

# A New Fast Density Evolution Method for LDPC Codes Using Higher Order Statistics

Soroush Akhlaghi and Amir K. Khandani

Coding & Signal Transmission Laboratory (www.cst.uwaterloo.ca)

E&CE Department, Univ. of Waterloo, Waterloo, ON, Canada, N2L 3G1  
 {soroush,khandani}@cst.uwaterloo.ca, Tel: (519) 885-1211

Abolfazl Falahati

Department of Electrical Engineering,

Iran University of Science & Technology (IUST), Tehran, IRAN

afalahati@iust.ac.ir

**Abstract**—Density Evolution (DE) is a technique for tracking the distribution of the Log Likelihood Ratio (LLR) messages exchanged between the variable nodes and the check nodes in a bipartite graph [1]. It is widely assumed that these distributions are close to Gaussian. However, in many scenarios, this assumption is not valid, e.g., the case that the Signal to Noise Ratio (SNR) is low, or the degree of variable nodes exceeds a certain threshold [2]. This article introduces a new (suboptimal) method for DE algorithm in Low-Density Parity-Check (LDPC) codes. We provide a more accurate model for the distribution of message bits (as compared to Gaussian) through matching the first  $n$  statistical moments. An iterative message passing algorithm is proposed to compute these moments from the graphical representation of the underlying code. We show that the proposed algorithm results in an improved estimate of the underlying EXIT chart as compared to using a Gaussian assumption. In this respect, the proposed method achieves a performance very close to that of the best earlier methods reported in [2] and [3], while it offers a much lower complexity

relies on Discretizing the exchange messages. The complexity of this method for each iteration is of the order  $O(2^{2k})$  due to the calculation at the check nodes, where  $k$  is the number of quantization bits [3]. This method is too complex for being used as the cost function since the optimization problem is not convex [9]. There are some sub-optimum DE algorithms which reduce the complexity at the price of a significant loss in the accuracy [10]–[12].

In this paper, we present a new sub-optimum method for DE that has a low complexity, and at the same time, is more accurate than the other known sub-optimum approaches. Unlike the previous methods reported in [2], [3] which try to estimate the output pdf of the check node message (with a complexity of  $O(2^{2k})$ ), the proposed method utilizes the corresponding moments of these messages in order to estimate the output pdf of the variable nodes. This results in a complexity order of  $O(n^3)$  where  $n$  is the number of required moments. We propose a nearly Gaussian model for the Probability Density Function (PDF) of the variable node messages that is consistent. Then, we force the first  $n$  moments of this PDF to match the first  $n$  moments of the exact messages generated from the message-passing algorithm by applying the Newton-Raphson algorithm [13]. For computing the exact moments, we need to solve some integrals that do not accept closed form solutions. With some manipulations, these integrals can be written in the form of  $\int f(x)e^{-x^2}$  which can be computed using the  $m$  point Gauss-Hermit quadrature formula, where  $m$  is the order of Hermit polynomial [14]. Then, we present a closed-form expression for computing the Message Error Probability (MEP) at each iteration.

The article is organized as follows: Section II provides the basic principles of LDPC codes. Section III contains a short overview and some background information on Gaussian approximation and consistency condition. Also, we study the condition that Gaussian approximation is not a good choice for modeling the distribution of the messages. In

## I. INTRODUCTION

LDPC codes, invented by Gallager in 1963 [4], did not receive much attention for more than three decades until turbo codes were introduced by Berrou et al. [5]. LDPC codes were rediscovered by MacKay et al. [6] and Spielman et al. [7]. Later, Luby in [8] introduced the Irregular LDPC codes. The excellent performance of this type of LDPC codes for various channels [1] has put them in the center of attention in the field of capacity approaching codes.

LDPC codes are designed by optimizing the degree structure of the underlying Tanner graph. Density Evolution (DE) is used as the cost function for this optimization problem [1]. In [3], the authors suggest an algorithm called Discretized Density Evolution (DDE) to compute the pdf of exchange messages in the corresponding Tanner graph. This algorithm

This work is financially supported by Communications and Information Technology Ontario (CITO), Nortel Networks, and Natural Sciences and Engineering Research Council of Canada (NSERC).

