Design of quasi-cyclic Tanner codes with low error floors

Gianluigi Liva[1], William E. Ryan[2], and Marco Chiani[1]

[1] Dipartimento di Elettronica, Informatica e Sistemistica, Università di Bologna, {gliva,mchiani}@deis.unibo.it
[2] Electrical and Computer Engineering Department, University of Arizona, Tucson, AZ, ryan@ece.arizona.edu

Abstract

Tanner codes represent a broad class of graph-based coding schemes, including low-density parity-check (LDPC) and turbo codes. Whereas many different classes of LDPC and turbo codes have been proposed and studied in the past decade, very little work has been performed on the broader class of Tanner codes. In this paper we propose a design technique which leads to efficiently encodable quasi-cyclic Tanner codes based on both Hamming and single parity check (SPC) nodes. These codes are characterized by fast message-passing decoders and can be encoded using shift-register-based circuits. The resulting schemes exhibit excellent performance in both the error floor and waterfall regions on the additive white Gaussian noise channel.

1 Introduction

TANNER codes [1] represent the broadest known class of iteratively decodable codes,including low-density parity-check (LDPC) codes [2] and turbo codes [3]. However, since the invention of turbo codes in the early 90's, only a few classes of Tanner codes have been studied [4]–[8]. On one hand, LDPC codes yield an efficient solution in the design of codes characterized by fast iterative decoding, low error floors, and decoding thresholds close to capacity. Their main limitation is the difficulty of designing low-rate codes (i.e., $R \leq 1/2$) which possess both low floors and good decoding thresholds. On the other hand, turbo codes possess fairly good performance in the low-rate regime, although parallel turbo codes tend to have high floors and serial turbo codes tend to have poor thresholds.

This paper explores other iteratively decodable code structures, focusing on the design of low-rate codes characterized by low error floors and fast decoding architectures. In particular, we will focus on Tanner codes with a quasi-cyclic structure [9], which simplifies the construction of the encoders and decoders. The decoding complexity of our codes is generally larger than that of quasi-cyclic LDPC codes. However, in many cases a Tanner code's decoder can take advantage of the use of simple (Hamming) component codes. In certain special cases, decoding can be simply the sum-product algorithm (SPA).

A Tanner code can be efficiently represented by a bipartite graph, with variable (or bit) nodes in one-to-one correspondence with the code bits, and constraint nodes which place collective constraints on their neighboring code bit nodes. LDPC codes are characterized by simple constraint nodes, single-parity-check (SPC) codes. Thus, the number of constraint nodes in the bipartite graph of a (n, k) LDPC code must be at least $(n - k)$. For (parallel and serial) convolutional turbo codes, the number of constraint nodes is lower, typically 2, but the nodes are more complex since they describe the constraints of a convolutional code. These two code classes represent extreme cases of Tanner codes: LDPC codes, with a large number of simple nodes; turbo codes, with a small number of complex nodes.

Other types of Tanner codes have attracted less attention. Block turbo codes (BTC) were proposed in [10] and constitute an application of the turbo decoding principle to product codes [11]. BTC are Tanner codes for which the trade-off between node count and node complexity lies midway between that of LDPC codes and convolutional turbo codes. In particular, a BTC is usually based on a relatively simple component code, which is used as both the row and column codes of the product code. Another example of Tanner codes with moderate constraint node complexities are generalized low density codes (GLDC) [4]. GLDC are a generalization of *regular* LDPC codes, where the constraint nodes based on SPC codes are replaced by more powerful nodes such as Hamming codes. Nevertheless, both BTC and GLDC constitute subclasses of Tanner codes with very specific characteristics: block turbo codes are constrained by the structure of product codes, while GLDC are based simply on regular bipartite graphs. Constructions of Tanner codes based on nodes of moderate complexity and *irregular* graphs are mostly unexplored.

The paper is organized as follows. Section 2 introduces some of the concepts and notation that underlie the Tanner codes that we study. A first example of a quasi-cyclic Tanner code is presented in that section. Section 3 includes a description of proposed Tanner code design techniques. Performance results of some low rate Tanner codes are presented in Section 4. Some concluding remarks are contained in Section 5.

