

convolutional code is periodic with period T (in submatrices) if $\mathbf{H}_i(t)$ is periodic, i.e., $\mathbf{H}_i(t) = \mathbf{H}_i(t + T), \forall i, t$, and if $\mathbf{H}_i(t) = \mathbf{H}_i, \forall i, t$, the code is time-invariant. An LDPC convolutional code is called *irregular* if its row and column weights are not constant.

III. PROTOGRAPH-BASED LDPC CONVOLUTIONAL CODES

A protograph is a small bipartite graph. Figure 1 shows a protograph and the associated protograph parity-check matrix.

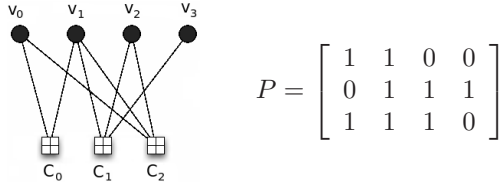


Fig. 1. An example protograph and associated parity-check matrix.

Suppose a given protograph has n_v variable nodes and n_c check nodes. An ensemble of protograph-based LDPC block codes can be created using the copy-and-permute operation [11]. The parity-check matrix H corresponding to a member of the resulting ensemble of protograph-based LDPC block codes can be obtained by replacing ones with $N \times N$ permutation matrices and zeros with $N \times N$ all-zero matrices in the underlying protograph parity-check matrix P , where the permutation matrices are chosen randomly and independently.

Using this construction, the sparsity condition of an LDPC parity-check matrix will be satisfied for large N . The code created by applying the copy-and-permute operation to an $n_c \times n_v$ protograph parity-check matrix P has block length $n = Nn_v$. In addition, the code has the same rate and degree distribution for each of its variable and check nodes as the underlying protograph.

We note that in this example the row and column weights of P are not constant, so P represents the parity-check matrix of an irregular protograph. The resulting ensemble of protograph-based block codes will thus also be irregular. Note that it is also possible to consider protograph parity-check matrices P with larger integer entries, which represent parallel edges in the base protograph. In this case, the resulting block in H consists of a sum of $N \times N$ permutation matrices [11].

A. Forming protograph-based convolutional codes

Suppose that we have an $n_c \times n_v$ protograph parity-check matrix P , where $\gcd(n_c, n_v) = y$. We then partition P as a $y \times y$ block matrix as follows:

$$P = \begin{bmatrix} P_{1,1} & \dots & P_{1,y} \\ \vdots & & \vdots \\ P_{y,1} & \dots & P_{y,y} \end{bmatrix},$$

where each block $P_{i,j}$ is of size $n_c/y \times n_v/y$. P can thus be separated into a lower triangular part, P_l , and an upper triangular part minus the leading diagonal, P_u . Explicitly,

$$P_l = \begin{bmatrix} P_{1,1} & & & \\ P_{2,1} & P_{2,2} & & \\ \vdots & \vdots & \ddots & \\ P_{y,1} & P_{y,2} & \dots & P_{y,y} \end{bmatrix} \text{ and } P_u = \begin{bmatrix} & P_{1,2} & \dots & P_{1,y} \\ & & \ddots & \vdots \\ & & & P_{y-1,y} \end{bmatrix},$$

where blank spaces correspond to zeros. This operation is called ‘cutting’ a protograph parity-check matrix.

Rearranging the positions of these two triangular matrices and repeating them indefinitely results in a parity-check matrix

P_{cc} of an unterminated, periodically time-varying convolutional code with rate $R = 1 - n_c/n_v$, constraint length $\nu_s = n_v$, and period $T = y$ given by¹

$$P_{cc} = \begin{bmatrix} P_l & & & \\ P_u & P_l & & \\ & P_u & P_l & \\ & & \ddots & \ddots \end{bmatrix}. \quad (2)$$

Note that when $\gcd(n_c, n_v) = 1$, we cannot form a square block matrix larger than 1×1 with equal size blocks. In this case, $P_l = P$ and P_u is the all-zero matrix of size $n_c \times n_v$. This trivial cut results in a convolutional code with syndrome former memory zero, with repeating blocks of the original protograph on the leading diagonal. In this case we must use one of two approaches. First, we can create a larger protograph parity-check matrix by applying the copy and permute operation M times to P . This results in an $Mn_c \times Mn_v = n'_c \times n'_v$ parity-check matrix for some small integer M . The $n'_c \times n'_v$ protograph parity-check matrix can then be cut following the procedure outlined above. Alternatively, we can use a nonuniform cut, as described in [8].

