

On the Minimum Trapping Distance of Repeat Accumulate Accumulate Codes

Jörg Kliewer

Klipsch School of Electrical and Computer Engineering,
New Mexico State University, Las Cruces, NM 88003, USA
Email: jkliewer@nmsu.edu

Kamil S. Zigangirov, Daniel J. Costello, Jr.

Department of Electrical Engineering,
University of Notre Dame, Notre Dame, IN 46556, USA
Email: {kzigangi, costello.2}@nd.edu

Abstract—We consider the ensemble of codes formed by a serial concatenation of a repetition code with two accumulators through uniform random interleavers. For this ensemble, asymptotic expressions for the normalized minimum trapping distance are derived. We employ a variant of the Gallager-Zyablov-Pinsker bit flipping decoding algorithm on a binary symmetric channel, where the analysis is based on the factor graph of the code. In particular, we show that the minimum trapping distance can be determined by solving a non-linear optimization problem. As a result we find that the minimum trapping distance grows linearly with block length for code rates of $1/3$ and smaller, albeit with very small growth rate coefficients.

I. INTRODUCTION

The serial concatenation of constituent codes combined with iterative decoding has led to powerful code constructions. The simplest construction is the repeat-accumulate (RA) code [1], which consists only of a repetition code, an interleaver, and an accumulator. These codes have lower decoding complexity than serially concatenated constructions with more complex constituent codes. An even more important benefit is that RA codes have an extremely low encoding complexity of $O(1)$, which makes them well suited for power-limited environments. However, as shown in [2–5], RA codes are not asymptotically good, i.e., their minimum distance does not grow linearly with block length, which has a significant impact on the performance in the error-floor regime.

In this paper, we consider repeat-accumulate-accumulate (RAA) codes, which are comprised of the serial concatenation of an outer repetition code with two accumulators connected through random interleavers. It was shown in [5], [6] that, for rates equal to $1/3$ or smaller, the resulting code ensemble is asymptotically good, where in [5] the exact growth rate coefficients are determined. While this result is useful in assessing the error-floor behavior for maximum-likelihood (ML) decoding, it does not provide an accurate characterization of the low bit error rate (BER) performance regime in conjunction with iterative decoding.

In particular, for low-density parity check (LDPC) codes, *stopping set* [7] and *trapping set* [8], [9] classifications have been proposed to characterize the error-floor properties for the binary erasure channel (BEC) in the former case and,

more generally, for non-erasure channels in the latter case, assuming belief-propagation (BP) iterative decoding. Trapping sets were first introduced as so-called *near codewords* in [8]. This concept was extended in [9], where trapping sets were used to predict the error floor performance in the very-low BER regime. An (a, b) trapping set for a given parity check matrix and its corresponding Tanner graph is defined in [9] as a collection of a variable nodes for which the subgraph induced by the a variable nodes contains b check nodes with odd degree. The intuition behind this definition is that, if $a > b$, potentially unreliable messages within the induced subgraph may not be corrected by reliable messages from outside the trapping set due to the relatively small number of odd degree check nodes within the subgraph. Further, the work in [10] proposes a method of deriving stopping and trapping set weight enumerators and analyzes the corresponding asymptotic behavior of a BP decoder. Recently, a variation of the trapping set definition has been stated in the form of *absorbing sets* [11], [12] in order to separate the effects of oscillatory behavior in the decoder from convergence to a non-codeword.

In the following, we present an analysis of the minimum trapping distance for an RAA code ensemble. As pointed out above, even though related work has been carried out in the context of LDPC codes in [10], such a characterization does not seem to exist for double serially concatenated ensembles such as the RAA ensemble¹. In this initial study, we focus solely on communicating over a binary symmetric channel (BSC). For decoding, we employ a variant of the majority-logic (bit flipping) iterative decoding algorithm introduced by Gallager [14] and Zyablov and Pinsker [15], resp., for LDPC codes on a factor graph representation of the RAA code. It was shown in [15] that, for LDPC codes of block length N , this algorithm has complexity $O(N \log N)$ and is able to correct $O(N)$ channel errors on a BSC. Thus, for small enough channel error probabilities, the block error probability goes to zero exponentially in N as $\exp(-O(N))$.

The paper is organized as follows. In Section II we address the encoder and the corresponding factor graph for the RAA code. Section III presents the bit flipping decoding algorithm,

This work was supported in part by NSF Grants CCF-0830666, CCF-0830650, and NASA Grant NNX07AK53G.

¹However, a stopping set analysis of RAA codes has recently been proposed in [13].

