Interleaver and Accumulator Design for Systematic Repeat-Accumulate Codes

Sarah Johnson & Steven Weller
School of Electrical Engineering and Computer Science
The University of Newcastle
Callaghan 2308, NSW Australia
email: {sarah,steve}@ee.newcastle.edu.au

***Abstract*— Recent promising theoretical results for repeat-accumulate (RA) codes, together with their extremely simple encoding, motivates this investigation into the design and implementation of practical RA codes. We consider two main issues: the construction of the interleaver in fixed length codes using combinatorial designs; and the improvement of the error floor performance of RA codes by modifying the accumulator.**

I. Introduction

Repeat-accumulate (RA) codes were first presented as a class of simple "turbo-like" codes for which coding theorems could be developed [1]. Indeed, Divsalar et al. were able to show that for RA codes decoded with maximum likelihood decoding, and sufficiently large signal-to-noise ratio, the block error rate approaches zero as the code length is allowed to increase to infinity [1]. It has since been realized that RA codes are powerful Turbo codes in their own right and, significantly, that they offer the performance and low decoding complexity of LDPC codes with the low encoding complexity of Turbo codes.

RA codes can be represented both as serially concatenated Turbo codes and as low-density parity-check (LDPC) codes. The implementation efficiency of RA codes derives from using their Turbo code representation to perform the encoding and their LDPC code representation to perform the decoding. When viewed as a Turbo code, the two constituent codes of an RA code are a rate-$\frac{1}{q}$ repetition code and a rate-1 $\frac{1}{1+D}$ convolutional code, called an accumulator, with a standard interleaver between them. When viewed as an LDPC code, the accumulator corresponds to weight two columns in the parity-check matrix of an RA code while the interleaver determines the structure of the remaining, weight q, columns in the parity-check matrix.

Interestingly, designers of LDPC codes, while considering how to arrange the large number of weight two columns required in LDPC codes by density evolution, have designed LDPC codes which are essentially RA codes (see e.g. [2]). The pattern of weight two columns in the parity-check matrix of an RA code is precisely the best way to add weight two columns to the parity-check matrix of an LDPC code without adding cycles.

Like LDPC codes, RA codes can also be designed with an irregular degree distribution [3], which is achieved by employing an irregular repetition code. The optimal degree distribution of an irregular RA (IRA) code can be determined using density evolution as for LDPC codes [4], as well as by approximation methods such as those proposed in [5]. Roumy et al. used approximation methods to demonstrate IRA codes with capacity thresholds competitive to those of the best known irregular LDPC codes [5].

More generally, Jin et. al have proven that for the binary erasure channel, IRA codes provide a code ensemble which can be decoded reliably in linear time at rates arbitrarily close to channel capacity [3]. Sason and Urbanke have further proven that with high probability the encoding and decoding complexity of certain systematic IRA ensembles on the binary erasure channel scales as $\ln \frac{1}{\varepsilon}$, as the block length of the code goes to infinity, where ε is the gap to capacity [6].

These promising theoretical results for RA codes, together with their extremely simple encoding, motivates this investigation into the design and implementation of practical RA codes. In particular we consider two main issues: the construction of the interleaver in fixed length codes, and the improvement of the error floor performance of RA codes by modifying the accumulator. We begin by introducing RA codes in Section II before presenting, in Section III, RA code interleavers designed using structures from combinatorics. Lastly, we present RA codes with a modified accumulator design in Section IV before concluding in Section V.

II. Repeat accumulate codes

RA codes are serially concatenated Turbo codes with a rate-$\frac{1}{q}$ repetition code and a rate-1 $\frac{1}{1+D}$ convolutional code, the accumulator, as the constituent codes, see Fig. 1. As for Turbo codes, the bits at the output of the repetition code are interleaved before being passed to the accumulator.

