Performance Bounds and Estimates for Quantized LDPC Decoders

Homayoon Hatami, *Member, IEEE*, David G. M. Mitchell, *Senior Member, IEEE*, Daniel J. Costello, Jr., *Life Fellow, IEEE*, and Thomas E. Fuja, *Fellow, IEEE*

***Abstract*—The performance of low-density parity-check (LDPC) codes at high signal-to-noise ratios (SNRs) is known to be limited by the presence of certain sub-graphs that exist in the Tanner graph representation of the code, for example trapping sets and absorbing sets. This paper derives a lower bound on the frame error rate (FER) of any LDPC code containing a given problematic sub-graph, assuming a particular message passing decoder and decoder quantization. A crucial aspect of the lower bound is that it is code-independent, in the sense that it can be derived based only on a problematic sub-graph and then applied to any code containing it. Due to the complexity of evaluating the exact bound, assumptions are proposed to approximate it, from which we can estimate decoder performance. Simulated results obtained for both the quantized sum-product algorithm (SPA) and the quantized min-sum algorithm (MSA) are shown to be consistent with the approximate bound and the corresponding performance estimates. Different classes of LDPC codes, including both structured and randomly constructed codes, are used to demonstrate the robustness of the approach.**

***Index Terms*—LDPC codes, absorbing sets, trapping sets, message passing decoders, decoder quantization, error-floor behavior.**

I. Introduction

LOW-DENSITY parity-check (LDPC) codes [1] are a class of error correcting codes with asymptotic performance approaching the Shannon limit. However, practical LDPC decoders, such as those that implement message-passing algorithms based on belief propagation (BP), can introduce an *error floor* that limits error probability at high signal-to-noise ratios (SNRs). A number of structures in a code's Tanner graph representation have been identified as significant factors in error floor performance, *e.g.*, *near-codewords* [2], *trapping sets* [3], and *absorbing sets* [4]. Absorbing sets are known to be problematic in a variety of LDPC codes and stable under bit flipping decoding [5], [6]. Other classes of trapping sets, such as *elementary trapping sets* and *leafless elementary trapping sets*, have been shown to be the dominant cause of the error floor for certain codes [7]–[10].

Several papers have addressed the problem of predicting the error floor performance of LDPC codes on the additive white Gaussian noise (AWGN) channel based on the existence of these problematic structures. In [3], Richardson proposed a variation of importance sampling to estimate the frame error rate (FER) of a code based on trapping sets. In [11], an error floor estimate was introduced based on the *dominant* absorbing sets (those empirically determined to cause most errors) in structured array-based codes, and the results were compared to those derived from importance sampling. In [12], a method similar to [11] was applied to the min-sum algorithm (MSA). In [13], the contribution of the shortest cycles in a code's graph was used to estimate its performance. Also, [14] and [15] developed a state-space model for a code's dominant absorbing sets to estimate its FER. Later, [16] applied this method to the case where the log-likelihood-ratios (LLRs) used for decoding are constrained to some maximum saturation value. Each of these references considered the problematic structures of a particular code. In contrast, the authors of [5] derived a real-valued threshold associated with a particular absorbing set irrespective of the code; the threshold indicates if the absorbing set can be "deactivated" and hence not contribute to the FER at high SNR in any code that contains it.

This paper obtains *sub-graph specific*, or *code-independent*, lower bounds on the performance of an LDPC code when a finite precision (quantized) LDPC decoder is used. These bounds are *general*, in that they apply to *any code* containing a particular problematic sub-graph; however, calculating the bound is complex, so we introduce assumptions and approximations to simplify its calculation, resulting in what we call an *approximate lower bound*. Given a description of a dominant problematic sub-graph and its multiplicity in a code, an estimate of the resulting FER performance is obtained. Extensive simulation results justify the validity of the assumptions and approximations used for various decoders, quantizers, problematic sub-graphs, and codes.

We first create a simplified model for the Tanner graph of a code containing a particular problematic sub-graph; this model captures the structure of the code outside the sub-graph with a single edge connected to each check node incident to a variable

Manuscript received April 24, 2019; revised September 10, 2019; accepted November 2, 2019. Date of publication November 13, 2019; date of current version February 14, 2020. This work was supported in part by the National Science Foundation under Grant Nos. ECCS-1710920 and OIA-1757207. This article was presented in part at the IEEE International Symposium on Information Theory, Barcelona, Spain, July 2016, and in part at the Allerton Conference on Communication, Control, and Computing, Monticello, IL, USA, October 2017. The associate editor coordinating the review of this article and approving it for publication was Q. Huang. *(Corresponding author: Homayoon Hatami.)*

H. Hatami was with the Department of Electrical Engineering, University of Notre Dame, Notre Dame, IN 46556 USA. He is now with Samsung Semiconductor Inc., San Diego, CA 92121 USA (e-mail: hhatami@nd.edu).

D. G. M. Mitchell is with the Klipsch School of Electrical and Computer Engineering, New Mexico State University, Las Cruces, NM 88003 USA (e-mail: dgmm@nmsu.edu).

D. J. Costello, Jr., and T. E. Fuja are with the Department of Electrical Engineering, University of Notre Dame, Notre Dame, IN 46556 USA (e-mail: costello.2@nd.edu; tfuja@nd.edu).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCOMM.2019.2953232

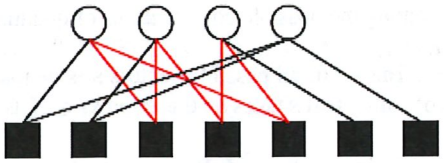

Fig. 1. An illustration of a $(4, 2)$ absorbing set with girth 6. This sub-graph can also be referred to as an elementary trapping set or a leafless elementary trapping set.

LDPC codes, for example, and we see that it contains a *cycle* of length six (the highlighted edges in the figure). The *girth* of an absorbing set is the length of its shortest cycle, and it can be readily observed that the girth of the absorbing set in Fig. 1 is six.

Other classifications of problematic sub-graphs have been referred to as elementary trapping sets (ETS), which contain only degree-1 and degree-2 check nodes [9], and leafless elementary trapping sets (LETS), in which each variable node is connected to at least two even-degree check nodes [7]. As such, Fig. 1 can also be referred to as a $(4, 2)$ ETS or LETS.

