ISIT 2009, Seoul, Korea, June 28 - July 3, 2009

Trapping Set Enumerators for Repeat Multiple Accumulate Code Ensembles

Christian Koller*, Alexandre Graell i Amat†, Jörg Kliewer‡, and Daniel J. Costello, Jr.*

*Department of Electrical Engineering, University of Notre Dame, Notre Dame, IN 46556, USA
Email: {dcostel1, ckoller}@nd.edu
† Department of Electronics, Institut TELECOM-TELECOM Bretagne, 29238 Brest, France
Email: alexandre.graell@telecom-bretagne.eu
‡Klipsch School of Electrical and Computer Engineering, New Mexico State University, Las Cruces, NM 88003, USA
Email: jkliewer@nmsu.edu

***Abstract*—The serial concatenation of a repetition code with two or more accumulators has the advantage of a simple encoder structure. Furthermore, the resulting ensemble is asymptotically good and exhibits minimum distance growing linearly with block length. However, in practice these codes cannot be decoded by a maximum likelihood decoder, and iterative decoding schemes must be employed. For low-density parity-check codes, the notion of trapping sets has been introduced to estimate the performance of these codes under iterative message passing decoding. In this paper, we present a closed form finite length ensemble trapping set enumerator for repeat multiple accumulate codes by creating a trellis representation of trapping sets. We also obtain the asymptotic expressions when the block length tends to infinity and evaluate them numerically.**

I. Introduction

Turbo-like codes [1], as well as LDPC codes [2], can perform close to the Shannon limit using suboptimal iterative decoding schemes. However, these codes typically exhibit an error floor at medium to high signal-to-noise ratios (SNRs). In [3], the height of the error floor of LDPC codes was linked to so-called "near codewords". Later, in [4], this concept was generalized to trapping sets, substructures in the Tanner graph of a code that may cause the iterative message passing decoder to fail. For certain LDPC codes, small trapping sets, rather than the minimum distance of the code, dominate the error floor performance.

Asymptotic spectra of trapping sets in LDPC code ensembles were computed in [5] for regular and irregular LDPC codes and in [6] for protograph-based codes. It was shown that there exist LDPC codes that exhibit a minimum trapping set size growing linearly with block length, for certain types of trapping sets.

In turbo-like codes, the concatenation of simple component codes through interleavers can lead to powerful code constructions. The simplest examples are repeat multiple accumulate (RMA) codes. These codes have a low encoding complexity of $O(1)$ and can be decoded using relatively few iterations. Furthermore, it has been shown in [7] and [8] that the double serially concatenated repeat accumulate accumulate (RAA) code of rate 1/3 or smaller is asymptotically good and exhibits minimum distance growing linearly with block length.

Like LDPC codes, turbo-like codes are decoded in an iterative fashion. Commonly, the component codes are decoded with a maximum a posteriori probability (MAP) decoding algorithm and the extrinsic information provided by a component decoder functions as a priori information for another. For RMA codes, the turbo decoder can be represented as a message passing decoder [9], similar to the belief propagation decoder, albeit with a different message passing schedule. Thus the turbo decoder may also be susceptible to trapping sets. To predict the error floor of a code one generally needs to have full knowledge of the trapping sets that dominate the error floor, i.e., one needs to know their graph structure and enumerate their multiplicities, and to find the probability that the decoder gets trapped in a particular set. The latter not only depends on the graph structure of the trapping set but also on the channel model, the decoding algorithm, and the particular decoder implementation that is used.

In this paper we address the first part of the problem, the enumeration of subgraphs in an RAA code. We derive a closed form trapping set enumerator (TSE) for general (a, b) trapping sets, as defined in [4] and [5]. A general (a, b) trapping set for a given Tanner graph is a set of a variable nodes that induces a subgraph containing b odd degree check nodes, which can be thought of as *unsatisfied* checks, and an arbitrary number of even degree check nodes. If there are only a few unsatisfied check nodes and a sufficiently large number of erroneous variable nodes, the iterative message passing decoder may not be able to correct the erroneous nodes. The TSE is the average number of (a, b) trapping sets in the ensemble composed of all possible interleaver realizations. We also derive asymptotic expressions for the TSE and analyze them.

