A New Method for Performance Evaluation of Turbo-Codes Using Statistics of the Log Likelihood Ratio

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#2004-01

February 11, 2004

A New Method for Performance Evaluation of Turbo-Codes Using Statistics of the Log Likelihood Ratio

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: 519-8884567, Fax: 519-8884338
e-mail: {ali, khandani}@cst.uwaterloo.ca

Abstract

In this paper[1], we present a new method for the performance evaluation of Turbo-Codes. The method is based on estimating the Probability Density Function (pdf) of the bit Log Likelihood Ratio (LLR) by using an exponential model. It is widely known that the pdf of the bit LLR is close to the normal density, and the proposed approach takes advantage of this property to simplify the calculations. The moment matching method is combined with the maximum entropy principle to estimate the parameters of the new model. We present a simple method for computing the confidence intervals for the estimated parameters, as well as for the Bit Error Rate (BER). The corresponding results are adopted to compute the number of samples that are required for a given precision of the estimated values. It is demonstrated that this method requires significantly fewer samples than the conventional Monte-Carlo (MC) simulation.

Index Terms

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

Bit Decoding, Log Likelihood Ratio, Maximum Entropy, Probability Density Function, Turbo-Code.

I. Introduction

In the application of channel codes, one of the most important issues is to develop an efficient method for performance evaluation, since the Monte-Carlo (MC) simulation is extremely time consuming for low Bit Error Rate (BER) values. In 1993, a new class of channel codes, called Turbo-Codes, were announced [1]. They have an astonishing performance, and at the same time, allow for a simple iterative decoding method by using the reliability information produced by a bit decoding algorithm. Since then, there have been numerous efforts devoted to the performance evaluation of Turbo-Codes. These approaches derive some bounds on the average performance of Turbo-Codes by assuming Maximum Likelihood (ML) decoding [2]–[4].

Some researchers have considered simplified cases of analytical BER calculations. An analytical method for computing the bit error probability of a two-state convolutional code with Maximum a Posteriori Probability (MAP) decoding is presented in [5]. The Pearson system of distributions is adopted in [6] to compute the error probability expectations, where moment matching is used to estimate the parameters of the model. Estimating the parameters of the generalized Gaussian Probability Density Function (pdf) by using entropy matching is considered in [7].

Other researchers have employed the Importance Sampling (IS) method to improve the performance of the MC simulation by increasing the weight of the rare error events. In this method, instead of choosing the samples from the original distribution, the samples are selected from a modified distribution which concentrates the points where the rare error events occur. This modified distribution is obtained from the original distribution by the application of a biasing function. This ensures a variance reduction if the biasing function is appropriately selected. The Gaussian Tail (GT) and Rayleigh Tail (RT) biasing functions are investigated in [8]. The IS method is applied to evaluate the performance of a digital communications system with Inter-Symbol Interference (ISI) in [9], and is extended to evaluate the performance of multi-hop satellite links in [10]. A general formulation of the IS method in probability space notation is introduced in [11]. The IS method is used in [12] to simulate the Viterbi decoder by examining the trellis structure in relation to the rare error events. A comparison between the Mean Translation (MT) technique and the Variance Scaling (VS) technique is performed in [13], where it is shown that the structure of the error regions determines the better method. Recently, [14] has revisited the IS method with the strategy to increase the rate by which the variance approaches zero, instead of reducing the variance itself.

The Turbo-Product Codes (of a small block length) are simulated in [15] by partitioning the error regions and by using MT for each sub region independently. This method becomes inefficient, as the complexity of the code increases. In the case of Turbo-Codes with a large block length, the search for the appropriate biasing functions may be lengthy, which renders this method even more complicated than the conventional MC simulation.

It is observed in [1], [4], [16] that the pdf of the bit Log Likelihood Ratio (LLR) is nearly Gaussian. This motivates us to propose an exponential model which has a polynomial in the exponent. The aforementioned model has the ability to efficiently capture the deviation of the desired pdf from Gaussian. We use the moments of the bit LLR to estimate the parameters for the proposed model, in this article.

This paper is organized as follows. We model the pdf of the bit LLR by using its symmetry properties in Section II. In Section III, the maximum entropy method to find the parameters of the proposed model is described. A method to compute the confidence intervals for the estimated parameters, as well as the estimated BER, is detailed in Section IV. The numerical results and conclusion are presented in Section V and Section VI, respectively.