Sections IV, the proposed algorithm is introduced and applied to both regular and irregular LDPC codes. Numerical results are provided in Section V showing high accuracy and low computational complexity for the proposed method. Finally, Section VI concludes the paper.

Throughout this paper, random variables are uppercase, a realization of random variables are lowercase, vectors are boldfaced lowercase, matrices are boldfaced uppercase, respectively, and  $\overline{(\cdot)}$  denote the expectation.

## II. LOW DENSITY PARITY-CHECK CODES

An LDPC code is a linear block code with a sparse parity-check matrix. Consider a low density parity check matrix  $H$  composed of  $M$  rows and  $N$  columns (the code rate is  $R = 1 - \frac{M}{N}$ ). This code can be represented with a Tanner graph consisting of  $N$  variable nodes,  $M$  check nodes, and a certain number of edges as shown in Fig. 1. Generally, an irregular LDPC code ensemble is specified by a degree distribution pair  $(\lambda, \rho)$  or its corresponding generating function as follows,

$$\begin{aligned}\Lambda(x) &= \sum_{i=2}^{l_{max}} \lambda_i x^{i-1} \\ \Omega(x) &= \sum_{i=2}^{r_{max}} \rho_i x^{i-1},\end{aligned}\quad (1)$$

where  $\lambda_i$  and  $\rho_i$  are the fractions of edges with left and right degree  $i$  and  $l_{max}$  and  $r_{max}$  are the maximal left and right degree of the edges, respectively. Also, the average left,  $\bar{l}$ , and right degree,  $\bar{r}$ , of the graph are [1],

$$\bar{l} = \frac{1}{\sum_i \frac{\lambda_i}{i}} = \frac{1}{\int_0^1 \Lambda(x) dx} \quad (2)$$

$$\bar{r} = \frac{1}{\sum_i \frac{\rho_i}{i}} = \frac{1}{\int_0^1 \Omega(x) dx}. \quad (3)$$

Therefore, the rate of the code is at least  $R = 1 - \frac{\bar{l}}{\bar{r}} = 1 - \frac{\int_0^1 \Omega(x) dx}{\int_0^1 \Lambda(x) dx}$ .

LDPC codes rely on the message passing algorithm for decoding. This algorithm can be interpreted in terms of passing messages throughout the Tanner graph as illustrated in Fig. 1. At the first iteration, the message passed from the  $n$ 'th variable node to each of its participating check nodes is the received LLR (initial message) of this variable node. The  $m$ 'th check node collects its incoming LLR messages (variable node messages), and passes an outgoing message (check node messages) which represents the LLR for the parity of the bits involved in the  $m$ 'th check node. Each variable node receives the check node messages and produces a new estimate for the variable LLR messages.

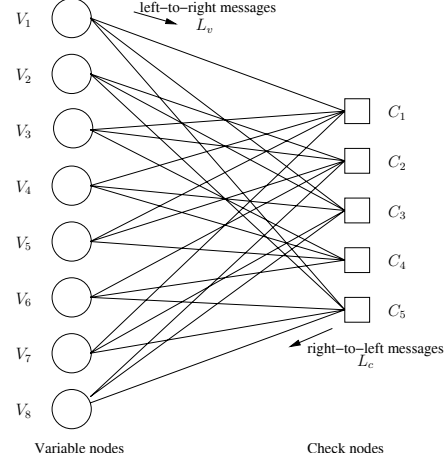


Fig. 1. Tanner Graph

## III. GAUSSIAN APPROXIMATION AND CONSISTENCY CONDITION

A distribution  $p$  defined on the set of real numbers  $R$  is called consistent if it satisfies  $p(-x) = e^{-x}p(x)$  for all  $x \geq 0$  [9]. In [9], it is shown that the consistency condition is preserved under DE. We consider an Additive White Gaussian Noise (AWGN) channel with input  $x \in \{-1, 1\}$ . We also assume that the bits zero and one are mapped to 1 and  $-1$ , respectively. As the code is linear, without loss of generality, one can assume that the all zero codeword is transmitted. For the received discrete-time signal, we have

$$y = x + z, \quad (4)$$

where  $z$  is a zero-mean Gaussian random variable with variance  $\sigma_z^2$ . The conditional PDF of the received signal is,

$$p(y|x) = \frac{1}{\sqrt{2\pi\sigma_z^2}} e^{-\frac{(y-x)^2}{2\sigma_z^2}}. \quad (5)$$

