Analysis of Low-Density Parity-Check codes based on EXIT functions

Eran Sharon, Alexei Ashikhmin, *Senior Member, IEEE,* Simon Litsyn, *Senior Member, IEEE*

***Abstract*— We exploit EXIT functions of single parity-check and repetition codes over the binary input AWGN (biAWGN) channel, derived in [16], for asymptotic performance analysis of belief propagation decoding of LDPC codes. The approach is based on a Gaussian Approximation (GA) of the Density Evolution (DE) algorithm using the mutual information measure. We show that our method allows more accurate prediction of the decoding threshold in the biAWGN channel than the earlier known GA methods.**

***Index Terms*— Iterative decoding, LDPC codes, performance analysis, EXIT functions.**

I. Introduction

Low-density parity-check (LDPC) codes have been the subject of much interest in recent years due to their excellent performance close to the Shannon limit, and low decoding complexity. Performance of LDPC codes exhibits a threshold phenomenon under iterative decoding - as the codeword length tends to infinity, the bit error probability becomes arbitrarily small if the noise level is below some constant threshold. Richardson *et al* [13],[14] developed the **Density Evolution** (DE) algorithm for determining the convergence threshold of message passing decoders over various channels by tracking the densities of messages passed in the decoding process.

The DE algorithm can be used for designing capacity approaching LDPC code ensembles [5]. However, due to its computational complexity, Gaussian Approximation (GA) algorithms for DE over the biAWGN channel were suggested. These algorithms convert the infinite-dimensional problem of iterative calculation of the message densities to a one-dimensional problem of tracking a single parameter of the Gaussian density. Several methods for the GA were investigated, differing in the way a non-Gaussian message is replaced by a Gaussian one. A method based on the mean value of the messages was suggested in [6]. A method based on the signal-to-noise (SNR) ratio was used in analysis of turbo codes in [9]. A method based on Bit Error Rate (BER) for analysis of LDPC and Product codes was proposed in [10]. A method based on the mutual information measure was suggested in [2],[3],[15]. This method uses EXIT functions of constituent decoders in the iterative coding scheme, that characterize how *a priori* information at the soft decoder input converts into extrinsic information at the soft decoder output. The exchange of extrinsic information between constituent decoders can be visualized as a decoding trajectory in an EXIT chart. EXIT charts provide a convenient tool for designing capacity approaching iterative coding systems by matching the EXIT functions of the constituent decoders based on the area property of the functions [1],[3],[4].

Since iterative decoding of LDPC codes can be seen as a series of elementary decodings of repetition and single parity-check constituent codes, analysis of LDPC codes depends on the EXIT functions of these codes. In previous research, the corresponding EXIT functions were determined by simulations or using as an approximation based on a duality property of EXIT functions over the Binary Erasure Channel (BEC) [1]. In [16], we have developed an analytical method for computing the EXIT functions for some block codes over biAWGN channel in the form of a series expansion with terms that are computed using the EXIT function of the same code over BEC. In this paper we use exact expressions for the EXIT functions of repetition and single-parity check codes for analysis of belief propagation decoding of regular and irregular LDPC codes over biAWGN. In particular, we generalize the EXIT charts method suggested in [2]. The new method gives a fast and most accurate approximation of the DE algorithm.

The paper is organized as follows. In Section II we provide a brief background on LDPC codes. In Section III we describe an algorithm corresponding to analysis of LDPC codes based on EXIT charts. In Section IV we describe an improved algorithm having better accuracy. In Section V we generalize the GA algorithms for analysis of irregular LDPC codes. We show that the GA algorithms based on the mutual information measure produce more accurate results than other GA algorithms that track parameters such as mean value, SNR or BER, especially for irregular LDPC codes.

II. Background

A regular (d_v, d_c)-LDPC code is a linear block code defined by a sparse parity-check matrix H, such that each column of H contains d_v 1's and each row in H contains d_c 1's. The code can also be represented by a bipartite graph, such that the left nodes (called variable node) correspond to the codeword bits and the right nodes (called check node) correspond to the parity-checks. Each variable node emanates d_v edges to the check nodes it participates in. Each check node emanates d_c edges to the variable nodes that participate in it. The designed rate of the code is given by $R = 1 - \frac{d_v}{d_c}$.

Regular LDPC codes can be generalized to irregular LDPC codes [11] that exhibit better performance under iterative decoding. An irregular LDPC code is represented by an irregular

S. Litsyn and E. Sharon are with Department of Electrical Engineering-Systems, Tel Aviv University, Ramat Aviv, 69978 Israel; e-mail: {litsyn, eransh}@eng.tau.ac.il

A. Ashikhmin is with Bell Laboratories, Lucent Technologies, 600 Mountain Avenue, Murray Hill, NJ 07974, USA; e-mail: aea@research.bell-labs.com

bipartite graph, where the degrees of left and right nodes vary. The ensemble of irregular LDPC codes is defined by the left and right degree distributions. Let $\lambda(x) = \sum_{i=2}^{d_v} \lambda_i x^{i-1}$ and $\rho(x) = \sum_{i=2}^{d_c} \rho_i x^{i-1}$ be the generating functions of the degree distributions for the variable and check nodes respectively, where λ_i and ρ_i are the fractions of edges belonging to degree-i variable and check node respectively, and d_v and d_c are the maximal left and right degrees respectively. Then, the designed rate of the code is given by $R = 1 - \frac{\int_0^1 \rho(x)dx}{\int_0^1 \lambda(x)dx}$. The degree distributions can be optimized in order to generate a capacity approaching code ensemble.

