Design of LDPC Codes Robust to Noisy Message-Passing Decoding

Alla Tarighati, Student Member, IEEE, Hamed Farhadi, Member, IEEE
and Farshad Lahouti, Senior Member, IEEE

Abstract—We address noisy message-passing decoding of low-density parity-check (LDPC) codes over additive white Gaussian noise channels. The internal decoder noise is motivated by the quantization noise of the messages in digital decoders or the intrinsic noise in analog decoders. We model the internal decoder noise as AWGN degrading exchanged messages. Using Gaussian approximation of the exchanged messages, we perform a two-dimensional density evolution analysis for the noisy LDPC decoder. This provides tracking both the mean, and the variance of the exchanged message densities, and hence, quantifying the threshold of the LDPC code in the presence of internal decoder noise. The numerical and simulation results are presented that quantify the performance loss due to the internal decoder noise. To partially compensate this performance loss, we propose a simple method, based on EXIT chart analysis, to design robust irregular LDPC codes. The simulation results indicate that the designed codes can indeed compensate part of the performance loss due to the internal decoder noise.

I. INTRODUCTION

Low-density parity-check (LDPC) codes – first discovered by Gallager [1] and rediscovered by Spielman et al. [2] and MacKay et al. [3], [4] – due to their outstanding performance have attracted much interest and have been studied quite extensively during the recent years. They have also been included in several wireless communication standards, e.g., DVB-S2 and IEEE 802.16e [5], [6]. Effective decoding of LDPC codes can be accomplished by using iterative message-passing schemes such as belief-propagation algorithm or sum product algorithm [7]. Moreover, strong analytical tools including EXIT chart analysis [8], [9] and density evolution analysis [10] are developed for designing LDPC codes and quantifying their performance limits.

The sum-product algorithm is based on elementary computations using sum-product modules [11]. These modules can be implemented using digital circuits or analog circuits. The digital implementation of sum-product modules for LDPC decoding, e.g., the one presented in [12], is subject to noisy message passing due to the quantization of messages. The quantization error is often modeled by an additive noise and under certain conditions is assumed Gaussian [13]. The impact of quantization error on LDPC decoder performance is investigated in [14], where the simulations results indicate a strong influence.

Loeliger et al. in [11] and Hagenauer and Winklhofer in [15] introduced soft gates in order to implement sum-product modules using simple analog transistor circuits. They have also shown that by variations of a single simple circuit, the entire family of sum-product modules can be implemented. Using these circuits, any network of sum-product modules, in particular, iterative decoding of LDPC codes can be directly implemented in analog very-large-scale integration (VLSI) circuits. In analog decoders, the exchanged messages are in general subject to an additive intrinsic noise which depends in part on the chip temperature [16]. To capture this phenomenon, in [16], a channel is considered that is subject to additive white Gaussian noise. This channel is motivated by point-to-point communication between two terminals that are embedded in the same chip. Therefore, in addition to the communication channel noise, the internal decoder noise may affect the communication of soft gates, and hence degrades the performance of the iterative analog decoder.

As discussed, practical digital or analog LDPC decoders are in general subject to an internal noise. Therefore, the impact of this noise on the performance of iterative decoding must be investigated. Performance analysis of noisy LDPC decoding has attracted extensive interest recently (see e.g. [17]–[21] and the references therein). Varshney analyzed the performance of LDPC bit-flipping decoding over a binary symmetric channel (BSC) using a noisy message-passing algorithm. In this setting, the decoder messages are exchanged over a binary symmetric internal channel between the variable nodes and the check nodes [17]. It is shown that the performance degrades smoothly as the decoder noise probability increases. Tabatabaei et al. studied the performance limits of LDPC decoder when it is subjected to transient processor error [18], [19]. This research was further generalized in [20] by considering both transient processor errors and permanent memory errors, using density evolution analysis for regular LDPC codes.

In this paper, we analyze the performance of LDPC codes over an AWGN communication channel, when a sum-product decoding algorithm is employed in which exchanged messages are degraded by independent additive white Gaussian noise. We first invoke a density evolution (DE) analysis to track the probability distribution of exchanged messages during decoding, and quantify the performance degradation due to
the decoder noise. We compute the density evolution equations for both regular and irregular LDPC codes. Also, we introduce an algorithm to find the EXIT curves of a noisy decoder. We further propose a simple algorithm for the design of robust irregular LDPC codes using EXIT chart for noisy decoders to partially compensate the performance loss due to the internal decoder noise.

This paper is organized as follows. In Section II we present the definitions and model for noisy message-passing decoder. In Section III we derive the density evolution equations for the noisy message-passing decoder. Next, numerical results of the density evolution analysis and simulation results of finite-length codes are presented. In Section IV EXIT chart analysis of the noisy decoder is presented. Using the EXIT charts, a method for designing robust codes for the noisy decoder is presented in Section V. Finally, Section VI concludes the paper.

