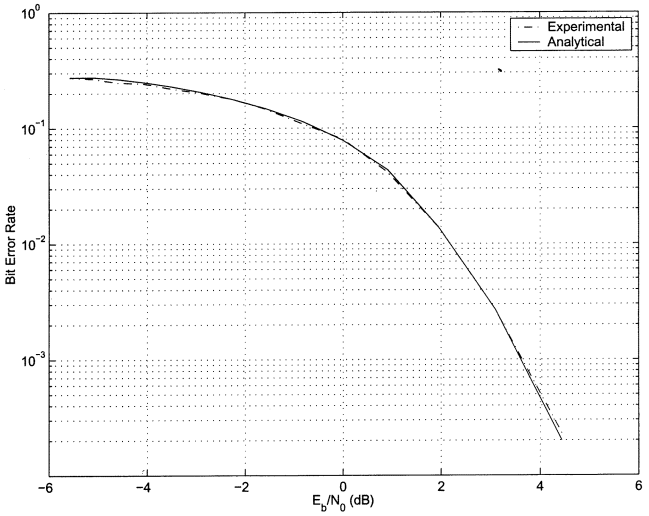


Fig. 3. Comparison between analytical and experimental BER for binary extended (24,12,8) Golay code.

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 Fig. 2.

The last example is the binary extended (24,12,8) Golay code. Its performance is shown in Fig. 3. The BER is calculated using the Gram–Charlier series with 14 terms.

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

VIII. Concluding Remarks

A method is presented for calculating BEP of binary linear block codes over an AWGN channel, using special weight enumerating functions of the code. A summary of the proposed method is presented here. Starting with calculation of special weight distribution functions defined in (37), proceed with a Taylor series of the LLR, as indicated in (27). Averaging this expansion will give us moments of the pdf of the bit LLR, which can be used to compute the coefficients of the Gram–Charlier series using (22). A closed-form expression (51) can be used to find the BEP. All these steps can be seen in Fig. 4. A possibility for future work is the extension of this method for performance evaluation of turbo codes. Although calculation of the required weight distribution functions for turbo codes is very complex, it can be approximated using the concept of uniform interleaving. Some existing approaches are bounds on the performance of turbo codes [35]–[37] under the assumption of ML decoding.

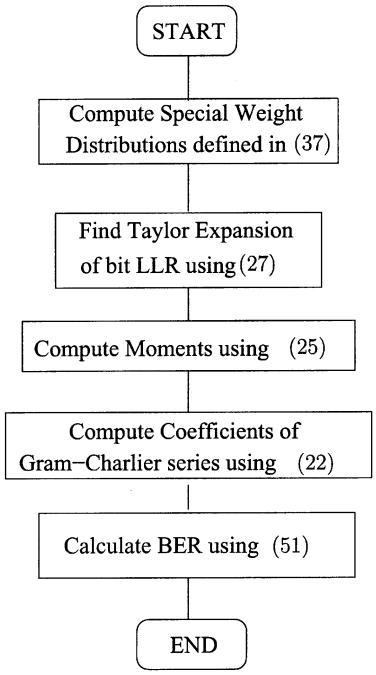


Fig. 4. Flow chart of the analytical method for performance evaluation of binary linear block codes.

Appendix
Proofs of Theorems

Theorem 4:

Proof: Using (34), one can write

$$\frac{\partial R_{\{q_1,\ldots,q_j\}}(\mathbf{n})}{\partial n_{q_i}} = \frac{\partial}{\partial n_{q_i}}\left[A^{-1}(\mathbf{n})F_{\{q_1,\ldots,q_j\}}(\mathbf{n})\right] \tag{59}$$

$$= A^{-1}(\mathbf{n})\frac{\partial F_{\{q_1,\ldots,q_j\}}(\mathbf{n})}{\partial n_{q_i}} + \frac{\partial A^{-1}(\mathbf{n})}{\partial n_{q_i}}F_{\{q_1,\ldots,q_j\}}(\mathbf{n}) \tag{60}$$

$$= A^{-1}(\mathbf{n})\frac{\partial F_{\{q_1,\ldots,q_j\}}(\mathbf{n})}{\partial n_{q_i}} - A^{-2}(\mathbf{n})\frac{\partial A(\mathbf{n})}{\partial n_{q_i}}F_{\{q_1,\ldots,q_j\}}(\mathbf{n}) \tag{61}$$