The most attractive feature of LDPC codes is their ability to achieve a significant fraction of the channel capacity using low complexity iterative message-passing decoding algorithms. These algorithms are based on the bipartite graph representation of the code. Decoding is initialized with estimates of the variable nodes based on channel observations. These estimates improve through iterative extrinsic message passing between variable and check nodes in the graph. An extrinsic message sent from a node on edge e is a function of all messages received by the node on edges other then e. In this work we analyze the Belief Propagation (BP) message passing algorithm [8],[12] over the biAWGN channel. Let $\underline{x}$ denote the BPSK modulated codeword $\underline{c}$, i.e. $x_j = (-1)^{c_j}$. Let $\underline{y}$ denote the output of a biAWGN channel. An estimate of a transmitted (modulated) bit X given an observation Y can be expressed in several ways, for example: 1) the bit's log likelihood ratio (LLR), $\log \frac{Pr(X=1|Y)}{Pr(X=-1|Y)}$. 2) the bit's expectation, also known as the "soft bit", $Pr(X=1|Y) - Pr(X=-1|Y)$.
The BP algorithm can be described in the LLR domain. Let $U_0 = \log \frac{Pr(x=1|y)}{Pr(x=-1|y)}$ denote an initial LLR channel message. For biAWGN channel, given that $x = 1$

$$U_0 = \frac{4}{No} y \sim \mathcal{N}(4\frac{E}{No}, 8\frac{E}{No}) = \mathcal{N}(m_{U_0}, 2m_{U_0}), \quad (1)$$

were E is the transmitted bit energy, $No/2$ is the one-sided spectral noise density of the biAWGN channel and m is the message's expectation. Let U and V denote the check-to-variable and the variable-to-check LLR messages respectively. We denote the "soft bit" message corresponding to the LLR message V by $T_V = \tanh(\frac{V}{2})$.

Under BP decoding, a degree d_v variable node implements the MAP decoding function of a $[d_v, 1]$ repetition code and the extrinsic variable-to-check message V is given by:

$$V = \sum_{i=0}^{d_v-1} U_i, \quad (2)$$

where U_i, $i = 1, \ldots, d_v - 1$, are the incoming messages from the neighbors of the variable node except the check node to which message V is sent and U_0 is the channel message associated with the variable node.

A degree d_c check node implements the MAP decoding function of a $[d_c, d_c - 1]$ parity-check code and the extrinsic check-to-variable message U is given by:

$$\tanh(U/2) = \prod_{j=1}^{d_c-1} \tanh(\frac{V_j}{2}) \Longleftrightarrow T_U = \prod_{j=1}^{d_c-1} T_{V_j} \quad (3)$$

where V_i, $i = 1, \ldots, d_c - 1$, are the incoming messages from the neighbors of the check node except the variable node to which the message U is sent.

Asymptotic performance analysis of LDPC codes is usually done assuming that the code's underlying bipartite graph is a tree-like graph. When the channel is memoryless, this assumption implies statistical independence of all incoming messages into a node in the graph. The tree assumption is justified by a general concentration theorem [13] showing that the decoder's performance on random graphs converges to its expected performance as the code length increases. This expected performance in the limit of infinitely long codes can be determined from the corresponding tree-like graph. Another assumption that simplifies analysis is that the all-zero codeword is transmitted. As shown in [13], due to symmetry conditions satisfied by the decoder its performance is independent of the transmitted codeword.

Asymptotic analysis of LDPC codes under BP decoding can be done using the DE algorithm by iteratively tracking message densities. LLR message densities satisfy a consistency condition under BP decoding, i.e. $f(x) = f(-x)e^x$ [14]. Utilizing this observation, in GA algorithms for DE, the messages' densities are assumed to be consistent Gaussian densities. For a Gaussian density, the consistency property is reduced to the condition $\sigma^2 = 2m$. Thus, only a single parameter of the Gaussian density should be tracked.

III. GA BASED ON THE MUTUAL INFORMATION MEASURE

Let us denote the means of U, U_0 and V by m_U, m_{U0} and m_V, respectively. Taking the expectation of both sides of equation (2) we get at the l'th iteration

$$m_V^{(l)} = m_{U_0} + (d_v - 1)m_U^{(l-1)}, \quad (4)$$

where $m_U^{(0)} = 0$ and m_{U_0} is determined according to (1). The all-zero codeword assumption guarantees identical means for all incoming U messages. The statistical independence of all incoming messages into the variable node guarantees that the consistency condition of V is maintained.

Deriving an update rule for $m_U^{(l)}$ is more tricky. Even if we assume that all incoming messages to a check node have consistent Gaussian densities and are independent, the output message of the check node U, which is given by (3) is not Gaussian due to the non-linearity of the check node computation rule (See Fig.1).

In order to keep tracking Gaussian messages, the output message of a check node U which is non-Gaussian should be replaced by a Gaussian message U_g. In the algorithm proposed in [6], the non-Gaussian message U is replaced by the Gaussian message U_g satisfying $\mathbf{E}[\tanh\frac{U_g}{2}] = \mathbf{E}[\tanh\frac{U}{2}]$. In the algorithm proposed in [10], the non-Gaussian message U is replaced by the Gaussian message U_g maintaining the same bit error rate. These schemes, although computationally simple, do not provide an accurate approximation. In this paper we are interested in methods that are based on tracking mutual information [2],[15]. Thus, we replace the non-Gaussian message U by the Gaussian message U_g which satisfies $I(X;U) = I(X;U_g)$, where X is the transmitted bit