The corresponding initial LLR of each received bit is calculated as,

$$L_m(y) = \ln \left( \frac{p(y|x=1)}{p(y|x=-1)} \right). \quad (6)$$

We have,

$$L_m(y) = \frac{2}{\sigma_z^2} y = \frac{2}{\sigma_z^2} (x + z), \quad (7)$$

with the conditional PDF as follows,

$$p_{L_m}(l_m|x=1) = \frac{1}{\sqrt{4\pi\mu_0}} e^{-\frac{(l_m-\mu_0)^2}{4\mu_0}}, \quad (8)$$

where  $\mu_0 = \frac{2}{\sigma_z^2}$ .  $p_{L_m}(l_m|x=1)$  has a Gaussian distribution in the form of  $N(\mu_0, 2\mu_0)$  which is a consistent PDF. In the DE method, the distribution of the initial messages,  $p_{L_m}(l_m|x=1)$ , is applied as the input to the algorithm. At each iteration, this PDF is updated via message passing

algorithm. Throughout this paper, for the sake of simplicity in notation, the conditional term in PDF is dropped, i.e.,  $p_{L_m}(l_m)$  is used instead of  $p_{L_m}(l_m|x = 1)$  as the initial messages and  $p_{L_v}(l_v)$  and  $p_{L_c}(l_c)$  represent the conditional PDF of variable and check node messages, respectively.

Wiberg in his Ph.D. dissertation [15] observes that the PDF of the LLR of the message bits is approximately Gaussian. With this idea, Chung has studied the convergence behavior of the sum-product decoding algorithm by relying on the Gaussian approximation [10], showing that this approximation is accurate at Variable nodes, but not at the check nodes. The method proposed by Chung has limitations in terms of the accuracy in handling low coding rates, and/or variable nodes with high degrees. Specifically, Chung in [10], designs good LDPC codes for rates in the range of 0.5 to 0.9, with a maximum variable node degree of 10. In [2], it is shown that relying on the Gaussian approximation at the output of the check nodes has two drawbacks. First, it is accurate only in the high SNR region when the check nodes have small degrees. In other words, the Gaussian approximation is not accurate in the analysis of low rate codes. Second, Gaussian approximation is inaccurate for large variable node degrees. The authors in [2] propose a semi-Gaussian approximation method which is based on applying DDE (introduced in [3]) at both check nodes and variable nodes. This is simplified by fitting a Gaussian (or Gaussian mixture) model to the output PDF of the variables nodes. The parameters of this model are selected such that the error probabilities of the message bits are equal to their true values (corresponding to the PDF computed by DDE). The complexity of the algorithm proposed in [2] is determined by the DDE for each point of the EXIT chart. The extensive computational complexity limits the application of the methods proposed in [2] and [3]. The primary motivation behind the current article is to reduce this computational complexity, while providing an accurate estimate of the EXIT chart.

#### IV. PROPOSED DE METHOD FOR ASYMPTOTIC ANALYSIS OF THE LDPC CODES

Gram-Charlier series is commonly used for expanding a nearly Gaussian distribution function over an orthonormal basis [16], [17]. A distribution,  $p_Y(y)$ , can be expanded as follows:

$$p_Y(y) \simeq \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y-\beta)^2}{2\sigma^2}} \sum_{i=0}^{\infty} \alpha_i T_i\left(\frac{y-\beta}{\sigma}\right), \quad (9)$$

where,

$$T_i(y) = \sum_{j=0}^{\lfloor \frac{i}{2} \rfloor} \gamma(i, j) y^{i-2j}, \quad (10)$$

is the Hermite Polynomial [17] of order  $i$ .