II. Modeling the pdf of the Bit LLR

A common tool to express the bit probabilities in bit decoding algorithms is based on the so-called LLR. The LLR of the k^{th} bit position is defined by the following equation:

$$LLR(k) = \log \frac{P(c_k = 1|\mathbf{x})}{P(c_k = 0|\mathbf{x})}, \tag{1}$$

where c_k is the value of the k^{th} bit in the transmitted code-word, $\mathbf{x}$ is the received vector, and $\log$ represents the natural logarithm. Let us define the random variable $Y = LLR(k)$ with its pdf denoted as $f(y)$. It is proved in [17] that the pdf of the bit LLR is independent of the transmitted code-word, as long as the value of the bit position under consideration remains unchanged. By using this result and without the loss of generality, we consider the case of sending the all-zero code-word. It is proved in [18] that the pdf of the bit LLR has the following symmetry property:

$$f(y) = e^{-y} f(-y). \tag{2}$$

Taking the logarithm from both sides of (2), we can write the following:

$$\log f(y) - \log f(-y) = -y. \tag{3}$$

Utilizing the power series, it easily follows that

$$\log f(y) = -y/2 + \sum_{i=0}^{\infty} a_i y^{2i}. \tag{4}$$

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

$$f(y) \simeq \exp(-y/2 + \sum_{i=0}^{N} a_i y^{2i}). \tag{5}$$

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

$$P_e \simeq \int_0^{\infty} f(y) dy. \tag{6}$$

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

III. MOMENT MATCHING USING THE MAXIMUM ENTROPY PRINCIPLE

There are various methods for parameter estimation. Typically, the unknown parameters of a pdf can be found by adopting moment matching, entropy matching, or ML. In this paper, we use the moment matching method with the maximum entropy principle simply because it is mathematically tractable, and has been successfully implemented in a variety of applications [19]. An attractive feature of the class of the distributions with the maximum entropy is that a simple iterative maximization technique can be employed to compute their parameters. The maximum entropy principle was first introduced by Jaynes [19] in 1982. Since then, it has been widely used in various applications. In this method, the search, while satisfying the constraints on the moments, is limited to the pdf with the maximum entropy. For more recent discussions on this method, refer to [20], [21]. We follow an approach that is similar to the one introduced in [22]. The maximum entropy density can be found by maximizing the following with respect to $\hat{f}(y)$:

$$\text{Maximize } -\int_{-\infty}^{+\infty} \hat{f}(y)\log[\hat{f}(y)]dy, \tag{7}$$

$$\text{Subject to: } \hat{\mu}_i = \mu_i, \quad i = 1, 2, \ldots, M, \tag{8}$$

with

$$\mu_i = \int_{-\infty}^{+\infty} y^i f(y)dy \tag{9}$$

and

$$\hat{\mu}_i = \int_{-\infty}^{+\infty} y^i \hat{f}(y)dy, \tag{10}$$

where M is the number of moments used in the parameter estimation. This maximization problem can be solved with the Lagrange multipliers λ_k, $k = 0, 1, \ldots, M$ by following the methods of the calculus of variations [23]. Let us define the Lagrangian as

$$\int_{-\infty}^{+\infty} \hat{f}(y)\log[\hat{f}(y)]dy + c\int_{-\infty}^{+\infty} \hat{f}(y)dy + \sum_{k=1}^{M} \lambda_k \int_{-\infty}^{+\infty} y^k \hat{f}(y)dy. \tag{11}$$

Setting the variations of (11) with respect to $\hat{f}(y)$ to zero, we have

$$\log[\hat{f}(y)] + \lambda_0 + \sum_{k=1}^{M} \lambda_k y^k = 0, \tag{12}$$

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

$$\hat{f}(y) = \exp(-\sum_{k=0}^{M} \lambda_k y^k). \tag{13}$$

From (5), it is clear that all of the odd coefficients, except λ_1, are zero. Hence, (13) can be reformulated with the new coefficients $a_k = \lambda_{2k}$, $k = 0, 1, \ldots, N = \lfloor \frac{M}{2} \rfloor$, as follows:

$$\hat{f}(y) = \exp(-y/2 - \sum_{k=0}^{N} a_k y^{2k}). \tag{14}$$

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

$$e^{a_0} = \int_{-\infty}^{+\infty} \exp(-y/2 - \sum_{k=1}^{N} a_k y^{2k}) dy. \tag{15}$$