II. LDPC CODES AND NOISY MESSAGE-PASSING DECODING PRINCIPLES

Consider a regular binary \( (d_v, d_c) \) LDPC code with length \( N \) and its parity check matrix \( \mathbf{H} \). The parity check matrix \( \mathbf{H} \) and the LDPC decoding can be described using a Tanner graph with \( N \) variable nodes and \( K = N \frac{d_c}{d_v} \) check nodes. Every variable node in the LDPC code is connected to \( d_v \) check nodes and every check node is connected to \( d_c \) variable nodes. Corresponding to the ones in the columns and rows of \( \mathbf{H} \), the variable nodes and the check nodes are connected to each other in the Tanner graph. Fig. 1 exemplifies a Tanner graph for a regular \((3,6)\) LDPC code with length \( N \). Variable node \( v_i \) and check node \( c_j \) are known as neighbors, if they are connected to each other.

In the message-passing decoding algorithm the outputs of the check nodes and the variable nodes are exchanged in an iterative manner. Specifically, every check node receives messages from its \( d_c \) neighbor variable nodes and sends the computed messages back. Similarly, the variable nodes do this by receiving the messages from their \( d_v \) neighbor check nodes.

We consider the variable nodes and the check nodes output messages as log-likelihood ratios (LLR) values, where the sign of a variable node message specifies the bit estimate and the magnitude indicates its level of reliability.

According to the sum-product decoding algorithm, the message at iteration \( l \) from a variable node to a check node, denoted by \( v^{(l)} \), is

\[
v^{(l)} = \sum_{i=0}^{d_v-1} u_i^{(l-1)},
\]

where \( u_i^{(l-1)} \), \( i = 1, \ldots, d_v - 1 \), are incoming LLRs from variable node neighbors at iteration \( l - 1 \), except the check node that is to receive the output message \( v^{(l)} \), and \( u_0 \) is the incoming LLR message from the communication channel. The message \( u^{(l)} \) from a check node to a variable node at iteration \( l \) can be obtained as follows

\[
\tanh \frac{u^{(l)}}{2} = \prod_{j=1}^{d_c} \tanh \frac{v^{(l)}}{2}.
\]

Fig. 2 shows message-passing through a variable node and a check node, respectively.

For a noisy decoder, the output messages of variable and check nodes are subject to additive white Gaussian noise. The conventional model shown in Fig. 2 can be extended to the one in Fig. 3, where \( n_i^v \) and \( n_j^c \) denote the additive white Gaussian noise affecting the output messages of check nodes and variable nodes, respectively. Hence, \( \gamma_j \) and \( \mu_i \) are noisy versions of \( v_j \) and \( u_i \), respectively. Therefore, the incoming messages to the variable nodes and the check nodes are

\[
\mu_i^{(l)} = u_i^{(l)} + n_i^v,
\]

\[
\gamma_j^{(l)} = v_j^{(l)} + n_j^c.
\]

where \( n_i^v \) and \( n_j^c \) are assumed to be independent and identically distributed (i.i.d.), i.e., \( n_i^v, n_j^c \sim \mathcal{N}(0, \sigma^2) \).

According to the sum-product algorithm, at iteration \( l \) the decoding is performed based on the following updating equations

\[
v^{(l)} = u_0 + \sum_{i=1}^{d_v-1} \mu_i^{(l-1)},
\]

\[
\tanh \frac{u^{(l)}}{2} = \prod_{j=1}^{d_c} \tanh \frac{\gamma_j^{(l)}}{2}.
\]

In order to generalize these equations to irregular case one can follow the same steps as the one described in [7]. In the next section, we propose an approach for the performance analysis of this noisy message-passing decoding algorithm.

Fig. 1. Tanner graph of a regular \((3,6)\) LDPC code, where squares denote check nodes and circles denote variable nodes.

Fig. 2. Message flow through a variable node (a), and through a check node (b).
III. DENSITY EVOLUTION ANALYSIS OF NOISY DECODER

In this Section, using Gaussian approximation, we will find the density evolution equations for the noisy decoder with regular variable and check degrees. We further generalize the results to the irregular case, and finally use the derived density evolution equations to evaluate the performance of noisy decoders.

A. Gaussian Approximation and Consistency

The density evolution analysis is an analytical method for tracking the densities of messages in iterative decoders. This can be used to predict the performance limits of an LDPC code measured by code’s threshold \( t_\text{c} \). The code’s threshold is the smallest (largest) communication channel SNR (noise variance) for which an arbitrarily small decoding bit-error probability can be achieved by sufficiently long codewords. For an AWGN communication channel and an LDPC sum-product decoder, the densities of the exchanged messages between the check nodes and the variable nodes can be approximated as Gaussian [7], [23]. Hence, these densities may be characterized only with their mean and variance. A Gaussian random variable whose variance is twice its mean is said to be consistent [17]. The consistency assumption simplifies density evolution as a one-dimensional recursive equation based on the mean (or the variance) of the messages. In [7], this assumption is used for the DE analysis of a noiseless LDPC decoder, and subsequently, quantifying the threshold of the code.