The qK bits at the output of the repetition code are q copies of the K message bits $\mathbf{m} = [m_1 \dots m_K]$, in the form

$$\mathbf{b} = [m_1, m_1, \dots, m_1 \quad \dots \quad m_K, m_K, \dots, m_K],$$

and so we have

$$b_i = m_{f(i)}, \qquad f(i) = \lceil i/q \rceil.$$

The interleaver pattern,

$$\Pi = [\pi_1, \pi_1, \dots, \pi_n],$$

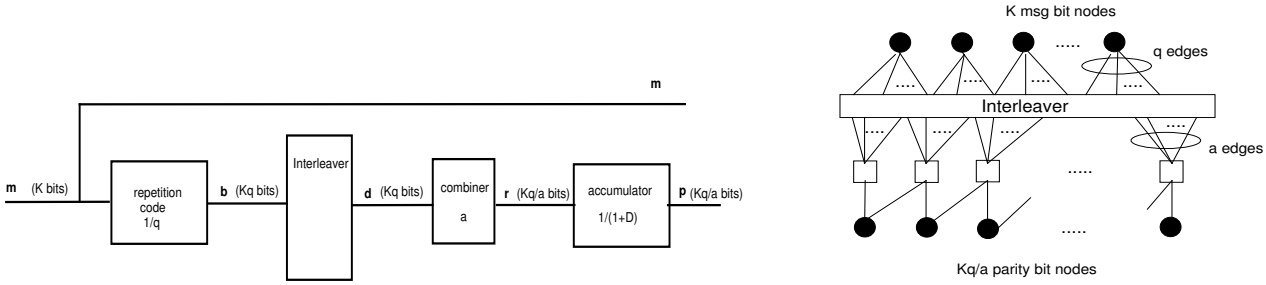

Fig. 1. A systematic RA code viewed as a tubo code, on the left, and as an LDPC code, on the right. Filled circles in the Tanner graph represent bit nodes, while open squares are check nodes.

defines the permutation of the input bits, $\mathbf{b} = [b_1, b_2, \ldots, b_n]$, to the output bits

$$\mathbf{d} = [d_1, d_2, \ldots, d_n] = [b_{\pi_1}, b_{\pi_2}, \ldots, b_{\pi_n}].$$

Thus two different interleaver patterns may describe the same RA code if the difference in the permutation pattern corresponds solely to a different copy of the same message bit.

The bits at the output of the interleaver are combined, modulo 2, in sets of a bits, before being passed to the accumulator. The Kq/a bits, $\mathbf{r}$, at the output of the combiner, are given by

$$r_i = d_{(i-1)a+1} \oplus d_{(i-1)a+2} \oplus \cdots \oplus d_{ia} \tag{1}$$

where $\oplus$ represents modulo 2 addition.

Finally, the kq/a parity bits, $\mathbf{p}$ at the output of the accumulator are described by

$$p_i = p_{i-1} \oplus r_i. \tag{2}$$

We consider systematic RA codes, that is codes for which both the original message bits and the parity bits are sent to the receiver, and so the final codeword is

$$\mathbf{c} = [m_1, m_2, \ldots m_K, p_1, p_2 \ldots, p_{Kq/a}],$$

and thus we have a code with length, N, and rate, R, given by:

$$N = K(1 + q/a), \qquad R = \frac{a}{a+q}.$$

An RA code can also be viewed as an LDPC code where the code parity-check matrix H has two parts;

$$H = [H_1, H_2]. \tag{3}$$

H_1 is a $Kq/a \times K$, column weight q, row weight a, matrix specified by the interleaver. That is, the rows of H_1 describe the equations in (1), i.e. if we have $r_i = m_{c1} + m_{c2}$ then the ith row of H_1 is '1' in the columns $c1$ and $c2$ and '0' elsewhere. H_2 is a $Kq/a \times Kq/a$ matrix which describes (2) and so has the form:

$$\begin{bmatrix} 1 & 0 & 0 & & 0 & 0 \\ 1 & 1 & 0 & & 0 & 0 \\ 0 & 1 & 1 & & 0 & 0 \\ & & & \ddots & & \\ 0 & 0 & 0 & & 1 & 0 \\ 0 & 0 & 0 & & 1 & 1 \end{bmatrix}. \tag{4}$$

The Tanner graph of an RA code is directly described by H and consists of Kq/a parity-check nodes and $K + Kq/a$ bit nodes. Unlike for a general LDPC code, the message bits in the codeword of a systematic RA code are easily distinguished from the parity bits. Fig. 1 shows the Tanner graph for a systematic RA code with the message bits appearing at the top of the graph and the parity bits at the bottom.