This work was funded in part by University of Bologna, Progetto Pluriennale, and by NASA-JPL grant 1264726.

2 Structured Tanner codes

In Fig. 1 the graphical structure (bipartite graph) of a Tanner code is depicted. Constraint nodes are represented as square nodes, variable (or code bit) nodes as circles. Square nodes marked by C_0, C_1, ..., C_{m_c-1} are based on linear block codes $\mathcal{C}^0$, $\mathcal{C}^1$, ..., $\mathcal{C}^{m_c-1}$.

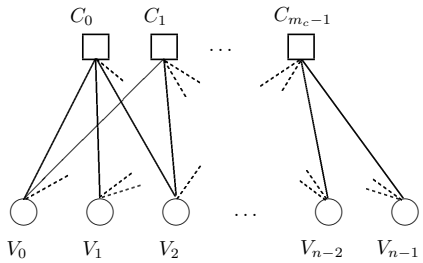

Fig. 1. Bipartite graph for a Tanner code.

It must be noted that, while for an LDPCC the bipartite graph gives a complete description of the code's structure, for a generic Tanner code the specifications of the component linear block codes are also required. Let us define m as the summation of all the redundancies introduced by the m_c constraint nodes, namely,

$$m = \sum_{i=0}^{m_c-1} (n_i - k_i),$$

where (n_i, k_i) represent the parameters of the component code underlying the i-th constraint node and where $n_i = k_i + 1$ when the i-th node corresponds to an SPC code. Thus, if n is the number of variable nodes in the bipartite graph representing the code, the number of information bits for the overall code is

$$k \geq n - m,$$

with equality if all the check equations derived from constraint nodes definitions are linearly independent. The code rate satisfies

$$R \geq 1 - \frac{m}{n}.$$

Let $V = \{V_j\}_{j=0}^{n-1}$ be the set of n variable nodes and $C = \{C_i\}_{i=0}^{m_c-1}$ be the set of m_c constraint nodes in the bipartite graph of a Tanner code. The connection between the nodes in V and C can be summarized in an $m_c \times n$ adjacency matrix $\mathbf{\Gamma}$. Whereas the adjacency matrix for an LDPC code serves as a parity-check matrix, for a Tanner code, construction of the parity-check matrix $\mathbf{H}$ requires also knowledge of the parity-check matrices of the component codes which reside at the constraint nodes.

To describe how $\mathbf{H}$ is contructed, it is convenient to choose an adjacency matrix $\mathbf{\Gamma}$ in block circulant form,

$$\mathbf{\Gamma} = \begin{bmatrix} \mathbf{\Gamma}^{(0)} \\ \mathbf{\Gamma}^{(1)} \\ \vdots \\ \mathbf{\Gamma}^{(M_c-1)} \end{bmatrix},$$

where $\mathbf{\Gamma}^{(i)} = [\pi_{i,0}\ \pi_{i,1}\ \ldots \pi_{i,N-1}]$ and $\pi_{i,j}$ is either a $q \times q$ circulant permutation matrix or a $q \times q$ zero matrix (note that M_c has not to be confused with m_c). We will call each row of permutation matrices a *block row* which we observe has q rows and $n = qN$ columns. The number of block rows M_c is chosen so that $m_c = qM_c$. We note that the number of nonzero permutation matrices in a block row is simultaneously equal to the degree of its corresponding constraint nodes and the length of the component codes for those nodes.

Supposing that there is one component code with parity-check matrix $\mathbf{H}_i$ for each block row, we need only discuss the i-th block row. Let $\mathbf{H}_i$ be $m_i \times n_i$. Then, for each row in the i-th block row, replace the n_i ones in the row by the corresponding n_i columns of $\mathbf{H}_i$. This expands the i-th block row from $q \times n$ to $qm_i \times n$ (for the special case of an SPC constraint node, $m_i = 1$ and the row block is not expanded). Once this process has been applied to each block row, the resulting parity-check matrix $\mathbf{H}$ for the Tanner code will be $\sum_i qm_i \times n$.