II. Trapping set enumerators for repeat accumulate accumulate code ensembles

The encoder structure of an RAA code $\mathcal{C}^{\text{RAA}}$ is shown in Fig. 1. It is a serial concatenation of a repetition code

This work was partly supported by NSF grants CCF05-15012, CCF08-30666, NASA grant NNX07AK536, and the Marie Curie Intra-European Fellowship within the 6th European Community Framework Programme.

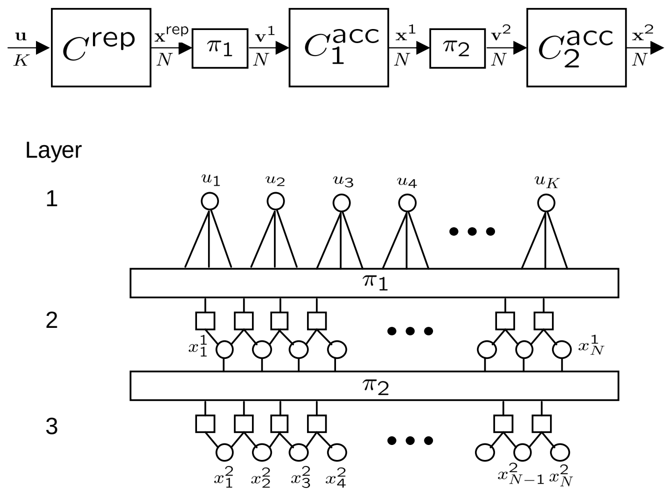


Fig. 1. Block diagram and factor graph ($q = 3$) of an RAA encoder. The circles represent variable nodes and the boxes check nodes, respectively.

$\mathcal{C}^{\text{rep}}$ of rate $R_{\text{rep}} = 1/q$ and two identical rate-1, memory-1, accumulate codes $\mathcal{C}_l^{\text{acc}}$, $l = \{1, 2\}$, with generator polynomials $g(D) = 1/(1+D)$, connected by interleavers π_1 and π_2. For the repetition code $\mathcal{C}^{\text{rep}}$, we denote the binary input sequence of length K by $\mathbf{u} = [u_1, \ldots, u_K]$ and the binary output sequence of length N by $\mathbf{x}^{\text{rep}} = [x_1^{\text{rep}}, \ldots, x_N^{\text{rep}}]$. Likewise, for encoder $\mathcal{C}_l^{\text{acc}}$, $\mathbf{v}^l = [v_1^l, \ldots, v_N^l]$ and $\mathbf{x}^l = [x_1^l, \ldots, x_N^l]$ denote the input sequence and the codeword, respectively, where both are of length N. Note that $\mathbf{v}^1 = \pi_1(\mathbf{x}^{\text{rep}})$ and $\mathbf{v}^2 = \pi_2(\mathbf{x}^1)$. The overall code rate is $R = K/N$.

The factor graph of an RAA code is also depicted in Fig. 1 for a repetition factor of $q = 3$. The circles represent variable nodes while the boxes represent check nodes. The information symbols $\mathbf{u}$ correspond to the variable nodes of the first layer and their degree is equal to the repetition factor q. The variable nodes of the second layer correspond to the output of the first accumulator $\mathbf{x}^1$, and the variable nodes of the third layer to the output of the second accumulator $\mathbf{x}^2$, respectively. The input bits of the accumulators are represented by the variable nodes of the next higher layer. Only the variable nodes in the third layer are transmitted through the channel. This implies that, initially, only variable nodes in the third layer can be in error, while the others have a neutral initial value. However, in the first decoding iteration, the variable nodes in layers 1 and 2 get values assigned based on the received sequence. If there are trapping sets containing variable nodes in those layers, erroneous values that were assigned during the first iteration may never be corrected and may cause the iterative decoder to fail. Therefore, we consider the whole graph when enumerating for trapping sets.