To create an ensemble of time-varying LDPC convolutional codes we follow the usual protograph construction technique. The ones are replaced with $N \times N$ permutation matrices (or summations of $N \times N$ permutation matrices for parallel edges) and the zeros with $N \times N$ all-zero matrices, where the permutation matrices are chosen randomly and independently. Choosing N to be sufficiently large guarantees the sparsity condition for an LDPC code.

IV. GENERAL TRAPPING SET ENUMERATORS FOR PROTOGRAPH-BASED CODES

Definition 1: An (a, b) general trapping set $\tau_{a,b}$ of a bipartite graph is a set of variable nodes of size a which induce a subgraph with exactly b odd-degree check nodes (and an arbitrary number of even-degree check nodes).

In order to calculate ensemble average general trapping set enumerators for protograph-based block codes, we make use of the combinatorial arguments previously presented in [12] and [13] for calculating ensemble average weight enumerators. The technique involves considering a two-part weight enumerator for a modified protograph with the property that any (a, b) trapping set in the original protograph is a codeword in the modified protograph. We now briefly describe the procedure introduced in [9]. An auxiliary ‘flag’ variable node is added to each check node, as displayed in Figure 2.

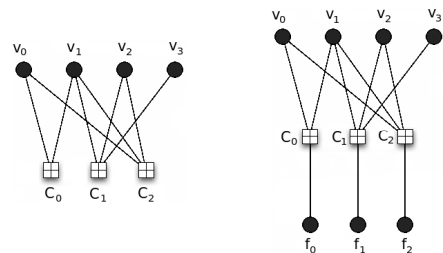


Fig. 2. An example protograph and modified version with auxiliary variable nodes.

¹Cutting certain protograph parity-check matrices may result in a smaller period $T = y'$, where $y' \in \mathbb{Z}^+$ divides y evenly. If $y' = 1$, then the resulting convolutional code is time-invariant.

Consider a subset S with cardinality a of the variable nodes $V = \{v_0, v_1, v_2, v_3\}$, for example, $a = 3$ and $S = \{v_0, v_1, v_2\}$. We now attach weight 1 to these variable nodes and weight 0 to the remaining nodes in $V/S = \{v_3\}$. We observe that check nodes c_0 and c_1 are satisfied, since they both have input weight 2, but that check node c_2 (with input weight 3) is unsatisfied. Thus there is $b = 1$ odd degree (unsatisfied) check node. This is an example of a $(3, 1)$ general trapping set. Thus $\tau_{3,1}$ contains the subset $S = \{v_0, v_1, v_2\}$.

For any subset of variable nodes we can satisfy any unsatisfied check nodes by assigning weight 1 to the corresponding auxiliary variable node. Note that the weight of the variable nodes V is $a = 3$ and the weight of the auxiliary nodes is $b = 1$ for this $(3, 1)$ general trapping set, which suggests that the general trapping sets of a protograph can be enumerated by applying a two-part weight enumerator analysis to the modified protograph. We thus consider a two-part weight enumerator over sets of variable nodes $V = \{v_0, v_1, \dots, v_{n_v-1}\}$ and auxiliary nodes $F = \{f_0, f_1, \dots, f_{n_f-1}\}$, where n_v is the number of variable nodes in the initial protograph and n_f is the number of auxiliary variable nodes (equal to the number of check nodes n_c). This method of enumerating trapping sets for protograph-based codes is presented in [9]. In the remainder of this section, we summarize the results of [9] and [12] on which our approach is based.

A. Finite length ensemble trapping set enumerators

Suppose that a modified protograph contains n_v variable nodes to be transmitted over the channel. Also, suppose that each of the n_v transmitted variable nodes has an associated weight d_i , where $0 \leq d_i \leq N$ for all i .² Let $S_d = \{(d_0, d_1, \dots, d_{n_v-1})\}$ be the set of all possible weight distributions such that $d_0 + \dots + d_{n_v-1} = a$. Finally, suppose that $S_f = \{(f_0, f_1, \dots, f_{n_f-1})\}$ is the set of all weight distributions such that $f_0 + \dots + f_{n_f-1} = b$, where $0 \leq f_i \leq N$ for all i . Then the two-part ensemble average trapping set enumerator for the modified protograph is given by

$$A_{a,b} = \sum_{\{d_k\} \in S_d} \sum_{\{f_i\} \in S_f} A_{\mathbf{d}}, \quad (3)$$

where $A_{\mathbf{d}}$ is the average number of codewords in the modified ensemble with weight distribution $\mathbf{d} = (d_0, d_1, \dots, d_{n_v-1}, f_0, f_1, \dots, f_{n_f-1})$.