C. Quantizers

Since quantized decoding may have different performance characteristics than unquantized decoding, considering the effect of quantization on decoder performance is of great importance:

- Uniform Quantization: Following convention, we let $\mathrm{Q}_{q_1.q_2}$ denote a quantizer that represents each message with $q = q_1 + q_2 + 1$ bits: q_1 bits to represent the integer part of the message, q_2 bits to represent the fractional part, and one bit to represent the sign. In this case, there are $t = 2^q$ quantization levels, where the levels (*i.e.*, the quantized message values) range from $\ell_1 = -2^{q_1}$ to $\ell_t = 2^{q_1} - 2^{-q_2}$, with step size $\Delta = 2^{-q_2}$ between levels. The quantizer thresholds are equidistant between the levels and range from b_1 to b_{t-1}, where $b_i = \frac{\ell_i + \ell_{i+1}}{2}$ for $i \in \{1, 2, \ldots, t-1\}$.
- Quasi-Uniform Quantization: In [17], the authors proposed a non-uniform quantizer, denoted as "quasi-uniform" due to its structure, which uses q bits for uniform quantization, thus maintaining precision, plus an extra bit to increase the range of the quantizer compared to a $q+1$ bit uniform quantizer. It is shown in [17] that the increased range of this quantizer improves the error-floor performance.

III. System Model

In this section, we propose a general model for representing a problematic sub-graph in an arbitrary code. We also formulate expressions for the quantized LLR values received at the variable nodes and check nodes in the sub-graph. As mentioned earlier, we focus on absorbing sets as our sub-graph of interest in the development of our system model; however, the system model can be generalized in a straightforward manner to any sub-graph.

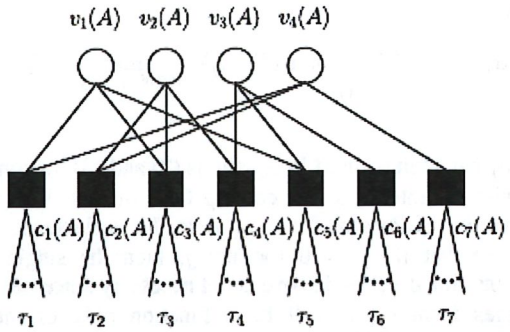

Fig. 2. An illustration of a $(4, 2)$ absorbing set with an unspecified number of edges connected to each check node.

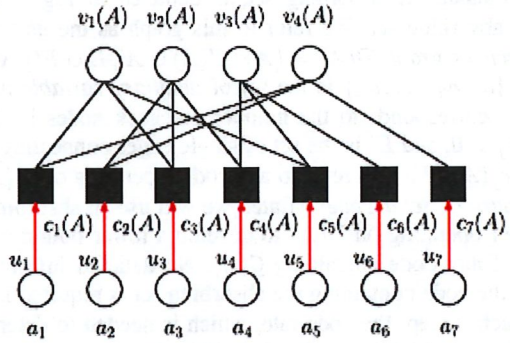

Fig. 3. An illustration of a $(4, 2)$ absorbing set decoder graph $\mathcal{D}(A)$ with single edges connected from auxiliary variable node a_j to each check node $c_j(A)$, where each u_j represents the LLR input to check node $c_j(A)$ from outside $\mathcal{G}(A)$.

A. Absorbing Set Model

We consider the general case of an (a, b) absorbing set with an unspecified number of edges connected to each of its check nodes. The variable nodes are represented by $A \subset V$. We partition the edges connected to each $c_j(A)$ into two groups depending on whether they connect to a variable node in A or $V \setminus A$. We denote the neighboring nodes of $c_j(A)$ in A as $N'(c_j(A))$ and the neighboring nodes of $c_j(A)$ in $V \setminus A$ as $N''(c_j(A))$. If there are $\rho_j \geq 1$ edges connected to A and $\tau_j \geq 0$ edges connected to $V \setminus A$, then $|N'(c_j(A))| = \rho_j$ and $|N''(c_j(A))| = \tau_j$. In Fig. 2, a $(4, 2)$ absorbing set is illustrated in which $\rho_j = 2$, $j = 1, 2, \ldots, 5$, $\rho_6 = \rho_7 = 1$, and τ_j is arbitrary for $j = 1, 2, \ldots, 7$ (note that τ_j can be zero).

To simplify the calculation of the LLRs sent from each check node $c_j(A)$ to the variable nodes $v_i(A) \in A$ in the case where $\tau_j > 0$, we represent the τ_j edges from the variable nodes in $N''(c_j(A))$ with a single edge (see Fig. 3). This edge has an LLR u_j that is a function of all the external LLRs coming from the set $N''(c_j(A))$ to $c_j(A)$ and can be derived as follows:

- SPA:

$$\begin{cases} \operatorname{sign}(u_j) = \prod_{v' \in N''(c_j)} \operatorname{sign}\left(\mathbb{V}_{v' \to j}\right), \\ \Phi_1\left(|u_j|\right) = \sum_{v' \in N''(c_j)} \Phi_1\left(|\mathbb{V}_{v' \to j}|\right). \end{cases} \tag{6}$$

The resulting array is deterministic, *i.e.*, it is not a function of the channel SNR. A pictorial representation of the decodability array is shown below:

$$\begin{array}{c} \\ \mathbf{W}_1 \\ \vdots \\ \mathbf{W}_k \\ \vdots \\ \mathbf{W}_{t^{\kappa p}} \end{array} \begin{array}{c} \begin{array}{cccccc} \mathbf{x}_1 & \cdots & \mathbf{x}_i & \cdots & \mathbf{x}_{t^a} \end{array} \\ \begin{pmatrix} \mathbb{I}_{(\mathbf{x}_1,\mathbf{W}_1)} & \cdots & \mathbb{I}_{(\mathbf{x}_i,\mathbf{W}_1)} & \cdots & \mathbb{I}_{(\mathbf{x}_{t^a},\mathbf{W}_1)} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \mathbb{I}_{(\mathbf{x}_1,\mathbf{W}_k)} & \cdots & \mathbb{I}_{(\mathbf{x}_i,\mathbf{W}_k)} & \cdots & \mathbb{I}_{(\mathbf{x}_{t^a},\mathbf{W}_k)} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \mathbb{I}_{(\mathbf{x}_1,\mathbf{W}_{t^{\kappa p}})} & \cdots & \mathbb{I}_{(\mathbf{x}_i,\mathbf{W}_{t^{\kappa p}})} & \cdots & \mathbb{I}_{(\mathbf{x}_{t^a},\mathbf{W}_{t^{\kappa p}})} \end{pmatrix} \end{array}. \tag{12}$$