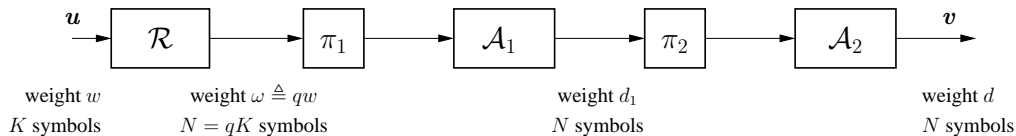


Fig. 1. Repeat accumulate-accumulate encoder, rate $R = 1/q$.

and in Section IV a lower bound on the minimum trapping distance is derived, and numerical results are presented.

II. FACTOR GRAPH REPRESENTATION OF RAA CODES

In this section we introduce a factor graph representation of RAA codes, where the RAA encoder is depicted in Fig. 1. The binary input sequence $\mathbf{u} = [u_1, u_2, \dots, u_K]$ of length K has weight w , \mathcal{R} denotes the repetition code of rate $R = 1/q$, which leads to a code sequence of weight qw and length $N = qK$. We consider the ensemble of all possible interleavers π_1 and π_2 by using the uniform interleaver approach [16], where each possible interleaver realization has the same probability $1/N!$. The permuted interleaver output sequences are applied to the recursive convolutional codes $\mathcal{A}_{1/2}$ with generator polynomials $g(D) = 1/(1 + D)$ (accumulator), leading to sequences of weight d_1 and d , respectively.

For decoding, we consider a factor graph representation of the RAA code, which is shown in Fig. 2 for $q = 3$. The small boxes represent check nodes, the circles variable nodes, and π_1 and π_2 denote the interleavers, respectively. The received code symbols r_n , $n = 1, 2, \dots, N$, appear as variable nodes in the first layer. At the output of the first layer, the sequence $\tilde{\mathbf{v}}_1 = [\tilde{v}_{1,1}, \tilde{v}_{1,2}, \dots, \tilde{v}_{1,N}]$ represents the inputs of accumulator \mathcal{A}_2 and the outputs of interleaver π_2 , which has $\mathbf{v}_1 = [v_{1,1}, v_{1,2}, \dots, v_{1,N}]$ as its input sequence. Further, both the input sequence of accumulator \mathcal{A}_1 and the output sequence of interleaver π_1 is denoted by $\tilde{\mathbf{u}} = [\tilde{u}_1, \tilde{u}_2, \dots, \tilde{u}_N]$. From Fig. 2 we obtain the following relations:

$$\tilde{v}_{1,n} = r_{n-1} + r_n, \quad (1)$$

$$r_n = \sum_{k=1}^n \tilde{v}_{1,k}, \quad (2)$$

$$\tilde{u}_n = v_{1,n-1} + v_{1,n}, \quad (3)$$

$$v_{1,n} = \sum_{k=1}^n \tilde{u}_k. \quad (4)$$

The k -th variable node, $k = 1, 2, \dots, K$, in the third level shares edges with q check nodes. Let the subset of these q values from $\tilde{\mathbf{u}}$ be denoted as $\tilde{\mathbf{u}}_k \triangleq [u_{k,1}, u_{k,2}, \dots, u_{k,q}]$ (see Fig. 2). We now introduce the value η_k corresponding to the k -th variable node as

$$\eta_k \triangleq \frac{w_H(\tilde{\mathbf{u}}_k)}{q}. \quad (5)$$

We also define the corresponding sequence $\boldsymbol{\eta} = [\eta_1, \eta_2, \dots, \eta_K]$.

The received sequence \mathbf{r} is a valid codeword if η_k , $k = 1, 2, \dots, K$, only takes on the values zero or one, i.e., if the

$\bar{\mathbf{u}}_k$ are either all-zero or all-one subsets. Further, we define so-called *syndrome* symbols s_k , $k = 1, 2, \dots, K$, as

$$s_k \triangleq \begin{cases} 0 & \text{if } \eta_k = 0 \text{ or } \eta_k = 1, \\ 1 & \text{otherwise,} \end{cases} \quad (6)$$

and the syndrome sequence as $\mathbf{s} = [s_1, s_2, \dots, s_k, \dots, s_K]$. Note that this definition is different from the usual definition of a syndrome. However, our definition has the analogous property that \mathbf{r} is a codeword only if the syndrome Hamming weight $w_H(\mathbf{s}) = 0$.

Lemma 1. *There are*

$$N! \frac{N!}{(q!)^K}$$

ways to choose the factor graph for the rate-1/q RAA code.