Let $\bar{A}_{a,b}^{\mathcal{C}^{\text{RAA}}}$ be the ensemble-average TSE of an RAA code ensemble, i.e., the average number of (a, b) trapping sets. With reference to Fig. 1, we denote by w the number of information bits that participate in an (a, b) trapping set of $\mathcal{C}^{\text{RAA}}$. Also, let a_l^i, a_l^o, and b_l be the number of variable nodes corresponding to input bits, the number of variable nodes corresponding to code bits, and the number of unsatisfied checks, respectively, of code $\mathcal{C}_l^{\text{acc}}$ involved in an (a, b) trapping set of $\mathcal{C}^{\text{RAA}}$.

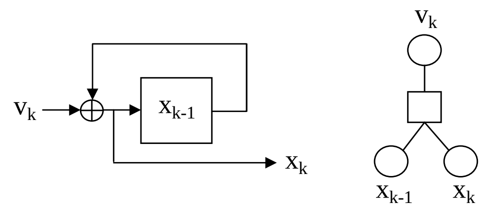


Fig. 2. Block diagram and factor graph representation of an accumulate code.

To proceed, we must define the trapping set enumerators of the component codes $A_{w,qw}^{\mathcal{C}^{\text{rep}}}$ and $A_{a_l^i,a_l^o,b_l}^{\mathcal{C}_l^{\text{acc}}}$, for $l = \{1, 2\}$. Since there are no check nodes in layer 1 of the factor graph, $A_{w,qw}^{\mathcal{C}^{\text{rep}}} = \binom{K}{w}$ is the input-output weight enumerator (IOWE) of the repetition code, giving the number of codewords in $\mathcal{C}^{\text{rep}}$ of input weight w and output weight qw, while $A_{a_l^i,a_l^o,b_l}^{\mathcal{C}_l^{\text{acc}}}$ is the input-output trapping set enumerator (IOTSE) of code $\mathcal{C}_l^{\text{acc}}$, denoting the number of trapping sets in $\mathcal{C}_l^{\text{acc}}$ consisting of a_l^i input variable nodes (i.e., variable nodes corresponding to information bits), a_l^o output variable nodes (i.e., variable nodes corresponding to code bits), and b_l unsatisfied checks.

With these definitions, the ensemble average TSE $\bar{A}_{a,b}^{\mathcal{C}^{\text{RAA}}}$ can be computed using the uniform interleaver concept [10] as:

$$\begin{aligned}\bar{A}_{a,b}^{\mathcal{C}^{\text{RAA}}} &= \sum_{\substack{w,a_1^o,a_2^o:\ w+a_1^o+a_2^o=a \\ b_1,b_2:\ b_1+b_2=b}} \frac{A_{w,qw}^{\mathcal{C}^{\text{rep}}} A_{qw,a_1^o,b_1}^{\mathcal{C}_1^{\text{acc}}} A_{a_1^o,a_2^o,b_2}^{\mathcal{C}_2^{\text{acc}}}}{\binom{N}{qw}\binom{N}{a_1^o}} \\ &= \sum_{\substack{w,a_1^o,a_2^o:\ w+a_1^o+a_2^o=a \\ b_1,b_2:\ b_1+b_2=b}} \bar{A}_{w,a_1^o,b_1,a_2^o,b_2}^{\mathcal{C}^{\text{RAA}}},\end{aligned} \tag{1}$$

where $\bar{A}_{w,a_1^o,b_1,a_2^o,b_2}^{\mathcal{C}^{\text{RAA}}}$ is called the ensemble-average conditional TSE.

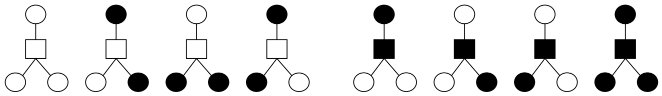
The evaluation of (1) requires the computation of $A_{a_l^i,a_l^o,b_l}^{\mathcal{C}_l^{\text{acc}}}$, which will be presented in the next section. The extension of (1) to more than two serially concatenated accumulators is straightforward.