As mentioned earlier, the message distribution at the variable nodes is nearly Gaussian and consistent. The consistency

condition makes it difficult to use the Gram-Charlier series. On the other hand, it can be shown that a consistent PDF can be decomposed as the product of another consistent PDF and an even function. By eliminating the odd terms in (9), the following consistent PDF (with  $N$  degrees of freedom) is used for the message distribution at the variable nodes:

$$p_{L_v}(l_v) = \frac{1}{\sqrt{4\pi\beta}} e^{-\frac{(l_v-\beta)^2}{4\beta}} \sum_{k=0}^{N-1} a_k l_v^{2k}. \quad (11)$$

As we will see later, this model lends itself to simplified computations to find the underlying parameters. Let us assume that the first  $N$  moments of the random variable  $L_v$ ,  $\nu_i$  for  $i = 1, \dots, N$ , are given. We follow the same approach in [13] to compute the corresponding  $N + 1$  unknown parameters at each iteration by relying on the Moment Matching Method. This method exploits the Newton-Raphson algorithm for matching the first  $N$  statistical moments of the underlying PDF and the measured moments.

At each variable node, the random variable  $L_v$  (the variable node message) is related to the initial message ( $L_m$ ) and the check node ( $L_c$ ) messages as follows:

$$l_v = l_m + \sum_{k=1}^{i-1} l_{c_k}. \quad (12)$$

Since the number of incoming check node messages,  $i$ , is another random variable with probability mass function  $\Lambda(x)$  as defined in (1), the Moment Generating Function (MGF) of the random variable  $L_v$  can be computed as:

$$\begin{aligned} G_v(s) = \overline{e^{sl_v}} &= \overline{e^{sl_m}} \sum_{i=2}^{l_{max}} \lambda_i (\overline{e^{sl_c}})^{i-1} \\ &= G_m(s) \Lambda(G_c(s)), \end{aligned} \quad (13)$$

where  $G_m(s)$  and  $G_c(s)$  are the MGF of the initial messages and the MGF of the check node messages, respectively, and  $\lambda_i$  is defined in (1). As a result

$$\begin{aligned} \nu_n &= \left. \frac{d^n (G_m(s) \Lambda(G_c(s)))}{ds^n} \right|_{s=0} \\ &= \sum_{k=0}^n \sum_{i=2}^{l_{max}} \binom{n}{k} \mu_k \lambda_i \left. \frac{d^{n-k} G_c^{i-1}(s)}{ds^{n-k}} \right|_{s=0}, \end{aligned} \quad (14)$$

where  $\mu_k$  is the  $k$ 'th moment of the initial Gaussian messages, and hence, can be computed by replacing  $m = \frac{2}{\sigma^2}$  in

$$\mu_k = \left. \frac{d^k}{ds^k} e^{(s+\sigma^2)m} \right|_{s=0}, \quad (15)$$

where the right hand side of (15) is the  $k$ 'th derivative of MGF for a Gaussian random variable with PDF,  $\mathcal{N}(m, 2m)$ . Also,  $\left. \frac{d^k G_c^n(s)}{ds^k} \right|_{s=0}$  can be easily computed with respect to the moments of the check node message distributions, i.e.,  $\omega_i$  for  $i = 1 \dots n$ . Numerical simulations show that only

the first four terms are needed to obtain an acceptable approximation and,

$$\begin{aligned}
\frac{dG_c^n(s)}{ds}\Big|_{s=0} &= n\omega_1 \\
\frac{d^2G_c^n(s)}{ds^2}\Big|_{s=0} &= n\omega_2 + n(n-1)\omega_1^2 \\
\frac{d^3G_c^n(s)}{ds^3}\Big|_{s=0} &= n\omega_3 + n(n-1)(3\omega_1\omega_2) \\
&\quad + n(n-1)(n-2)\omega_1^3 \\
\frac{d^4G_c^n(s)}{ds^4}\Big|_{s=0} &= n\omega_4 + n(n-1)(4\omega_1\omega_3 + 3\omega_2^2) \\
&\quad + n(n-1)(n-2)(4\omega_1^2\omega_2) \\
&\quad + n(n-1)(n-2)(n-3)\omega_1^4. \quad (16)
\end{aligned}$$

As noted earlier,  $\omega_n$  is the  $n$ 'th moment of the random variable  $L_c$  which is the LLR of the check node messages. For a check node of degree  $k$ , this random variable is determined using the so-called ‘‘tanh rule’’ [1]:

$$\tanh\left(\frac{l_c^{(l)}}{2}\right) = \prod_{i=1}^{k-1} \tanh\left(\frac{l_{v_i}^{(l-1)}}{2}\right), \quad (17)$$

where  $l_{v_1}^{(l-1)}, l_{v_2}^{(l-1)}, \dots, l_{v_{k-1}}^{(l-1)}$  are variable node messages after  $(l-1)$ 'th iteration, each with MGF  $G_v(s)$ . In what follows, for the sake of simplicity, the superscript  $l$  is dropped.