Proof: We first consider all possible edges between the K variable nodes on the third level and the N check nodes on the second level. The first variable node of the third level can be connected to q check nodes on the second level in

$$\frac{N(N-1) \dots (N-q+1)}{q!}$$

ways, the second variable node of the third layer can be connected in

$$\frac{(N-q)(N-q-1) \dots (N-2q+1)}{q!}$$

ways, and so on. In total, we have $N!/(q!)^K$ possible edges between the second and the third layer. Combining this with the fact that the edges between the N check nodes on the first level and the N variable nodes on the second level can be arranged in $N!$ different ways, the lemma is proved. ■

III. DECODING ALGORITHM

In order to decode the noisy RAA codewords, we consider a variant of the Gallager-Zyablov-Pinsker bit flipping decoding algorithm for LDPC codes [15]. The motivation for employing this decoding algorithm is that a very intuitive trapping set characterization can be obtained, as explained below.

The decoding is carried out on the factor graph of the code (see Fig. 2). In the first decoding iteration, those symbols in \mathbf{r} are marked for which flipping reduces the syndrome weight $w_H(\mathbf{s})$. After flipping those symbols simultaneously, the resulting flipped sequence is considered as a new received sequence \mathbf{r} , and the next decoding iteration starts. In particular, the decoder carries out the following steps:

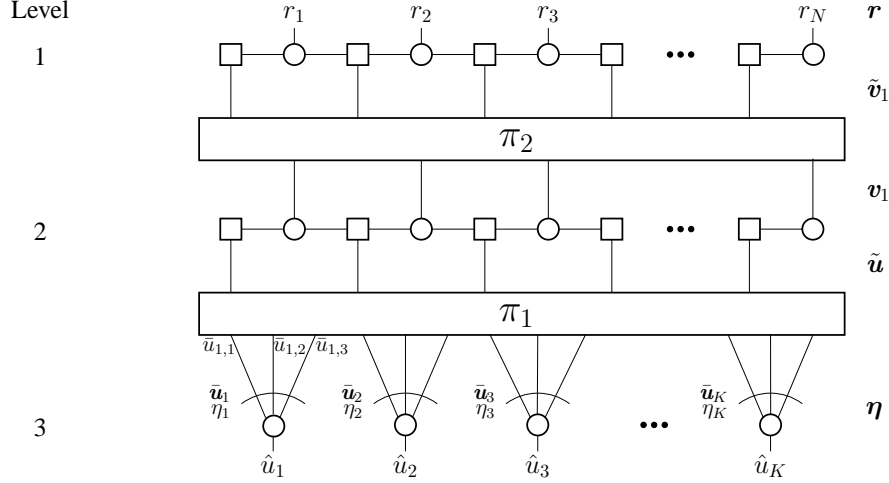


Fig. 2. Factor graph for an RAA code with $q = 3$. The small boxes represent check nodes, the circles variable nodes, and π_1 and π_2 denote the interleavers, respectively.

- 1) Based on the received sequence \mathbf{r} , the sequence $\tilde{\mathbf{v}}_1$ is computed according to (1). Then, knowing the structure of the interleaver π_2 , \mathbf{v}_1 can be obtained.
- 2) The sequence $\tilde{\mathbf{u}}$ is then determined via (3), and finally $\boldsymbol{\eta}$ and \mathbf{s} are computed by evaluating (5) and (6). If $w_H(\mathbf{s}) = 0$, decoding is stopped.
- 3) The decoder finds candidate received symbols r_n that will decrease the syndrome weight $w_H(\mathbf{s})$ if flipped. Then, all candidate symbols are flipped simultaneously. If there are no candidate symbols available, decoding is stopped and a decoding failure is declared.
- 4) The resulting flipped sequence is considered as a new received sequence. Go to Step 1).

Definition 1. *The sequence \mathbf{r} is called a trapping set if there are no candidate symbols available that lead to a reduction of the syndrome weight.*

Note that this definition of trapping sets is different from the definitions in [9], [10] given for LDPC codes in the sense that it identifies exactly those conditions that lead to a decoding failure.

In the following, without loss of generality, we consider the case when the *transmitted* information sequence \mathbf{u} of length K is the all-zero sequence, and we define the trapping distance of a trapping set \mathbf{r} as its Hamming weight.

Definition 2. *The parameter d_{trap} is called the minimum trapping distance if there are no trapping sets with trapping distance $d_0 < d_{\text{trap}}$, and there is at least one trapping set with trapping distance d_{trap} .*

In the next section we derive a lower bound on the minimum trapping distance by computing ensemble-average weight spectra for trapping sets.

IV. TRAPPING DISTANCE LOWER BOUND FOR THE RAA ENSEMBLE

We assume that the all-zero codeword has been sent, the received sequence \mathbf{r} of length qK has Hamming weight d , the decoded sequence $\tilde{\mathbf{u}}$ has Hamming weight ω , and the decoded sequences \mathbf{v}_1 and $\tilde{\mathbf{v}}_1$ both have Hamming weight d_1 . We can then state the following Lemmas.