B. Asymptotic trapping set spectral shape function

The two-part normalized logarithmic asymptotic trapping set spectral shape function of a code ensemble can be written as $r(\alpha, \beta) = \lim_{n \rightarrow \infty} \sup r_n(\alpha, \beta)$, where $r_n(\alpha, \beta) = \frac{\ln(A_{a,b})}{n}$, $\alpha = a/n$, $\beta = b/n$, a and b are Hamming weights, n is the block length, and $A_{a,b}$ is the two-part ensemble average weight distribution.

Suppose now we are interested in the ratio of b to a for a general (a, b) trapping set enumerator. Let $\Delta = b/a = \beta/\alpha$, $\Delta \in [0, \infty)$. As proposed in [9], we may now classify the trapping sets as $\tau_{\Delta} = \{\tau_{a,b} | b = \Delta \cdot a\}$. For each Δ , we define $d_{ts}(\Delta)$ to be the Δ -trapping set number, which is the size of the smallest, non-empty trapping set in τ_{Δ} . Now consider

²Since we use N copies of the protograph, the weight associated with a particular variable node in the protograph can be as large as N .

fixing Δ and plotting the normalized weight α against the two-part asymptotic spectral shape function $r(\alpha, \beta) = r(\alpha, \Delta\alpha)$. Suppose $\alpha > 0$ and the first zero-crossing of $r(\alpha, \beta)$ occurs at $\alpha = \delta_{ts}(\Delta)$. If $r(\alpha, \beta)$ is negative in the range $0 < \alpha < \delta_{ts}(\Delta)$, then the first zero-crossing $\delta_{ts}(\Delta)$ is called the Δ -trapping set growth rate of the code ensemble. If $\delta_{ts}(\Delta)$ exists, and if the probability

$$\mathbb{P}(a < \delta_{ts}(\Delta)n) = \sum_{a=1}^{\delta_{ts}(\Delta)n-1} A_{a,b} \ll 1$$

as the block length n grows, we can say with high probability that the majority of codes in the ensemble have a Δ -trapping set number that increases linearly with n , i.e., $d_{ts}(\Delta) = n\delta_{ts}(\Delta)$. This implies that, for sufficiently large n , a typical member of the ensemble has no small trapping sets.

V. GENERAL TRAPPING SET ENUMERATORS FOR LDPC CONVOLUTIONAL CODES

In this section, we present a method for obtaining a lower bound on the Δ -trapping set number of an ensemble of unterminated, asymptotically good, periodically time-varying LDPC convolutional codes derived from protograph-based LDPC block codes. To proceed, we introduce a family of tail-biting LDPC convolutional codes with incremental increases in block length. The tail-biting codes are then used as a tool to obtain the desired bound on the Δ -trapping set number of the unterminated codes.

A. Tail-biting convolutional codes

Consider the parity-check matrix P_{cc} of the protograph-based unterminated convolutional code introduced in Section III-A. We now introduce the notion of tail-biting convolutional codes by defining an ‘unwrapping factor’ λ as the number of times the sliding convolutional structure is repeated before applying tail-biting termination. For $\lambda \geq 1$, the parity-check matrix $P_{tb}^{(\lambda)}$ of the desired tail-biting protograph-based convolutional code with block length λn_v can be written as

$$P_{tb}^{(\lambda)} = \begin{bmatrix} P_l & & & & & & & & P_u \\ P_u & P_l & & & & & & & \\ & P_u & P_l & & & & & & \\ & & & \ddots & \ddots & & & & \\ & & & & & P_u & P_l & & \end{bmatrix}_{\lambda n_c \times \lambda n_v}. \quad (4)$$

Note that the tail-biting convolutional code for $\lambda = 1$ is simply the original protograph-based block code.