While H is not regular in the sense defined for LDPC codes (since H has column weights of q, 2 and 1), an RA code is called regular if the weight of all the rows of H_1 are the same and the weight of all the columns of H_1 are the same. An irregular RA code will have an irregular column weight distribution in H_1, with H_2 the same as for a regular code.

RA codes can be decoded by sum-product decoding on the code's Tanner graph in exactly the same way as for LDPC codes (see e.g. [7] for a description of sum-product decoding). As with LDPC codes, convergence to a valid codeword is easily detected and so it is possible to both halt decoding once a valid codeword has been found and to distinguish between detected and undetected errors.

An RA code is completely specified by N, a, q and the interleaver pattern Π. Thus designing an RA code involves designing the interleaver, or equivalently, the parity-check matrix H_1. For example the length 10 RA code with $q = 3$, $a = 1$ and interleaver

$$\Pi = [1, 7, 4, 10, 2, 5, 8, 11, 3, 9, 6, 12]$$

has the parity-check matrix and Tanner graph given in Fig 2. If a is non-zero the RA interleaver must be constrained in order to avoid repeated edges in the code Tanner graph. For example, an interleaver $\Pi = [1, 2 \cdots]$ for this code will give a repeated edge between the first bit node and first parity-check node and so fails to give a code that can be decoded correctly as an LDPC code.

III. Combinatorial interleavers

Designing a regular RA code requires that we design an interleaver, Π. Since we are decoding the code using sum-product decoding on the code's Tanner graph, rather than by turbo decoding, the performance of the code is determined by the properties of H, as for an LDPC code. In particular small cycles in the codes Tanner graph should be avoided. A cycle in the Tanner graph is a sequence of connected nodes which

$$H = \begin{bmatrix} 1 & . & 1 & . & 1 & . & . & . & . & . \\ . & 1 & . & 1 & 1 & 1 & . & . & . & . \\ 1 & 1 & . & . & . & 1 & 1 & . & . & . \\ . & . & 1 & 1 & . & . & 1 & 1 & . & . \\ 1 & . & 1 & . & . & . & . & 1 & 1 & . \\ . & 1 & . & 1 & . & . & . & . & 1 & 1 \end{bmatrix}$$

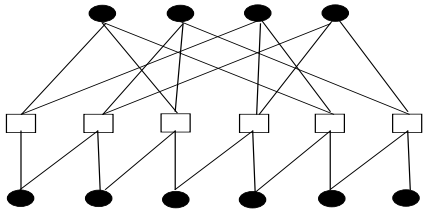

Fig. 2. A length 10 RA code with $q = 3$, $a = 2$ and interleaver $\Pi = [1, 7, 4, 10, 2, 5, 8, 11, 3, 9, 6, 12]$. Zero entries in the parity-check matrix are represented by dots. Filled circles in the Tanner graph represent bit nodes, while open squares are check nodes.

start and end at the same node and contain no other node more than once. A cycle of size 4, called a 4-cycle, will occur if two columns of the parity-check matrix both have a '1' entry in the same two rows.

We propose then to design a good parity-check matrix H, with no 4-cycles, while also constraining our design of H such that it fits the format required of an RA code. The interleaver can then be determined directly from H as shown in the previous section.

Using the intuition gained from the combinatorial design of LDPC codes (see e.g. [8], [9]) we use combinatorics to design H and then show that we can transform H into the form of (3), (4) to produce the interleaver and accumulator of an RA code.

A. Interleavers from combinatorial designs

We will consider RA codes with $q = 3$, and so focus on designs called Steiner triple systems (STS) to construct our codes. However for larger values of q other designs can be used similarly.