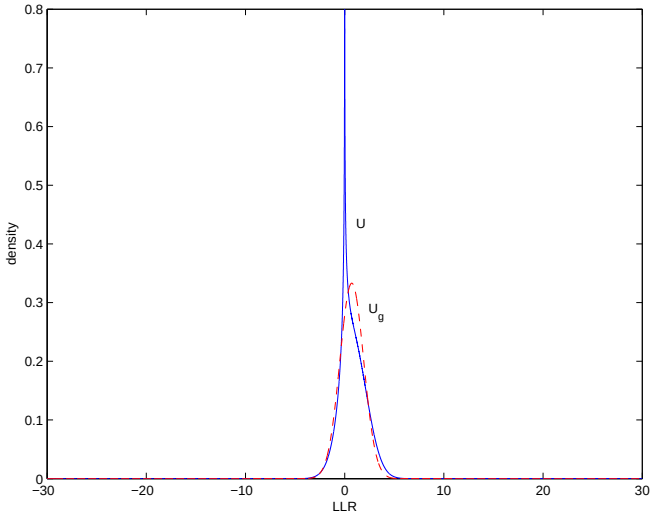


Fig. 1. Check node output message's pdf of a (3,6)-LDPC after the first iteration over a biAWGN channel ($\frac{Eb}{No} = 3dB$) - exact pdf (-) and the Gaussian approximation (- -)

corresponding to the message U. We use the following update rule for m_U:

$$m_U^{(l)} = J^{-1}(I(X;U^{(l)})), \tag{5}$$

where $J(m)$, defined by

$$J(m) = I(X;L) = \int_{\mathcal{R}} \frac{1}{\sqrt{4\pi m}} e^{-\frac{(l-m)^2}{4m}} (1 - \log_2(1+e^{-l}) dl, \tag{6}$$

is the mutual information $I(X;U_g)$ between a consistent Gaussian random variable $U_g|X = 1 \sim \mathcal{N}(m_U, 2m_U)$ and a binary random variable $X \sim \{1 \text{ w.p } 0.5, -1 \text{ w.p } 0.5\}$ as a function of m_U [1]. Now, it remains to show how to compute $I(X;U^{(l)})$ from $m_V^{(l)}$.

Proposition 3.1: $I(X;U) = I(X;T_U)$

Proof: $I(X;U) = I(X;\tanh\frac{U}{2}) = I(X;T_U)$. Here the first equality is due to the invertibility of the tanh function and the *data processing inequality* [7]. □

Since the check nodes implement MAP decoding of single parity-check codes, and since under the tree assumption and the symmetry conditions maintained by the BP decoder [13] the "soft bit" inputs to the check node can be considered as originating from a symmetric, memoryless channel, we can use Theorem 4.5 from [16] to compute

$$I(X;U) = \frac{1}{\ln 2} \sum_{i=1}^{\infty} \frac{1}{(2i)(2i-1)} [\mathbf{E}(T_V^{2i})]^{d_c - 1}, \tag{7}$$

By assuming that the check node input messages are Gaussian - $\mathcal{N}(m_V, 2m_V)$, we can compute $\mathbf{E}(T_V^{2i})$ using the following function [16]:

$$\Phi_i(m) \doteq \mathbf{E}(T^{2i}) = \int_{-1}^{+1} \frac{2t^{2i}}{(1-t^2)\sqrt{4\pi m}} e^{-\frac{(\ln\frac{1+t}{1-t} - m)^2}{4m}} dt. \tag{8}$$

Together, (5), (7) and (8) define the update rule for $m_U^{(l)}$ from $m_V^{(l)}$. To summarize, we obtain the following update rules for m_V and m_U:

$$m_V^{(l)} = m_{U_0} + (d_v - 1) m_U^{(l-1)} \tag{9}$$

$$m_U^{(l)} = J^{-1}\left(\frac{1}{\ln 2} \sum_{i=1}^{\infty} \frac{1}{(2i-1)(2i)} [\Phi_i(m_V^{(l)})]^{d_c - 1}\right) \tag{10}$$

Using a finite number of terms in the series of (7) we can approximate $I(X;U)$ as accurate as we wish.

Remark: Expression (7) can also be written in the form of a sign alternating series $\frac{1}{\ln 2} \sum_{i=1}^{\infty} (-1)^{i+1} \frac{1}{i} \mathbf{E}(T_V^i)$, thus as explained in [16] its convergence is assured by the *Leibnitz test*. The convergence rate of the series depends on the actual probability density function of T. However, since the series is sign alternating, upper and lower bounds on $I(X;U)$ can be obtained by truncating the series appropriately. In general, the convergence of the series becomes slower as $I(X;U)$ increases. Fig. 2 shows an example of the series convergence rate for a length 6 single parity-check code.

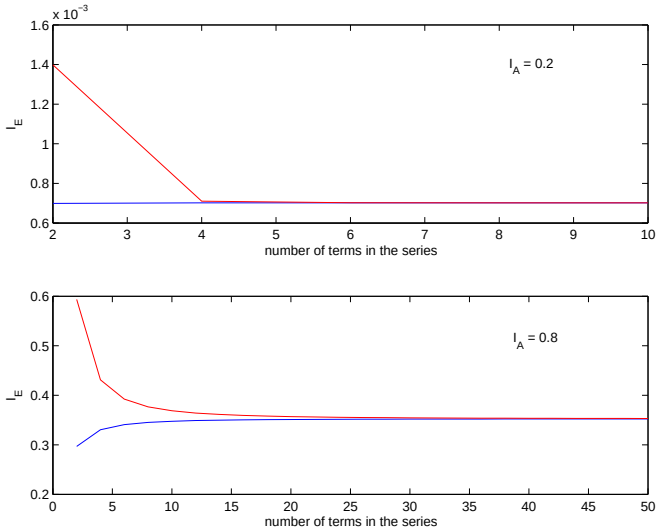


Fig. 2. Upper and lower bounds on $I(X;U)$ for a length 6 single parity-check code as a function of the number of terms in series of (7), for *a-priori* information of $I(X;V) = 0.2$ and $I(X;V) = 0.8$.

The functions $J(m)$ and $\Phi_i(m), i = 1, \ldots \infty$, can be computed using a preprocessed table, or by an approximation of the functions. Thus, the complexity of the algorithm remains constant per iteration.