Lemma 2. *If $w_H(\mathbf{r}) = d$ and $w_H(\tilde{\mathbf{v}}_1) = d_1$, then the sequence \mathbf{r} can be chosen in*

$$\binom{d-1}{\lceil \frac{d_1}{2} \rceil - 1} \binom{N-d}{\lfloor \frac{d_1}{2} \rfloor}$$

different ways.

The proof can be found in [1].

Lemma 3. *Given the sequence $\tilde{\mathbf{v}}_1$ of weight d_1 , the sequence \mathbf{v}_1 can be chosen in $d_1!(N-d_1)!$ different ways.*

Proof: There are $d_1!$ ways to connect the symbols with value one in the sequences \mathbf{v}_1 and $\tilde{\mathbf{v}}_1$ in the factor graph, and $(N-d_1)!$ ways to connect the symbols with the value zero. ■

Now consider a corrupted received sequence \mathbf{r} at the decoder input. At every decoder iteration, each of the possible values of η_k is specified in (5). We define the set $\boldsymbol{\nu} \triangleq (\nu_0, \nu_1, \dots, \nu_q)$, where each ν_ℓ counts the number of occurrences of the values ℓ/q , $\ell = 0, 1, \dots, q$, in the sequence $\boldsymbol{\eta} = [\eta_1, \eta_2, \dots, \eta_K]$. The set $\boldsymbol{\nu}$ is called the *constraint composition* and satisfies the following relations:

$$\sum_{\ell=0}^q \nu_\ell = K \quad \text{and} \quad \sum_{\ell=0}^q \ell \nu_\ell = \omega. \quad (7)$$

Lemma 4. *A sequence $\boldsymbol{\eta}$ with constraint composition $\boldsymbol{\nu}$ can be chosen in*

$$\binom{K}{\nu_0, \nu_1, \dots, \nu_q} = \frac{K!}{\nu_0! \nu_1! \dots \nu_q!}$$

different ways.

The proof follows from the definition of $\boldsymbol{\nu}$. By combining (7) and Lemma 4 we obtain the following lemma.

Lemma 5. For a given sequence $\tilde{\mathbf{u}}$ of weight ω , the sequence $\boldsymbol{\eta}$ can be chosen in

$$\sum_{\boldsymbol{\nu}: \sum_{\ell=0}^q \nu_{\ell}=K, \sum_{\ell=0}^q \ell \nu_{\ell}=\omega} \binom{K}{\nu_0, \nu_1, \dots, \nu_q}$$

different ways.

We now prove the following key lemma.

Lemma 6. Given a sequence $\tilde{\mathbf{u}}$ of weight ω and a sequence $\boldsymbol{\eta}$ with constraint composition $\boldsymbol{\nu}$, there are

$$\frac{\omega!(N-\omega)!}{\prod_{\ell=0}^q (\ell!(q-\ell)!)^{\nu_{\ell}}}$$

ways to arrange the edges connecting the second and third levels in the factor graph of the code.

Proof: The proof is analogous to the proof of Lemma 1. First, the variable nodes corresponding to $\eta_k = 0$ can only be connected to check nodes associated with symbols $\tilde{u}_k = 0$ in

$$\frac{(N-\omega)(N-\omega-1)\cdots(N-\omega-q\nu_0+1)}{(q!)^{\nu_0}} \quad (8)$$

different ways. We fix these edges in the factor graph. Then, the variable nodes associated with $\eta_k = 1/q$ consist of a single edge associated with $\tilde{u}_k = 1$ and $q-1$ edges with $\tilde{u}_k = 0$, so we have

$$\frac{(N-\omega-q\nu_0)(N-\omega-q\nu_0-1)\cdots(N-\omega-q\nu_0-(q-1)\nu_1+1)}{((q-1)!)^{\nu_1}} \cdot \frac{\omega(\omega-1)\cdots(\omega-\nu_1+1)}{(1!)^{\nu_1}} \quad (9)$$

different sets of edges. Now, we continue with the values $\eta_k = \ell/q$, $\ell = 2, \dots, q-1$ in the same way, and by fixing all those edges, for $\eta_k = 1$ we end up with

$$\frac{\left(\omega - \sum_{\ell=1}^{q-1} \nu_{\ell}\right) \cdots 2 \cdot 1}{(q!)^{\nu_q}} \quad (10)$$

different ways of arranging the edges between the variable nodes associated with $\eta_k = 1$ and the corresponding check nodes for $\tilde{u}_k = 1$. By combining the cases represented by (8), (9), and (10), and the corresponding equations for $\ell = 2, \dots, q-1$, the lemma is proved. ■