III. Input-output trapping set enumerator for the accumulate code

In the following, we address the computation of the IOTSE $A_{a_l^i,a_l^o,b_l}^{\mathcal{C}_l^{\text{acc}}}$ of an accumulate code by considering an equivalent trellis representation of trapping sets in the factor graph. In Fig. 2, the block diagram of an accumulate code and a single section of the corresponding factor graph are depicted. From the figure we obtain the following relation:

$$v_k = x_{k-1} + x_k. \tag{2}$$

Four different 3-tuples (v_k, x_{k-1}, x_k) are possible, namely $(0, 0, 0)$, $(0, 1, 1)$, $(1, 0, 1)$, and $(1, 1, 0)$, such that the parity check is satisfied. Their factor graph representations are shown in Fig. 3(a), where black circles represent non-zero symbols and empty circles represent zero symbols. Now consider an (a, b) trapping set of $\mathcal{C}^{\text{RAA}}$, and assume that (some) of the variable nodes of accumulate code $\mathcal{C}^{\text{acc}}$ corresponding to (v_k, x_{k-1}, x_k) participate in the trapping set and cause an unsatisfied check. Again, only four different configurations are possible. They are depicted in Fig. 3(b), where black circles



(a) (b)

Fig. 3. Factor graph representations of an accumulate code.

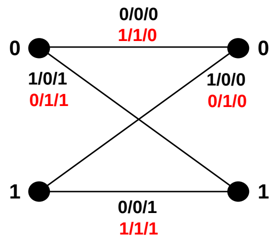


Fig. 4. Extended Trellis Section.

correspond to erroneous symbols and a black box means that the check is unsatisfied. Note that all possible trapping sets can be obtained by properly combining the eight factor graph sections of Fig. 3.

For enumeration purposes, it is simpler to refer to an equivalent trellis representation. Assign to the variable nodes and the check nodes in Fig. 3 that participate in a trapping set (the black circles and boxes) the value 1. Then the eight factor graph sections in Fig. 3 can be conveniently represented by the equivalent trellis section of Fig. 4. We call this the *extended trellis section* since it extends the standard trellis section of an accumulate code to include all possible trapping sets. Each edge between two trellis states is labeled with a binary 3-tuple $s_i/c/s_o$, where s_i denotes the input symbol, s_o denotes the output symbol, and c is 1 if the check node in the corresponding equivalent factor graph representation is unsatisfied. The four labels in black correspond to the four configurations of Fig. 3(a) and define the standard trellis section of an accumulate code, while the four labels in red correspond to the four configurations of Fig. 3(b). Now the IOTSE of the accumulate code can be computed from the trellis representation of Fig. 4 by considering a trellis consisting of N concatenated trellis sections like the one in Fig. 4 and enumerating all possible paths. The IOTSE is given in closed form in the following Theorem.

Theorem 1. *Let (a^i, a^o, b) be a trapping set with a^i information variable nodes, a^o code variable nodes, and b unsatisfied checks. The input-output trapping set enumerator (IOTSE) for the rate-1, memory-1, convolutional encoder $\mathcal{C}^{\text{acc}}$ with generator polynomial $g(D) = 1/(1+D)$, terminated to the all-zero state at the end of the trellis, and with input and output block length N, can be given in closed form as:*

$$A_{a^i,a^o,b} = \sum_m \sum_n \binom{N-a^o}{m}\binom{a^o-1}{m-1} \cdot \binom{a^o-m}{\frac{a^i+b}{2}-n-m}\binom{N-a^o-m}{n}\binom{2m}{\frac{a^i-b}{2}+m}, \tag{3}$$

where m and n must satisfy the constraints

$$\begin{aligned} m &\geq \frac{|a^i-b|}{2}, \quad m \leq \min\{a^o, N-a^o\}, \\ n &\geq \frac{a^i+b}{2} - a^o, \quad n \leq N - a^o - m. \end{aligned} \tag{4}$$

Proof: Consider the extended trellis section of the encoder $g(D) = 1/(1+D)$ in Fig. 4. Denote by n the number of length-one error events $1/1/0$ from the zero state to the zero state, called type-1 error events, and by m the number of error events that leave the zero state, and remerge later to the zero state, called type-2 error events. Further, let a^i, a^o, and b be the number of information variable nodes, the number of code variable nodes, and the number of unsatisfied checks, respectively, participating in the trapping set. Also, let w^t denote the total input weight associated with the transitions $0 \to 1$ (from state zero to state one) and $1 \to 0$ (from state one to state zero) in the m type-2 error events.