To compute  $\omega_n$ , we use the following two operations: First, we compute  $\omega_n$  from the moments of the random variable  $u$  which is related to  $l_c$  using the following formula,

$$\begin{aligned}
\tanh\left(\frac{l_c}{2}\right) = u &\Rightarrow l_c^n = f_n(u) \\
&= \left(\ln \frac{1+u}{1-u}\right)^n. \quad (18)
\end{aligned}$$

Second, we compute the moments of  $u$  from the moments of variable node messages in the previous iteration. For the first step, we use the Taylor series expansion of the function  $l_c^n = f_n(u)$ , expanded around the mean of  $u$  as given in (18)

$$\begin{aligned}
l_c^n &= f_n(\bar{u}) + (u - \bar{u})f_n'(u) \\
&\quad + (u - \bar{u})^2 \frac{f_n''(u)}{2!} + \dots \quad (19)
\end{aligned}$$

Taking expectation and noting that the right degree distribution of check nodes are as (1), we obtain,

$$\omega_n = E[l_c^n] \simeq \sum_{i=2}^{r_{max}} \rho_i \left( f_n(\bar{u}_i) + \frac{1}{2} f_n''(\bar{u}_i) \sigma_{u_i}^2 \right), \quad (20)$$

To solve (20), we need to determine  $\bar{u}_i$  and  $\sigma_{u_i}$ . At the second step, these parameters are computed as follows:

$$\bar{u}_i = \left( \overline{\tanh\left(\frac{l_v}{2}\right)} \right)^{i-1}, \quad (21)$$

$$\sigma_{u_i}^2 = \left( \overline{\tanh^2\left(\frac{l_v}{2}\right)} \right)^{i-1} - \bar{u}_i^2. \quad (22)$$

By using (11),  $\overline{\tanh\left(\frac{l_v}{2}\right)}$  would be

$$\begin{aligned}
\overline{\tanh\left(\frac{l_v}{2}\right)} &= \int_{-\infty}^{\infty} \tanh\left(\frac{l_v}{2}\right) p_{L_v}(l_v) dl_v \\
&= \sum_{i=0}^{N-1} a_i \int_{-\infty}^{\infty} l_v^{2i} \tanh\left(\frac{l_v}{2}\right) \frac{1}{\sqrt{4\pi\beta}} e^{-\frac{(l_v-\beta)^2}{4\beta}} dl_v. \\
&= \sum_{i=0}^{N-1} a_i h_i(\beta), \quad (23)
\end{aligned}$$

where,

$$\begin{aligned}
h_i(\beta) &= \int_{-\infty}^{\infty} l_v^{2i} \tanh\left(\frac{l_v}{2}\right) \frac{1}{\sqrt{4\pi\beta}} e^{-\frac{(l_v-\beta)^2}{4\beta}} dl_v \\
&= \int_{-\infty}^{\infty} e^{-x^2} \frac{1}{\sqrt{\pi}} (\beta + 2\sqrt{\beta}x)^{2i} \tanh(\sqrt{\beta}x + \frac{\beta}{2}) dx \\
&= \int_{-\infty}^{\infty} e^{-x^2} \varphi(x) dx. \quad (24)
\end{aligned}$$

$$\varphi(x) = \frac{1}{\sqrt{\pi}} (\beta + 2\sqrt{\beta}x)^{2i} \tanh(\sqrt{\beta}x + \frac{\beta}{2}) dx. \quad (25)$$

Using Gauss-Hermite quadrature method [14], we have,

$$\int_{-\infty}^{\infty} e^{-x^2} \varphi(x) dx = \sum_{i=1}^q \eta_i \varphi(x_i), \quad (26)$$

where  $\eta_i = \frac{2^{q-1} q! \sqrt{\pi}}{q^2 [H_{q-1}(x_i)]^2}$ ,  $x_i$ 's are the roots of the Hermite polynomial of degree  $q$ ,  $H_q(x)$ , defined as,

$$H_q(x) = (-1)^q e^{x^2} \frac{d^q}{dx^q} (e^{-x^2}). \quad (27)$$

The corresponding approximation error,  $\zeta$ , is

$$\zeta = \frac{q! \sqrt{\pi}}{2^q (2q)!} \varphi^{2q}(\xi), \quad 0 < \xi < 1. \quad (28)$$