We now define the *absorbing region* of an absorbing set decoder as the set of all pairs $(\mathbf{x}_t, \mathbf{W}_k)$ with '1' entries in the decodability array.[2] Letting ψ represent the absorbing region, *i.e.*, $\psi = \{(\mathbf{x}_t, \mathbf{W}_k) \,|\, \Pr(\xi(A)|\mathbf{s} = \mathbf{x}_t, \mathbf{U} = \mathbf{W}_k) = 1\}$, $\Pr(\xi(A))$ in (10) can be written as

$$\Pr(\xi(A)) = \sum_{(\mathbf{x}_i,\mathbf{W}_k)\in\psi} \Pr(\mathbf{s} = \mathbf{x}_t, \mathbf{U} = \mathbf{W}_k), \tag{13}$$

where (8) and (9) indicate the dependence of $\Pr(\xi(A))$ on SNR. Evaluating (13) is computationally complex, since the size of the decodability array $t^{\kappa p} \times t^a$ is typically extremely large. In the rest of this section, we propose an approach to simplify the problem of finding the probability $\Pr(\xi(A))$ of the absorbing region.

We proceed by proposing to lower bound $\Pr(\xi(A))$. Assuming that $\mathbf{s}$ and $\mathbf{U}$ are chosen independently, (13) becomes

$$\Pr(\xi(A)) = \sum_{(\mathbf{x}_i,\mathbf{W}_k)\in\psi} \Pr(\mathbf{s} = \mathbf{x}_t) \cdot \Pr(\mathbf{U} = \mathbf{W}_k), \tag{14}$$

where we note that, in an absorbing set decoder, we are independently choosing an $\mathbf{s}$ and a $\mathbf{U}$, running the decoder to see if it is decoded incorrectly, which results in a "1" in the decodability array, and then repeating this process for every possible combination in the array. After the process is complete, each entry in the array is either a "1" or a "0".

We now define the following sets, which can be understood by referring to the decodability array. First, for a given $\mathbf{W}_k$ (row of the decodability array), denote the set of all $\mathbf{x}_t$ (columns of the decodability array) for which the $(\mathbf{x}_t, \mathbf{W}_k)$ pairs cannot be decoded correctly as $\Psi(\mathbf{W}_k)$, *i.e.*, $\Psi(\mathbf{W}_k) = \{\mathbf{x}_t | (\mathbf{x}_t, \mathbf{W}_k) \in \psi\}$. This is equivalent to the set of all columns with entries '1' in a given row $\mathbf{W}_k$ of the decodability array. Additionally, we let $\Psi(W)$ denote the set of all columns in the decodability array with '1' entries in every row, *i.e.*, $\Psi(W) = \{\mathbf{x}_t | (\mathbf{x}_t, \mathbf{W}_k) \in \psi, \forall\, \mathbf{W}_k \in W\}$, where we note that

$$\Psi(W) = \bigcap_{k=1}^{t^{\kappa p}} \Psi(\mathbf{W}_k). \tag{15}$$

In (14), the error probability is a function of $\Pr(\mathbf{U} = \mathbf{W}_k)$, which involves computing the probability of a particular set of κ check node inputs (from outside $\mathcal{G}(A)$) for each of the p iterations. If we are interested only in a lower bound on the probability of $(\mathbf{x}_t, \mathbf{W}_k)$ belonging to the absorbing region, this term can be eliminated from the calculation by including in the sum only entries whose columns have a '1' in every row, *i.e.*, the set $\Psi(W)$, which results in the following lower bound

$$\begin{aligned} \Pr(\xi(A)) &= \sum_{(\mathbf{x}_i,\mathbf{W}_k)\in\psi} \Pr(\mathbf{s} = \mathbf{x}_t) \cdot \Pr(\mathbf{U} = \mathbf{W}_k) \\ &\geq \sum_{\mathbf{x}_i\in\Psi(W),\mathbf{W}_k\in W} \Pr(\mathbf{s} = \mathbf{x}_t) \cdot \Pr(\mathbf{U} = \mathbf{W}_k) \\ &= \sum_{\mathbf{x}_i\in\Psi(W)} \Pr(\mathbf{s} = \mathbf{x}_t) \sum_{\mathbf{W}_k\in W} \Pr(\mathbf{U} = \mathbf{W}_k) \\ &= \sum_{\mathbf{x}_i\in\Psi(W)} \Pr(\mathbf{s} = \mathbf{x}_t). \end{aligned} \tag{16}$$

The lower bound in (16) implies that

$$\Pr(\xi(A)) \geq \lambda(A) \triangleq \sum_{\mathbf{x}_i\in\Psi(W)} \Pr(\mathbf{s} = \mathbf{x}_t), \tag{17}$$

so that instead of including all the pairs in the decodability array with '1' entries, we only need to include the columns with all '1' entries, which leads to the removal of the term $\Pr(\mathbf{U} = \mathbf{W}_k)$ from the expression for $\Pr(\xi(A))$. This makes the evaluation of the lower bound in (17) dependent only on the absorbing set A and not on the structure of the code containing A.[3]

V. Bounding the FER of an LDPC Code

In this section, we begin by deriving a lower bound on the FER of any LDPC code whose Tanner graph representation contains at least one instance of a given (a, b) absorbing set $\mathcal{G}(A)$ in Section V-A. We then provide a series of approximations in Section V-B to reduce the complexity of evaluating the bound. Finally, in Section V-C we provide some remarks concerning the application, evaluation, and merits of a code-independent bound on the FER of an LDPC code.