If (15) is substituted for e^{a_0} in (14), then,

$$\hat{f}(y) = \exp\left\{-y/2 - \sum_{k=1}^{N} a_k y^{2k} - \log\left[\int_{-\infty}^{+\infty} \exp(-z/2 - \sum_{k=1}^{N} a_k z^{2k}) dz\right]\right\}. \tag{16}$$

The objective is to estimate the parameters a_k, $k = 0, 1, \ldots, N$, where a_0 can be computed using (15). As we will see later, one can estimate the parameters a_k, $k = 1, \ldots, N$ using the first N moments of the bit LLR. In practice, the statistical estimates of the moments are used instead of the true moments. Using (10), we have

$$\hat{\mu}_i = \int_{-\infty}^{+\infty} y^i \exp\left\{-y/2 - \sum_{k=1}^{N} a_k y^{2k} - \log\left[\int_{-\infty}^{+\infty} \exp(-z/2 - \sum_{k=1}^{N} a_k z^{2k}) dz\right]\right\} dy, \tag{17}$$

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

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

$$G_i(a_1, a_2, \ldots, a_N) = \hat{\mu}_i - \mu_i$$
$$= \int_{-\infty}^{+\infty} (y^i - \mu_i) \exp\left\{-y/2 - \sum_{k=1}^{N} a_k y^{2k} - \log\left[\int_{-\infty}^{+\infty} \exp(-z/2 - \sum_{k=1}^{N} a_k z^{2k}) dz\right]\right\} dy = 0, \tag{18}$$

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

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

$$\mathbf{a}^{(r+1)} = \mathbf{a}^{(r)} + \Delta\mathbf{a}^{(r)}. \tag{19}$$

This signifies that

$$G_i(\mathbf{a}^{(r)} + \Delta\mathbf{a}^{(r)}) \simeq G_i(\mathbf{a}^{(r)}) + \sum_{k=1}^{N} \frac{\partial G_i(\mathbf{a}^{(r)})}{\partial a_k^{(r)}} \Delta a_k^{(r)}, \quad i = 1, 2, \ldots, N. \tag{20}$$

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

$$G_i(\mathbf{a}^{(r)}) = \hat{\mu}_i^{(r)} - \mu_i = \sum_{k=1}^{N} \left[-\frac{\partial G_i(\mathbf{a}^{(r)})}{\partial a_k^{(r)}}\right] \Delta a_k^{(r)}, \quad i = 1, 2, \ldots, N, \tag{21}$$

where notation $\hat{\mu}_i^{(r)}$ is employed to point out that the estimated moments are updated by replacing $\mathbf{a}^{(r)}$ in (17) after the r^{th} iteration. Differentiating (18) with respect to a_k yields

$$\frac{\partial G_i(\mathbf{a}^{(r)})}{\partial a_k^{(r)}} = \left.\frac{\partial G_i(\mathbf{a})}{\partial a_k}\right|_{\mathbf{a}=\mathbf{a}^{(r)}} = \hat{\mu}_{2k}^{(r)}.\hat{\mu}_i^{(r)} - \hat{\mu}_{2k+i}^{(r)}. \tag{22}$$

The algorithm[2] is summarized in the following steps:

- *Step1:* Start with an initial value $\mathbf{a}^{(0)} = \{a_1^{(0)}, a_2^{(0)}, \ldots, a_N^{(0)}\}$.
- *Step2:* Compute the estimated moments by replacing $\mathbf{a}^{(r)}$ into (17).
- *Step3:* Plug the estimated moments into (21) to find $\Delta\mathbf{a}^{(r)}$.
- *Step4:* Compute the new parameters $\mathbf{a}^{(r+1)} = \mathbf{a}^{(r)} + \Delta\mathbf{a}^{(r)}$.

[2] This algorithm is adopted from [22].

- *Step5:* Go to *Step 2,* if $\|\Delta\mathbf{a}^{(r)}\| > \epsilon$, where ϵ is the desired precision and $\|.\|$ denotes the norm of a vector.