An STS design is an arrangement of a set of v *points* into $b = v(v-1)/6$ subsets, called *blocks* such that [10]:

- D1 there are exactly 3 points in each block,
- D2 there are exactly $\rho = (v-1)/2$ blocks which contain each point, and
- D3 every pair of points of the design appear in exactly one block together.

A STS design can be described by a binary $v \times b$ incidence matrix $\mathcal{N}$ where each column in $\mathcal{N}$ represents a block B_j of the design and each row a point P_i:

$$\mathcal{N}_{i,j} = \begin{cases} 1 & \text{if } P_i \in B_j, \\ 0 & \text{otherwise.} \end{cases}$$

Lemma 1: Any permutation of the point set of the design or, equivalently, any permutation of the rows of $\mathcal{N}$ is also an STS design. Similarly, any reordering of the blocks or, equivalently, any permutation of the columns of $\mathcal{N}$ is also an STS design. The proof of Lemma 1 is simply that the properties of an STS design (D1-D3) are independent of the ordering of points and blocks.

By Property D3 we have

$$\forall i \in \{1, \cdots, v-1\} \;\; \exists \;\; B_j \in \mathcal{N} \;\; : \;\; P_i, P_{i+1} \in B_j. \quad (5)$$

We denote by $B(i)$ the block containing both P_i and P_{i+1}. Then if

$$B(i) \neq B(i+1) \;\; \forall i \quad (6)$$

the columns $B(1) \cdots B(v)$ are distinct. Otherwise we need to apply a permutation to the rows of $\mathcal{N}$ so that (6) holds.

Lemma 2: For any $\mathcal{N}$, the incidence matrix of an STS(v) design, we can find some row permutation π so that (6) holds.

Proof: Take any column j of $\mathcal{N}$. There are

$$3!(v-2)(v-3)!$$

permutations of the rows of $\mathcal{N}$ that will place the three non-zero entries of column j into three consecutive rows. We call these the "bad" permutations for for column j. Thus there are

$$v! - 3!(v-2)(v-3)! = (v(v-1)-6)(v-2)!$$

permutations which are "good" for this column. If we assume the worst case, that is that a permutation which is bad for one column is good for all the others, there are $v(v-1)/6$ columns in $\mathcal{N}$ and hence

$$3!(v-2)(v-3)!v(v-1)/6 = v!$$

bad permutations in total. i.e. all of the $v!$ possible permutations produce a bad column. However, we can prove that a good permutation exists by proving that our worst case assumption is in fact incorrect. If we start with two disjoint columns then the set of $3!(v-3)!$ permutations which put the '1's of the first column into the first three rows, includes every possible permutation of the remaining $v-3$ rows. Since the second column has its '1's in these $v-3$ rows, $3!(v-5)(v-6)!$ of these permutations are simultaneously bad for the second column and the proof follows. ■

By Lemma 1, (5) holds for any permutation of the rows of $\mathcal{N}$, and by Lemma 2 we can always find a permutation of the rows of $\mathcal{N}$ such that (6) holds. In practice such a permutation is extremely simple to find. Replacing the third entry in each column $B(1) \cdots B(v)$ (that is the '1' entry which is not in the ith or $i+1$th rows) with '0' and permuting the columns of $\mathcal{N}$ so that the last v columns of $\mathcal{N}$ are $B(1) \cdots B(v)$ gives us the parity-check matrix $H = \mathcal{N}$, of a length b, rate $(b-v)/b$, RA code with $q = 3$. Since every pair of points occurs in exactly one block together, no two columns of $\mathcal{N}$ have a non-zero entry in more than one row in common. Removing entries from $\mathcal{N}$ can't change this, and thus the RA code constructed from $H = \mathcal{N}$ has a Tanner graph free of 4-cycles.