We now enumerate all possible combinations of weights $w_H(\mathbf{r}) = d$, $w_H(\mathbf{v}_1) = d_1$, and $w_H(\tilde{\mathbf{u}}) = \omega$ for a given information sequence \mathbf{u} with constraint composition $\boldsymbol{\nu}$ using an input-output weight enumerator function (IOWEF) $\bar{A}_{d,d_1,\omega}(\boldsymbol{\nu})$. Here, we assume that $\boldsymbol{\nu}$ satisfies the conditions of (7). An expression for the ensemble average IOWEF

$\bar{A}_{d,d_1,\omega}(\boldsymbol{\nu})$ can then be obtained by combining Lemmas 1, 2, 3, 4, and 6, which gives rise to the following theorem.

Theorem 1. The ensemble-average IOWEF $\bar{A}_{d,d_1,\omega}$ for an RAA code ensemble with rate $R = 1/q$ and a given constraint composition $\boldsymbol{\nu} = (\nu_0, \nu_1, \dots, \nu_q)$ equals

$$\bar{A}_{d,d_1,\omega}(\boldsymbol{\nu}) = \frac{\binom{d_1-1}{\lceil \frac{\omega}{2} \rceil - 1} \binom{N-d_1}{\lfloor \frac{\omega}{2} \rfloor} \binom{d-1}{\lceil \frac{d_1}{2} \rceil - 1} \binom{N-d}{\lfloor \frac{d_1}{2} \rfloor}}{\binom{N}{\omega} \binom{N}{d_1}} \cdot \frac{(q!)^K}{\prod_{\ell=0}^q (\ell!(q-\ell)!)^{\nu_{\ell}}} \quad (11)$$

The following corollary immediately follows from the above theorem.

Corollary 1. The ensemble-average IOWEF $\bar{A}_{d,d_1,q,w}(\boldsymbol{\nu})$ with $\boldsymbol{\nu} = (K-w, 0, \dots, 0, w)$ from Theorem 1 is equivalent to the (unconstrained) ensemble-average IOWEF for the RAA code derived in [5].

In the following, we employ Theorem 1 to derive a lower bound on d_{trap} . It can be seen from the factor graph in Fig. 2 that each flipping of a symbol r_n in the first level causes the flipping of at most four symbols in the sequence $\tilde{\mathbf{u}}$. Since we have assumed that the transmitted sequence is the all-zero sequence and the received sequence \mathbf{r} is a BSC error pattern of weight d , we do not need to consider the case that a flipping candidate r_n is connected to a variable node on the third level of the factor graph corresponding to $\eta_k = 0$, which would lead to an increase of the syndrome weight. Thus, only the following cases need to be considered:

- 1) There is a connection between a flipping candidate r_n and a variable node corresponding to the value $\eta_k = 1/q$. In this case, since the all-zero codeword has been sent, we again only need to address the case where the syndrome weight is decreased, i.e., where the corresponding symbol \tilde{u}_n changes its value to zero.
- 2) There is a connection between a flipping candidate r_n and a variable node corresponding to the value $\eta_k = \ell/q$, $2 \leq \ell \leq q-1$. In this case, the syndrome weight cannot increase and we assume in the following that the syndrome weight is always unchanged².
- 3) There is a connection between a flipping candidate r_n and a variable node corresponding to the value $\eta_k = 1$, whose flipping increases the syndrome weight.

Since case 2) above does not lead to an increase of the syndrome weight, we only need to focus on cases 1) and 3). Thus, if the number of check nodes $\tilde{u}_n = 1$ sharing an edge with a variable node $\eta_k = 1/q$ is larger than the number of

²Here we neglect the case that the ℓ corresponding values in $\tilde{\mathbf{u}}$ could be changed to zero at the same time by a single flip of r_n and therefore decrease the syndrome weight, since the probability of this event typically is very small.

check nodes $\tilde{u}_n = 1$ sharing an edge with a variable node $\eta_k = 1$, i.e.,

$$\nu_1 - q\nu_q > 0, \quad (12)$$

then at least one received symbol r_n can always be identified as a flipping candidate such that the syndrome weight is decreased. As a consequence, (12) represents a sufficient condition for the absence of a trapping set.