Only type-2 error events are responsible for the weight at the output of the accumulator. From [11], we know that the number of permutations of m type-2 error events resulting in an output weight of a^o is

$$\binom{N-a^o}{m}\binom{a^o-1}{m-1}.$$

(Here, the transitions away from and back to the zero state are not only caused by the input weight w^t but also by $2m - w^t$ unsatisfied check nodes.) The m type-2 error events include $a^o - m$ transitions from the one state to the one state ($1 \to 1$), and the input weight associated with the transitions $1 \to 1$ is

$$w_{1\to1} = a^i - n - w^t. \tag{5}$$

Moreover, the following equality holds:

$$w^t = \frac{a^i - b}{2} + m. \tag{6}$$

Also, due to termination, $a^i + b$ is even. From (5) and (6) it now follows that $w_{1\to1} = \frac{a^i+b}{2} - n - m$. This weight can be ordered in $\binom{a^o-m}{\frac{a^i+b}{2}-n-m}$ different ways, which gives the third binomial coefficient in (3). On the other hand, there are $N - a^o - m$ transitions from the zero state to the zero state with an associated input weight n. Therefore, we obtain the term $\binom{N-a^o-m}{n}$. The last binomial coefficient in (3) results from the ordering of the w^t ones in the $2m$ transitions $0 \to 1$ and $1 \to 0$, in $\binom{2m}{\frac{a^i-b}{2}+m}$ ways.

To summarize, the number of paths in the extended trellis consisting of n type-1 error events and m type-2 error events is given by:

$$\binom{N-a^o}{m}\binom{a^o-1}{m-1}\binom{a^o-m}{\frac{a^i+b}{2}-n-m}\binom{N-a^o-m}{n}\binom{2m}{\frac{a^i-b}{2}+m}.$$

The result for the encoder $g(D) = 1/(1+D)$ follows by summing over all possible values of n and m. ■

Corollary 1. *For $b = 0$, the expression in (3) reduces to the well-known IOWE for the rate-1, memory-1, accumulate code [11].*

IV. Asymptotic ensemble trapping set enumerator

In order to determine the asymptotic spectral shape of the trapping sets associated with a particular code ensemble, as the block length N tends to infinity, we define the normalized logarithmic asymptotic TSE $r^{\mathcal{C}}(\alpha,\beta)$ of a code ensemble $\mathcal{C}$ as

$$r^{\mathcal{C}}(\alpha,\beta) = \limsup_{N\to\infty} \frac{\ln \bar{A}^{\mathcal{C}}_{a,b}}{N}, \tag{7}$$

where $\alpha = a/N$, $\beta = b/N$, and the supremum is taken over all intermediate variables. We also define the functions $f^{\mathcal{C}^{\text{rep}}}$ and $f^{\mathcal{C}_l^{\text{acc}}}$ as the asymptotic behavior of the IOWE of a repeat code and the asymptotic behavior of the IOTSE of an accumulate code $\mathcal{C}_l^{\text{acc}}$, respectively:

$$\begin{aligned} f^{\mathcal{C}^{\text{rep}}}(\omega) &= \lim_{N\to\infty} \frac{\ln A^{\mathcal{C}^{\text{rep}}}_{w,qw}}{N} \\ f^{\mathcal{C}_l^{\text{acc}}}(\alpha_l^i,\alpha_l^o,\beta_l) &= \lim_{N\to\infty} \frac{\ln A^{\mathcal{C}_l^{\text{acc}}}_{a_l^i,a_l^o,b_l}}{N}, \quad l=1,2, \end{aligned} \tag{8}$$

where $\omega = w/K$, $\alpha_l^i = a_l^i/N$, $\alpha_l^o = a_l^o/N$, and $\beta_l = b_l/N$.