A process analogous to this one can be followed for the case when $\mathbf{\Gamma}$ is not block circulant. However, when it is block circulant and the rows belonging to a block row share the same component code description, the resulting matrix $\mathbf{H}$ can also be put in a block-circulant form (thus, the Tanner code will be quasi-cyclic). To do this, first observe that $\mathbf{H}$ so constructed will have the form

$$\mathbf{H} = \begin{bmatrix} \mathbf{H}^{(0)} \\ \mathbf{H}^{(1)} \\ \vdots \\ \mathbf{H}^{(M_c-1)} \end{bmatrix}$$

where $\mathbf{H}^{(i)}$ is the i-th expanded block row. To obtain the block-circulant form of $\mathbf{H}$, we re-order the rows within each $\mathbf{H}^{(i)}$ by taking row 0, then row m_i, then row $2m_i$, ..., then row $(q-1)m_i$; then take row 1, then row m_i+1, then row $2m_i+1$, ..., then row $(q-1)m_i+1$; and so on. Symbolically, if we let $s = 0, 1, ..., qm_i - 1$ be a row index for $\mathbf{H}^{(i)}$, then the index for the re-ordered version of $\mathbf{H}^{(i)}$ is

$$l = q(s \mod m_i) + \lfloor s/m_i \rfloor .$$

Example. We consider a code with $n = 2190$ variable nodes and $m_c = 292$ constraint nodes. Half of these constraint nodes correspond to the (15,11) Hamming code having parity-check matrix

$$\mathbf{H}_i = [\mathbf{M}_1\ \mathbf{M}_2] = \tag{1}$$

$$= \begin{bmatrix} 1\,0\,1\,0\,1\,0\,1\,0 & 1\,0\,1\,0\,1\,0\,1 \\ 0\,1\,1\,0\,0\,1\,1\,0 & 0\,1\,1\,0\,0\,1\,1 \\ 0\,0\,0\,1\,1\,1\,1\,0 & 0\,0\,0\,1\,1\,1\,1 \\ 0\,0\,0\,0\,0\,0\,0\,1 & 1\,1\,1\,1\,1\,1\,1 \end{bmatrix}, \tag{2}$$

$i = 0, 1, ..., 145$, and the other half have the parity-check matrix

$$\mathbf{H}_i = [\mathbf{M}_2\ \mathbf{M}_1], \tag{3}$$

$i = 146, 147, ..., 291$ (where the definitions of $\mathbf{M}_1$ and $\mathbf{M}_2$ are evident). The 292×2190 adjacency matrix (with $M_c = 2$ and $q = 146$) is depicted in Fig. 2 and the re-ordered $\mathbf{H}$ matrix is shown in Fig. 3. Observe that an alternative approach for obtaining the re-ordered matrix $\mathbf{H}$ is to replace each 1 in the rows of the matrix in (2) (the matrix in (3)) by the corresponding permutation matrices of the first (second) block row of the adjacency matrix in Fig. 2, and to then stack the first resulting matrix on the second. ■

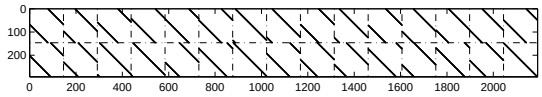

Fig. 2. Adjacency matrix of the (2190,1022) Tanner code.

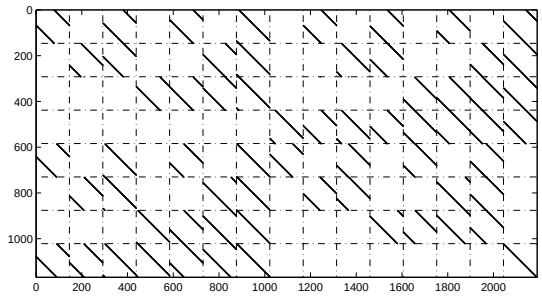

Fig. 3. Block-circulant form for the parity-check matrix $\mathbf{H}$ of the (2190,1022) Tanner code.

3 Design of structured Tanner codes via doping

The design of Tanner codes can be based on the same principles as the design of LDPC codes. For a given code rate, a density evolution analysis can be exploited to derive the decoding thresholds of bipartite graphs, whose nodal distributions are optimized with the aim of minimizing the signal-to-noise ratio (*SNR*) threshold. However, Tanner codes deal with a larger set of possible node types than do LDPC codes. Thus, the search for a good mixture of component codes can be impractical. To facilitate the code design, one can place restrictions on the number of component codes that can be used or on the regularity of the Tanner graph. The code presented in the previous section is a case in point: its graph is regular and only two component codes are used.