B. Tail-biting LDPC convolutional code ensembles

Given a protograph parity-check matrix P , we generate a family of tail-biting convolutional codes with parity check matrices $P_{tb}^{(\lambda)}$ and increasing block lengths λn_v , $\lambda = 1, 2, \dots$, using the process described above. Since tail-biting convolutional codes are themselves block codes, we can treat the Tanner graph of $P_{tb}^{(\lambda)}$ as a protograph for each value of λ . Replacing the entries of this matrix with either $N \times N$ permutation matrices or $N \times N$ all-zero matrices, as discussed in Section III, creates an ensemble of LDPC codes that can be analysed asymptotically as N goes to infinity, where the sparsity condition of an LDPC code is satisfied for large N . Each tail-biting LDPC code ensemble, in turn, can be unwrapped and repeated indefinitely to form an ensemble of unterminated periodically time-varying LDPC convolutional

codes³ with rate $R = 1 - n_c/n_v$, constraint length $\nu_s = Nn_v$, and, in general, period $T = \lambda y$.

To study the average general trapping set enumerators of these block codes, we add auxiliary flag variables following the procedure detailed in Section IV. The resulting protograph-based parity-check matrix is given by

$$\tilde{P}_{tb}^{(\lambda)} = \left[\begin{array}{ccc|cc} P_l & & & P_u & \\ P_u & P_l & & & \\ & P_u & P_l & & \\ & & \ddots & \ddots & \\ & & & P_u & P_l \end{array} \right]_{\lambda n_c \times \lambda(n_v+n_c)},$$

where \mathbb{I}_n is the $n \times n$ identity matrix. For any λ , we can now follow the procedure detailed in Section IV to calculate the Δ -trapping set number $d_{ts}^{(\lambda)}$ for the ensemble of LDPC tail-biting convolutional codes based on the protograph parity-check matrix $P_{tb}^{(\lambda)}$.

C. A lower bound on the convolutional Δ -trapping set number

Theorem 1: Consider forming a family of unterminated, periodically time-varying LDPC convolutional codes with rate $R = 1 - Nn_c/Nn_v = 1 - n_c/n_v$, constraint length $\nu_s = Nn_v$, and period $T = \lambda y$ from (4), as described in Section V-B. Let $d_{cts}^{(\lambda)}(\Delta)$ be the Δ -trapping set number of the code with $T = \lambda y$, and let $d_{cts}(\Delta) = \max_{\lambda > 0} d_{cts}^{(\lambda)}(\Delta)$, which we call the Δ -trapping set number of the unterminated convolutional code family. Then, for any tail-biting termination with unwrapping factor λ , $d_{cts}(\Delta)$ is bounded below by $d_{ts}^{(\lambda)}(\Delta)$ for any λ , i.e.,

$$d_{cts}(\Delta) \geq d_{ts}^{(\lambda)}(\Delta). \quad (5)$$

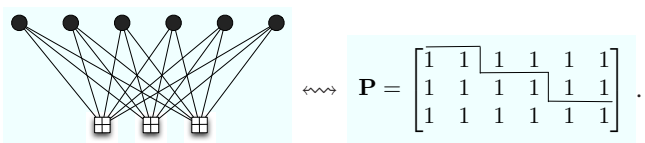
Proof. The proof, omitted here for brevity, is presented in [14].

Intuitively, as λ increases, the tail-biting code becomes a better representation of the associated unterminated convolutional code, with $\lambda \rightarrow \infty$ corresponding to a non-periodically time-varying convolutional code. This is reflected in the average general trapping set enumerators, and it is shown in Section VI that increasing λ provides us with Δ -trapping set growth rates $\delta_{ts}^{(\lambda)}(\Delta)$ that converge to a lower bound on $\delta_{cts}(\Delta)$, which we call the Δ -trapping set growth rate of the unterminated convolutional code family.

VI. TRAPPING SET ANALYSIS

We now present a trapping set analysis for two asymptotically good ensembles, one regular and one irregular, of unterminated periodically time-varying LDPC convolutional codes. As described in Section V, we make use of ensembles of tail-biting LDPC convolutional codes to obtain a lower bound on the desired Δ -trapping set growth rate of the associated unterminated convolutional code ensemble.

Example 1: Consider a (3,6)-regular LDPC code with the following protograph and associated parity-check matrix:



³In this case, the submatrices of P_l and P_u are of size $Nn_c/y \times Nn_v/y$.