As a result, the right hand side of (26) converges to the left hand side as  $q \rightarrow \infty$ , if  $\varphi^{2q}(\xi)$  has a limit. One can easily show that  $\varphi(x)$ , defined in (25), has this property. As a result, (24) is approximately equal to,

$$h_i(\beta) \simeq \sum_{i=1}^q \eta_i \varphi(x_i). \quad (29)$$

One can compute  $\overline{\tanh^2\left(\frac{l_v}{2}\right)}$  in the same way as  $\overline{\tanh\left(\frac{l_v}{2}\right)}$ .

Now, we can compute the MEP at each iteration as follows:

$$\begin{aligned}
p_e &= \int_{-\infty}^0 p_{L_v}(l_v) dl_v \\
&= \int_{-\infty}^0 \frac{1}{\sqrt{4\pi\beta}} e^{-\frac{(x-\beta)^2}{4\beta}} \sum_{i=0}^{N-1} a_i x^{2i} dx \\
&= \sum_{i=0}^{N-1} a_i \int_{\frac{\beta}{2}}^{\infty} (\sqrt{2\beta}x - \beta)^{2i} \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} dx \\
&= \sum_{i=0}^{N-1} a_i T_{2i}(\beta), \tag{30}
\end{aligned}$$

where  $T_i(\beta)$  obeys the following recursion:

$$\begin{aligned}
T_i(\beta) &= \int_{\frac{\beta}{2}}^{\infty} (\sqrt{2\beta}x - \beta)^i \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} dx \\
&= \sqrt{2\beta} \int_{\frac{\beta}{2}}^{\infty} (\sqrt{2\beta}x - \beta)^{i-1} x e^{-\frac{x^2}{2}} dx \\
&\quad - \beta T_{i-1}(\beta) \\
&= 2\beta(i-1)T_{i-2}(\beta) - \beta T_{i-1}(\beta), \tag{31}
\end{aligned}$$

and

$$\begin{aligned}
T_0(\beta) &= Q\left(\sqrt{\frac{\beta}{2}}\right) \\
T_1(\beta) &= \sqrt{2\beta}e^{-\frac{\beta}{4}} - \beta Q\left(\sqrt{\frac{\beta}{2}}\right), \tag{32}
\end{aligned}$$

Noting above, the MEP in (30) can be computed using (31) and (32). Equation (30) results in a closed-form expression for computing MEP at each iterations. The algorithm can be summarized in the following steps:

- 1) Compute the initial messages from the received signal and set  $\omega_i = 0$  for  $i = 1, \dots, N$ , and  $K = \text{Maximum Number of Iterations}$ .
- 2) Compute the moments  $\nu_i$  for  $i = 1, \dots, N$  using (14), (16).
- 3) Update the unknown parameters  $[\beta, \mathbf{a}]$  in (11) using the Newton-Raphson Algorithm.
- 4) Compute the MEP using (30) and (31).
- 5) Compute  $\tanh\left(\frac{L_v}{2}\right)$  and  $\tanh^2\left(\frac{L_v}{2}\right)$  for  $i = 1, \dots, N$  using (23) and (24).
- 6) Compute  $\bar{u}_i$  and  $\sigma_{u_i}$  using (21), (22).
- 7) Compute  $\omega_i$  for  $i = 1, \dots, N$  using (20).
- 8) If  $K > 0$ ,  $K = K - 1$  and go to the step 2, unless otherwise stop.