The key assumption in density evolution analysis of noise-free decoders is that the code block length is sufficiently large, based on which it may be assumed that the Tanner graph of the LDPC code is cycle free. Since the code is linear and the communication channel is symmetric, we consider the transmission of an all-one codeword using a BPSK modulation. Thus, the LLRs received over an AWGN communication channel are Gaussian distributed. The mean and the variance of the received LLRs are respectively equal to \( m_0 = \frac{2}{\sigma_n^2} \) and \( \sigma_0^2 = \frac{4}{\sigma_n^2} \), where \( \sigma_n^2 \) is the channel noise variance [17]. Let assume that the variables \( u, v, u_i \) and \( v_j \) are all Gaussian. First, we check whether the messages of a noisy sum-product LDPC decoder are consistent. To this end, we consider the expected values of both sides of (3) and (5) and obtain

\[
m_v^{(l)} = m_0 + (d_v - 1)m_u^{(l-1)},
\]

where \( m_v^{(l)} \) and \( m_u^{(l-1)} \) denote the mean of output messages of variable nodes and check nodes, respectively. The index \( i \) is omitted since \( u_i, i = 1, \ldots, d_v - 1 \), are i.i.d. Next, by computing the variances of both sides of (3) we have

\[
\sigma_v^{2(l)} = \sigma_{u_i}^2 + \sigma_d^2.
\]

Using (5), we obtain the variance of the variable node output

\[
\sigma_v^{2(l)} = \sigma_0^2 + \text{var}(\sum_{i=1}^{d_v-1} \mu_i^{(l-1)}) + 2\text{cov}(u_0, \sum_{i=1}^{d_v-1} \mu_i^{(l-1)}),
\]

where \( \text{var}(X) \) denotes the variance of random variable \( X \), and \( \text{cov}(X, Y) \) is the covariance of random variables \( X \) and \( Y \). Since \( \mu_i, i = 1, \ldots, d_v - 1 \), are i.i.d. Gaussian random variables, we have

\[
\text{var}(\sum_{i=1}^{d_v-1} \mu_i^{(l-1)}) = \sum_{i=1}^{d_v-1} \text{var}(\mu_i^{(l-1)}) = (d_v - 1)\sigma_\mu^{2(l-1)}.
\]

The last term in (9) is zero, as the Tanner graph of the code is assumed to be cycle-free and \( u_0 \) is independent of the noisy messages. Therefore, the variance of a variable node output message at iteration \( l \) can be simplified as follows

\[
\sigma_v^{2(l)} = \sigma_0^2 + (d_v - 1)\sigma_u^{2(l-1)} + (d_v - 1)\sigma_d^2.
\]

To verify the consistency of the noisy decoder, we plug in \( \sigma_v^{2(l)} = 2m_v^{(l)} \) and \( \sigma_u^{2(l-1)} = 2m_u^{(l-1)} \) into (11) and compare it with (7). It is clear that as long as \( \sigma_d^2 \) is non-zero, the two are not equal and hence the noisy LDPC sum-product decoder is not consistent. As a result, it does not suffice to track only the mean values of the nodes’ output messages. Instead, it is required to track both the mean and the variance of nodes’ output messages. A similar situation is shown in simulation results of [24], when there is an incorrect estimate of the communication channel SNR at a (noiseless) LDPC decoder. However, since the code is linear and the communication channel is symmetric, sending all-one codeword is sufficient for statistical performance evaluation of the code [24].

For an irregular LDPC code, the degree distribution of variable nodes is \( \lambda(x) = \sum_{i=2}^{D_v} \lambda_i x^{i-1} \) and that of the check nodes is \( \rho(x) = \sum_{i=2}^{D_c} \rho_i x^{i-1} \), where \( \lambda_i \) and \( \rho_i \) are the percentage of edges connected to variable nodes and check nodes of degree \( i \), respectively. \( D_v \) is the maximum degree of variable nodes and \( D_c \) is the maximum degree of check nodes. In this case, by similar steps as the ones for the regular case, it can be shown that for the mean and the variance of a variable node of degree \( i \) at iteration \( l \)

\[
m_{v,i}^{(l)} = m_0 + (i - 1)m_u^{(l-1)},
\]

\[
\sigma_{v,i}^{2(l)} = \sigma_0^2 + (i - 1)\sigma_u^{2(l-1)} + (i - 1)\sigma_d^2,
\]

where \( m_{v,i}^{(l)} \) and \( \sigma_{v,i}^{2(l)} \) are the mean and the variance of a variable node of degree \( i \) at iteration \( l \), respectively. From (12) and (13)
and it can be inferred that the consistency is similarly not valid for the irregular case.