$$= A^{-1}(\mathbf{n})\frac{\partial F_{\{q_1,\ldots,q_j\}}(\mathbf{n})}{\partial n_{q_i}} - A^{-2}(\mathbf{n})F_{\{q_i\}}(\mathbf{n})F_{\{q_1,\ldots,q_j\}}(\mathbf{n}) \tag{62}$$

$$= A^{-1}(\mathbf{n})\frac{\partial F_{\{q_1,\ldots,q_j\}}(\mathbf{n})}{\partial n_{q_i}} - \left[A^{-1}(\mathbf{n})F_{\{q_1,\ldots,q_j\}}(\mathbf{n})\right] \times \left[A^{-1}(\mathbf{n})F_{\{q_i\}}(\mathbf{n})\right] \quad (63)$$

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

Using (31) and noting that $m^2(c_{q_i}^l) = 1$, we have

$$\frac{\partial F_{\{q_1,\ldots,q_j\}}(\mathbf{n})}{\partial n_{q_i}} = \begin{cases} \sigma^{-4}F_{\{q_1,\ldots,q_{i-1},q_{i+1},\ldots,q_j\}}(\mathbf{n}), & \text{if } q_i \in \{q_1,\ldots,q_j\} \\ F_{\{q_1,\ldots,q_j,q_i\}}(\mathbf{n}), & \text{otherwise.} \end{cases} \quad (65)$$

Substituting (65) in (64), and using (34), completes the proof. ■

Theorem 5:

Proof: We consider two different cases. If $q_i \in \{q_1,\ldots,q_j\}$, using (40), one can write

$$\frac{\partial^2 R_{\{q_1,\ldots,q_j\}}(\mathbf{n})}{\partial n_{q_i}^2} = \frac{\partial}{\partial n_{q_i}}\left[\sigma^{-4}R_{\{q_1,\ldots,q_{i-1},q_{i+1},\ldots,q_j\}}(\mathbf{n}) - R_{\{q_1,\ldots,q_j\}}(\mathbf{n})R_{\{q_i\}}(\mathbf{n})\right] \quad (66)$$

$$= \sigma^{-4}\frac{\partial}{\partial n_{q_i}}\left[R_{\{q_1,\ldots,q_{i-1},q_{i+1},\ldots,q_j\}}(\mathbf{n})\right] - \frac{\partial}{\partial n_{q_i}}\left[R_{\{q_1,\ldots,q_j\}}(\mathbf{n})R_{\{q_i\}}(\mathbf{n})\right] \quad (67)$$

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

$$= \sigma^{-4}\left[R_{\{q_1,\ldots,q_j\}}(\mathbf{n}) - R_{\{q_1,\ldots,q_{i-1},q_{i+1},\ldots,q_j\}}(\mathbf{n})R_{\{q_i\}}(\mathbf{n})\right] - \left[\sigma^{-4}R_{\{q_1,\ldots,q_{i-1},q_{i+1},\ldots,q_j\}}(\mathbf{n}) - R_{\{q_1,\ldots,q_j\}}(\mathbf{n})R_{\{q_i\}}(\mathbf{n})\right]R_{\{q_i\}}(\mathbf{n}) - R_{\{q_1,\ldots,q_j\}}(\mathbf{n})\left[\sigma^{-4} - R_{\{q_i\}}^2(\mathbf{n})\right] \quad (69)$$

$$= -2\sigma^{-4}R_{\{q_1,\ldots,q_{i-1},q_{i+1},\ldots,q_j\}}(\mathbf{n})R_{\{q_i\}}(\mathbf{n}) + 2R_{\{q_1,\ldots,q_j\}}(\mathbf{n})R_{\{q_i\}}^2(\mathbf{n}) \quad (70)$$

$$= -2R_{\{q_i\}}(\mathbf{n})\left[\sigma^{-4}R_{\{q_1,\ldots,q_{i-1},q_{i+1},\ldots,q_j\}}(\mathbf{n}) - R_{\{q_1,\ldots,q_j\}}(\mathbf{n})R_{\{q_i\}}(\mathbf{n})\right]$$

$$= -2R_{\{q_i\}}(\mathbf{n})\frac{\partial R_{\{q_1,\ldots,q_j\}}(\mathbf{n})}{\partial n_{q_i}}. \quad (71)$$