A. A Lower Bound on the FER of an LDPC Code

We define $\mathcal{E}(V)$ as the event that there is at least one bit error in the set of variable nodes V after the quantized received vector $\mathbf{r}$ is decoded using a quantized decoder operating on the full code graph for p iterations. Then the FER of the LDPC code can be written as

$$\text{FER} = \Pr(\mathcal{E}(V)) = \sum_{k=1}^{t^n} \Pr(\mathcal{E}(V)|\mathbf{r} = \mathbf{z}_k) \cdot \Pr(\mathbf{r} = \mathbf{z}_k), \tag{18}$$

since there are t^n possible realizations of $\mathbf{r}$.

[2] A related definition of an absorbing region was defined in [11]. We note that, generally, the decodability array can be constructed in this way for any problematic sub-graph and the corresponding "absorbing region" would refer to the portion of the array with '1' entries.

[3] If every column of the array has at least one "0" entry, that means that every possible input to the "absorbing set" can be decoded with some combination of check node inputs and we would obtain the trivial bound $\lambda = 0$; however, since such an object isn't problematic by our definition, a lower bound of zero makes sense.

Furthermore, since

$$\Pr\left(\bigcup_{i=1}^{N}\mathcal{E}(A_i)\right) \geq \sum_{i=1}^{N}\Pr\left(\mathcal{E}(A_i)\right) - \sum_{1\leq i<j\leq N}\Pr\left(\mathcal{E}(A_i)\cap\mathcal{E}(A_j)\right), \quad (29)$$

(27) and (29) can be combined to give the following lower bound

$$\text{FER} \geq \sum_{i=1}^{N}\Pr\left(\mathcal{E}(A_i)\right) - \sum_{1\leq i<j\leq N}\Pr\left(\mathcal{E}(A_i)\cap\mathcal{E}(A_j)\right). \quad (30)$$

We now assume that any two error events $\mathcal{E}(A_i)$ and $\mathcal{E}(A_j)$ associated with the same (a,b) absorbing set are independent, *i.e.*,

$$\Pr\left(\mathcal{E}(A_i)\cap\mathcal{E}(A_j)\right) = \Pr\left(\mathcal{E}(A_i)\right)\cdot\Pr\left(\mathcal{E}(A_j)\right). \quad (31)$$

This assumption is made for simplicity and is based on the observation that most pairs of a given absorbing set appearing in a code are disjoint, in the sense that they do not have any nodes in common. Using this assumption, the right hand side of (30) can be written as

$$N\Pr\left(\mathcal{E}(A)\right) - \binom{N}{2}\left(\Pr\left(\mathcal{E}(A)\right)\right)^2. \quad (32)$$

Further, as noted in [11], the fact that the channel LLRs in the error floor region are typically large implies that the chance of more than one absorbing set $\mathcal{G}(A)$ receiving low channel LLRs, and thus causing decoding errors, is small. This, combined with the fact that the second term in (32) will not have a significant impact (since $\Pr(\mathcal{E}(A))$ will be small and thus $(\Pr(\mathcal{E}(A)))^2 \ll \Pr(\mathcal{E}(A))$ in the error floor) and can thus be neglected, results in the following approximate lower bound on the FER in the error floor region of an LDPC code containing N instances of the absorbing set $\mathcal{G}(A)$:

$$\text{FER} \gtrapprox N\Pr\left(\mathcal{E}(A)\right) \geq N\lambda(A), \quad (33)$$

where the accuracy of the approximate bound in (33) depends on the tightness of the bound in (17). Furthermore, if $\mathcal{G}(A)$ is the *most harmful* or *dominant* absorbing set in a code, $N\lambda(A)$ represents an estimate of its FER performance in the error floor region.[5]

Expressions (28) and (33) represent a true lower bound and an approximate lower bound, respectively, valid in the error floor region, in terms of $\lambda(A)$, defined in (17). The multiplicities of the different absorbing sets needed to evaluate (33) may be derived either using analytical or semi-analytical methods, such as those given in [4], [18], [19],

B. Approximating the Lower Bound on FER

In this section, we propose a reduced complexity method to approximate $\lambda(A)$. Although the term $\Pr(\mathbf{U}=\mathbf{W}_k)$ was eliminated from the expression for $\Pr(\xi(A))$ in (16), thus making the lower bound code-independent and simplifying the expression, calculating $\lambda(A)$ in (17) still depends on finding $\Psi(W)$, which, in-turn requires examining all $\mathbf{W}_k \in W$ as shown in (15). In other words, all $t^{\kappa p}$ rows of the decodability array should be examined for each of the t^a columns $\mathbf{x}_i$. Therefore, instead of finding $\Psi(W)$, we consider the less computationally complex set

$$\hat{\Psi}(W) = \bigcap_{m=1}^{M}\Psi\left(\mathbf{W}_{k_m}\right), k_1, k_2, \ldots, k_m \in \{1,2,\ldots,t^{\kappa p}\}, \quad (34)$$

which involves examining only a subset of M rows of the decodability array. By properly choosing the M rows and finding the columns with all '1' entries in these rows, it is possible to obtain a good approximation to the set of columns with '1' entries in every row, allowing us to compute

$$\hat{\lambda}(A) \triangleq \sum_{\mathbf{x}_i\in\hat{\Psi}(W)}\Pr\left(\mathbf{s}=\mathbf{x}_i\right) \approx \lambda(A), \quad (35)$$

which results in the *approximate lower bound*[6]

$$\text{FER} \gtrapprox N\hat{\lambda}(A). \quad (36)$$

In the following, we explain how the approximate lower bound $\hat{\lambda}(A)$ is calculated. We first assume that M rows of the decodability array, denoted by $\mathbf{W}_{k_m}$ for $m=1,2,\ldots,M$, have been selected. The calculation of (35) then involves two steps:

1) Finding the set $\hat{\Psi}(W)$. This is achieved by operating the absorbing set decoder on A for each $(\mathbf{s}=\mathbf{x}_i, \mathbf{U}=\mathbf{W}_{k_m})$. Then, using (34), if the decoder fails to correctly decode $\mathbf{s}=\mathbf{x}_i$ for all the $\mathbf{W}_{k_m}, m=1,2,\ldots,M$, it follows that $\mathbf{s}=\mathbf{x}_i \in \hat{\Psi}(W)$. Otherwise, $\mathbf{s}=\mathbf{x}_i$ is discarded.
2) Summing the $\Pr(\mathbf{s}=\mathbf{x}_i)$ for all $\mathbf{s}=\mathbf{x}_i \in \hat{\Psi}(W)$, where $\Pr(\mathbf{s}=\mathbf{x}_i)$ is obtained using (8) and (9).