The described algorithm is equivalent to the EXIT charts analysis of LDPC codes, and the two update rules actually describe the decoding trajectory that can be visualized on an EXIT chart with the following two EXIT functions:

Variable node EXIT function:

$$\begin{aligned} I_{E_v} &= I(X;V) = I(X;U_0, U_1, \ldots, U_{d_v - 1}) \\ &= f(I(X;U_0), I(X;U)) = f(I_{ch}, I_{A_v}) \\ &= J(J^{-1}(I_{ch}) + (d_v - 1) J^{-1}(I_{A_v})) \end{aligned} \tag{11}$$

where, I_{E_v} is the average extrinsic information at the output of the variable node decoder, $I_{ch} = I(X;U_0)$ is the average channel information and $I_{A_v} = I(X;U)$ is the average *a priori* information at the input of the variable node decoder.

Check node EXIT function:

$$\begin{aligned} I_{E_c} &= I(X;U) = I(X;V_1,\ldots,V_{d_c-1}) = f(I(X;V)) \\ &= f(I_{A_c}) = \frac{1}{\ln 2}\sum_{i=1}^{\infty}\frac{1}{(2i-1)(2i)}[\Phi_i(J^{-1}(I_{A_c})]^{d_c-1} \end{aligned} \quad (12)$$

where, I_{E_c} is the average extrinsic information at the output of the check node decoder and $I_{A_c} = I(X;V)$ is the average *a priori* information at the input of the check node decoder.

Since the variable node's *a priori* information is the check node's extrinsic information, and vice versa, we have: $I_{E_v} = I_{A_c}$ and $I_{E_c} = I_{A_v}$.

Fig. 3 shows an example of an EXIT chart for a $(3,6)$-regular LDPC code over a biAWGN channel. The asymptotic decoding trajectory is shown on the EXIT chart.

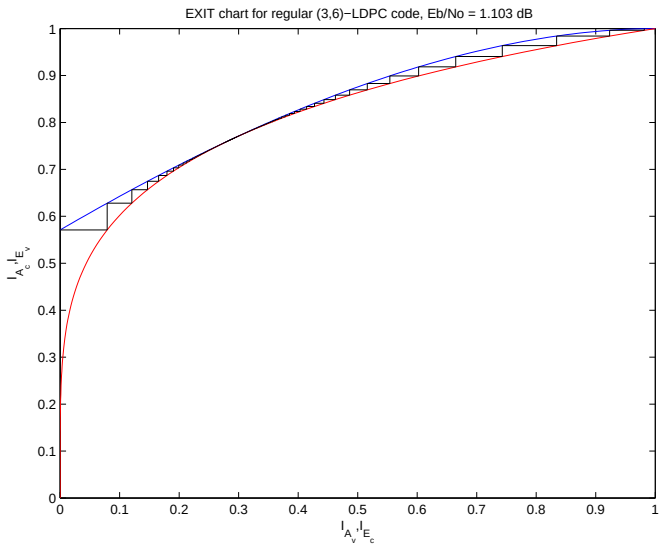


Fig. 3. EXIT chart for a (3,6)-LDPC code over a biAWGN channel with $E_b/N_o = 1.103dB$ generated using (11) and (12).

IV. GA BASED ON THE MUTUAL INFORMATION MEASURE - IMPROVED ALGORITHM

To improve the accuracy of the algorithm we can compute the expected mutual information of a variable-to-check message at the l'th iteration $I(X;V^l)$ directly from its expected mutual information in the previous iteration $I(X;V^{l-1})$. We then replace V^l by a Gaussian message V_g^l which satisfies $I(X;V_g^l) = I(X;V^l)$. This way we make a Gaussian assumption only on the variable-to-check message V without making a Gaussian assumption on the check-to-variable message U. This provides a better GA since unlike a check-to-variable message pdf, the variable-to-check message pdf is very close to Gaussian.

Proposition 4.1:

$$\begin{aligned} I(X;V) &= I(X;U_0,\ldots,U_{d_v-1}) \\ &= [\sum_{i=0}^{d_v-1} I(X;U_i)] - [\sum_{i=1}^{d_v-1} I(U_i;U_0,\ldots,U_{i-1})]. \end{aligned}$$

Proof: Under the tree assumption, the variable node implements MAP decoding, hence

$$I(X;V = U_0+U_1+\ldots+U_{d_v-1}) = I(X;U_0,U_1,\ldots,U_{d_v-1}).$$

If U_a and U_b given X are independent random variables then

$$\begin{aligned} I(X;U_a|U_b) &= H(U_a|U_b) - H(U_a|X,U_b) \\ &= H(U_a|U_b) - H(U_a|X) \\ &= H(U_a) - H(U_a|X) - [H(U_a) - H(U_a|U_b)] \\ &= I(X;U_a) - I(U_a;U_b) \end{aligned}$$

Using this expression and recursion, we obtain

$$\begin{aligned} & I(X;U_0,\ldots,U_{d_v-1}) \\ = \; & I(X;U_0) + I(X;U_1|U_0) + I(X;U_2|U_1,U_0) + \ldots \\ = \; & [\sum_{i=0}^{d_v-1} I(X;U_i)] - [\sum_{i=1}^{d_v-1} I(U_i;U_0,\ldots,U_{i-1})]. \end{aligned}$$

□

The term $I(X;V)$ is hard to compute in general (due to the term $\sum_{i=1}^{d_v-1} I(U_i;U_0,\ldots,U_{i-1})$). However, for $d_v = 2$ the variable node performs MAP decoding of the length 3 repetition code and $I(X;V)$ remains simple to compute. This is not surprising since for the length 3 repetition code the pMAP decoder defined in [16] and the MAP decoder coincide (see Example 3 in [16]), producing the following decoding rule for the variable node:

$$V = U_0 + U_1 \quad \Leftrightarrow \quad 1 + T_V = \frac{(1+T_{U_0})(1+T_{U_1})}{1+T_{U_0}T_{U_1}} \quad (13)$$