Using Stirling's approximation for binomial coefficients $\binom{n}{k} \overset{n\to\infty}{\longrightarrow} e^{n\mathbb{H}\left(\frac{k}{n}\right)}$, where $\mathbb{H}(\cdot)$ is the binary entropy function with natural logarithms, the functions in (8) can be written as:

$$f^{\mathcal{C}^{\text{rep}}}(\omega) = \frac{1}{q}\mathbb{H}(\omega), \tag{9}$$

and

$$\begin{aligned} f^{\mathcal{C}_l^{\text{acc}}}(\alpha_l^i,\alpha_l^o,\beta_l) = \sup_{\mu_l,\nu_l} \quad & (1-\alpha_l^o)\mathbb{H}\left(\frac{\mu_l}{1-\alpha_l^o}\right) + \\ + \alpha_l^o\mathbb{H}\left(\frac{\mu_l}{\alpha_l^o}\right) &+ (\alpha_l^o-\mu_l)\mathbb{H}\left(\frac{\alpha_l^i+\beta_l-2(\nu_l+\mu_l)}{2(\alpha_l^o-\mu_l)}\right) + \\ + (1-\alpha_l^o-\mu_l)\mathbb{H}\left(\frac{\nu_l}{1-\alpha_l^o-\mu_l}\right) &+ 2\mu_l\mathbb{H}\left(\frac{\alpha_l^i-\beta_l+2\mu_l}{4\mu_l}\right), \end{aligned} \tag{10}$$

where we have defined the normalized quantities $\mu_l = m_l/N$ and $\nu_l = n_l/N_l$.

Then, using (8-10) and (1) in (7), the asymptotic TSE of a code ensemble $\mathcal{C}^{\text{RAA}}$ can be written as:

$$\begin{aligned} r^{\mathcal{C}^{\text{RAA}}}(\alpha,\beta) = \sup_{\substack{\alpha=\omega/q+\alpha_1^o+\alpha_2^o \\ \beta=\beta_1+\beta_2}} & f^{\mathcal{C}^{\text{rep}}}(\omega) + f^{\mathcal{C}_1^{\text{acc}}}(\omega,\alpha_1^o,\beta_1) + \\ & + f^{\mathcal{C}_2^{\text{acc}}}(\alpha_1^o,\alpha_2^o,\beta_2) - \mathbb{H}(\omega) - \mathbb{H}(\alpha_1^o), \end{aligned} \tag{11}$$

with the constraints $\alpha = \frac{\omega}{q} + \alpha_1^o + \alpha_2^o$ and $\beta = \beta_1 + \beta_2$.

V. Numerical Evaluation

In this section, we present a numerical evaluation of (11). Following [6], in the curves for the asymptotic TSE that we present, we keep the ratio $\Delta = \beta/\alpha$ of unsatisfied check nodes to erroneous variable nodes constant and compute $r(\alpha,\Delta\alpha)$ for varying values of α. In Fig. 5, the unsatisfied checks in the RAA code ensemble are equally distributed between the middle and inner accumulator, i.e., $\beta_1 = \beta_2 = \beta/2$. For $\Delta = 0$, when no unsatisfied checks are present in the factor graph, the spectral shape $r(\alpha,0)$ exhibits a zero stretch in the beginning and turns positive when the number of codewords with normalized weight α starts to grow exponentially in

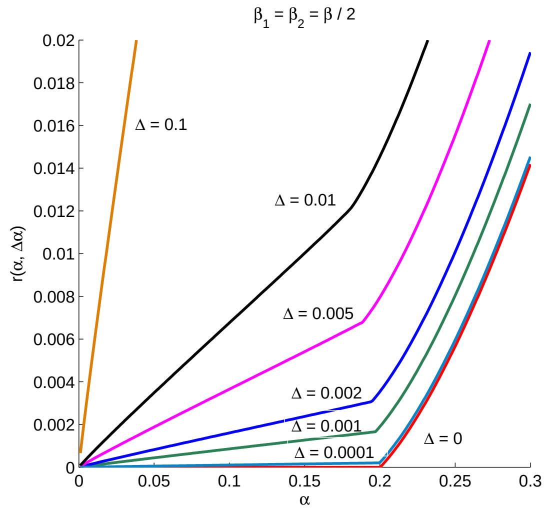


Fig. 5. Asymptotic TSE for different values of Δ and $\beta_1 = \beta_2 = \beta/2$.