In order to obtain a computationally efficient approximation, we should choose rows expected to have a small number of '1's, since they eliminate more columns than rows with a large number of '1's. In other words, the M rows should be chosen as a set of check node input matrices $\mathbf{U}=\mathbf{W}_{k_m}$ that we expect to result in a small number of input vectors to the absorbing set $\mathbf{s}=\mathbf{x}_i$ that cannot be decoded correctly. Rows which we expect will lead to incorrect decoding of most input vectors $\mathbf{s}=\mathbf{x}_i$, on the other hand, are not useful. Therefore, we try to avoid such rows. Before proceeding, we review some important facts regarding the dynamics of absorbing sets in the high SNR region (with highly reliable input channel values). For such absorbing sets, after a certain number of iterations, it is common for the LLRs received by the check nodes in $C(A)$ from the variable nodes in $V\setminus A$ to grow rapidly and reach the maximum quantizer level ℓ_t (or the saturation level) within a few iterations [20]. For example, the analysis in [5] starts from the point where all the LLRs have already converged to ℓ_t. This motivates our choice of *Row Set I*, where we consider only the row

[5]In the case where more than one absorbing set is believed to be dominant, the maximum of all the lower bounds can be used to form an error estimate.

[6]We use this term to emphasize the fact that approximations are used in calculating $\lambda(A)$.

the probability of correct decoding. Therefore, we conclude that Assumption 1, which is based on ML decoders, is not necessarily valid for all MP decoders. This suggests choosing $\mathbf{U} = \mathbf{W}'$, where $\mathbf{W}'$ is a check node input matrix corresponding to some other row of the decodability array, can lead to correct decoding of some absorbing set input vectors $\mathbf{s} = \mathbf{x}_i$ when $\mathbf{U} = \mathbf{W}_{\max}$ does not lead to correct decoding. In [14]–[16], the authors model the dynamics of an absorbing set by applying Density Evolution (DE) to the messages coming from outside the absorbing set, where a Gaussian distribution for the LLRs received by the check nodes in $C(A)$ at each iteration is assumed. These distributions are represented by their mean and variance, which are shown to be increasing with iteration number. Here, we make use of those results and extend them to our code-independent framework by considering a check node input matrix $\mathbf{U} = \mathbf{W}_{\text{inc}}$ for which the LLRs increase gradually until reaching the maximum quantizer levels, thereby slowing down the convergence speed of the LLRs passed along the edges of the absorbing set decoder. To this end, the elements of $\mathbf{U}^1_{(\kappa\times 1)}$ are set equal to the lowest positive quantizer level. We then let this value increase with the iteration number μ, so that each of the $\frac{t}{2}$ positive quantizer levels is used h times before moving to the next larger level, for a total of $p = h\left(\frac{t}{2}\right)$ iterations, resulting in the check node input matrix[9]

$$\mathbf{U} = \mathbf{W}_{\text{inc}} = \left[\left[\ell_{\frac{t}{2}+1}\right]_{(\kappa\times h)} \cdots \left[\ell_t\right]_{(\kappa\times h)}\right]_{(\kappa\times p)}, \quad (39)$$

and the set

$$\hat{\Psi}(W) = \Psi\left(\mathbf{W}_{\text{inc}}\right). \quad (40)$$

As in the case of Row Set I, $\hat{\lambda}(A)$ is significantly less complex to calculate than $\lambda(A)$. The choice of h and the general trajectory of the increasing quantizer levels give us some options for choosing $\mathbf{U} = \mathbf{W}_{\text{inc}}$. According to our experience for a $(6,0)$ absorbing set with a 5-bit quasi-uniform quantizer, increasing h beyond 3 did not improve the approximate lower bound based on $\mathbf{U} = \mathbf{W}_{\text{inc}}$.

3) Row Set III: Finally, we can apply (34) to the two proposed sets $\Psi\left(\mathbf{W}_{\text{inc}}\right)$ and $\Psi\left(\mathbf{W}_{\max}\right)$ to obtain

$$\hat{\Psi}(W) = \Psi(\mathbf{W}_{\text{inc}}) \cap \Psi\left(\mathbf{W}_{\max}\right). \quad (41)$$

$\hat{\Psi}(W)$ again yields a $\hat{\lambda}(A)$ that is significantly less complex to calculate than $\lambda(A)$. The procedure to find the proposed $\hat{\Psi}(W)$ is described in Algorithm 1.

As noted previously, the calculation of $\hat{\lambda}(A)$ can be seen as a two-step process: finding the set $\hat{\Psi}(W)$ by operating the absorbing set decoder and then calculating the probability of $\hat{\Psi}(W)$ using (35). In Fig. 6, for the $(6,0)$ absorbing set, the approximate lower bound of (36) based on the set $\hat{\Psi}(W) = \Psi(\mathbf{W}_{\text{inc}}) \cap \Psi\left(\mathbf{W}_{\max}\right)$, the bound based only on the set $\Psi\left(\mathbf{W}_{\max}\right)$, and the simulated performance are shown for a $(3,61)$ array code [22] with a 5-bit quasi-uniform quantizer and an MSA decoder. We observe that in this case the approximate lower bound based on Row Set III gives a better result than the one obtained using only $\Psi\left(\mathbf{W}_{\max}\right)$, *i.e.*, Row Set I. It is worth noting that, to reduce the complexity of applying Algorithm 1, we start with $\mathbf{W}_{\max}$, since it is likely to eliminate the most input vectors $\mathbf{s} = \mathbf{x}_i$. Then we look for other rows that might succeed where $\mathbf{W}_{\max}$ fails, so that, after checking $\mathbf{W}_{\max}$, it is only necessary to run the absorbing set decoder for those $\mathbf{x}_i$'s with a '1' in the row of the decodability array associated with $\mathbf{W}_{\max}$.