Under the tree assumption and the symmetry conditions maintained by the BP decoder the "soft bit" inputs to the variable node can be considered as originating from a symmetric, memoryless channel. Thus, applying Proposition 4.3 and Theorem 4.4 from [16] for a length 3 repetition code we get:

$$\begin{aligned} & I(X;V) = I(X;U_0,U_1) \\ & = \frac{1}{\ln 2}\sum_{i=1}^{\infty}\frac{1}{(2i)(2i-1)}[E(T_{U_0}^{2i}) + E(T_{U_1}^{2i}) - E(T_{U_0}^{2i})E(T_{U_1}^{2i})] \end{aligned} \quad (14)$$

Note that (14) and Proposition 4.1 imply that

$$I(U_0;U_1) = \frac{1}{\ln 2}\sum_{i=1}^{\infty}\frac{1}{(2i)(2i-1)}E(T_{U_0}^{2i})E(T_{U_1}^{2i}) \quad (15)$$

Taking the expectation of both sides of (3) and noting that the check node input messages are i.i.d (due to the tree and zero codeword assumptions) we have:

$$\mathbf{E}(T_{U_1}^{2i}) = \mathbf{E}((\prod_{j=1}^{d_c-1} T_{V_j})^{2i}) = \mathbf{E}(\prod_{j=1}^{d_c-1} T_{V_j}^{2i}) = \prod_{j=1}^{d_c-1}\mathbf{E}(T_{V_j}^{2i}) = [\mathbf{E}(T_V^{2i})]^{d_c-1} \quad (16)$$

$\mathbf{E}(T_{U_0}^{2i})$ and $\mathbf{E}(T_V^{2i})$ can be computed using (8) with m_{U_0} and m_V respectively, and substituted into (14) to obtain

$$\begin{aligned} & I(X;U_0,U_1) = I(X;U_0) + I(X;U_1) - I(U_1;U_0) \\ & = \frac{1}{\ln 2}\sum_{i=1}^{\infty}\frac{1}{(2i-1)(2i)}[\Phi_i(m_{U_0}) + [\Phi_i(m_V)]^{d_c-1} - \\ & \quad -\Phi_i(m_{U_0})[\Phi_i(m_V)]^{d_c-1}] \end{aligned} \quad (17)$$

Using (17) we can obtain a good approximation for $I(X;V) = I(X;U_0 + \ldots + U_{d_v-1})$. This can be done by computing $I(X;V)$ recursively. Let $V_j = U_0 + \ldots + U_j$. Assume we know $I(X;V_{d_v-2}) = I(X;U_0 + \ldots + U_{d_v-2})$. Then, we can replace V_{d_v-2} by the Gaussian variable $V^g_{d_v-2}$ that satisfies $I(X;V^g_{d_v-2}) = I(X;V_{d_v-2})$, i.e. $V^g_{d_v-2}$ is a consistent Gaussian variable with expectation $m_{V_{d_v-2}} = J^{-1}(I(X;V_{d_v-2}))$. Then, we use (17) to compute $I(X;V_{d_v-1} = V_{d_v-2} + U_{d_v-1}) = I(X;V^g_{d_v-2}) + I(X;U_{d_v-1}) - I(V^g_{d_v-2};U_{d_v-1})$, where, $I(V^g_{d_v-2};U_{d_v-1})$ is computed using (17) with $V^g_{d_v-2}$ in the role of U_0. Another way to understand the recursive procedure described above is to think of a computation of (2) as if it were done in d_v-1 steps by the decoder, i.e.: $V = \sum_{i=0}^{d_v-1} U_i = (\cdots(U_0 + U_1) + U_2) + \cdots) + U_{d_v-1})$, and after each step of addition the resulting non-Gaussian variable is replaced by a Gaussian variable having the same mutual information with X. The non-Gaussian variables that are being replaced by Gaussian ones are very close to Gaussian as can be seen on Fig.4. Thus the GA is better than the one that was suggested in the previous section where we replaced U_i which is less similar to Gaussian by a Gaussian variable.

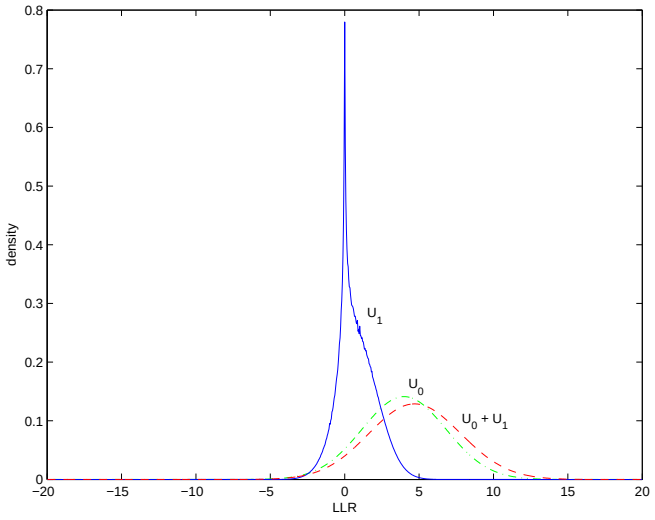


Fig. 4. The pdf of U_0 (-.),U_i (-) and $U_0 + U_i$ (–)

After computing $I(X;V^l)$ by the recursive procedure described above using m_V^{l-1} and m_{U_0}, we update m_V^l:

$$m_V^l = J^{-1}(I(X;V^l)), \tag{18}$$

The complexity of this GA algorithm is $O(d_v-1)$ per iteration. The improved mutual information tracking GA algorithms described in Sections III and IV were used for computing the channel noise standard deviation threshold values σ_{MI1} and σ_{MI2} respectively, for various regular (j,k)-LDPC codes over biAWGN channel and BP decoder. The threshold values were compared to the approximated threshold value σ_{MEAN} and σ_{BER} computed by the mean and BER tracking GA algorithms proposed in [6] and [10] respectively and to the exact threshold value σ_{exact} computed by the DE algorithm [5]. The results are summarized in Table I. The table also shows the inaccuracy Δ, measured in decibels, of the thresholds computed by the GA algorithms compared to the exact threshold computed by the DE algorithm.