The convexity of this maximization problem guarantees that if a stationary point is found for some finite values of $a_1, \ldots, a_N$, it must be a unique absolute minimum [24]. However, the convexity alone does not imply that such a minimum should exist. More discussions on the convexity of the problem and existence of the solution can be found in [24], [25].

IV. Confidence Interval Statements

In the following, we first propose a method to compute the confidence intervals for the estimated parameters of the model in terms of the covariance matrix of the estimated moments. Subsequently, we derive a relationship between the confidence interval on the BER and the confidence interval on the parameters.

A. Confidence Interval for the Estimated Parameters

If the moment estimator satisfies a set of mild conditions, it follows that the estimated parameters are asymptotically normal with a derivable covariance matrix [26]. This allows for the confidence interval statements to be made concerning $\hat{f}(y)$. In the following, we present a method to compute the covariance matrix of the estimated parameters in terms of the covariance matrix of the moments. The covariance matrix of the moments can be easily computed by using the method described in Appendix A.

We can rewrite (21) as follows:

$$\hat{\mu}_i^{(r)} - \mu_i = \sum_{k=1}^{N} h_{ik}^{(r)} \Delta a_k^{(r)}, \tag{23}$$

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

$$u_{ij}^{(r)} = \text{cov}(\hat{\mu}_i^{(r)}, \hat{\mu}_j^{(r)}) = E[(\hat{\mu}_i^{(r)} - \mu_i)(\hat{\mu}_j^{(r)} - \mu_j)], \tag{24}$$

where it is assumed[3] that $\mu_i = E[\hat{\mu}_i^{(r)}]$ and $\mu_j = E[\hat{\mu}_j^{(r)}]$. Using (23), we have

$$u_{ij}^{(r)} = E\left[\left(\sum_{k=1}^{N} h_{ik}^{(r)} \Delta a_k^{(r)}\right)\left(\sum_{m=1}^{N} h_{jm}^{(r)} \Delta a_m^{(r)}\right)\right] \tag{25}$$

$$= \sum_{k=1}^{N}\sum_{m=1}^{N} h_{ik}^{(r)} h_{jm}^{(r)} E[\Delta a_k^{(r)} \Delta a_m^{(r)}] \tag{26}$$

$$= \sum_{k=1}^{N}\sum_{m=1}^{N} h_{ik}^{(r)} h_{jm}^{(r)} \text{cov}(\Delta a_k^{(r)}, \Delta a_m^{(r)}) \tag{27}$$

$$= \sum_{k=1}^{N} h_{ik}^{(r)} \sum_{m=1}^{N} h_{jm}^{(r)} c_{mk}^{(r)} \tag{28}$$

$$= \sum_{k=1}^{N} h_{ik}^{(r)} d_{jk}^{(r)}. \tag{29}$$

In the matrix notation, the following is defined:

$$D = \{d_{ij}^{(r)}\}, \quad \text{where } d_{ij}^{(r)} = \sum_{m=1}^{N} h_{im}^{(r)} c_{mj}^{(r)}, \tag{30}$$

$$H = \{h_{ij}^{(r)}\}, \quad \text{where } h_{ij}^{(r)} = -\frac{\partial G_i(\mathbf{a}^{(r)})}{\partial a_j^{(r)}}, \tag{31}$$

$$U = \{u_{ij}^{(r)}\}, \quad \text{where } u_{ij}^{(r)} = \text{cov}(\hat{\mu}_i^{(r)}, \hat{\mu}_j^{(r)}), \tag{32}$$

and

$$C^{(r)} = \{c_{ij}^{(r)}\}, \quad \text{where } c_{ij}^{(r)} = \text{cov}(\Delta a_i^{(r)}, \Delta a_j^{(r)}). \tag{33}$$

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

$$U = H.(H.C^{(r)})^T = H.(C^{(r)})^T.H^T. \tag{34}$$

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

$$C^{(r)} = (H^{-1}.U.(H^T)^{-1})^T = H^{-1}.U^T.(H^T)^{-1}. \tag{35}$$

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

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

$$A = \sum_{r=1}^{R} C^{(r)}, \tag{36}$$

where R is the total number of iterations. With the covariance matrix of the parameters, the desired confidence intervals for the parameters can be easily computed.