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

$$\frac{\partial^2 R_{\{q_1,\ldots,q_j\}}(\mathbf{n})}{\partial n_{q_i}^2} = \frac{\partial}{\partial n_{q_i}}\left[R_{\{q_1,\ldots,q_j,q_i\}}(\mathbf{n}) - R_{\{q_1,\ldots,q_j\}}(\mathbf{n})R_{\{q_i\}}(\mathbf{n})\right] \quad (72)$$

$$= \frac{\partial}{\partial n_{q_i}}\left[R_{\{q_1,\ldots,q_j,q_i\}}(\mathbf{n})\right] - \frac{\partial}{\partial n_{q_i}}\left[R_{\{q_1,\ldots,q_j\}}(\mathbf{n})R_{\{q_i\}}(\mathbf{n})\right] \quad (73)$$

$$= \sigma^{-4}R_{\{q_1,\ldots,q_j\}}(\mathbf{n}) - R_{\{q_1,\ldots,q_j,q_i\}}(\mathbf{n})R_{\{q_i\}}(\mathbf{n}) - \frac{\partial}{\partial n_{q_i}}\left[R_{\{q_1,\ldots,q_j\}}(\mathbf{n})\right]R_{\{q_i\}}(\mathbf{n}) - R_{\{q_1,\ldots,q_j\}}(\mathbf{n})\frac{\partial}{\partial n_{q_i}}\left[R_{\{q_i\}}(\mathbf{n})\right] \quad (74)$$

$$= \sigma^{-4}R_{\{q_1,\ldots,q_j\}}(\mathbf{n}) - R_{\{q_1,\ldots,q_j,q_i\}}(\mathbf{n})R_{\{q_i\}}(\mathbf{n}) - \left[R_{\{q_1,\ldots,q_j,q_i\}}(\mathbf{n}) - R_{\{q_1,\ldots,q_j\}}(\mathbf{n})R_{\{q_i\}}(\mathbf{n})\right] \times R_{\{q_i\}}(\mathbf{n}) - R_{\{q_1,\ldots,q_j\}}(\mathbf{n})\left[\sigma^{-4} - R_{\{q_i\}}^2(\mathbf{n})\right] \quad (75)$$

$$= -2R_{\{q_1,\ldots,q_j,q_i\}}(\mathbf{n})R_{\{q_i\}}(\mathbf{n}) + 2R_{\{q_1,\ldots,q_j\}}(\mathbf{n})R_{\{q_i\}}^2(\mathbf{n}) \quad (76)$$

$$= -2R_{\{q_i\}}(\mathbf{n}) \times \left[R_{\{q_1,\ldots,q_j,q_i\}}(\mathbf{n}) - R_{\{q_1,\ldots,q_j\}}(\mathbf{n})R_{\{q_i\}}(\mathbf{n})\right]$$

$$= -2R_{\{q_i\}}(\mathbf{n})\frac{\partial R_{\{q_1,\ldots,q_j\}}(\mathbf{n})}{\partial n_{q_i}}. \quad (77)$$

It can be seen from (71) and (77) that both cases ended up with the same expression as the one in (41), which completes the proof. ■

Proof of Property (48)

We can expand the right-hand side of (48) as follows:

$$ye^{-\frac{y^2}{2}}T_{i-1}(y) - e^{-\frac{y^2}{2}}\frac{d}{dy}T_{i-1}(y) \quad (78)$$

$$= e^{-\frac{y^2}{2}}\left[yT_{i-1}(y) - \frac{d}{dy}T_{i-1}(y)\right]. \quad (79)$$

Using (20), we have

$$T_{i-1}(y) = (-1)^{i-1}e^{\frac{y^2}{2}}\frac{d^{i-1}}{dy^{i-1}}\left[e^{-\frac{y^2}{2}}\right]. \quad (80)$$

It follows that:

$$\frac{d}{dy}T_{i-1}(y) = (-1)^{i-1}\left(ye^{\frac{y^2}{2}}\frac{d^{i-1}}{dy^{i-1}}\left[e^{-\frac{y^2}{2}}\right] + e^{\frac{y^2}{2}}\frac{d^i}{dy^i}\left[e^{-\frac{y^2}{2}}\right]\right) \quad (81)$$

$$= yT_{i-1}(y) - T_i(y). \quad (82)$$

Substituting (82) in (79), we have

$$e^{-\frac{y^2}{2}}T_i(y) \quad (83)$$

which is the left-hand side of (48).