A family of rate $R = 1/2$ tail-biting LDPC convolutional code ensembles can be generated according to the displayed cut. We now proceed to calculate the Δ -trapping set growth rate $\delta_{ts}^{(\lambda)}(\Delta)$ for the modified tail-biting convolutional code ensembles with base parity-check matrices $\tilde{P}_{tb}^{(\lambda)}$ for various fixed values of Δ and increasing values of the unwrapping factor λ . Note that setting $\Delta = \beta/\alpha = 0$ corresponds to the minimum distance growth rate problem [7]. Thus, for $\lambda = 1$, which corresponds to the (3,6)-regular block code ensemble, $\delta_{ts}^{(1)}(0) = \delta_{min} = 0.023$, where δ_{min} is the minimum distance growth rate for the (3,6)-regular ensemble, as originally calculated by Gallager [15]. Further, for larger values of λ , the value for $\delta_{ts}^{(\lambda)}(0)$ agrees with the earlier results for minimum distance growth rates for tail-biting convolutional codes given in [7].

As Δ ranges from 0 to ∞ , the points $(\delta_{ts}^{(\lambda)}(\Delta), \Delta \cdot \delta_{ts}^{(\lambda)}(\Delta))$ trace out the so-called *zero-contour curve* for a protograph-based block code ensemble [9]. The zero-contour curves for Example 1 are shown in Figure 3, and the Δ -trapping set growth rates are highlighted for $\Delta = 0.02$.

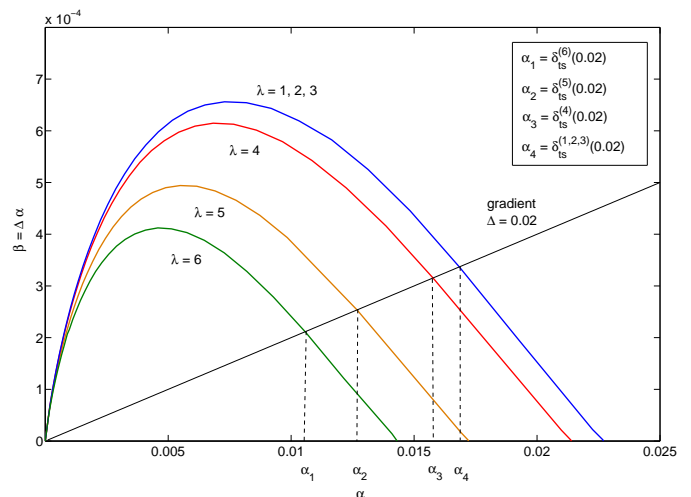


Fig. 3. Zero-contour curves for Example 1.

The zero-contour curve is key to understanding the role of trapping sets in iterative decoding. Code ensembles with large Δ -trapping set numbers $d_{ts}^{(\lambda)}(\Delta)$ are of primary interest, since small trapping sets dominate iterative decoding performance in the error floor [2]. Thus we want the Δ -trapping set growth rate $\delta_{ts}^{(\lambda)}(\Delta)$ to exist and be as large as possible for each value of Δ . We observe in Fig. 3 that $\delta_{ts}^{(\lambda_1)}(\Delta) \leq \delta_{ts}^{(\lambda_2)}(\Delta)$ for any $\lambda_1 > \lambda_2$. This is analogous to the decrease in the minimum distance growth rate with increasing λ observed in [7]. If a zero-contour curve of ensemble A is always below the zero-contour curve of ensemble B, then, in general, we would expect a code drawn from ensemble A to exhibit poorer error floor performance than one drawn from ensemble B. Thus we expect worse error floor performance with increasing λ for the tail-biting convolutional code ensembles.⁴

The Δ -trapping set growth rates of the tail-biting ensembles for each λ are given by

$$\delta_{ts}^{(\lambda)}(\Delta) = \frac{d_{ts}^{(\lambda)}(\Delta)}{n} = \frac{d_{ts}^{(\lambda)}(\Delta)}{\lambda N n_v} = \frac{d_{ts}^{(\lambda)}(\Delta)}{\lambda \nu_s}, \quad (6)$$