**B. Density Evolution with Gaussian Approximation for Noisy Message-Passing Decoder**

In the case of a noisy LDPC decoder, we have shown that consistency does not hold and we should track both the mean and the variance of nodes’ output messages. To this end, we use the key equations (3)-(11). By computing the expected value of both sides of equation (6), and noting that \(\gamma_j^{(l)}\), \(j = 1, \ldots, d_c - 1\) are i.i.d., we have

\[
\mathbb{E}\left[\tanh\frac{u_i^{(l)}}{2}\right] = \mathbb{E}\left[\prod_{j=1}^{d_c-1} \tanh\frac{\gamma_j^{(l)}}{2}\right] = \left(\mathbb{E}\left[\tanh\frac{\gamma_j^{(l)}}{2}\right]\right)^{d_c-1},
\]

where \(\gamma_j^{(l)}\) defined in (4) has the following distribution

\[
\gamma_j^{(l)} \sim N(m_i^{(l)}, \sigma_v^{2(l)} + \sigma_d^2).
\]

Next, by computing the expected value of squared tanh rule, we obtain the second major equation as follows

\[
\mathbb{E}\left[\tanh^2\frac{u_i^{(l)}}{2}\right] = \left(\mathbb{E}\left[\tanh^2\frac{\gamma_j^{(l)}}{2}\right]\right)^{d_c-1}.
\]

The density evolution can be obtained by simultaneously solving equations (14) and (16). Specifically, representing \(\gamma_j^{(l)}\) using (15) and \(m_i^{(l)}, \sigma_v^{2(l)}\) from (7) and (11), we obtain the DE equations for check nodes as follows

\[
f(m_u^{(l)}, \sigma_u^{2(l)}) = \left(\prod_{j=2}^{d_c-1} f\left(\frac{m_v^{(l)} + \sigma_v^{2(l)} + \sigma_d^2}{2}\right)\right)^{d_c-1},
\]

\[
g(m_u^{(l)}, \sigma_u^{2(l)}) = \left(\prod_{j=2}^{d_c-1} g\left(\frac{m_v^{(l)} + \sigma_v^{2(l)} + \sigma_d^2}{2}\right)\right)^{d_c-1}.
\]

The auxiliary functions \(f(m, \sigma^2)\) and \(g(m, \sigma^2)\) are defined as follows

\[
f(m, \sigma^2) \triangleq \mathbb{E}\left[\tanh\frac{X}{2}\right],
\]

\[
g(m, \sigma^2) \triangleq \mathbb{E}\left[\tanh^2\frac{X}{2}\right],
\]

where \(X \sim N(m, \sigma^2)\). These equations can be used to track \(m_u^{(l)}\) and \(\sigma_v^{2(l)}\) in the decoding iterations of a regular \((d_c, d_v)\) LDPC for given values of communication channel noise variance and internal decoder noise variance.

![Fig. 4. Error probability performance of (3, 6) regular finite length code with decoder noise variance \(\sigma_d^2\).](image_url)

Similarly, for irregular LDPC codes, the message distributions are approximated by Gaussian mixture (7) and for each check node of degree \(i\) at iteration \(l\) we have

\[
f\left(\frac{m_v^{(l)}}{\sigma_v^{2(l)} + \sigma_d^2}\right) = \sum_{j=2}^{d_c-1} \lambda_j f\left(\frac{m_v^{(l)} + \sigma_v^{2(l)} + \sigma_d^2}{2}\right)^{i-1},
\]

\[
g\left(\frac{m_v^{(l)}}{\sigma_v^{2(l)} + \sigma_d^2}\right) = \sum_{j=2}^{d_c-1} \lambda_j g\left(\frac{m_v^{(l)} + \sigma_v^{2(l)} + \sigma_d^2}{2}\right)^{i-1},
\]

and from Gaussian mixture equations, the density of check node in iteration \(l\) has the following mean and variance values

\[
m_u^{(l)} = \sum_{i=2}^{D_c} \rho_i m_{u,i}^{(l)},
\]

\[
\sigma_u^{2(l)} = \sum_{i=2}^{D_c} \rho_i \left(\frac{\sigma_v^{2(l)}}{\rho_i} + \left(m_{u,i}^{(l)}\right)^2 - \left(m_{u}^{(l)}\right)^2\right).
\]

Therefore, the DE can be found by solving (19) for a check node with degree \(i\) and the distribution of check node is then can be found using (20) and (21), iteratively.