We demonstrate in Fig. 3 the performance, on an additive white Gaussian noise (AWGN) channel, of two RA codes constructed from STS designs, compared to RA codes with randomly constructed interleavers. The performance of random interleavers is averaged over the ensemble of all possible interleavers excluding those interleaver patterns which result

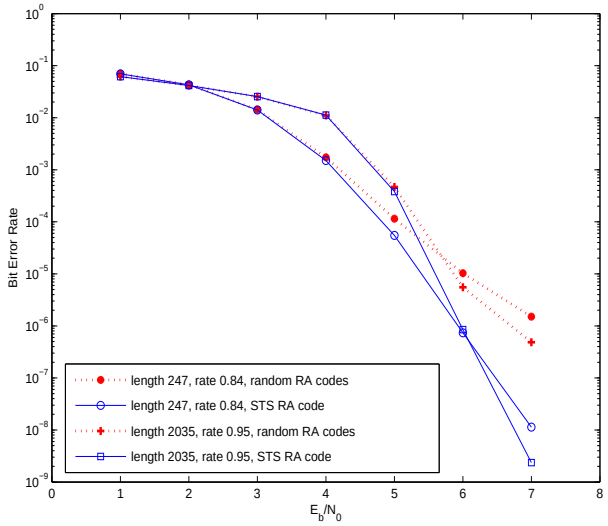

Fig. 3. The error correction performance on an AWGN channel of RA codes from STS designs.

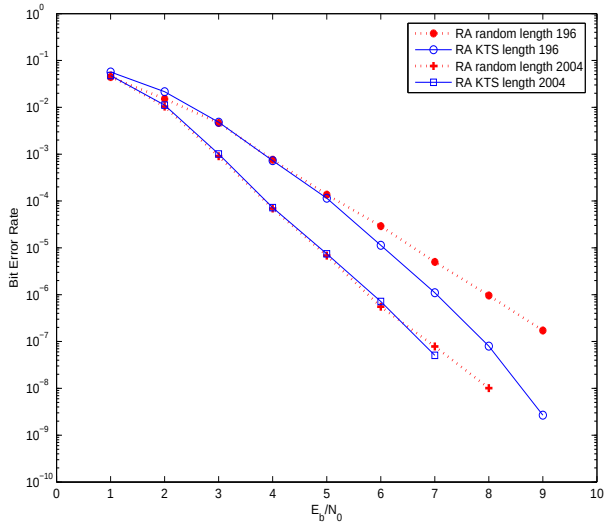

Fig. 4. The error correction performance on an AWGN channel of rate-$1/4$ RA codes.

in repeated edges in the Tanner graph. For both the STS and randomly constructed codes we design codes with fixed q, and row degrees as regular as possible. We see that the codes from STS designs significantly outperform randomly constructed RA codes.

The construction of RA codes using STS designs gives us codes with high rates, comparable to those likely to be used in applications such as magnetic recording, for which good randomly constructed interleavers are the most difficult to find. To produce codes with lower rates, for applications such as wireless communications, we propose in the following section a construction modification employing resolvable designs.

B. Flexible interleavers from designs

Kirkman triple system (KTS) designs are STS designs which are resolvable, that is the blocks of the design can be arranged into r groups, called resolution classes, such that the $\frac{v}{3}$ blocks of each resolution class are disjoint, and each class contains every point precisely once. If the columns of the incidence matrix of a KTS design are arranged into its resolution classes, each set of $v/3$ columns in $\mathcal{N}$ will contain a '1' entry in every row. Constructions for KTS designs on v points, denoted KTS(v), exist for all $v \equiv 3 \pmod 6$ [10, p. 89, Theorem 6.7].