A different approach to Tanner codes design is based on doping techniques. *Code doping* was introduced in [6] and consists of the substitution of selected SPC nodes in a LDPC code bipartite graph with some more powerful nodes based on less trivial linear block codes. (Note that the term "code doping" was already used, with a different meaning, in [12].) Supposing that the fraction of nodes that are doped is small, the resulting Tanner code will be called a *doped LDPC code* (D-LDPCC). If the doping codes are (shortened/extended) Hamming codes, the code will be referred as a *Hamming-doped LDPC code* (HD-LDPCC).

The code doping approach begins with an LDPC code bipartite graph characterized by a good decoding threshold, followed by the substitution of a fraction of the check nodes with stronger constraint nodes, and then the evaluation of final graph's decoding performance through density evolution and computer simulations. The choice of the fraction of doped nodes and of the component codes takes into account requirements on the desired code rate and block length. The set of component codes used for doping is limited to codes with low decoding complexity. This pragmatic approach is less thorough than the density evolution-based search used in the design of LDPC codes. However, it permits the design of codes with both good decoding thresholds and low floors.

Classes of LDPC codes that are particularly suited for code doping are ones based on protographs [13]. A protograph is a relatively small bipartite graph from which a larger graph can be obtained by a copy-and-permute procedure: the protograph is copied q times, and then the edges of the individual replicas are permuted among the q replicas (under restrictions described below) to obtain a single, large graph. Suppose the protograph possesses n_p variable nodes and m_p constraint nodes. Then the *derived graph* will consist of $n = n_p q$ variable nodes and $m = m_p q$ constraint nodes.

Note that the edge permutations cannot be arbitrary. In particular, the nodes of the protograph are labelled so that if variable node A is connected to constraint node B in the protograph, then variable node A in a replica can only connect to one of the q replicated B constraint nodes. Doing so preserves the decoding threshold properties of the protograph. If the edge permutations are organized in a cyclic manner such that the final adjacency matrix is in block-circulant form, the derived graph describes a code that can be put in quasi-cyclic form. A protograph can possess parallel edges, i.e., two nodes can be connected by more than one edge. The copy-and-permute procedure must eliminate such parallel connections in order to obtain a derived graph appropriate for a parity-check matrix.

Code doping is conveniently applied at the protograph level: a conventional SPC node in a protograph can be replaced by a more powerful constraint node. The doping will be inherited by the derived graph, which will possess q replicas of the constraint node. Moreover, if the edge permutations are organized in a cyclic manner, the final adjacency matrix will describe a quasi-cyclic Tanner code.

Among LDPCC possessing a protograph representation, accumulate-repeat-accumulate (ARA) codes [14]

TABLE I
CODE RATES ACHIEVABLE BY PUNCTURING CODE $\mathcal{C}^I$ PROTOGRAPH.

Rate	Punc. nodes	Threshold	Capacity	Gap
1/6	-	-0.3 dB	-1.07 dB	0.77 dB
1/5	s_0	-0.54 dB	-0.96 dB	0.42 dB
1/4	s_0, p_4	-0.32 dB	-0.8 dB	0.48 dB

are among the most interesting for code doping purposes. ARA codes are characterized by good decoding thresholds and simple protograph representations. As an example of code doping on ARA code, we choose a protograph of an ARA code with rate $1/4$ [14], depicted in Fig. 4(a). This code has a decoding threshold at $(E_b/N_0)^* = 0.34$ dB, which can be easily computed from the protograph. By puncturing the nodes corresponding to s_0, the code rate becomes $1/3$, and the threshold is reduced to $(E_b/N_0)^* = -0.048$ dB, less than 0.5 dB away from the capacity for the binary-input AWGN channel.