B. Confidence Interval for the Estimated BER

In the following, we use the previous results to compute the confidence interval on the BER. Let us assume that the $c\%$ confidence interval on each parameter a_i is equal to some positive α_i, i.e.,

$$p(|\delta a_1| < \alpha_1, \ldots, |\delta a_N| < \alpha_N) = \frac{c}{100}, \tag{37}$$

where δa_i represents the error in the computation of the parameters. Using this notation, we can rewrite the BER integral from (6) as follows:

$$P_e + \Delta P_e(\delta a_1, \ldots, \delta a_N) = \int_0^\infty \exp[-y/2 + \sum_{i=0}^{N}(a_i + \delta a_i)y^{2i}]dy \tag{38}$$

$$= \int_0^\infty \exp(-y/2 + \sum_{i=0}^{N} a_i y^{2i}) \exp(\sum_{i=0}^{N} \delta a_i y^{2i})dy \tag{39}$$

$$\simeq \int_0^\infty \exp(-y/2 + \sum_{i=0}^{N} a_i y^{2i})(1 + \sum_{i=0}^{N} \delta a_i y^{2i})dy \tag{40}$$

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

$$+ \int_0^\infty \exp(-y/2 + \sum_{i=0}^{N} a_i y^{2i}) \sum_{i=0}^{N} \delta a_i y^{2i} dy \tag{41}$$

$$= P_e + \sum_{i=0}^{N} \delta a_i \int_0^\infty y^{2i} \exp(-y/2 + \sum_{i=0}^{N} a_i y^{2i})dy \tag{42}$$

$$= P_e + \sum_{i=0}^{N} m_i \delta a_i, \tag{43}$$

where

$$m_i = \int_0^{\infty} y^{2i} \exp(-y/2 + \sum_{i=0}^{N} a_i y^{2i}) dy. \tag{44}$$

It can be seen that ΔP_e, the error in the BER estimation, is a linear combination of m_i's, which can be estimated during the procedure of the moment computation by considering the positive samples only. Recalling the confidence interval statement (37) for the parameters, and noting that ΔP_e is a linear combination of δa_i's, we can present a similar statement for the BER as follows:

$$p(|\Delta P_e(\delta a_1, \ldots, \delta a_N)| < \Delta P_e(\alpha_1, \ldots, \alpha_N)) = \frac{c}{100}. \tag{45}$$

This analysis enables us to make confidence interval statements on the estimated BER in terms of the confidence intervals for the model parameters.

V. Numerical Results

A Turbo-Code of the length 100 and rate $1/2$ is employed to perform the simulations. In Table 1, variances of the BER estimations are computed for both methods. The number of samples and the variance to the mean ratio of the BER are denoted as $n(.)$ and $v(.)$, respectively. The variance of the MC method can be computed analytically (refer to Appendix B), although this analysis is very complex for the proposed method and we need to estimate the variances with numerical methods. The variance of the proposed method can be computed by repeating the experiment for J times (generating J independent sets of moments), and computing the variance of the resulting sequence of the BER values, denoted as p_i, $i = 1, \ldots, J$, as follows:

$$E[\hat{P}_e] = \frac{1}{J} \sum_{i=1}^{J} p_i \tag{46}$$

and

$$\text{var}[\hat{P}_e] = -(E[\hat{P}_e])^2 + \frac{1}{J} \sum_{i=1}^{J} p_i^2. \tag{47}$$

In the computations of Table 1, we set $J = 1000$ to obtain a reasonable approximation, and at the same time, render the analysis feasible in the sense of the required time.

We use the relative gain G in Table 1 as a measure to compare the two methods. To incorporate both the variance reduction and the sample reduction advantage of the new method, and noting that $v(\mathrm{MC})$ is inversely proportional[4] to $n(\mathrm{MC})$, we define G as follows:

$$G = \frac{v(\mathrm{MC})}{v(\text{our method})} . \frac{n(\mathrm{MC})}{n(\text{our method})} . \tag{48}$$

Simulation results are shown in Figure 1, where we have used the same number of samples as indicated in Table 1. It is evident that increasing the number of moments (the order of approximation) that are involved from two to five significantly improves the approximation.