We now employ the result in (12) to define the ensemble average weight enumerating function (WEF) for received sequences of weight d that constitute trapping sets as

$$\bar{A}_d = \sum_{\omega=1}^N \sum_{d_1=1}^N \sum_{\substack{\nu: \sum_{\ell=0}^q \nu_\ell = K \\ \sum_{\ell=0}^q \ell \nu_\ell = \omega \\ \nu_1 - q\nu_q \leq 0}} \bar{A}_{d,d_1,\omega}(\boldsymbol{\nu}). \quad (13)$$

From (13), we obtain

$$\bar{A}_d \leq N^2 \max_{1 \leq d_1 \leq N, 1 \leq \omega \leq N} \sum_{\substack{\nu: \sum_{\ell=0}^q \nu_\ell = K \\ \sum_{\ell=0}^q \ell \nu_\ell = \omega \\ \nu_1 - q\nu_q \leq 0}} \bar{A}_{d,d_1,\omega}(\boldsymbol{\nu}), \quad (14)$$

and we can derive an upper bound for the sum in (14) as

$$\begin{aligned} & \sum_{\substack{\nu: \sum_{\ell=0}^q \nu_\ell = K \\ \sum_{\ell=0}^q \ell \nu_\ell = \omega \\ \nu_1 - q\nu_q \leq 0}} \bar{A}_{d,d_1,\omega}(\boldsymbol{\nu}) \leq \\ & \min_{s, \lambda < 0} \underbrace{\sum_{\nu} \bar{A}_{d,d_1,\omega}(\boldsymbol{\nu}) \exp \left[s \left(\sum_{\ell=0}^q \ell \nu_\ell - \omega \right) \right] \exp[\lambda(\nu_1 - q\nu_q)]}_{\triangleq \bar{M}_{d,d_1,\omega}(s, \lambda)}, \end{aligned} \quad (15)$$

where s and $\lambda < 0$ are auxiliary variables and the summation is carried out over all $\boldsymbol{\nu}$ such that $\sum_{\ell=0}^q \nu_\ell = K$. In order to obtain a tight upper bound, the right hand side in (15) must be minimized over s and $\lambda < 0$. The quantity $\bar{M}_{d,d_1,\omega}(s, \lambda)$ in (15) represents a modified ensemble-average IOWEF. Thus, the ensemble-average WEF \bar{A}_d can be upper bounded as

$$\bar{A}_d \leq N^2 \max_{1 \leq d_1 \leq N, 1 \leq \omega \leq N} \min_{s, \lambda < 0} \bar{M}_{d,d_1,\omega}(s, \lambda). \quad (16)$$

For the sake of brevity, we now restrict ourselves to the case $q = 3$. By combining (11) and (15) and subsequently employing the multinomial theorem, $\bar{M}_{d,d_1,\omega}(s, \lambda)$ can be written as

$$\begin{aligned} \bar{M}_{d,d_1,\omega}(s, \lambda) = & \frac{\binom{d_1-1}{\lceil \frac{\omega}{2} \rceil - 1} \binom{N-d_1}{\lfloor \frac{\omega}{2} \rfloor} \binom{d-1}{\lceil \frac{d_1}{2} \rceil - 1} \binom{N-d}{\lfloor \frac{d_1}{2} \rfloor}}{\binom{N}{\omega} \binom{N}{d_1}} \\ & e^{-s\omega} \cdot \left(1 + 3e^{s+\lambda} + 3e^{2s} + e^{3(s-\lambda)} \right)^K. \end{aligned} \quad (17)$$

Using the asymptotic form of Stirling's approximation, (17) yields

$$\bar{M}_{d,d_1,\omega}(s, \lambda) = \exp[(F(\alpha, \beta, \rho, s, \lambda) + o(N))N], \quad (18)$$

where $\alpha \triangleq \omega/N$, $\beta \triangleq d_1/N$, and $\rho \triangleq d/N$ are normalized weights, and the function $F(\cdot)$ is defined as

$$F(\alpha, \beta, \rho, s, \lambda) \triangleq f(\alpha, \beta, \rho) + g(\alpha, s, \lambda), \quad (19)$$

where

$$\begin{aligned} f(\alpha, \beta, \rho) = & -H(\beta) - H(\rho) + H\left(\frac{\beta - \alpha/2}{1 - \alpha}\right)(1 - \alpha) \\ & + \alpha \ln 2 + H\left(\frac{\rho - \beta/2}{1 - \beta}\right)(1 - \beta) + \beta \ln 2 \end{aligned} \quad (20)$$

and

$$g(\alpha, s, \lambda) = \frac{1}{3} \ln \left(1 + 3e^{s+\lambda} + 3e^{2s} + e^{3(s-\lambda)} \right) - s\alpha. \quad (21)$$

The derivation of $F(\alpha, \beta, \rho, s, \lambda)$ can be extended to the case $q > 3$ in a straightforward way. The only difference is that the function $g(\alpha, s, \lambda)$ must be modified: for example, for $q = 4$, $g(\alpha, s, \lambda)$ becomes

$$g(\alpha, s, \lambda) = \frac{1}{4} \ln \left(1 + 4e^{s+\lambda} + 6e^{2s} + 4e^{3s} + e^{4(s-\lambda)} \right) - s\alpha. \quad (22)$$