## V. SIMULATION RESULTS

In this section, we provide numerical results to demonstrate the accuracy of the proposed method as compared to the Gaussian approximation [10] and DDE [3]. In [3], it is shown that the 11-bit quantization of messages in DDE provides an accurate probability of error. However, to have a more accurate result, we have used 13-bit quantization of

messages in DDE method to compare with the proposed method. The EXIT chart based on Message Error Rate reported in [2] is used in our simulation. Figs. 2 and 4 show the EXIT chart for (3,6) regular LDPC code with  $\Lambda(x) = x^2, \Omega(x) = x^5, R = 1/2$  and irregular LDPC code<sup>1</sup> with  $\Lambda(x) = .267818x + .204657x^2 + .0774591x^5 + .204181x^7 + .245886x^9, \Omega(x) = x^5, R = 1/3$ , respectively. Table I shows the error in approximation of threshold for both the regular and irregular codes using the Gaussian approximation and the proposed method with respect to the DDE. Figure 3 shows that the proposed method outperforms the Gaussian approximation. Figures 4 and 5 show that there is a gap between the EXIT chart of the proposed method as compared to DDE, while the proposed method still outperforms the Gaussian approximation. Unlike the Gaussian approximation, the proposed method is not sensitive to the variable node degrees which makes it suitable for analyzing capacity approaching codes.

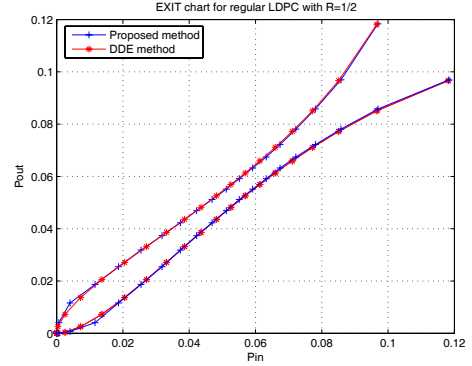


Fig. 2. Convergence behavior of Regular LDPC code,  $R = 1/2$

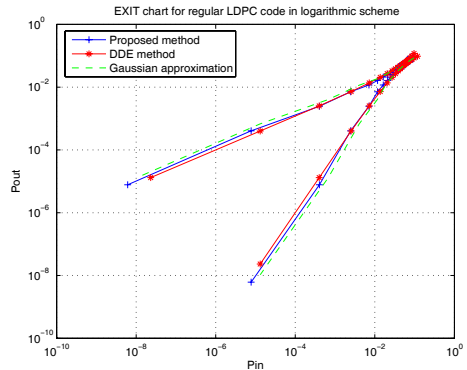


Fig. 3. Convergence behavior of Regular LDPC code in Log Scheme,  $R = 1/2$

<sup>1</sup>this code approaches the Shannon limit within .079dB

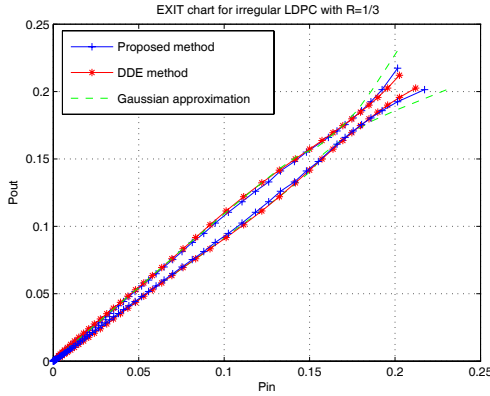


Fig. 4. Convergence behavior of Irregular LDPC code,  $R = 1/3$

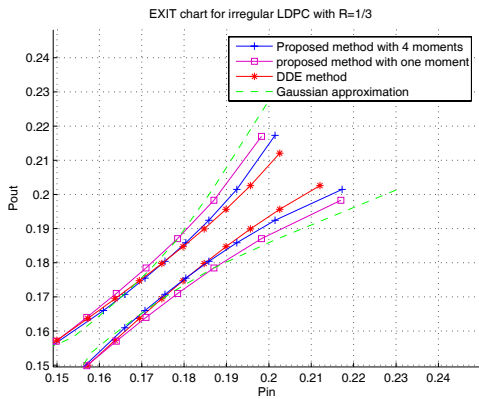


Fig. 5. Convergence behavior of Irregular LDPC code for 1 and 4 moments,  $R = 1/3$

## VI. CONCLUSIONS

An analytical low complex DE method with acceptable accuracy is proposed. We provide a better model for the distribution of message bits through matching the first  $n$  statistical moments. These moments are update using an iterative message passing algorithm based on the graphical representation of the underlying code. Numerical results show that the performance of the proposed method is close to that of the DDE method, while it offers a lower complexity. For low SNR values and high check node degrees, the PDF of check node messages tends to have a narrow peak [3], and thus, one needs more moments (Higher Order Statistics)

TABLE I  
THE ERROR IN APPROXIMATION OF THRESHOLD

| LDPC code            | Error of Gaussian Approximation | Error of proposed method with 4 moments |
|----------------------|---------------------------------|---|
| regular( $R=1/2$ )   | .06 dB                          | .02 dB                                  |
| irregular( $R=1/3$ ) | .25 dB                          | .082 dB                                 |

to get the better estimation of the final PDF. The number of required moments depends on the SNR and the maximum check node degree.