V. GA based on the Mutual Information measure for irregular LDPC codes

Analysis tools for LDPC codes are of great importance for irregular LDPC codes, where they can be used for design of the left and right degree distributions ($\lambda(x)$ and $\rho(x)$) of capacity approaching irregular LDPC codes ensembles. We show here that the GA algorithm based on the mutual information measure produces very accurate results in predicting the performance and convergence of the decoding process for irregular codes as well. This is especially important since the mean and BER tracking GA algorithms proposed in [6],[10] become inaccurate for irregular LDPC codes.

The algorithm described in Section III can be generalized for irregular codes by assuming that the messages at the check node input are Gaussian messages with extrinsic information equal to the average extrinsic information of the messages emanating from the variable nodes. The same assumption is used for the messages at the input of a variable node. The generalized update rules for m_V and m_U are given by:

$$m_V^{(l)} = J^{-1}\left(\sum_{j=2}^{d_v} \lambda_j J\left(m_{U_0} + (j-1)m_U^{(l-1)}\right)\right) \tag{19}$$

$$m_U^{(l)} = J^{-1}\left(\sum_{j=2}^{d_c} \rho_j \frac{1}{\ln 2}\sum_{i=1}^{\infty}\frac{1}{(2i-1)(2i)}[\Phi_i(m_V^{(l)})]^{j-1}\right) \tag{20}$$

Again, this is equivalent to the EXIT charts analysis of irregular LDPC codes with the following EXIT functions:
Variable node EXIT function:

$$\begin{aligned} I_{E_v} &= f(I_{ch}, I_{A_v}) = \sum_{j=2}^{d_v} \lambda_j I_{E_{v_j}} \\ &= \sum_{j=2}^{d_v} \lambda_j J(J^{-1}(I_{ch}) + (j-1)J^{-1}(I_{A_v})) \end{aligned} \tag{21}$$

Check node EXIT function:

$$\begin{aligned} I_{E_c} &= f(I_{A_c}) = \sum_{j=2}^{d_c} \rho_j I_{E_{c_j}} \\ &= \sum_{j=2}^{d_c} \rho_j \frac{1}{\ln 2}\sum_{i=1}^{\infty}\frac{1}{(2i-1)(2i)}[\Phi_i(J^{-1}(I_{A_c})]^{j-1} \end{aligned} \tag{22}$$

The algorithm described above is quite accurate, however an additional improvement of the algorithm can be made. Note that we introduce additional inaccuracy by assuming that the input messages of a node are Gaussian instead of Gaussian mixtures. Unlike the BEC, where a mixture of BEC densities is a BEC density, for AWGN channel this is not true, i.e. a mixture of Gaussian densities is not a Gaussian density. Thus, even if the messages emanating from the nodes are Gaussian, our analysis will still be inaccurate because we assume that the input messages of a node are Gaussian. This additional inaccuracy can be avoided by tracking the extrinsic information of each degree j nodes separately instead

TABLE I

APPROXIMATE AND EXACT THRESHOLD VALUES FOR VARIOUS REGULAR LDPC CODES

j	k	$rate$	σ_{BER}	σ_{MEAN}	σ_{MI1}	σ_{MI2}	σ_{EXACT}	$\Delta_{BER}[dB]$	$\Delta_{MEAN}[dB]$	$\Delta_{MI1}[dB]$	$\Delta_{MI2}[dB]$
3	6	0.5	0.8483	0.8747	0.8784	0.8812	0.8809	0.3275	0.0613	0.0247	-0.0030
4	8	0.5	0.7984	0.8323	0.8355	0.8381	0.8376	0.4166	0.0553	0.0218	-0.0049
5	10	0.5	0.7543	0.7910	0.7918	0.7939	0.7936	0.4413	0.0285	0.0197	-0.0033
3	5	0.4	0.9637	1.0003	1.0064	1.0099	1.0093	0.4014	0.0778	0.0246	-0.0052
4	6	1/3	0.9508	1.0035	1.0085	1.0118	1.0109	0.5323	0.0638	0.0206	-0.0077
3	4	0.25	1.1909	1.2517	1.2637	1.2682	1.2667	0.5361	0.1035	0.0206	-0.0103
4	10	0.6	0.7183	0.7440	0.7462	0.7483	0.7481	0.3530	0.0477	0.0221	-0.0023
3	9	2/3	0.6903	0.7051	0.7066	0.7082	0.7082	0.2221	0.0381	0.0196	0.0002
3	12	0.75	0.6194	0.6297	0.6309	0.6319	0.6320	0.1747	0.0317	0.0151	0.0014

of tracking only the average extrinsic information. This way, we can accurately compute the extrinsic information at the output of a node assuming the input messages have a Gaussian mixture distribution.