For this protograph, the sub-graph comprising nodes s_0, p_0 and p_1 represents the protograph of an eIRA code [15]. The remaining check node is connected to node p_2, p_2 being the modulo-2 sum of two information bits. Thus, the overall scheme can be considered a parallel concatenation of an eIRA code and an SPC code (where input bits are repeated 2 times). We chose to dope this protograph by replacing the SPC code with a more powerful component code. In order to preserve a reasonable decoding complexity, we focused on the use of a short Hamming code. Specifically, the shortened $(6, 3)$ Hamming code was used to dope this protograph, necessitating the addition of two more variable nodes, p_3 and p_4, as depicted in Fig. 4(b). The extra input edge required by the Hamming node was connected to p_0. In the following, the rate-$1/6$ code derived from the protograph in Fig. 4(b) (and higher rate codes obtained by puncturing) will be referred as *Tanner code I* (denoted $\mathcal{C}^I$). Higher code rates can be achieved by puncturing the protograph of code $\mathcal{C}^I$. Examples of higher-rate protographs and their corresponding thresholds are listed in Table I.

A modified protograph of the one in Fig. 4(b) is given in Fig. 5. Here, one more edge has been drawn, connecting s_0 to a constraint node of the eIRA subgraph. The code obtained by expansion of this modified protograph will be called *Tanner code II* ($\mathcal{C}^{II}$). Also in this case, higher rates are achievable by puncturing. The corresponding thresholds are listed in Table II. Although this modified protograph possesses slightly worse thresholds than the one in Fig. 4(b), it leads to codes with lower floors, as will be shown in the next section.

The threshold values listed in Tables I, II were obtained through a density evolution analysis which takes into account the protograph of the code.

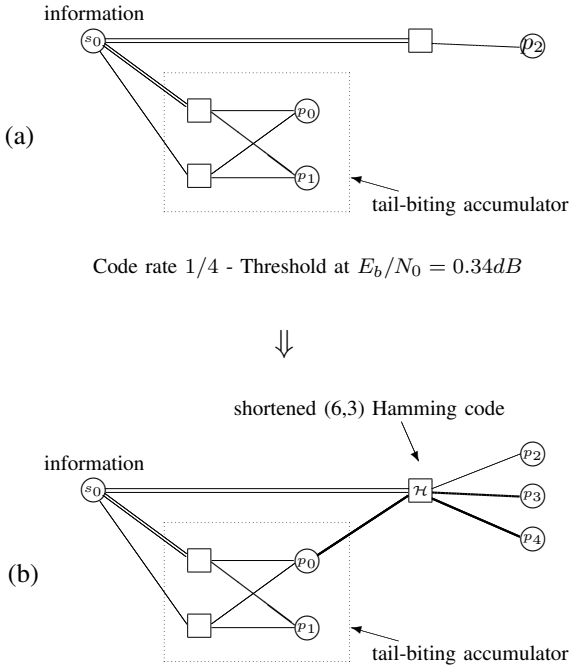

Fig. 4. (a) Protograph for the rate 1/4 ARA code and (b) protograph for the rate 1/6 Tanner code I.

4 Numerical Results

In this section simulation results are provided for quasi-cyclic Tanner codes over the binary-input AWGN channel. The decoder is based on the log-domain message-passing algorithm described in [6]. For the constraint nodes, soft-outputs are computed using a BCJR decoder [16] working on the BCJR trellis [17] of the component code. We first focus on the design of a rate $1/2$, $k = 1022$ code, obtained by puncturing the (2190,1022)

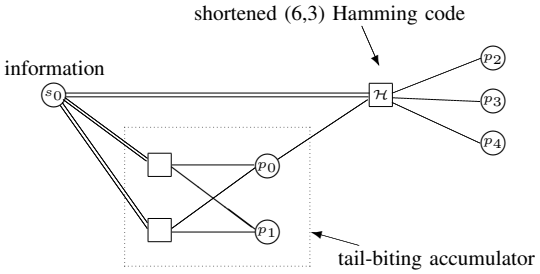

Fig. 5. Protograph for the rate 1/6 Tanner code II.

TABLE II
CODE RATES ACHIEVABLE BY PUNCTURING CODE $\mathcal{C}^{II}$ PROTOGRAPH.