In addition we compute the confidence intervals by using the proposed method in Section IV for this example. This confidence interval is closely related to n, the number of samples used to compute it. In Table 2, this relation is presented for three different values of n at E_b/N_0=2dB. We compute[5] $p(|\Delta P_e| < \theta)$ for the different values of n, where $\theta = \Delta P_e(\alpha|a_1|, \ldots, \alpha|a_N|)$. When we compare the proposed method with the MC simulation in Table 1 and Table 2, the number of samples required for the BER calculations indicate a significant reduction for our method. It can be seen that the proposed method is more accurate than the MC simulation even by using significantly fewer samples.

VI. Concluding Remarks

In this paper, we have proposed a new method for the performance evaluation of Turbo-Codes. Although our focus is on Turbo-Codes, the application of the proposed method is not necessarily restricted to this class of channel codes. The problem of finding the BER in high signal to noise ratio regions can be solved with this method, since the MC simulation may not be feasible. We take advantage of the symmetry properties of the pdf of the bit LLR to propose a suitable model for this unknown density. The moment

[4]Refer to Appendix B.

[5]Refer to Appendix C for more details on the confidence interval for the MC simulation.

matching method is employed to find the density with the maximum entropy which satisfies the moment constraints. A simple method is introduced to make confidence interval statements both for the parameters of the model and the BER integral, which enables us to compute the BER values accurately. It is demonstrated that significantly fewer samples, compared to those required in the MC simulation, are necessary to compute the statistical moments that are accurate enough.

APPENDIX

A. Covariance Matrix of the Moments

Let us define the k^{th} moment of the random variable Y (which corresponds to the bit LLR) as,

$$\mu_k = E[Y^k], \tag{49}$$

which can be estimated by statistical averaging as follows:

$$\tilde{\mu}_k = \frac{1}{n}\sum_{i=1}^{n} y_i^k, \tag{50}$$

where y_i is one instance of the random variable Y, and n is the number of samples. The covariance matrix of the moments can be computed as follows:

$$\begin{aligned}
\text{cov}(\tilde{\mu}_k, \tilde{\mu}_m) &= E[(\tilde{\mu_k} - \mu_k)(\tilde{\mu_m} - \mu_m)] && (51)\\
&= E[(\frac{1}{n}\sum_{i=1}^{n} y_i^k - \mu_k)(\frac{1}{n}\sum_{j=1}^{n} y_j^m - \mu_m)] && (52)\\
&= \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n} E[y_i^k y_j^m] - \mu_k\mu_m && (53)\\
&= \frac{1}{n^2}\left[n\mu_{k+m} + (n^2 - n)\mu_k\mu_m\right] - \mu_k\mu_m && (54)\\
&= \frac{1}{n}(\mu_{k+m} - \mu_k\mu_m). && (55)
\end{aligned}$$

B. Variance of Monte-Carlo (MC) Simulation

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

$$e_i = \begin{cases} 1, & b_i \neq \hat{b}_i, \\ 0, & \text{otherwise.} \end{cases} \tag{56}$$

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

$$\hat{P}_e = \frac{1}{n}\sum_{i=1}^{n} e_i. \tag{57}$$

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

$$\begin{aligned}
\text{var}[\hat{P}_e] &= E[(\frac{1}{n}\sum_{i=1}^{n} e_i)^2] - (E[\frac{1}{n}\sum_{i=1}^{n} e_i])^2 && (58)\\
&= \frac{1}{n^2}\left(E[\sum_{i=1}^{n} e_i^2] + E[\sum_{i=1}^{n}\sum_{i\neq j=1}^{n} e_i e_j]\right) - P_e^2 && (59)\\
&= \frac{1}{n^2}\sum_{i=1}^{n} E[e_i^2] + \frac{1}{n^2}\sum_{i=1}^{n}\sum_{i\neq j=1}^{n} E[e_i]E[e_j] - P_e^2 && (60)\\
&= \frac{P_e}{n} + \frac{n(n-1)}{n^2}P_e^2 - P_e^2 && (61)\\
&= \frac{P_e}{n}(1 - P_e). && (62)
\end{aligned}$$

In practice, an estimation of var$[\hat{P}_e]$ is obtained by substituting P_e with $\hat{P}_e$ in (62).