**C. Numerical Results of Density Evolution**

We solve the density evolution equations iteratively for a \((3, 6)\) regular LDPC code considering \(m_u^{(0)} = 0\) and \(\sigma_u^{2(0)} = 0\) as initial conditions. This provides the mean and the variance of check nodes’ outputs and allows for the computation of the threshold for the given variances of internal decoder noise and communication channel noise.

Table III shows the relation between the threshold and the internal decoder noise variance \(\sigma_d^2\) resulting from (17). It can be observed that the SNR threshold \(\text{SNR}_{\text{th}} = \frac{\sigma_d^2}{\rho_0}\) increases as the internal decoder noise variance increases. This is in line

| \(\sigma_d^2\) | SNR threshold \(\text{SNR}_{\text{th}}\) | \((\sigma_u)_{\text{th}}\) |
|---|---|---|
| 0 | 1.163 dB | 0.8744 |
| 1 | 2.835 dB | 0.7215 |
| 2 | 3.635 dB | 0.6580 |
| 3 | 4.185 dB | 0.6177 |
with a similar observation in [17] on the performance of bit-flipping LDPC decoding in the presence of noisy message-passing over BSC channels, where the performance deteriorates as the cross over probability of the internal decoder noise increases.

Simulation results confirm that our analytical results accurately predict the performance of finite-length codes as well. Fig. 4 depicts the simulation results for the performance of a finite-length (3, 6) regular code with length $N = 1008$. It is evident that the threshold of this finite length code is fairly the same as our analytical threshold for various values of the internal decoder noise variance $\sigma^2_d$.

One can use (20) and (21) to also track the density of irregular LDPC codes for noisy decoder. In general, the presented analysis could be used to design irregular LDPC codes for noisy decoders. Since the problem of designing by density evolution is not a convex problem, finding a good degree distribution requires complex computations and extensive search [25]. Therefore, in the remaining sections after investigating the performance limits of the noisy decoder by means of EXIT chart, we will introduce a simple and effective method to design robust LDPC codes for the noisy decoder.

IV. EXIT CHART ANALYSIS OF NOISY DECODER

The EXIT chart analysis, first introduced in the pioneering work of ten Brink [26], is a powerful tool for analyzing the performance of iterative turbo techniques. It is mainly based on keeping track of the mutual information of channel input bits and variable node and check node outputs.

Let $X$ be a binary random variable denoting the BPSK modulated AWGN communication channel input which takes ±1 values with equal probabilities. If $f(y)$ is the probability density function (pdf) of the communication channel soft output $Y$, then, the mutual information of $X$ and $Y$ for a symmetric channel [27], [28] is

$$I(X; Y) = \frac{1}{2} \sum_{z = \pm 1} \int_{-\infty}^{\infty} f(y|x) \log \left( \frac{f(y|x)}{f(y)} \right) dy.$$  (22)

In order to find the EXIT function of a noiseless decoder, the variable node and the check node EXIT functions are obtained by defining a $J$-function [8], which is directly resulted from the consistency assumption of the decoder. Since this assumption is violated in the case of a noisy decoder, to compute the a priori and extrinsic mutual information, the definition of mutual information is directly used.

To obtain the EXIT curves for the noisy LDPC decoder, we use Algorithm 11 and the empirical distributions computed by running simulations for each degree of variable node and check node. Specifically for each decoder component, we compute the extrinsic mutual information $I_{E}$ corresponding to a priori mutual information $I_{A}$ for the noisy variable (check) node decoder. To maintain the desired structure of the iterative decoder [8] for the noisy decoder, we modified the variable nodes and check nodes as shown in Fig. 5. In fact, we have considered decoder noise as a part of variable (check) nodes and call them noisy variable (check) node decoders, NVND (NCND). In Algorithm 11 the proposed method for finding the EXIT curve of NVND is described. The procedure of finding EXIT curves for the noisy check node is similar to that of the noisy variable node; however, for the check node the communication channel output is fed to the check node, i.e., the only input of the noisy check node is a priori messages. The presented algorithm is an extension of the algorithm 7.4 in [29] for Turbo codes.

Fig. 6 illustrates the evolution of EXIT charts of (3, 6) regular LDPC code for different values of decoder noise variance $\sigma^2_d$ and for communication channel SNR 3dB. For $\sigma^2_d = 0$, there is an open tunnel between the curves, and increasing $\sigma^2_d$ to one makes the tunnel tighter. However, for $\sigma^2_d = 2$ and $\sigma^2_d = 3$ the variable and the check EXIT curves cross each other. The EXIT charts in Fig. 6 illustrate that the

![Fig. 5. Modification of decoder components (VND or CND) to noisy decoder components (NVND or NCND).]