Rate	Punc. nodes	Threshold	Capacity	Gap
1/6	-	-0.33 dB	-1.07 dB	0.74 dB
1/5	s_0	-0.34 dB	-0.96 dB	0.62 dB
1/4	s_0, p_4	-0.24 dB	-0.8 dB	0.56 dB

code of Section 2. Puncturing the first 146 bits of the codeword (corresponding to the first column of circulants of $\mathbf{\Gamma}$) leads to a (2044,1022) quasi-cyclic code. The frame error rate (FER) performance of this code is depicted in Fig. 6. To achieve sufficient reliability, 30 error events were collected for each E_b/N_0 value; only at the highest SNR point ($E_b/N_0 = 2.5$ dB) was the decoder stopped after 15 error events (obtained after more than 300 million transmitted codewords). For the simulations, the maximum number of iterations was set to $I_{max} = 50$. The Tanner code does not display a floor down to $FER \simeq 5 \cdot 10^{-8}$. This is a remarkable result for a medium length code: down to such low error rate, the code's performance is still within 1 dB from the random coding bound [18] for (2044,1022) block codes. We point out that the Tanner codes' software decoder developed for the simulations was able to process approximatively 200K bps on a 3 GHz Pentium IV platform. This observation supports the potential of high-speed decoding of Tanner codes based on simple Hamming codes.

Further simulations have been performed on the AWGN channel to test the performance of quasi-cyclic low-rate codes constructed from the protographs presented in Section 3. A rate-$1/6$ code with information block length $k = 1024$ was designed according the protograph of Tanner code II, depicted in Fig. 5. In Fig. 7, the performance of the (6144,1024) Tanner code II was investigated using two different decoding algorithms. The first decoder is a belief propagation decoder applied to the adjacency matrix which employs the BCJR algorithm at each Hamming node. The second is a belief propagation decoder applied to the $\mathbf{H}$ matrix, using the standard LDPC code sum-product algorithm. The $\mathbf{H}$ matrix was derived from the adjacency matrix as described in Section 2. Its bipartite graph is suited for SPA decoding, since it does not have length-4 loops. This is due to the fact that the parity-check matrix of the shortened (6,3) Hamming code

$$\mathbf{H} = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}$$

and the bipartite graph corresponding to the adjacency matrix are both cycle-4 free. Both simulations used $I_{max} = 200$. The SPA decoder suffers a loss in the waterfall region of only about 0.1 dB.

Remarkable results (Fig. 8) are achieved also for a rate-$1/4$ code obtained, as suggested in Section 3, by puncturing the variable nodes (related to the s_0 and p_4 nodes of the protograph of Fig. 5) of the previous rate-$1/6$ code. In this case, the FER curve (using the BCJR decoding for the Hamming nodes) is about 0.7 dB from the bound, with $I_{max} = 200$.

Two quasi-cyclic rate-$1/5$ codes with input block length $k = 1792$ have been constructed from the protographs of Tanner codes I and II, the higher rate obtained by puncturing the systematic bits. In Fig. 9, the frame error rates for these two codes are compared (with $I_{max} = 200$) to the rate $1/6$ convolutional turbo code standardized by CCSDS [19], for which $k = 1786$. For the turbo code, the maximum number of iterations was set to 10 [20]. That value is usually sufficient to achieve a coding gain near the maximum possible for such turbo schemes. For the low-rate quasi-cyclic Tanner codes, a higher number of iterations is necessary to achieve most of the coding gain possible (because their lower decoding convergence speeds). However, the larger number of iterations need not imply a slower decoder, since Tanner code constraint nodes are less complex than those of the turbo code. Further, parallel processing is more amenable in a Tanner code.

The Tanner code I is characterized by astonishing performance in the waterfall region: at a $FER = 10^{-3}$ the code is less than 0.5 dB away from the random coding bound, and exhibits almost the same performance as the rate $1/6$ turbo code (yet the Tanner code has a higher rate). The protograph of Tanner code II has an additional edge relative to Tanner code I, which degrades the threshold of the code ensemble, as shown in Tables I and II. However, the introduction of this edge lowers the error floor. As seen in Fig 9, Tanner code I is 0.2 dB superior to Tanner code II at $FER = 10^{-3}$, but it has an error floor just below this error rate. Tanner code II reaches a FER close to 10^{-6} without a floor, within 0.9 dB from the bound.