C. Computing Confidence Intervals on Monte-Carlo (MC) Simulation

Let us define the $c\%$ confidence interval on MC, denoted as α, as follows:

$$p(|P_e - \hat{P}_e| < \alpha) = \frac{c}{100}, \tag{63}$$

where $P_e, \hat{P}_e$ are the true and the estimated values of the BER. Following the same notation and definitions as Appendix B, for a large n and some integers m and a, we can represent P_e and α as $\frac{m}{n}$ and $\frac{a}{n}$, respectively. We can find the confidence interval for the MC simulation as follows:

$$p(|P_e - \hat{P}_e| < \frac{a}{n}) = p(\frac{m-a}{n} < \frac{1}{n}\sum_{i=1}^{n} e_i < \frac{m+a}{n}) \tag{64}$$

$$= p(\sum_{i=1}^{n} e_i < m+a) - p(\sum_{i=1}^{n} e_i \leq m-a) \tag{65}$$

$$= \sum_{j=0}^{m+a-1} p(j \text{ errors among } n \text{ bits}) - \sum_{j=0}^{m-a} p(j \text{ errors among } n \text{ bits}) \tag{66}$$

$$= \sum_{j=0}^{m+a-1} \binom{n}{j} P_e^j (1-P_e)^{n-j} - \sum_{j=0}^{m-a} \binom{n}{j} P_e^j (1-P_e)^{n-j} \tag{67}$$

$$= \sum_{j=m-a+1}^{m+a-1} \binom{n}{j} P_e^j (1-P_e)^{n-j}. \tag{68}$$

In practice, an estimation of (64) is obtained by substituting P_e with $\hat{P}_e$ in (68).

REFERENCES

[1] C. Berrou, A. Glavieux and 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] T. M. Duman and M. Salehi, "New Performance Bounds for Turbo Codes," *IEEE Transactions on Communications*, vol. 46, no. 6, pp. 717-723, June 1998.

[3] 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 Transactions on Information Theory*, vol.46, No.1, pp. 24-47, January 2000.

[4] 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.

[5] H. Yoshikawa, "Theoretical Analysis of Bit Error Probability for Maximum a Posteriori Probability Decoding," *Proceedings of the IEEE International Symposium on Information Theory (ISIT 2003)*, Yokohama, Japan, July 2003.

[6] M. R. D. Rodrigues, J. E. Mitchell, I. Darwazeh and J. J. O'Reilly, "Error Probability Evaluation with a Limited Number of Moments," *Proceedings of the IEEE International Symposium on Information Theory (ISIT 2003)*, Yokohama, Japan, July 2003.

[7] B. Aiazzi, L. Alparone and S. Baronti, "Estimation Based on Entropy Matching for Generalized Gaussian pdf Modeling," *IEEE Signal Processing Letters*, vol. 6, no. 6, pp. 138-140, June 1999.

[8] N. C. Beaulieu, "An Investigation of Gaussian Tail and Rayleigh Tail Density Functions for Importance Sampling Digital Communication System Simulation," *IEEE Transactions on Communications*, vol. 38, no. 9, pp. 1288-1292, September 1990.

[9] K. S. Shanmugam and P. Balaban, "A Modified Monte-Carlo Simulation Technique for the Evaluation of Error Rate in Digital Communication Systems," *IEEE Transactions on Communications*, vol. 28, no. 11, pp. 1916-1924, November 1980.

[10] M. C. Jeruchim, "On the Application of Importance Sampling to the Simulation of Digital Satellite and Multi-hop Links," *IEEE Transactions on Communications*, vol. 32, no. 10, pp. 1088-1092, October 1984.

[11] Q. Wang and V. K. Bhargava, "On the Application of Importance Sampling to BER Estimation in the Simulation of Digital Communication Systems," *IEEE Transactions on Communications*, vol. 35, no. 11, pp. 1231-1233, November 1987.

[12] J. S. Sadowsky, "A New Method for Viterbi Decoder Simulation Using Importance Sampling," *IEEE Transactions on Communications*, vol. 38, no. 9, pp. 1341-1351, September 1990.

[13] J. C. Chen, D. Lu, J. S. Sadowsky and K. Yao, "On Importance Sampling in Digital Communications. I. Fundamentals," *IEEE Journal on Selected Areas in Communications*, vol. 11, no. 3, pp. 289-299, April 1993.