### Algorithm 1 EXIT curve for NVND

1. **Input:** $I_A$, $d_v$, $\sigma^2_n$, $\sigma^2_d$, $N$
2. **Output:** $I_E$
3. $X = 1$, [length $N$ BPSK modulated codeword]
4. for $k = 1 : N$ do
   5. compute decoder LLR inputs
   6. $Y(k) = \frac{2}{\sigma^2_n} (1 + n_c)$, $n_c \sim \mathcal{N}(0, \sigma^2_n)$
5. end for
6. for $i = 1 : \text{length}(I_A)$ do
   7. $\sigma_A = J^{-1}(I_A(i))$, where
   8. $J(\sigma) = 1 - \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x^2-\sigma^2/2)^2}{2\sigma^2}} \log_2(1+e^{-x}) dx$
9. for $j = 1 : N$ do
   10. compute a priori LLRs
   11. $AP(j) = \sigma^2_n/d_n + n_c \sim \mathcal{N}(0, \sigma^2_A)$
9. end for
8. end for
9. [compute noisy variable node outputs using (5)]
   10. $E = \text{NVND}(d_v, AP, Y, \sigma^2_d)$
10. [Compute extrinsic mutual information using (22)]
   11. $I_E(i) = I(X, E)$
11. end for
SNR threshold of (3, 6) regular LDPC code is less than 3 dB when $\sigma^2_d = 0$ and $\sigma^2_d = 1$ and is greater than 3 dB when $\sigma^2_d = 2$ and $\sigma^2_d = 3$. This is in line with the results obtained from the density evolution analysis, as shown in Table III.

In the following Section, we will use the EXIT curves resulting from Algorithm 1 to design robust LDPC codes for noisy decoders.

V. DESIGN OF ROBUST IRREGULAR LDPC CODES FOR NOISY DECODER

In this Section, our goal is to design robust irregular LDPC codes for noisy decoder. The design procedure of LDPC codes by means of EXIT charts, consists of curve fitting of designed variable and check EXIT curves. In [8], for a noiseless decoder, right-regular LDPC codes are designed for a fixed check node degree of $d_c$, and fitting a weighted sum of the EXIT curves of variable nodes to the check node EXIT curve. In this work, we design irregular LDPC codes and allow more than one degree for both variable and check nodes. Our benchmark for comparisons are irregular LDPC codes with the same rates designed for a noiseless decoder in [27]. We verify the performance of the designed codes by simulations of finite-length LDPC codes.

It is noteworthy that when the decoder is not consistent, the weighted sum of the EXIT chart curves of variable node and check node are not the exact curves of the irregular code. Still, our simulation results indicate that this approach is accurate enough for the case with a noisy decoder.

A. Code Design Algorithm

Using the EXIT curves resulting from the previous Section, we propose a simple method for the design of irregular codes for the noisy decoder. Similar to the code design approach proposed in [27], we restrict our attention to irregular codes with two consecutive check node degrees as

$$\rho(x) = \alpha x^{d_c-1} + (1-\alpha) x^{d_c},$$

and variable node degrees as

$$\lambda(x) = \sum_{i=2}^{D_v} \lambda_i x^{i-1}.$$  \hspace{1cm} (24)

The effective EXIT curve of a noisy irregular check (variable) node $I_{A,NCND}$ ($I_{A,NVND}$) is obtained by averaging over the EXIT curves of check (variable) nodes with constituent check (node) degrees [9].

**Algorithm 2 Robust LDPC code design**

1. **Input**: $d_c$, $D_v$, $r$, $\sigma^2_d$, $\Delta$, $S$
2. **Output**: code $C^* = (\rho(x)^*, \lambda(x)^*)$
3. initialize $\text{SNR}_{th}$
4. $\sigma^2_n = (2r\text{SNR}_{th})^{-1}$
5. for $i = 1 : M + 1$ do
6. $\alpha = S(i)$
7. [compute EXIT curves of check nodes with degree $d_c - 1$ and degree $d_c$ using Algorithm 1]
8. [compute effective EXIT curve $I_{A,NCND}$ using $\rho(x)$] in (23)
9. [compute EXIT curves of variable nodes with degrees $d_v = 2$ to $d_v = D_v$ using Algorithm 1]
10. [check if there is a $\lambda(x)$ such that $C = (\rho(x), \lambda(x))$ satisfies the constraints in (25)]
11. if code $C$ exists then
12. $\text{SNR}_{th} = \text{SNR}_{th} - \Delta$
13. $C^* = C$
14. go to step 4
15. end if
16. end for

In this paper, we refer to a candidate code $C = (\rho(x), \lambda(x))$ of rate $r$ as a **successful code** for a given SNR, if it satisfies the following constraints

$$\begin{align*}
  i) & \quad \lambda(1) = 1, \\
  ii) & \quad \rho(1) = 1, \\
  iii) & \quad r = 1 - \int_0^1 \rho(x)dx \int_0^1 \lambda(x)dx, \\
  iv) & \quad I_{E,NVND} > I_{A,NCND},
\end{align*} \hspace{1cm} (25)$$

where the last constraint implies that the effective EXIT curve of its variable node lies above the effective EXIT curve of its check node.