5 Conclusion

In this paper a technique for designing quasi-cyclic Tanner codes is presented, permitting the construction of Tanner codes amenable to efficient encoder and decoder implementations. The Tanner codes constructed possess remarkable performance in both the error floor and waterfall regions. A pragmatic approach for finding good codes was proposed which is based on the insertion of powerful constraint nodes in an LDPC code bipartite graph. Such a doping technique is most easily performed on the protograph of an LDPC code. The performance of low-rate codes constructed in this manner were presented. The results in this paper motivate further investigation into the design of quasi-cyclic Tanner codes (via protographs) for a wider range of code rates and channels.

6 Acknowledgments

We would like to acknowledge Dr. Christopher Jones (NASA Jet Propulsion Lab) for his feedback.

References

[1] R. Tanner, "A recursive approach to low complexity codes," *IEEE Trans. Inform. Theory*, vol. 27, pp. 533–547, Sept. 1981.

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

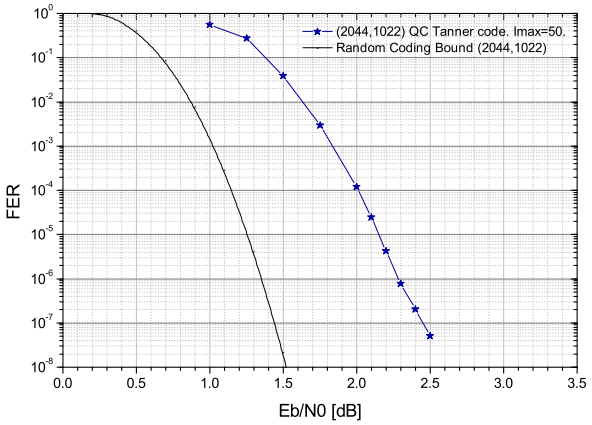

Fig. 6. Frame error rate for the (2044,1022) quasi cyclic Tanner code, compared to the random coding bound. I_{max} was set to 50.

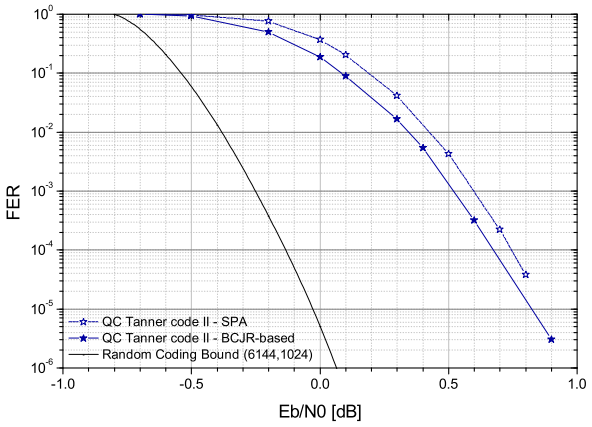

Fig. 7. Frame error rates of rate $1/6$ (6144,1024) Tanner code II with BCJR-based and SPA decoding. I_{max} was set to 200.

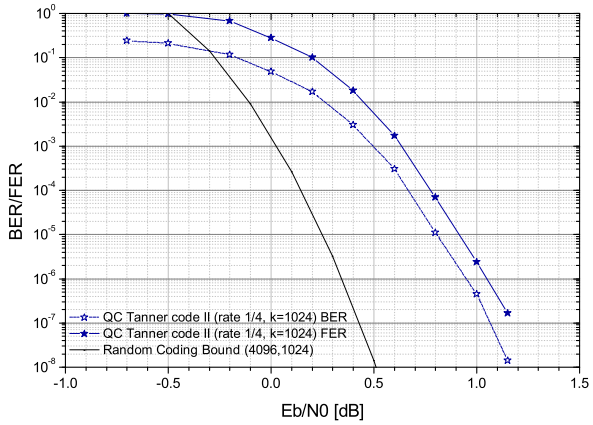

Fig. 8. Frame and bit error rates of rate $1/4$ (4096,1024) Tanner code II (obtained by puncturing of the rate $1/6$ code), compared to the random coding bound. I_{max} was set to 200.