We can generalize the algorithm described in Section IV for the case of irregular LDPC codes. The generalized algorithm tracks the extrinsic information of each degree j nodes separately, and performs the extrinsic information computation assuming a node's input messages have Gaussian mixture distributions. Since now the input message to a check node has a Gaussian mixture distribution and by linearity of expectation we can write the "soft bit" expectation at the input of a check node as:

$$\mathbf{E}(T_V^{2i}) = \sum_{j=2}^{d_v} \lambda_j \mathbf{E}(T_{V_j}^{2i}) = \sum_{j=2}^{d_v} \lambda_j \Phi_i(m_{V_j}), \qquad (23)$$

where T_{V_j} is the soft bit message of a degree j variable node, corresponding to the LLR message $\sim \mathcal{N}(m_{V_j}, 2m_{V_j})$. Following the same arguments, and substituting (23) into (16) we get:

$$\begin{aligned} \mathbf{E}(T_U^{2i}) &= \sum_{h=2}^{d_c} \rho_h \mathbf{E}(T_{U_h}^{2i}) = \sum_{h=2}^{d_c} \rho_h \left[\sum_{j=2}^{d_v} \lambda_j \mathbf{E}(T_{V_j}^{2i}) \right]^{h-1} \\ &= \sum_{h=2}^{d_c} \rho_h \left[\sum_{j=2}^{d_v} \lambda_j \Phi_i(m_{V_j}) \right]^{h-1}. \qquad (24) \end{aligned}$$

Substituting (24) into equations (7) and (15) we get:

$$I(X;U) = \frac{1}{\ln 2} \sum_{i=1}^{\infty} \frac{1}{(2i)(2i-1)} \sum_{h=2}^{d_c} \rho_h \left[\sum_{j=2}^{d_v} \lambda_j \Phi_i(m_{V_j}) \right]^{h-1} \qquad (25)$$

$$I(U_0;U) = \frac{1}{\ln 2} \sum_{i=1}^{\infty} \frac{1}{(2i-1)(2i)} \Phi_i(m_{U_0}) \sum_{h=2}^{d_c} \rho_h \left[\sum_{j=2}^{d_v} \lambda_j \Phi_i(m_{V_j}) \right]^{h-1} \qquad (26)$$

The resulting algorithm is summarized in Table II

The GA algorithms defined by (19) and (20) and by Table II were used for computing the channel noise standard deviation threshold values σ_{MI3} and σ_{MI4} respectively, for various irregular LDPC codes (defined in Table III) over biAWGN channel and BP decoder. The threshold values were compared to the approximated threshold value σ_{MEAN} and σ_{BER} computed by the mean and BER tracking GA algorithms proposed in [6] and [10] respectively and to the exact threshold value σ_{exact} computed by the DE algorithm [5]. The results are summarized in Table IV.

TABLE II

IMPROVED GA ALGORITHM: GA_{MI4}

iteration (l)**:**

1) $m_{V_1}^{(l)} = m_{U_0} = 4\frac{E}{No}$ (assume channel parameter is $\frac{E}{No}$)
2) $a_i = \sum_{h=2}^{d_c} \rho_h \left[\sum_{j=2}^{d_v} \lambda_j \Phi_i(m_{V_j^{(l-1)}}) \right]^{h-1}, i = 1, \ldots, \infty;$
3) $I(X;U^{(l)}) = \frac{1}{\ln 2} \sum_{i=1}^{\infty} \frac{1}{(2i-1)(2i)} a_i;$
4) for $j = 2 : d_v$
5) $\quad I(V_{j-1}^{(l)};U^{(l)}) = \frac{1}{\ln 2} \sum_{i=1}^{\infty} \frac{1}{(2i-1)(2i)} \Phi_i(m_{V_{j-1}^{(l)}}) a_i;$
6) $\quad I(X;V_j^{(l)}) = I(X;V_{j-1}^{(l)}) + I(X;U^{(l)}) - I(V_{j-1}^{(l)};U^{(l)});$
7) $\quad m_{V_j}^{(l)} = J^{-1}(I(X;V_j^{(l)}));$
8) end of loop.

The convergence process of BP decoding of Code 1 (from Table III) at $Eb/No = 0.3dB$, as predicted by the DE algorithm (with 10-bit quantization of the messages) and by GA_{MI4} algorithm is shown on Figs. 5 and 6.

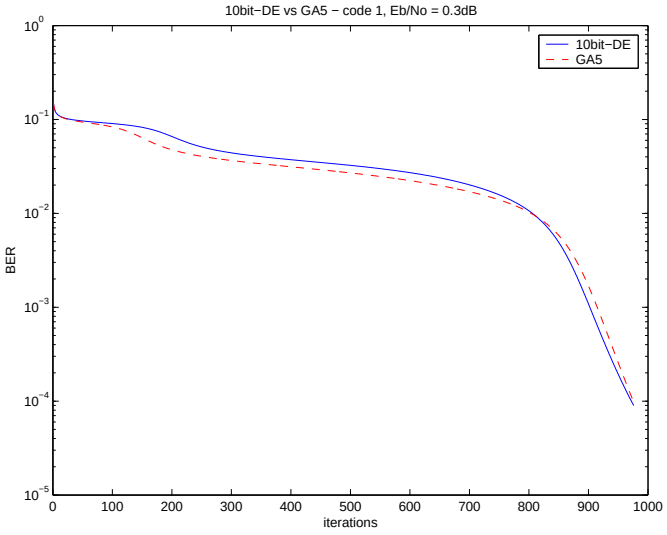


Fig. 5. Bit Error Rate (BER) versus the number of iterations as predicted by the DE and GA_{MI4} algorithms for Code 1 over a biAWGN with $Eb/No = 0.3dB$.