[9]We do not use negative quantizer values because we are interested in rows that can decode most of the input patterns, and check node input matrices with negative values typically have a small probability of correct decoding (*e.g.*, see [17]).

Algorithm 1 Calculate $\hat{\lambda}(A) \triangleq \sum_{\mathbf{x}_i \in \hat{\Psi}(W)} \Pr(\mathbf{s} = \mathbf{x}_i)$

$\hat{\Psi}(W) \leftarrow \varnothing$ (the empty set)
for all $\mathbf{x}_i \in X$ **do**
  The absorbing set decoder tries to decode $\mathbf{x}_i$ with $\mathbf{U} = \mathbf{W}_{\max}$;
  if the absorbing set decoder fails **then**
    The absorbing set decoder tries to decode $\mathbf{x}_i$ with $\mathbf{U} = \mathbf{W}_{\text{inc}}$;
    if the absorbing set decoder fails **then**
      $\hat{\Psi}(W) = \hat{\Psi}(W) \cup \mathbf{x}_i$;
    end if
  end if
end for
return

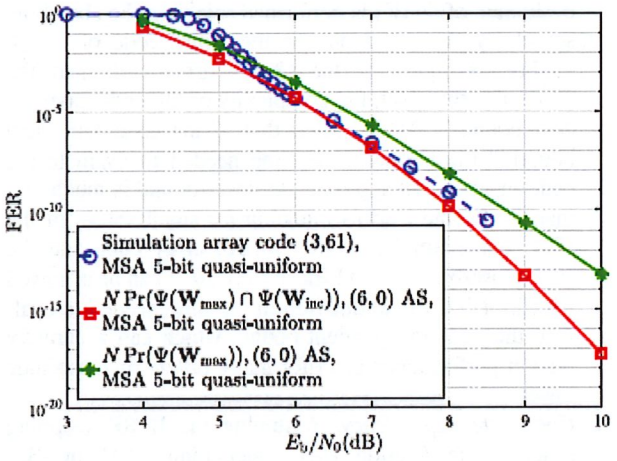

Fig. 6. Approximate lower bound of (36) based on Row Sets I and III for a $(6,0)$ absorbing set in a $(3,61)$ array code with a 5-bit quasi-uniform quantizer and an MSA decoder ($N = 2, 195, 390$).

C. Remarks

Due to its generality and simplicity, the code-independent approximate lower bound on the FER in (36) is a useful tool in predicting the high SNR performance of quantized LDPC decoders based on the presence of a given absorbing set (or general problematic sub-graph). Below we summarize this concept and pinpoint its strengths.

- *Application:* The lower bound $\lambda(A)$ indicates that any code containing at least one instance of a given absorbing set A cannot achieve an FER lower than that value. This statement, although not strictly true for the approximate

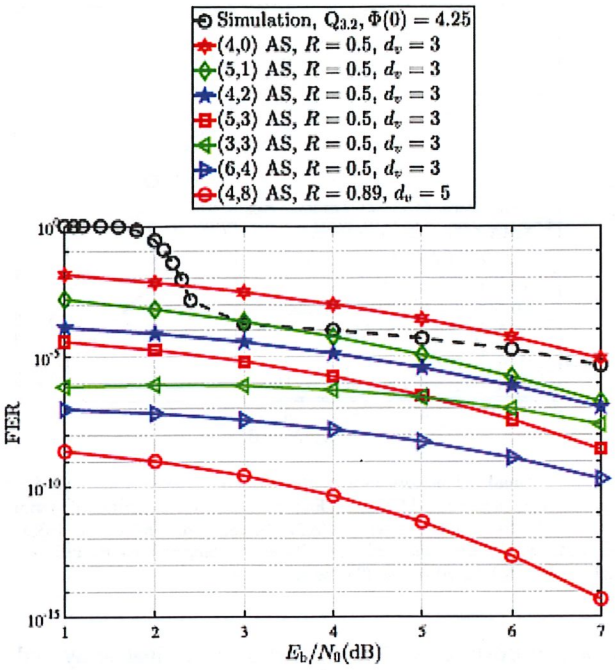

Fig. 7. Approximate lower bound $\hat{\lambda}$ based on various absorbing sets with different variable node degrees d_v and different rates R. The SPA is used and the quantization scheme is $Q_{3.2}$ with $\Phi(0) = 4.25$.

i.e., each variable node in the absorbing set is connected to 3 check nodes of the absorbing set. The results demonstrate that increasing a or b leads to a lower $\hat{\lambda}$, as expected. The simulated performance of the randomly constructed $R = 0.5$, $(3,6)$-regular code with length $n = 4000$ from [25] is also depicted in Fig. 7 for the same decoder and quantizer parameters. A $(4,2)$ absorbing set (or LETS), as shown in Fig. 1, exists in this code, and we see that the calculated $\hat{\lambda}$ provides a lower bound of its performance.

We have also computed $\hat{\lambda}$ for some absorbing sets with fixed variable node degrees $d_v > 3$. The results confirm that these absorbing sets typically have a lower $\hat{\lambda}$ than for absorbing sets with $d_v = 3$. (Again, this is to be expected, since larger variable node degrees generally correspond to stronger codes.) As an illustration, the bound for the $(4,8)$ absorbing set with $d_v = 5$, which was identified as the dominant absorbing set for the length $n = 2209$ and rate $R = 42/47 = 0.89$, $(5,47)$ array code using the $Q_{3.2}$ quantizer with $\Phi(0) = 4.25$ in [20], is seen to be much lower than any of the bounds for the $d_v = 3$ absorbing sets shown in Fig. 7.[12]

B. FER Performance Estimates $N\hat{\lambda}$ for Various Codes

In this section, the randomly constructed code of [25], several array codes [22], a Tanner code [26], and a Euclidean Geometry (EG) code [27] are considered. Based on the dominant absorbing set for each code, their estimated FER performance $N\hat{\lambda}$ is evaluated.[13]