## REFERENCES

- [1] Thomas J. Richardson, M. Amin Shokrollahi and Rüdiger L. Urbanke , "Design of capacity-approaching irregular low-density parity-check codes," *IEEE Trans. Inform. Theory*, vol. 47, February 2001.
- [2] Masoud Ardakani and Frank R. Kschischang, "A more accurate one-dimensional analysis and design of irregular ldpc codes," *IEEE Trans on Communications*, vol. 52, pp. 2106–2114, December 2004.
- [3] S.-Y. Chung, G. D. Forney, Jr., T. J. Richardson, and R. Urbanke , "On the design of low-density parity-check codes within 0.0045 db of the shannon limit," *IEEE Communications Letters*, vol. 5, pp. 58–60, February 2001.
- [4] R.G. Gallager, *Low-Density Parity-Check Codes*, MIT Press, 1963.
- [5] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near shannon limit error correcting coding and decoding: Turbo-codes," *In Proc. IEEE Int. Conf. Commun. Geneva, Switzerland*, pp. 1064–1070, May 1993.
- [6] D. J. C. MacKay and R. M. Neal, "Near shannon limit performance of low density parity check codes," *IEEE Electronics Letters*, vol. 32, no. 18, pp. 1645–1646, Aug 1996.
- [7] M. Sipser and D. A. Spielman, "Expander codes," *IEEE Trans. Inform.Theory*, vol. 42, no. 6, pp. 1710–1722, Nov 1996.
- [8] M. G. Luby, M. Mitzenmacher, M. A. Shokrollahi, and D. A. Spielman , "Improved low-density parity-check codes using irregular graphs," *IEEE Trans. Inform.Theory*, vol. 47, no. 2, pp. 585–598, Feb 2001.
- [9] T. J. Richardson and R. L. Urbanke, "The capacity of low-density parity-check codes under message passing decoding," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 599–618, Feb 2001.
- [10] S.-Y. Chung, T. J. Richardson, and R. L. Urbanke , "Analysis of sum-product decoding of low-density parity-check codes using a gaussian approximation," *IEEE Trans. Inform.Theory*, vol. 47, no. 2, pp. 657–670, Feb 2001.
- [11] F. Lehmann and G. M. Maggio, "An approximate analytical model of the message passing decoder of ldpc codes," in *Proc. IEEE International Symposium on Information Theory,Lausanne, Switzerland*, July 2002.
- [12] S ten. Brink , "Design of low-density parity-check codes for modulation and detection," *IEEE Trans. Commun.*, vol. 52, no. 670-678, April 2004.
- [13] Ali Abedi and Amir K. Khandani, "A new method for performance evaluation of bit decoding algorithms using statistics of the log likelihood ratio," *Technical report is available at: www.cst.uwaterloo.ca.*, vol. 103, no. 1, pp. 193–205, May 1998.
- [14] Holtzman, J.M. , "On using perturbation analysis to do sensitivity analysis: derivatives vs. differences," *IEEE Trans. Inform. Theory*, vol. 3, pp. 2018–2023, December 1989.
- [15] N.Wiberg, *Codes and decoding on general graphs*, Ph.D. dissertation, Linköping University, Sweden, 1996.
- [16] Ali Abedi and Amir. K. Khandani, "An analytical method for approximate performance evaluation of binary linear block codes," *IEEE Trans. Commun.*, vol. 52, pp. 228–235, Feb 2004.
- [17] S. Blinnikov and R. Moessner,, "Expansions for nearly gaussian distributions," *Astron. Astrophys. Supplement Ser.*, vol. 103, no. 1, pp. 193–205, May 1998.