Acknowledgment

The authors would like to thank M. H. Baligh, A. Heunis, and M. Thompson for their helpful discussions and comments. They also acknowledge the detailed comments of the reviewers which have been extremely helpful in improving the quality of this article.

REFERENCES

[1] A. Abedi and A. K. Khandani, "An analytical method for performance analysis of binary linear block codes," in *Proc. IEEE Int. Symp. Information Theory*, Lausanne, Switzerland, July 2002, p. 403.

[2] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Inform. Theory*, vol. IT-20, pp. 284–287, Mar. 1974.

[3] L. Ping and K. L. Yeung, "Symbol-by-symbol decoding of the Golay code and iterative decoding of concatenated Golay codes," *IEEE Trans. Inform. Theory*, vol. 45, pp. 2558–2562, Nov. 1999.

[4] Y. Liu, S. Lin, and M. P. C. Fossorier, "MAP algorithms for decoding linear block codes based on sectionalized trellis diagrams," *IEEE Trans. Commun.*, vol. 48, pp. 577–586, Apr. 2000.

[5] S. Riedel, "Symbol-by-symbol MAP decoding algorithm for high-rate convolutional codes that use reciprocal dual codes," *IEEE J. Select. Areas Commun.*, vol. 16, pp. 175–185, Feb. 1998.

[6] C. R. P. Hartmann and L. D. Rudolph, "An optimum symbol-by-symbol decoding rule for linear codes," *IEEE Trans. Inform. Theory*, vol. IT-22, pp. 514–517, Sept. 1976.

[7] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: turbo codes (1)," in *Proc. IEEE Int. Conf. Communications*, Geneva, Switzerland, May 1993, pp. 1064–1070.

[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 Trans. Inform. Theory*, vol. 42, pp. 1562–1568, Sept. 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 Trans. Inform. Theory*, vol. 41, pp. 441–447, Mar. 1995.

[11] C. R. P. Hartmann, L. D. Rudolph, and K. G. Mehrotra, "Asymptotic performance of optimum bit-by-bit decoding for the white Gaussian channel," *IEEE Trans. Inform. Theory*, vol. IT-23, pp. 520–522, July 1977.

[12] E. Baccarelli, R. Cusani, and G. D. Blasio, "Performance bound and trellis-code design criterion for discrete memoryless channels and finite-delay symbol-by-symbol decoding," *IEEE Trans. Commun.*, vol. 45, pp. 1192–1199, Oct. 1997.

[13] D. C. Schleher, "Generalized Gram–Charlier series with application to the sum of log-normal variates," *IEEE Trans. Inform. Theory*, vol. IT-23, pp. 275–280, Mar. 1977.

[14] G. L. Cariolaro and S. G. Pupolin, "Considerations on error probability in correlated-symbol systems," *IEEE Trans. Commun.*, vol. COM-25, pp. 462–467, Apr. 1977.

[15] S. Blinnikov and R. Moessner, "Expansions for nearly Gaussian distributions," *Astron. Astrophys. Supplement Ser.*, vol. 130, no. 1, pp. 193–205, May 1998.

[16] M. A. Najib and V. K. Prabhu, "Analysis of equal-gain diversity with partially coherent fading signals," *IEEE Trans. Veh. Technol.*, vol. 49, 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 Trans. Commun.*, vol. 38, pp. 1463–1474, Sept. 1990.

[18] C. Tellambura and A. Annamalai, "Further results on the Beaulieu series," *IEEE Trans. Commun.*, vol. 48, pp. 1774–1777, Nov. 2000.

[19] I. T. Monroy and G. Hooghiemstra, "On a recursive formula for the moments of phase noise," *IEEE Trans. Commun.*, vol. 48, pp. 917–920, June 2000.

[20] M. Kavehrad and M. Joseph, "Maximum entropy and the method of moments in performance evaluation of digital communications systems," *IEEE Trans. Commun.*, vol. COM-34, pp. 1183–1189, Dec. 1986.

[21] A. Abedi, P. Chaudhari, and A. K. Khandani, "On some properties of bit decoding algorithms," in *Proc. Canadian Workshop on Information Theory (CWIT 2001)*, Vancouver, BC, Canada, June 2001, pp. 106–109.