⁴We observe from the zero-contour curves of Example 1 that increasing λ results in smaller Δ -trapping set growth rates for $\lambda \geq 3$. However, we must be careful in this case to remember that the block lengths also increase and the Δ -trapping set number is $d_{ts}^{(\lambda)}(\Delta) = n \delta_{ts}^{(\lambda)}(\Delta) = N \lambda n_v \delta_{ts}^{(\lambda)}(\Delta)$.

and the Δ -trapping set growth rate of the associated rate $R = 1/2$ ensemble of unterminated periodically time-varying LDPC convolutional codes is $\delta_{ccts}(\Delta) = d_{ccts}(\Delta)/\nu_s$, as discussed above. Thus (5) gives us the lower bound

$$\delta_{ccts}(\Delta) = \frac{d_{ccts}(\Delta)}{\nu_s} \geq \frac{d_{ts}^{(\lambda)}(\Delta)}{\nu_s} = \lambda \delta_{ts}^{(\lambda)}(\Delta), \quad (7)$$

for all $\lambda \geq 1$. These Δ -trapping set growth rates are plotted in Fig. 4 for $\Delta = 0, 0.01$, and 0.05 .

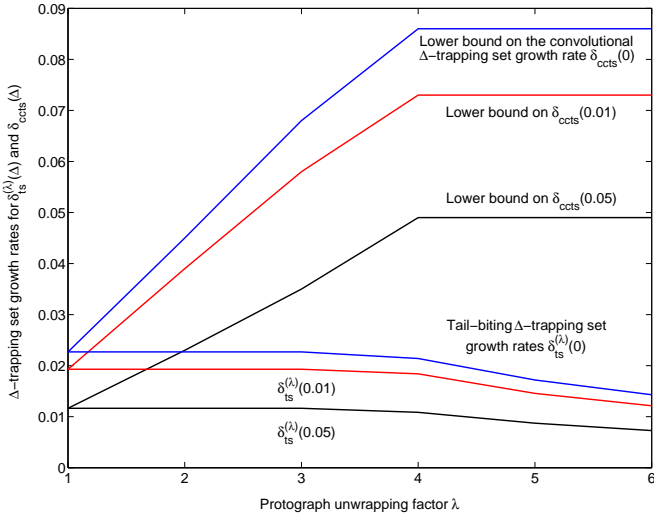


Fig. 4. Δ -trapping set growth rates for Example 1.

We observe that, once the unwrapping factor λ of the tail-biting convolutional code ensemble exceeds 3, the lower bound on $\delta_{ccts}(\Delta)$ levels off for each distinct value of Δ . We also observe a significant increase in the value of $\delta_{ccts}(\Delta)$ compared to $\delta_{ts}^{(1)}(\Delta)$, the Δ -trapping set growth rate of the underlying block code ensemble.

Example 2: The following irregular protograph is from the Repeat Jagged Accumulate [16] (RJA) family. It was shown to have a good iterative decoding threshold ($\gamma_{iter} = 1.0$ dB) while maintaining linear minimum distance growth ($\delta_{min} = 0.013$). We display below the associated P matrix and cut used to generate the family of tail-biting LDPC convolutional code ensembles.



We observe that, as in Example 1, the Δ -trapping set growth rates calculated for increasing λ provide us with a lower bound on the Δ -trapping set growth rate of the unterminated convolutional code ensemble for each value of Δ . The bounds calculated for several values of Δ are given below. Recall that $\delta_{ts}^{(1)}(\Delta)$ is the Δ -trapping set growth rate of the protograph-based LDPC block code ensemble.

Δ	$\delta_{ts}^{(1)}(\Delta)$	lower bound on $\delta_{ccts}(\Delta)$
0	0.013	0.057
0.01	0.010	0.046
0.04	0.005	0.026
0.12	0.001	0.005

VII. CONCLUSIONS

In this paper, asymptotic methods were used to calculate a lower bound on the Δ -trapping set number that grows

linearly with constraint length for several ensembles of unterminated, protograph-based periodically time varying LDPC convolutional codes. Further, it was shown that the Δ -trapping set growth rates of the LDPC convolutional code ensembles exceed the growth rates of the corresponding LDPC block code ensembles on which they are based. These large trapping set growth rates suggest that LDPC convolutional codes will exhibit good iterative decoding performance in the error floor.