VI. CONCLUSION

We proposed several approximation methods of the DE algorithm for BP decoding of LDPC codes, based on different

TABLE III

SOME RATE $1/2$ IRREGULAR LDPC CODES

code1				code2				code3			
j	λ_j	h	ρ_h	j	λ_j	h	ρ_h	j	λ_j	h	ρ_h
2	0.23403	8	0.71875	2	0.1712	9	0.3362	2	0.170031	10	0.0625
3	0.21242	9	0.28125	3	0.21053	10	0.08883	3	0.16046	11	0.9375
6	0.1469			4	0.00273	11	0.57497	6	0.112837		
7	0.10284			7	0.00009			7	0.047489		
20	0.30381			8	0.15269			10	0.011481		
				9	0.09227			11	0.091537		
				10	0.02802			26	0.152978		
				15	0.01206			27	0.036131		
				30	0.07212			100	0.217056		
				50	0.2583						

TABLE IV

APPROXIMATE AND EXACT THRESHOLD VALUES FOR VARIOUS IRREGULAR LDPC CODES

code	σ_{BER}	σ_{MEAN}	σ_{MI3}	σ_{MI4}	σ_{EXACT}	$\Delta_{BER}[dB]$	$\Delta_{MEAN}[dB]$	$\Delta_{MI3}[dB]$	$\Delta_{MI4}[dB]$
1	0.8955	0.9446	0.9704	0.96685	0.9669	0.6660	0.2028	-0.0313	0.00046
2	0.8873	0.9496	0.9759	0.9715	0.9718	0.7900	0.2009	-0.0364	0.0029
3	0.8907	0.9489	0.9804	0.9746	0.9759	0.7936	0.2437	-0.0396	0.0115

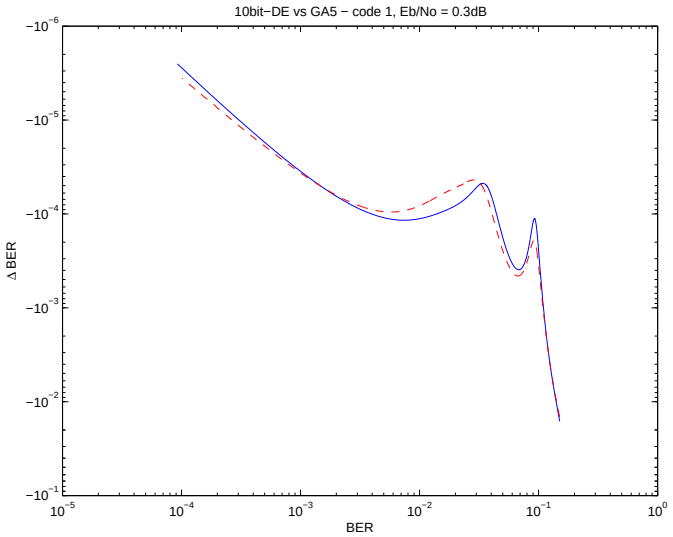


Fig. 6. decrease in BER versus BER as predicted by the DE and GA_{MI4} algorithms for Code 1 over a biAWGN with $Eb/No = 0.3dB$.

strategies of substituting non-Gaussian messages with Gaussian ones having the same mutual information. These methods are simpler than the DE algorithm, and provide better accuracy in prediction of the convergence threshold compared to other GA algorithms, especially for irregular LDPC codes.

REFERENCES

[1] A. Ashikmin, G.Kramer, S.ten Brink, "Extrinsic Information Transfer Functions: Model and Erasure Channel Property", *IEEE Trans. Inform. Theory*, vol. 50, 11, pp. 2657-2673, 2004.

[2] S.ten Brink, "Convergence of iterative decoding", *Electron. Lett*, vol. 35, 10, pp. 806-808, 1999.

[3] S.ten Brink, "Designing iterative decoding schemes with Extrinsic Information Transfer Charts", *AEU International Journal of Electronics and Communications*, vol. 54, 6, pp. 389-398, 2000.

[4] S.ten Brink, G.Kramer, A. Ashikmin "Detection and Decoding of Multi-Antenna Signals with Low-Density Parity-Check Codes", *Trans. on Commun.*, vol. 52, 4, pp. 670-678, 2004.

[5] 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 Commun. Lett*, vol. 5, 2, pp.58–60, 2001.

[6] S.-Y.Chung, T.J.Richardson and R.Urbanke, "Analysis of sum-product decoding of low-density parity-check codes using Gaussian approximation", *IEEE Trans. Inform. Theory*, vol. 47, 2, pp. 657-670, 2001.

[7] M.C.Cover and J.A. Thomas *Elements of Information Theory.* Wiley, New York, 1991.

[8] R.G.Gallager, *Low-Density Parity-Check Codes*, Cambridge, MA: MIT Press 1963.

[9] H. El Gamal and A. R.Hammons JR. "Analyzing the turbo decoder using the Gaussian approximation", *IEEE Trans. Inform. Theory*, vol. 47, 2, pp. 671-686, 2001.

[10] F.Lehmann and G.M.Maggio, "Analysis of the Iterative Decoding of LDPC and Product Codes Using the Gaussian Approximation", *IEEE Trans. Inform. Theory*, vol. 49, 11, 2993-3000 , 2003.

[11] M.G.Luby, M.Mitzenmather, M.A. Shokrollahi, D.A. Spielman, "Analysis of low-density codes and improved designs Using Irregular Graphs", in *Proc. 30th Annu. ACM Symp. Theory of Computing*, pp. 249-258, 1998.

[12] J.Pearl *Probabilistic Reasoning in Intelligent Systems* San Mateo,CA: Morgan Kaufman, 1988.

[13] T.J.Richardson and R.Urbanke, "The capacity of low-density parity-check codes under message passing decoding", *IEEE Trans. Inform. Theory*, vol. 47, 2, pp. 599-618, 2001.

[14] T.J.Richardson, A. Shokrollahi and R.Urbanke, "Design of capacity approaching irregular low-density parity-check codes", *IEEE Trans. Inform. Theory*, vol. 47, 2, pp. 619-637, 2001.

[15] E.Sharon, "Analysis of belief propagation decoding of LDPC codes over the biAWGN channel using improved Gaussian approximation based on the mutual information measure", *Proc. IEEE - 22nd Convention* Tel-Aviv, Israel, pp. 262-264. , 2002.

[16] E. Sharon, A. Ashikhmin, and S. Litsyn, "EXIT Functions for Binary input Memoryless Symmetric Channels," submitted to *Trans. on Commun.*