We now define the function

$$r_{\text{trap}}(\rho) \triangleq \lim_{N \rightarrow \infty} \frac{1}{N} \ln \bar{A}_{\rho N}, \quad (23)$$

where $r_{\text{trap}}(\rho)$ has the following interpretation: if the function is negative for some ρ , $\rho_{\text{trap}} > \rho > 0$, then crosses zero and is positive for $\rho > \rho_{\text{trap}}$, it follows that, for almost all codes in the ensemble, the minimum trapping distance is lower bounded by $d_{\text{trap}} \geq \rho_{\text{trap}}N$ as the block length N tends to infinity.

By combining (16), (18), and (19), and inserting the resulting expression for \bar{A}_d in (23), we obtain the following theorem and corollary.

Theorem 2. *The function $r_{\text{trap}}(\rho)$ can be upper bounded as*

$$\begin{aligned} r_{\text{trap}}(\rho) & \leq \hat{r}_{\text{trap}}(\rho) \\ & = \max_{0 < \alpha < 0.5, 0 < \beta < 0.5} \left(f(\alpha, \beta, \rho) + \min_{s, \lambda < 0} g(\alpha, s, \lambda) \right), \end{aligned} \quad (24)$$

where the function $f(\cdot)$ is given by (20) and the function $g(\cdot)$ is given by (21) (or the corresponding expressions for $q > 3$).

Corollary 2. *Let $\rho_0 > 0$ be such that $\hat{r}_{\text{trap}}(\rho_0) < 0$. Then almost all RAA codes in the ensemble do not contain trapping sets of weight $\rho_0 N$ or less.*

In order to compute a lower bound ρ'_{trap} on the asymptotic normalized minimum trapping distance ρ_{trap} , we must find the zero crossing of $\hat{r}_{\text{trap}}(\rho)$ by numerically solving (24) with constrained non-linear optimization techniques. Table I shows the resulting lower bounds for $q \geq 3$. We see that the minimum trapping distance $d_{\text{trap}} = \rho_{\text{trap}}N$ grows linearly with block length, even though the growth coefficients are very small.

The reason is that the employed bit flipping decoder has only moderate error correcting capabilities and is inferior to performing BP decoding on the factor graph or iterative decoding on the constituent codes. Note that, for $q = 2$, we are

TABLE I
LOWER BOUND ρ'_{TRAP} ON THE ASYMPTOTIC NORMALIZED MINIMUM TRAPPING DISTANCES FOR THE RAA CODE ENSEMBLE, A BIT FLIPPING DECODER, AND DIFFERENT RATES $R = 1/q$.

q	ρ'_{trap}
3	$3.00 \cdot 10^{-8}$
4	$9.50 \cdot 10^{-8}$
5	$1.05 \cdot 10^{-6}$
6	$4.70 \cdot 10^{-6}$
7	$1.33 \cdot 10^{-5}$
8	$2.70 \cdot 10^{-5}$
9	$4.41 \cdot 10^{-5}$
10	$6.15 \cdot 10^{-5}$

not able to obtain linear minimum trapping distance growth, which is consistent with the fact that linear minimum distance growth cannot be proved either in this case [5]. Finally, Fig. 3 displays the values of ρ'_{trap} from Table I versus the inverse code rate $q = 1/R$.

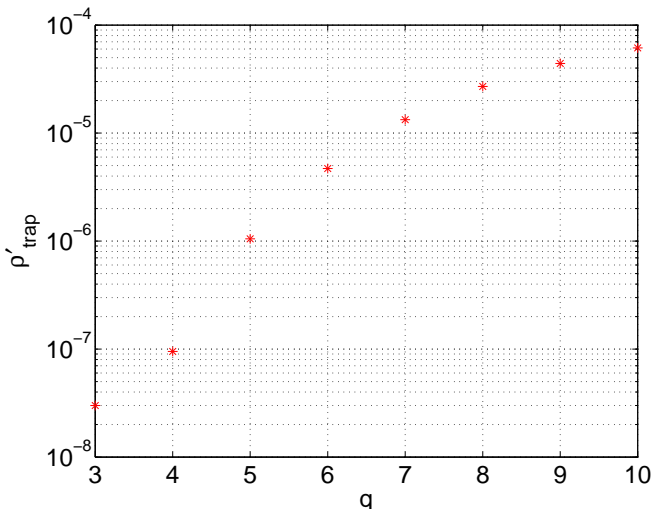


Fig. 3. Lower bound ρ'_{trap} on the normalized asymptotic trapping distance for the RAA code ensemble versus the inverse rate $q = 1/R$.