We use Kirkman triple systems to construct RA codes with

$$N = (a+3)\frac{v}{3}, \quad R = \frac{a}{a+3},$$

for any $a = \{1, \cdots, (v-7)/2\}$ as follows:

Construction 1:

- Step 1 Order the columns of $\mathcal{N}$ into its resolution classes.
- Step 2 For $i = 1, \ldots, v-1$
  a) Find the minimum $l \geq i$ such that the lth column of $\mathcal{N}$ contains a '1' in row i.
  b) If $l \leq v$ and there exists a '1' in the jth row of column l such that $j \geq i+1$, swap the lth and ith columns of $\mathcal{N}$ and swap the jth and $i+1$th rows of $\mathcal{N}$. Otherwise find the column $B(i)$ in $\mathcal{N}$ and swap this column with the ith column of A. In the second case the resolution class which contained $B(i)$ must now be excluded from use in H_1 as it is no longer regular.
  c) If the third non-zero entry of column i is in position $i+2$ randomly swap this row with some other row r with $r > i+3$ (so that a pair of consecutive points is not deleted in step d).
  d) Replace the remaining entry of '1' in the ith column (i.e. the entry not in rows i and $i+1$) with '0'.

  The first v columns of $\mathcal{N}$ now give H_2
- Step 3 Take the blocks from a of the unused resolution classes in $\mathcal{N}$ to make up the columns of H_1. If there are insufficient full resolution classes remaining choose the columns to keep H_1 as regular as possible.

In practice very few columns outside of the first v are needed to form H_2 and so it is possible to construct a completely regular RA code for most values of a. Again, because we start with $\mathcal{N}$ the incidence matrix of an KTS design (which is a special case of an STS design), we have codes with Tanner graphs free of 4-cycles.

Figs. 4-6 show the performance, on an AWGN channel, of several KTS RA codes of varying rate and length compared to RA codes with randomly chosen interleavers. We consider codes with rates and lengths likely to be used in wireless applications such as IEEE802.11n and IEEE802.16d. Both the KTS and the randomly constructed codes have fixed a, and fixed $q = 3$. Using a combinatorial construction offers performance advantages for the RA codes, particularly at high rates where the large a means that a much greater number of the parity-check matrix columns are determined by the interleaver rather than by the accumulator.

IV. Column weight three accumulators

Since the accumulator corresponds to weight two columns in the parity-check matrix of an RA code, the fraction $(1-R)$ of the columns of a rate R code must be weight two. This large portion of weight two columns is, in low rate codes, even more than the number typically recommended by density evolution

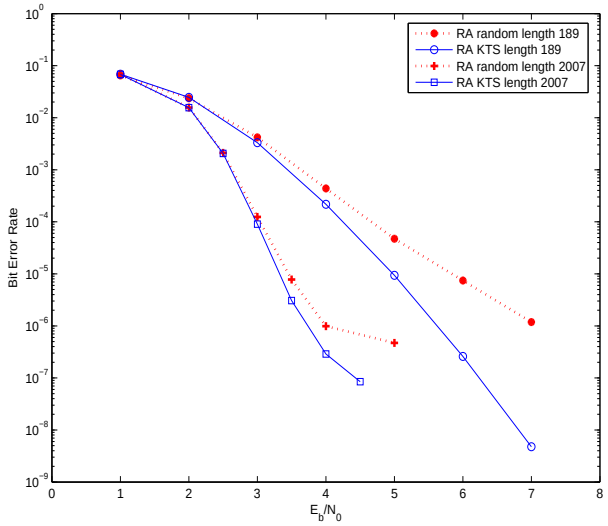

Fig. 5. The error correction performance on an AWGN channel of rate-$2/3$ RA codes.

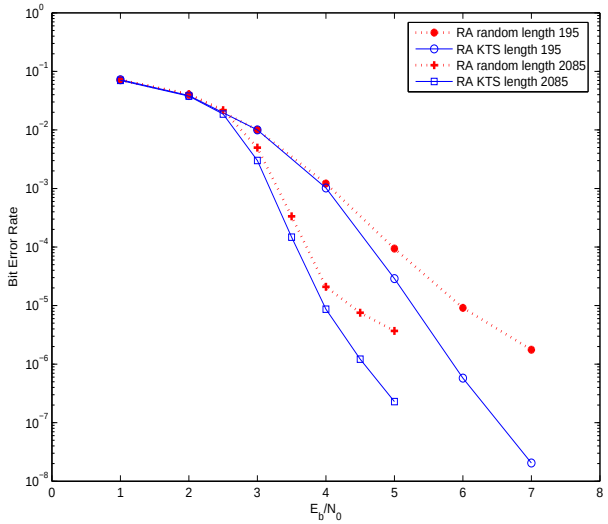

Fig. 6. The error correction performance on an AWGN channel of rate-$4/5$ RA codes.