For a given code rate $r$, we are looking for the code $C^* = (\rho(x)^*, \lambda(x)^*)$ corresponding to the threshold SNR, $\text{SNR}_{th}$, i.e. the minimum SNR for which there is a successful code.

In Algorithm 2 we set the check node degree $d_c$, the maximum variable node degree $D_v$, and design rate $r$. In order for algorithm to run fast, we let $\alpha$ take limited values in $S = [0 : 1/M : 1]$. Then, for a given $\text{SNR}_{th}$, by varying $\alpha$ from zero to one, we check if there is a variable degree distribution $\lambda(x)$, as in (24), such that the constraints in
and (25) are satisfied. If such code $C = (\rho(x), \lambda(x))$ exists, we reduce $\text{SNR}_{th}$ by $\Delta$ and the algorithm does another iteration; otherwise, it terminates and the successful code from the previous iteration is chosen.

The EXIT curves of a check node do not depend on $\text{SNR}_{th}$; By changing $\text{SNR}_{th}$, the EXIT curves of variable nodes, and consequently $I_{E,\text{NVND}}$ are affected. Using this fact, the complexity of the proposed algorithm may be further reduced by pre-computing $I_{E,\text{NCND}}$ for each value of $\alpha$ once and storing them before running the algorithm. In the following Section, we will present some results illustrating the application of the proposed algorithm in the design of robust LDPC codes for noisy decoders.

**B. Design Examples**

From Table 1 of [27], for the maximum variable degree of four, the introduced code of rate one-half has two types of check nodes with degrees five and six. The threshold of this code is $0.8085$ dB and has degree distribution as

$$\lambda(x) = 0.384x + 0.042x^2 + 0.574x^3,$$

$$\rho(x) = 0.241x^4 + 0.759x^5.$$  

However, when the decoder is not perfect, the threshold and also the performance of this code degrades. For instance, in noisy decoder the threshold has increased by about $1.7$dB and $2.5$dB for decoder noise variances $\sigma_d^2 = 0.5$ and $\sigma_d^2 = 1$, respectively.

Considering the same constraints in degree distribution and maximum variable degree, we have designed an irregular one-half rate LDPC code for the noisy decoder with $\sigma_d^2 = 0.5$. Using Algorithm 2 we obtained code A with the following degree distributions

$$\lambda(x) = 0.453x + 0.547x^3,$$

$$\rho(x) = 0.451x^4 + 0.549x^5.$$  

whose threshold is approximately $0.15$ dB smaller than the code in [27] decoded by the noisy decoder. Fig. 7 shows the simulation results for a noisy decoder with $\sigma_d^2 = 0.5$ and when the code length is $10^4$. It is noteworthy that the Tanner graph of both of these codes are free from cycles of length four or less.

As another example of code design, considering the same constraints on degree distribution, we have designed an irregular code (code B) for noisy decoder with $\sigma_d^2 = 1$. The degree distribution of this code is

$$\lambda(x) = 0.4808x + 0.5192x^3,$$

$$\rho(x) = 0.553x^4 + 0.447x^5.$$  

In comparison to the code in [27], the proposed code B achieves a $0.2$dB smaller threshold, when the decoder is noisy with $\sigma_d^2 = 1$. Fig. 8 shows the simulation result of these codes, for the code length of $10^4$ when for both of these codes the Tanner graph has no cycle less than or equal to four. The simulation results in Fig. 7 and Fig. 8 confirm that the performance of the code can be improved if we consider the decoder noise during the design procedure.

**VI. Conclusions**

We considered a noisy belief propagation scheme for the decoding of LDPC codes over AWGN communication channels. We modeled the internal decoder noise as additive white Gaussian and observed the inconsistency of the exchanged message densities in the iterative decoder. For the inconsistent LDPC decoder, a density evolution scheme was formulated to track both the mean and the variance of message. The results quantify the increase of the decoding threshold SNR as a consequence of the internal decoder noise. Using EXIT chart analysis, the performance of the noisy decoder was analyzed and it was shown to be in harmony with that obtained from the density evolution. Next using EXIT charts, an algorithm was introduced to design robust irregular LDPC codes in order to
partially compensate the performance loss due to the decoder noise. In this work, we modeled the decoder noise as AWGN on the exchanged messages, however, an interesting future step is to incorporate other noise models possibly directly obtained from practical decoder implementations.