[14] J. A. Bucklew and R. Radeke, "On the Monte-Carlo Simulation of Digital Communication Systems in Gaussian Noise," *IEEE Transactions on Communications*, vol. 51, no. 2, pp. 267-274, February 2003.

[15] M. Ferrari and S. Bellini, “Importance Sampling Simulation of Turbo-Product Codes,” *Proceedings of IEEE International Conference on Communications (ICC 2001)*, vol. 9, pp. 2773-2777, June 2001.

[16] H. El-Gamal and 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.

[17] A. Abedi and A. K. Khandani, “Some Properties of Bit Decoding Algorithms for Binary Linear Block Codes,” *IEEE Transactions on Information Theory*, Revised, February 2004.

[18] T. J. Richardson, M. A. Shokrollahi and R. L. Urbanke, “Design of Capacity Approaching Irregular Low-Density-Parity-Check Codes,” *IEEE Transactions on Information Theory*, Vol.47, No.2, pp. 619-637, February 2001.

[19] E. T. Jaynes, “On The Rationale of Maximum-Entropy Methods,” *Proceedings of the IEEE*, Vol. 70, No. 9, pp. 939-952, September 1982 (Invited paper).

[20] M. Grendar Jr. and M. Grendar, “Maximum Entropy: Clearing up Mysteries,” Journal of Entropy, Vol. 3, pp. 58-63, 2001.

[21] P.D. Grunwald and A. P. Dawid, “Game Theory, Maximum Generalized Entropy, Minimum Discrepancy, Robust Bayes and Pythagoras,” *Information Theory Workshop*, Bangalore, India, pp. 94-07, October 2002.

[22] M. Kavehrad and M. Joseph, “Maximum Entropy and the Method of Moments in Performance Evaluation of Digital Communications Systems,” *IEEE Transactions on Communications*, Vol.34, No.12, pp. 1183-1189, December 1986.

[23] R. Weinstock, *Calculus of Variations with Applications to Physics and Engineering*, New York: Dover, 1974.

[24] L. R. Mead and N. Papanicolaou, “Maximum entropy in the Problem of Moments,” *Journal of Mathematical Physics*, Vol. 25, pp. 2404-2417, Aug. 1984.

[25] N. Agmon, Y. Alhassid and R. D. Levine, “An Algorithm for Finding the Distribution of Maximal Entropy,” *Journal of Comp. Physics*, Vol. 30, pp. 250, 1979.

[26] S. Kullback, *Information Theory and Statistics*, New York: Wiley, 1959.

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

Table 1 : Comparison of the proposed method and the MC simulation, where the variances are computed as described in Section V.

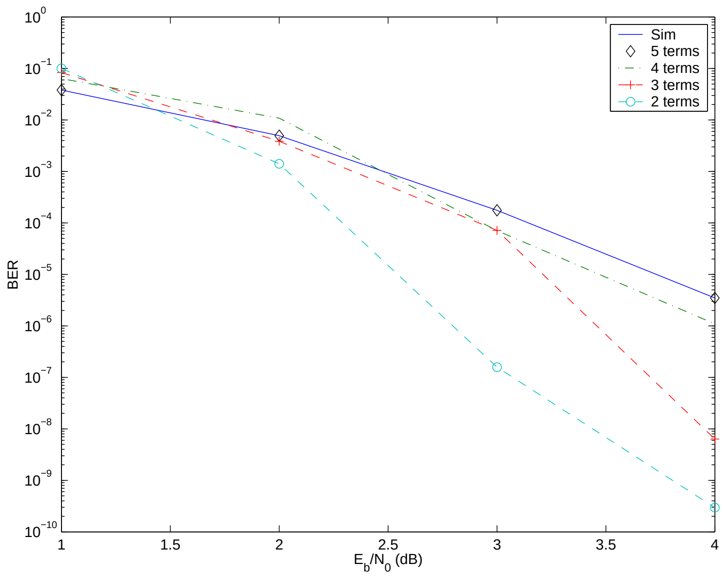

Fig. 1. BER curves for Turbo-Code of the length 100 and rate $1/2$ in comparison with the MC simulation.

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

Table 2 : Relation between n and confidence interval at E_b/N_0=2dB for the new method and the MC simulation.