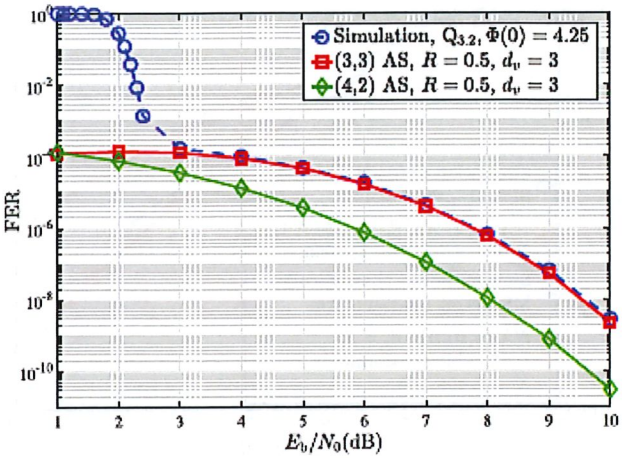

Fig. 8. Simulation results (dashed blue line) for a $(3,6)$ code with length $n = 4000$ and rate $R = 0.5$ of [2] and estimated FER performance $N\hat{\lambda}$ (solid lines) based on a $(4,2)$ absorbing set and a $(3,3)$ absorbing set with multiplicities $N = 1$ and $N = 171$, respectively, decoded using the SPA with a $Q_{3.2}$ uniform quantizer.

We first consider again the $(3,6)$ randomly constructed code of length $n = 4000$ and rate $R = 0.5$ [25]. Fig. 8 shows the simulated FER performance obtained with an SPA decoder and a 6-bit uniform quantizer. Also shown are the FER estimates $N\hat{\lambda}$ for the $(4,2)$ and $(3,3)$ absorbing sets, where the multiplicities $N = 1$ and $N = 171$, respectively, were obtained from [19]. It is observed that, even though a single $(3,3)$ absorbing set (also classified as a LETS) is much less harmful than a single $(4,2)$ absorbing set (see Fig. 7), when the multiplicities are considered the $(3,3)$ absorbing set is dominant in the error floor region for this decoder.[14]

In Fig. 9, the simulated FER performance (dashed blue circles) is shown for the $(3,61)$ array code [22] of length $n = 3721$ and rate $R = 0.9514$ when decoded using the SPA and a 6-bit uniform quantizer. Also shown is the estimated FER performance $N\hat{\lambda}$ (solid red dots), where $\hat{\lambda}$ was computed for a $(4,2)$ absorbing set and the same decoder with multiplicity $N = 334,890$ obtained from [4]. We see that the performance estimate is accurate, since this absorbing set is dominant. The simulated performance is also shown for the MSA decoder and a 5-bit quasi-uniform (dashed blue crosses). We observe here that the MSA outperforms the SPA, as previously noted in [17], for this code. The estimated FER performance $N\hat{\lambda}$ is shown (solid red triangles), where $\hat{\lambda}$ was computed for a $(6,0)$ absorbing set and the same MSA decoder, with multiplicity $N = 2,195,390$ obtained from [18].

[12]These results do not yet consider the multiplicity of the absorbing set, which plays a significant role for structured codes.

[13]A code might have different dominant absorbing sets, depending on the decoding algorithm. Also, the dominant absorbing set might depend on SNR. Here, we consider the absorbing sets that are dominant in the error floor (high SNR) region.

[14]We see from the results in Fig. 8 that it is not unusual for the simulated performance to diverge from the estimated performance in the waterfall region, where the concept of a single dominant absorbing set is no longer relevant.

the binary-input AWGN channel. The bounds and estimates are general, in the sense that they apply to any code containing a given absorbing set and only depend on the rate of the code. For a given absorbing set, we used the concept of an absorbing set decoder to derive a lower bound on its FER by finding the set of channel input patterns to the absorbing set that are not correctly decoded under any circumstances imposed by the LLRs coming from outside the absorbing set. We then showed that, instead of considering all possible realizations of the LLRs coming into the check nodes of the absorbing set, it suffices to examine only a few realizations to obtain a good approximate lower bound on the FER, thus making its computation much simpler. We also showed that, if the multiplicity of the dominant absorbing set in a code is known, an accurate estimate of the code's FER performance in the error floor (high SNR) region is obtained. Finally, using various examples, we showed that the approximate lower bound and the FER performance estimates, which can be evaluated much faster than performing conventional Monte-Carlo simulations, are useful tools in predicting the high SNR behavior of quantized LDPC decoders.

REFERENCES

[1] R. Gallager, "Low-density parity-check codes," *IRE Trans. Inf. Theory*, vol. IT-8, no. 1, pp. 21–28, Jan. 1962.

[2] D. J. C. MacKay and M. S. Postol, "Weaknesses of Margulis and Ramanujan-Margulis low-density parity-check codes," *Electron. Notes Theor. Comput. Sci.*, vol. 74, no. 10, pp. 97–104, Oct. 2003.

[3] T. Richardson, "Error floors of LDPC codes," in *Proc. 41st Annu. Allerton Conf. Commun., Control, Comput.*, Oct. 2003, pp. 1426–1435.

[4] L. Dolecek, Z. Zhang, V. Anantharam, M. J. Wainwright, and B. Nikolic, "Analysis of absorbing sets and fully absorbing sets of array-based LDPC codes," *IEEE Trans. Inf. Theory*, vol. 56, no. 1, pp. 181–201, Jan. 2010.

[5] A. Tomasoni, S. Bellini, and M. Ferrari, "Thresholds of absorbing sets in low-density parity-check codes," *IEEE Trans. Commun.*, vol. 65, no. 8, pp. 3238–3249, Aug. 2017.

[6] G. B. Kyung and C.-C. Wang, "Finding the exhaustive list of small fully absorbing sets and designing the corresponding low error-floor decoder," *IEEE Trans. Commun.*, vol. 60, no. 6, pp. 1487–1498, Jun. 2012.

[7] M. Karimi and A. H. Banihashemi, "On characterization of elementary trapping sets of variable-regular LDPC codes," *IEEE Trans. Inf. Theory*, vol. 60, no. 9, pp. 5188–5203, Sep. 2014.