REFERENCES

[1] R. Gallager, “Low-density parity-check codes,” IRE Trans. Inf. Theory, vol. 8, no. 1, pp. 21–28, Jan 1962.
[2] M. Sipser and D. Spielman, “Expander codes,” IEEE Trans. Inf. Theory, vol. 42, no. 6, pp. 1710–1722, Nov 1996.
[3] D. J. MacKay and R. M. Neal, “Near shannon limit performance of low density parity check codes,” Electronics letters, vol. 32, no. 18, pp. 1645–1646, 1996.
[4] D. J. MacKay, “Good error-correcting codes based on very sparse matrices,” IEEE Trans. Inf. Theory, vol. 45, no. 2, pp. 399–431, 1999.
[5] M. Eroz, F. Sun, and L. Lee, “DVB-S2 low-density parity-check codes with near shannon limit performance,” Int. Journal of Satellite Communications and Networking, vol. 22, no. 3, 2004.
[6] “Part 16: Air interface for fixed and mobile broadband wireless access systems amendment for physical and medium access control layers for combined fixed and mobile operation in licensed bands,” IEEE P802.16e, Oct 2004.
[7] 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. Inf. Theory, vol. 47, no. 2, pp. 657–670, 2001.
[8] S. ten Brink, G. Kramer, and A. Ashikhmin, “Design of low-density parity-check codes for modulation and detection,” IEEE Trans. Commun., vol. 52, no. 4, pp. 670–678, April 2004.
[9] A. Ashikhmin, G. Kramer, and S. ten Brink, “Extrinsic information transfer functions: model and erasure channel properties,” IEEE Trans. Inf. Theory, vol. 50, no. 11, pp. 2657–2673, Nov 2004.
[10] T. J. Richardson and R. L. Urbanke, “The capacity of low-density parity-check codes under message-passing decoding,” IEEE Trans. Inf. Theory, vol. 47, no. 2, pp. 599–618, Feb 2001.
[11] H.-A. Loeliger, F. Lustenberger, M. Helfenstein, and F. Tarköy, “Probability propagation and decoding in analog VLSI,” IEEE Trans. Inf. Theory, vol. 47, no. 2, pp. 837–843, 2001.
[12] T. Pegoraro, F. Gomes, R. Lopes, R. Gallo, J. Panaro, M. Paiva, F. Oliveira, and F. Lumertz, “Design, simulation and hardware implementation of a digital television system: LDPC channel coding,” in IEEE 9th Int. Symp. Spread Spectrum Techniques and Applications, Aug 2006, pp. 203–207.
[13] D. Lee and D. Neuhoff, “Asymptotic distribution of the errors in scalar and vector quantizers,” IEEE Trans. Inf. Theory, vol. 42, no. 2, pp. 446–460, Mar 1996.
[14] Z. Zhang, L. Dolecek, M. Wainwright, V. Anantharam, and B. Nikolic, “Quantization effects in low-density parity-check decoders,” in IEEE Int. Conf. Commun. (ICC), June 2007, pp. 6231–6237.
[15] J. Hagenauer and M. Winklhofer, “The analog decoder,” in IEEE Int. Symp. Inf. Theory, Aug 1998, p. 145.
[16] T. Koch, A. Lapidoth, and P. Sotiriadis, “Channels that heat up,” IEEE Trans. Inf. Theory, vol. 55, no. 8, pp. 3594–3612, Aug 2009.
[17] L. Varshney, “Performance of LDPC codes under faulty iterative decoding,” IEEE Trans. Inf. Theory, vol. 57, no. 7, pp. 4427–4444, July 2011.
[18] S. Tabatabaei Yazdi, C.-H. Huang, and L. Dolecek, “Optimal design of a Gallager B noisy decoder for irregular LDPC codes,” IEEE Commun. Lett., vol. 16, no. 12, pp. 2052–2055, Dec 2012.
[19] S. Tabatabaei Yazdi, H. Cho, and L. Dolecek, “Gallager B decoder on noisy hardware,” IEEE Trans. Commun., vol. 61, no. 5, pp. 1660–1673, May 2013.
[20] C.-H. Huang, Y. Li, and L. Dolecek, “Gallager B LDPC decoder with transient and permanent errors,” IEEE Trans. Commun., vol. 62, no. 1, pp. 15–28, Jan 2014.
[21] A. Tarighati, H. Farhadi, and F. Lahouti, “Performance analysis of noisy message-passing decoding of low-density parity-check codes,” in Proc. 6th IEEE Int. Symp. Turbo Codes and Iterative Inform. Processing, 2010.
[22] T. K. Moon, Error Correction Coding: Mathematical Methods and Algorithms. Wiley-Interscience, 2005.
[23] H. El Gamal and A. R. Hammons Jr, “Analyzing the turbo decoder using the Gaussian approximation,” IEEE Trans. Inf. Theory, vol. 47, no. 2, pp. 671–686, Feb 2001.