Related work includes an important subset of general trapping set enumerators, called *elementary trapping set enumerators* [2], which require a slightly different analytical model, as described in [9]. Monte Carlo simulations (see, e.g., [2]) have shown that in fact most of the decoding failures in iterative decoding correspond to elementary trapping sets. Similar results to those presented here for elementary trapping sets are the subject of ongoing research.

ACKNOWLEDGEMENTS

This work was partially supported by NSF Grants CCF05-15012 and CCF08-30650 and NASA Grant NNX07AK536. In addition, the authors acknowledge the support of the Scottish Funding Council for the Joint Research Institute with the Heriot-Watt University, which is a part of the Edinburgh Research Partnership.

REFERENCES

- [1] D. J. C. MacKay and M. S. Postol, "Weaknesses of Margulis and Ramanujan-Margulis low-density parity-check codes", *Elec. Not. in Theoretical Comp. Sci.*, Vol. 74, 2003.
- [2] T. Richardson, "Error-floors of LDPC codes", *Proc. Forty-first Annual Allerton Conference*, Allerton, IL, Sep. 2003.
- [3] O. Milenkovic, E. Soljanin, and P. Whiting, "Asymptotic spectra of trapping sets in regular and irregular LDPC code ensembles", *IEEE Trans. on Inform. Theory*, IT-53, pp.39-55, Jan. 2007.
- [4] A. Jiménez-Felström and K. Sh. Zigangirov, "Time-varying periodic convolutional codes with low-density parity-check matrices", *IEEE Trans. Inform. Theory*, IT-45, pp.2181-2191, Sept. 1999.
- [5] D. J. Costello, Jr., A. E. Pusane, S. Bates, and K. Sh. Zigangirov, "A comparison between LDPC block and convolutional codes", *Proc. Inform. Theory and App. Workshop*, San Diego, CA, Feb. 2006.
- [6] D. J. Costello, Jr., A. E. Pusane, C. R. Jones, and D. Divsalar, "A comparison of ARA- and protograph-based LDPC block and convolutional codes", *Proc. Inform. Theory and App. Workshop*, San Diego, CA, Feb. 2007.
- [7] D. G. M. Mitchell, A. E. Pusane, K. Sh. Zigangirov, and D. J. Costello, Jr., "Asymptotically good LDPC convolutional codes based on protographs", *Proc. IEEE Int. Symp. on Inform. Theory*, Toronto, Canada, July 2008.
- [8] D. G. M. Mitchell, A. E. Pusane, N. Goertz, and D. J. Costello, Jr., "Free distance bounds for protograph-based regular LDPC convolutional codes", *Proc. Int. Symp. on Turbo Codes and Related Topics*, Lausanne, Switzerland, Aug. 2008.
- [9] S. Abu-Surra, W. E. Ryan, and D. Divsalar, "Ensemble trapping set enumerators for protograph-based LDPC codes", *Proc. Forty-fifth Annual Allerton Conference*, Allerton, IL, Sep. 2007.
- [10] M. Tavares, K. Sh. Zigangirov, and G. Fettweis, "Tail-Biting LDPC Convolutional Codes", *Proc. IEEE Int. Symp. on Inform. Theory*, Nice, France, June 2007.
- [11] J. Thorpe, "Low-Density Parity-Check (LDPC) codes constructed from protographs", *JPL INP Progress Report*, Vol. 42-154, Aug. 2003.
- [12] D. Divsalar, "Ensemble weight enumerators for protograph LDPC codes", *Proc. IEEE Int. Symp. on Inform. Theory*, Seattle, WA, Jul. 2006.
- [13] S. L. Fogal, R. McEliece, and J. Thorpe, "Enumerators for protograph ensembles of LDPC codes", *Proc. IEEE Int. Symp. on Inform. Theory*, Adelaide, Australia, Sept. 2005.
- [14] D. G. M. Mitchell, A. E. Pusane, and D. J. Costello, Jr., "Asymptotic trapping set analysis of regular protograph-based LDPC convolutional codes", *Proc. Inform. Theory and App. Workshop*, San Diego, CA, Feb. 2009.
- [15] R. G. Gallager, "Low-density parity-check codes", *IRE Trans. Inform. Theory*, IT-8: 21-28, Jan. 1962.
- [16] D. Divsalar, S. Dolinar, and C. Jones, "Construction of protograph LDPC codes with linear minimum distance", *Proc. IEEE Int. Symp. on Inform. Theory*, Seattle, WA, Jul. 2006.