[8] S. Laendner, T. Hehn, O. Milenkovic, and J. B. Huber, "The trapping redundancy of linear block codes," *IEEE Trans. Inf. Theory*, vol. 55, no. 1, pp. 53–63, Jan. 2009.

[9] O. Milenkovic, E. Soljanin, and P. Whiting, "Asymptotic spectra of trapping sets in regular and irregular LDPC code ensembles," *IEEE Trans. Inf. Theory*, vol. 53, no. 1, pp. 39–55, Jan. 2007.

[10] Y. Zhang and W. E. Ryan, "Toward low LDPC-code floors: A case study," *IEEE Trans. Commun.*, vol. 57, no. 6, pp. 1566–1573, Jun. 2009.

[11] L. Dolecek, P. Lee, Z. Zhang, V. Anantharam, B. Nikolic, and M. Wainwright, "Predicting error floors of structured LDPC codes: Deterministic bounds and estimates," *IEEE J. Sel. Areas Commun.*, vol. 27, no. 6, pp. 908–917, Jun. 2009.

[12] H. Hatami, D. G. M. Mitchell, D. J. Costello, and T. Fuja, "Lower bounds for quantized LDPC min-sum decoders based on absorbing sets," in *Proc. 55th Annu. Allerton Conf. Commun., Control, Comput.*, Oct. 2017, pp. 694–699.

[13] H. Xiao, A. H. Banihashemi, and M. Karimi, "Error rate estimation of low-density parity-check codes decoded by quantized soft-decision iterative algorithms," *IEEE Trans. Commun.*, vol. 61, no. 2, pp. 474–484, Feb. 2013.

[14] J. Sun, O. Y. Takeshita, and M. P. Fitz, "Analysis of trapping sets for LDPC codes using a linear system model," in *Proc. 42nd Annu. Allerton Conf.*, Monticello, IL, USA, Sep./Oct. 2004, pp. 1701–1702.

[15] S. Zhang and C. Schlegel, "Controlling the error floor in LDPC decoding," *IEEE Trans. Commun.*, vol. 61, no. 9, pp. 3566–3575, Sep. 2013.

[16] B. K. Butler and P. H. Siegel, "Error floor approximation for LDPC codes in the AWGN channel," *IEEE Trans. Inf. Theory*, vol. 60, no. 12, pp. 7416–7441, Dec. 2014.

[17] X. Zhang and P. H. Siegel, "Quantized iterative message passing decoders with low error floor for LDPC codes," *IEEE Trans. Commun.*, vol. 62, no. 1, pp. 1–14, Jan. 2014.

[18] H. Liu, L. Ma, and J. Chen, "On the number of minimum stopping sets and minimum codewords of array LDPC codes," *IEEE Commun. Lett.*, vol. 14, no. 7, pp. 670–672, Jul. 2010.

[19] Y. Hashemi and A. H. Banihashemi, "New characterization and efficient exhaustive search algorithm for leafless elementary trapping sets of variable-regular LDPC codes," *IEEE Trans. Inf. Theory*, vol. 62, no. 12, pp. 6713–6736, Dec. 2016.

[20] Z. Zhang, L. Dolecek, B. Nikolic, V. Anantharam, and M. J. Wainwright, "Design of LDPC decoders for improved low error rate performance: Quantization and algorithm choices," *IEEE Trans. Commun.*, vol. 57, no. 11, pp. 3258–3268, Nov. 2009.

[21] F. Angarita, J. Valls, V. Almenar, and V. Torres, "Reduced-complexity min-sum algorithm for decoding LDPC codes with low error-floor," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 61, no. 7, pp. 2150–2158, Jul. 2014.

[22] J. L. Fan, "Array codes as low-density parity-check codes," in *Proc. 2nd Int. Symp. Turbo Codes*, 2000, pp. 545–546.

[23] H. Hatami, D. G. M. Mitchell, D. J. Costello, and T. Fuja, "A modified min-sum algorithm for quantized LDPC decoders," in *Proc. IEEE Int. Symp. Inf. Theory*, Jul. 2019, pp. 2434–2438.

[24] B. Vasic. *Trapping Set Ontology*. Accessed: Nov. 14, 2019. [Online]. Available: http://www2.engr.arizona.edu/~vasiclab/project.php?id=9

[25] D. MacKay. *Gallager Code Resources*. Accessed: Nov. 14, 2019. [Online]. Available: http://www.inference.phy.cam.ac.uk/mackay/S0.html

[26] R. M. Tanner, D. Sridhara, A. Sridharan, T. E. Fuja, and D. J. Costello, Jr., "LDPC block and convolutional codes based on circulant matrices," *IEEE Trans. Inf. Theory*, vol. 50, no. 12, pp. 2966–2984, Dec. 2004.

[27] Y. Kou, S. Lin, and M. P. C. Fossorier, "Low-density parity-check codes based on finite geometries: A rediscovery and new results," *IEEE Trans. Inf. Theory*, vol. 47, no. 7, pp. 2711–2736, Nov. 2001.

Homayoon Hatami (S'14–M'19) was born in Kermanshah, Iran, in 1987. He received the B.Sc. degree in electrical engineering-communications and the M.Sc. degree in electrical engineering-communication systems from the University of Tehran, Tehran, Iran, in 2010 and 2013, respectively, and the Ph.D. degree in electrical engineering-communication systems from the University of Notre Dame, Notre Dame, IN, USA, in 2019. Since then, he has been with Samsung Semiconductor Inc., San Diego, CA, USA, where he is currently a Senior Engineer.

David G. M. Mitchell (M'10–SM'16) received the Ph.D. degree in electrical engineering from the University of Edinburgh, U.K., in 2009. From 2009 to 2015, he held Post-Doctoral Research Associate and then Visiting Assistant Professor positions with the Department of Electrical Engineering, University of Notre Dame, USA. Since 2015, he has been an Assistant Professor with the Klipsch School of Electrical and Computer Engineering, New Mexico State University, USA. His research interests lie in the area of digital communications, with emphasis on error control coding and information theory.