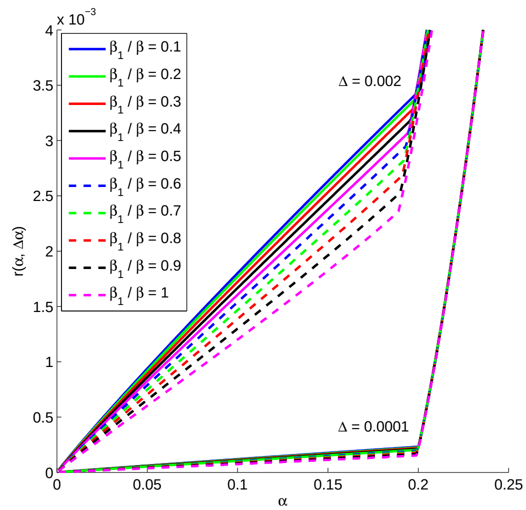


Fig. 6. Asymptotic TSE for different fractions β_1/β.

N with increasing α. The presence of unsatisfied checks in the factor graph results in a positive initial slope, and we observe a quasi-linear increasing first section of the curve, until there is a discontinuity in the slope. In the second section, the slope of the curve is similar for all values of Δ, and the curve shifts to the left with increasing Δ. Also, as the fraction of unsatisfied checks Δ increases, the slope in the first section also increases. Because of the large number of parameters involved in taking the supremum in (11), it is difficult to draw general conclusions about the trapping set structures that are most likely to cause decoding failures. The structure of a trapping set is greatly influenced by the choice of these parameters. We are primarily concerned with trapping set configurations that lead to decoding errors and this requires $\omega > 0$. The choice of the parameters β_1 and β_2 determines how many unsatisfied checks are associated with the middle and inner accumulator, respectively. For instance, in the extreme case of $\beta_1 = \beta$ and $\beta_2 = 0$, all the unsatisfied checks are associated with the middle accumulator, and there are no unsatisfied checks in the graph of the inner accumulator.

In Fig. 6 we vary the ratio β_1/β, the fraction of unsatisfied check nodes associated with the middle accumulator in the

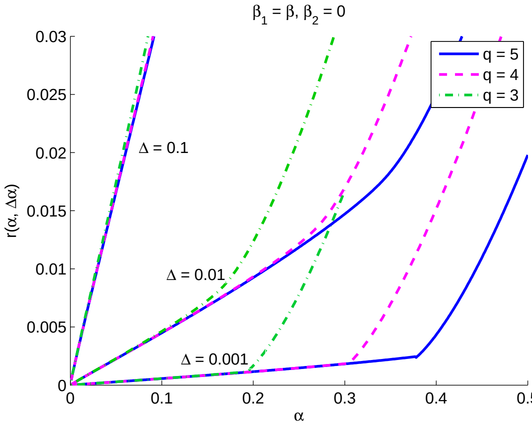


Fig. 7. Asymptotic TSE of the RAA code ensemble for different repetition factors q.

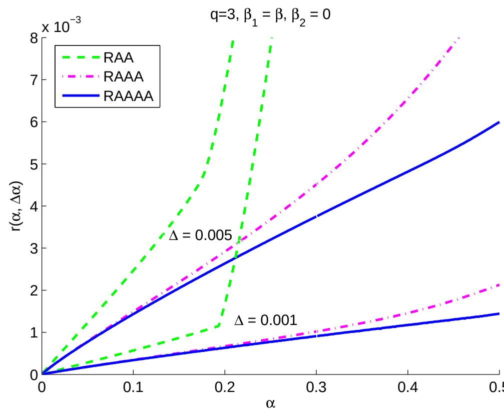


Fig. 8. Asymptotic TSE for the RAA, RAAA, and RAAAA code ensembles.

RAA code ensemble. For larger β_1/β, when relatively more unsatisfied check nodes are present in the middle accumulator, the slope in the first section is smaller and the influence of varying β_1/β on the slope becomes greater as β_1/β gets closer to one. However, the influence that varying Δ has on the slope is much greater than the influence of varying β_1/β. Also, for larger values of Δ, the variance of the curves with β_1/β is greater.