for irregular LDPC codes, and results in a low error floor in the decoding performance of the code (see e.g. [7] for a discussion of column weight and code performance). In this section we propose an alternative accumulator design which dramatically reduces this error floor. We then propose interleaver designs which provide 4-cycle free codes with this new accumulator.

For a weight 3 accumulator we propose the use of a rate-1

$$\frac{1}{1+D+D^{g+1}}$$

convolutional code as the accumulator, where g becomes a design parameter of the new RA codes. Translating from the accumulator to the parity-check matrix, the ith column of the accumulator will be non-zero in the ith, $i+1$th and $i+1+g$th rows. Thus g specifies the number of rows between the second and third entry in the columns of the accumulator as well as the number of weight two columns remaining in the accumulator. While reducing g will increase the average column weight, g cannot be reduced too far as the accumulator adds cycles of size $2(g+1)$. For example, Fig. 7 shows the code of Fig. 2 now with a weight three accumulator specified by $g=2$.

With columns of weight three in the accumulator, there are now three times as many row pairs present in H_2 and so it becomes more difficult to construct interleavers without adding 4-cycles into the code. However, we see that using the properties of KTS designs it is still relatively straightforward to construct 4-cycle free codes.

Using the incidence matrix, $\mathcal{N}$, of KTS design on v points we construct a 4-cycle free length $(a+3)*\frac{v}{3}$ rate-$\frac{a}{a+3}$ RA code, with all but $g+1$ of the columns of the accumulator weight 3, as follows:

Construction 2: Replace parts b) to d) of Construction 1 with the following:

b) If $l \leq v$ and there exists a '1' in the jth row of column l such that $j \geq i+1+g$, swap the lth and ith columns of $\mathcal{N}$ and swap the jth and $i+1$th rows of $\mathcal{N}$. Otherwise find the column $B(i)$ in $\mathcal{N}$ and swap this column with the ith column of A. In the second case the resolution class which contained $B(i)$ must now be excluded from use in H_1 as it is no longer regular.

c) (When $i < v-g-1$) If the third entry of the new ith column is in row $j \geq i+g$, switch the jth and $i+g+1$th rows of $\mathcal{N}$. Otherwise place a zero in entry $\mathcal{N}(j,i)$ and a one in entry $\mathcal{N}(i+g+1,i)$. In the second case the two columns with the pairs $\{i, i+g+1\}$ and $\{i+1, i+g+1\}$ respectively must now be excluded from use in H_1 in order to avoid all 4-cycles.

c) (When $i \geq v-g-1$) If the third non-zero entry of column i is in position $i+3$ randomly swap this row with some other row $r > i+3$. Replace the third non-zero entry in column i with '0'.

Using our new accumulator along with KTS designs we can now construct LDPC codes, with only a small number of columns with weight less than three, which are free of 4-cycles and have extremely simple encoding circuits.

Figs. 8-10 show the performance, on an AWGN channel, of the new RA codes for the same rates as simulated in the previous section. We show only RA codes constructed using designs in these figures as the KTS codes are as good as or better than all of the randomly constructed codes that we have simulated. The new RA codes, labeled w3RA codes, all have a constant a, and a constant $q=3$. We see that, especially for high rate codes which previously had a large proportion of weight 2 columns, the error floor can be significantly lowered by using the new accumulator design. Using KTS designs is an efficient way of constructing RA codes with weight 3 accumulators and without small cycles in the code Tanner graph.