[22] H. El-Gamal and A. R. Hammons, Jr., "Analyzing the turbo decoder using the Gaussian approximation," *IEEE Trans. Inform. Theory*, vol. 47, pp. 671–686, Feb. 2001.

[23] D. Divsalar, S. Dolinar, and F. Pollara, "Iterative turbo decoder analysis based on density evolution," *IEEE J. Select. Areas Commun.*, vol. 19, pp. 891–907, May 2001.

[24] R. S. Freedman, "On Gram–Charlier approximations," *IEEE Trans. Commun.*, vol. COM-29, pp. 122–125, Feb. 1981.

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

[26] J. F. Kenney and E. S. Keeping, *Mathematics of Statistics*, 2nd ed. Princeton, NJ: Van Nostrand, 1951, pt. 2.

[27] A. W. Van der vaart, *Asymptotic Statistics*. Cambridge, U.K.: Cambridge Univ. Press, 1998.

[28] H. Cramer, *Mathematical Methods of Statistics*. Princeton, NJ: Princeton Univ. Press, 1957.

[29] ——, *Random Variables and Probability Distributions*, 3rd ed. Cambridge, U.K.: Cambridge Univ. Press, 1970.

[30] ——, "On some classes of series used in mathematical statistics," in *Proc. 6th Scandinavian Congr. Mathematicians*, Copenhagen, Denmark, 1925, pp. 399–425.

[31] E. T. Whittaker and G. N. Watson, *A Course of Modern Analysis*, 4th ed. Cambridge, U.K.: Cambridge Univ. Press, 1962.

[32] G. Szego, *Orthogonal Polynomials*. Providence, RI: RI Amer. Math. Soc., 1967.

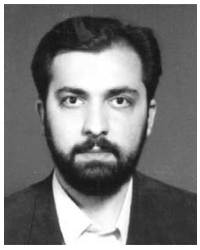
[33] P. V. Oneil, *Advanced Engineering Mathematics*, 4th ed. New York: Thomson, 1995.

[34] I. S. Gradshteyn and I. M. Ryzhik, *Table of Integrals, Series, and Products*, 5th ed. New York: Academic, 1994.

[35] T. M. Duman and M. Salehi, "New performance bounds for turbo codes," *IEEE Trans. Commun.*, vol. 46, pp. 717–723, June 1998.

[36] D. Divsalar, "A Simple Tight Bound on Error Probability of Block Codes With Application to Turbo Codes," NASA, JPL, Pasadena, CA, TMO Progress Rep. 42-139, 1999.

[37] I. Sason and S. Shamai, "Improved upper bounds on the ML decoding error probability of parallel and serial concatenated turbo codes via their ensemble distance spectrum," *IEEE Trans. Inform. Theory*, vol. 46, pp. 24–47, Jan. 2000.



Ali Abedi (S'98) was born in 1975 in Tehran, Iran. He received the B.Sc. and M.Sc. degrees in electrical engineering and communications systems, with highest academic rank, from Sharif University of Technology, Tehran, Iran, in 1996 and 1998, respectively. He is currently working toward the Ph.D. degree at the Coding and Signal Transmission Laboratory, Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada.

From 1998 to 2000, he was with the Department of Electrical Engineering, Iranian Air Force University, Tehran, Iran, as a Lecturer and Director of the Research Center. His research interests include performance evaluation of bit decoding algorithms for wireless digital communications systems.

Mr. Abedi is the recipient of the award for academic excellence from Natural Sciences and Engineering Research Council of Canada, Ontario Graduate Scholarship, and Graduate Incentive Award.

Amir K. Khandani (M'02) received the B.A.Sc. and M.A.Sc. degrees from the University of Tehran, Tehran, Iran, and the Ph.D. degree from McGill University, Montreal, QC, Canada, in 1985 and 1992, respectively.

He was a Research Associate with INRS-Telecommunication, Montreal, QC, Canada, for one year. In 1993, he joined the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada, where he is currently a Professor.

Dr. Khandani has received a number of awards and recognitions, including a Certificate of Achievement by Nortel Networks. He is currently serving as an Associate Editor for the IEEE TRANSACTIONS ON COMMUNICATIONS in the area of coding and communication theory.