In Figs. 7 and 8 we display the influence of the repetition factor q and the number of concatenated accumulators, respectively, on the shape of the asymptotic TSE. In both cases the greatest effect on the slope in the first section was observed when $\beta_1 = \beta$, i.e., when all unsatisfied check nodes are associated with the outermost accumulator. In the other extreme case, when all the unsatisfied check nodes are associated with the inner accumulator, the slope in the first section did not change. In Fig. 7, the reduction in slope in the first section caused by increasing the repetition factor q is only marginal and the curves almost lie on top of each other. However, increasing the number of serially concatenated accumulators decreases the slope in the first section, as can be seen in Fig. 8. Finally, we note that increasing the repetition factor q or adding more accumulators increases the minimum distance of the code, and thus the transition from the quasi-linear section of the asymptotic TSE to the more steeply increasing section takes place at higher values of α.

VI. Conclusions

We have presented a simple closed form method to enumerate general (a, b) trapping sets for RAA code ensembles. The trapping set enumerator is first obtained for finite block lengths N and its asymptotic expression is derived by letting N go to infinity. Similar to [5] and [6], we observe that, when unsatisfied check nodes are present in the factor graph, the asymptotic TSE lies strictly above the asymptotic spectral shape for the case when no unsatisfied check nodes exist in the graph. Although the RAA code ensemble is asymptotically good and exhibits minimum distance growing linearly with block length, in contrast to regular and some protograph-based irregular LDPC codes, there exists no region where the minimum trapping set size grows linearly with block length. It can, at best, grow only sublinearly in the block length, since the asymptotic TSE of the RAA code ensemble is always positive if unsatisfied check nodes are present in the graph. While the method presented in this paper allows us to enumerate all general (a, b) trapping sets, the influence that these trapping sets have on the error floor must still be evaluated separately. As noted earlier, the probability that the decoder gets stuck in particular types of trapping sets depends on the channel, the decoding algorithm, and the particular decoder implementation. In future work we hope to evaluate this probability for the turbo decoder and the belief propagation decoder, in order to obtain a reliable estimate of the height of the error floor for RMA codes.

References

[1] C. Berrou, A. Glavieux, and P. Thitimajshima, “Near Shannon limit error-correcting coding and decoding: Turbo-codes,” in *Proc. IEEE Int. Conf. Commun. (ICC)*, (Geneva, Switzerland), pp. 1064–1070, May 1993.

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

[3] D. J. C. MacKay and M. Postol, “Weaknesses of Margulis and Ramanujan-Margulis low-density parity-check codes,” *Electronic Notes in Theoretical Computer Science*, vol. 74, 2003.

[4] T. J. Richardson, “Error floors of LDPC codes,” in *Proc. 41st Annual Allerton Conf. on Commun., Contr., and Comp.*, pp. 1426–1435, 2003.

[5] 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, pp. 39–55, Jan. 2007.

[6] S. Abu-Surra, W. E. Ryan, and D. Divsalar, “Ensemble trapping set enumerators for protograph-based LDPC codes,” in *Proc. 45th Annual Allerton Conf. on Commun., Control, and Computing*, (Monticello, IL), pp. 201–210, Sept. 2007.

[7] H. D. Pfister, *On the Capacity of Finite State Channels and the Analysis of Convolutional Accumulate-m Codes*. San Diego, CA: Ph.D. Thesis, University of California, 2003.

[8] 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 Conf. on Commun., Control, and Computing*, (Monticello, IL), Sept. 2007.

[9] F. Kschischang, B. Frey, and H.-A. Loeliger, “Factor graphs and the sum-product algorithm,” *IEEE Trans. Inf. Theory*, vol. 47, pp. 498–519, Feb. 2001.

[10] 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, pp. 909–926, May 1998.

[11] D. Divsalar, H. Jin, and R. J. McEliece, “Coding theorems for ’turbo-like’ codes,” in *Proc. 36th Annual Allerton Conf. on Commun., Control, and Computing*, (Monticello, IL), pp. 201–210, Sept. 1998.