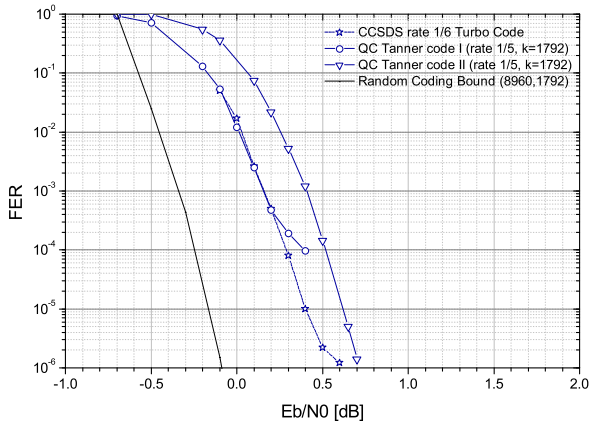

Fig. 9. Frame error rate of rate $1/5$ codes obtained by puncturing (10752,1792) Tanner codes I and II, compared to the CCSDS turbo code with rate $1/6$ (whose performance was taken from [20]) and to the random coding. I_{max} was set to 200 for the Tanner codes.

[3] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes," in *Proc. ICC'93*, Geneva, Switzerland, May 1993.

[4] J. Boutros, O. Pothier, and G. Zemor, "Generalized low density (Tanner) codes," in *Proc. ICC'99*, Vancouver, Canada, June 1999.

[5] S. Dolinar, "Design and iterative decoding of networks of small codes," in *Proc. IEEE International Symposium on Information Theory*, June 2003.

[6] G. Liva and W. E. Ryan, "Short low-error-floor Tanner codes with Hamming nodes," in *Proc. IEEE Milcom*, Oct. 2005.

[7] J. Chen and R. Tanner, "A hybrid coding scheme for the Gilbert-Elliott channel," in *Proc. Allerton Conf. on Communication, Control, and Computing*, 2004.

[8] P.O.Vontobel and H.A.Loeliger, "Irregular codes from regular graphs," in *Proc. IEEE International Symposium on Information Theory*, Lausanne, Switzerland, 2002.

[9] W. W. Peterson and E. J. Weldon Jr, *Error-Correcting Codes*. Cambridge, MA: M.I.T. Press, 1972.

[10] R. Pyndiah, "Near optimum decoding of product codes: Block turbo codes," *IEEE Trans. Commun.*, vol. 46, pp. 1003–1010, Aug. 1998.

[11] P. Elias, "Error-free coding," *IRE Trans.Inform.Theory*, vol. PGIT-4, pp. 29–37, Sept. 1954.

[12] S. ten Brink, "Code doping for triggering iterative decoding convergence," in *Proc. IEEE International Symposium on Information Theory*, Washington, D.C., USA, 2001.

[13] J. Thorpe, "Low-density parity-check (LDPC) codes constructed from protographs," JPL INP, Tech. Rep., Aug. 2003, 42-154.

[14] A. Abbasfar, K. Yao, and D. Disvalar, "Accumulate repeat accumulate codes," in *Proc. IEEE Globecomm*, Dallas, Texas, Nov. 2004.

[15] M. Yang, Y. Li, and W. Ryan, "Design of efficiently encodable moderate-length high-rate irregular LDPC codes," *IEEE Trans. Commun.*, vol. 52, pp. 564–571, Apr. 2004.

[16] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Inform. Theory*, vol. 20, pp. 284–287, March 1974.

[17] R. J. McEliece, "On the BCJR trellis for linear block codes," *IEEE Trans. Inform. Theory*, vol. 42, pp. 1072–1092, July 1996.

[18] R. G. Gallager, *Information Theory and Reliable Communication*. New York: Wiley, 1968.

[19] *TM Synchronization and Channel Coding*, Blue Book, Issue 1, Consultative Committee for Space Data Systems (CCSDS) Recommendation for Space Data System Standard 131.0.B.1, Sept. 2003. [Online]. Available: http://www.ccsds.org/CCSDS/documents/131x0b1.pdf

[20] *Telemetry Channel Coding*, Green Book, Draft Rev. 5, Consultative Committee for Space Data Systems (CCSDS) Recommendation for Space Data System Standard 1xx.0-G-1, Sept. 2002.