V. CONCLUSION

In this paper we have addressed the asymptotic minimum trapping distance of an RAA code ensemble, where a variant of the Gallager-Zyablov-Pinsker bit flipping decoding algorithm on the factor graph of the code was employed. In particular, we have proposed a complete trapping set characterization for the bit flipping decoder on the BSC. The lower bound on the normalized minimum trapping distance for the RAA code ensemble is obtained by employing a uniform interleaver approach and combining the resulting expression with a decoder condition for the existence of trapping sets. Finally, a numerical evaluation is obtained via a constrained nonlinear

optimization approach. The main result is that, for all rates smaller than or equal to $R = 1/3$, the minimum trapping distance grows linearly with block length, although the growth rate coefficients are very small.

Future work will include a further analysis of the bit flipping decoder; in particular, investigating the effects of introducing errors by the flipping of correctly received symbols. We will also focus on extending the results to AWGN channels, and on performing trapping set analyses for different decoders, such as BP and iterative decoding of the individual constituent codes, for which the results of this paper could serve as initial performance benchmarks.

REFERENCES

- [1] D. Divsalar, H. Jin, and R. J. McEliece, "Coding theorems for 'turbo-like' codes," in *Proc. 36th Annual Allerton Conf. Commun., Control, Computing*, Monticello, IL, Sept. 1998, pp. 201–210.
- [2] L. Bazzi, M. Mahdian, and D. A. Spielman, "The minimum distance of turbo-like codes," Submitted to *IEEE Trans. Inf. Theory*, May 2003.
- [3] N. Kahale and R. Urbanke, "On the minimum distance of parallel and serially concatenated codes," in *Proc. IEEE Int. Symposium on Inform. Theory*, Cambridge, MA, Aug. 1998, p. 31.
- [4] A. Perotti and S. Benedetto, "An upper bound on the minimum distance of serially concatenated convolutional codes," *IEEE Trans. Inf. Theory*, vol. 52, no. 12, pp. 5501–5509, Dec. 2006.
- [5] J. Kliewer, K. S. Zigangirov, and D. J. Costello, Jr., "New results on the minimum distance of repeat-multiple-accumulate codes," in *Proc. 45th Annual Allerton Conference on Communication, Control, and Computing*, Monticello, IL, September 2007, pp. 1097–1102.
- [6] H. D. Pfister and P. H. Siegel, "The serial concatenation of rate-1 codes through uniform random interleavers," *IEEE Trans. Inf. Theory*, vol. 49, no. 6, pp. 1425–1438, June 2003.
- [7] C. Di, D. Proietti, E. Telatar, T. J. Richardson, and R. Urbanke, "Finite-length analysis of low-density parity check codes on the binary erasure channel," *IEEE Trans. Inf. Theory*, vol. 48, no. 6, pp. 1570–1579, June 2002.
- [8] D. J. C. MacKay and M. Postol, "Weaknesses of Margulis and Ramanujan-Margulis low-density parity-check codes," *Electron. Notes Theor. Comput. Sci.*, vol. 74, 2003.
- [9] T. J. Richardson, "Error floors of LDPC codes," in *Proc. 41st Annual Allerton Conference on Communication, Control, and Computing*, Monticello, IL, September 2003, pp. 1426–1435.
- [10] 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, January 2007.
- [11] Z. Zhang, L. Dolecek, B. Nikolić, V. Anantharam, and M. Wainwright, "Investigation of error floors of structured low-density parity-check codes via hardware simulation," in *Proc. GLOBECOM*, San Francisco, CA, November 2006, pp. 1–6.
- [12] L. Dolecek, Z. Zhang, V. Anantharam, M. Wainwright, and B. Nikolić, "Analysis of absorbing sets for array-based LDPC codes," in *Proc. IEEE Int. Conf. on Commun.*, Glasgow, U.K., July 2007, pp. 6261–6268.
- [13] A. Graell i Amat and E. Rosnes, "Stopping set analysis of hybrid concatenated codes," in preparation, 2008.
- [14] R. G. Gallager, *Low-density parity-check codes*, MIT Press, Cambridge, MA, 1963.
- [15] V. V. Zyablov and M. S. Pinsker, "Estimates of the error-correction complexity of Gallager's low-density codes," *Problems of Information Transmission*, vol. 11, no. 1, pp. 18–28, January-March 1976.
- [16] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara, "Serial concatenation of interleaved codes: performance analysis, design, and iterative decoding," *IEEE Trans. Inf. Theory*, vol. 44, no. 3, pp. 909–926, May 1998.