V. Conclusion

In this paper we have considered the design and implementation of practical RA codes. In particular we provide construction methods for the interleaver of fixed length codes, and improve the error floor performance of RA codes by suggesting a new accumulator design. In the terminology of LDPC codes we have, equivalently, provided a construction

$$H = \begin{bmatrix} 1 & . & 1 & . & 1 & . & . & . & . & . \\ . & 1 & . & 1 & 1 & 1 & . & . & . & . \\ 1 & 1 & . & . & . & 1 & 1 & . & . & . \\ . & . & 1 & 1 & 1 & . & 1 & 1 & . & . \\ 1 & . & 1 & . & . & 1 & . & 1 & 1 & . \\ . & 1 & . & 1 & . & . & 1 & . & 1 & 1 \end{bmatrix}$$

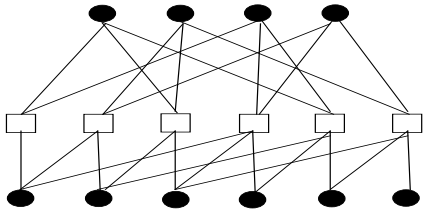

Fig. 7. A length 10 RA code with $q = 3$, $a = 2$, interleaver $\Pi = [1, 7, 4, 10, 2, 5, 8, 11, 3, 9, 6, 12]$, and the new accumulator with $g = 2$. Zero entries in the parity-check matrix are represented by dots. Filled circles in the Tanner graph represent bit nodes, while open squares are check nodes.

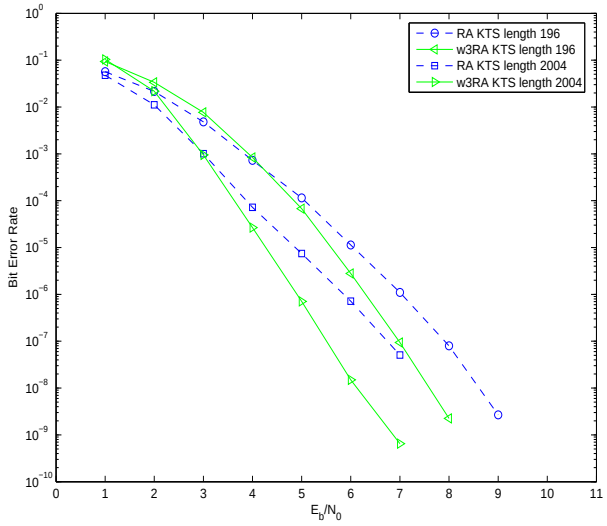

Fig. 8. The error correction performance on an AWGN channel of rate-$1/4$ RA codes with weight two (RA) and weight three (w3RA) accumulators.

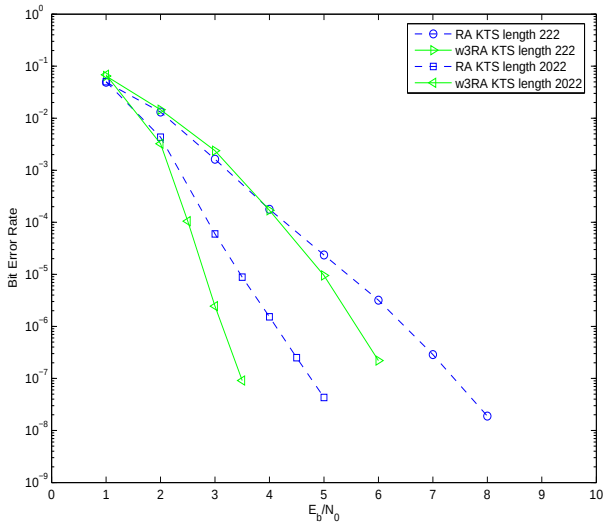

Fig. 9. The error correction performance on an AWGN channel of rate-$1/2$ RA codes with weight two (RA) and weight three (w3RA) accumulators.

method for nearly $(3, r)$-regular LDPC codes which have an extremely simple encoding circuit.

A possible extension, to be considered in future work, is to investigate further modifications to the interleaver design in order to improve the minimum distance of the RA codes. Such extensions could be considered equally for randomly constructed interleavers as well as for interleavers constructed from KTS designs.

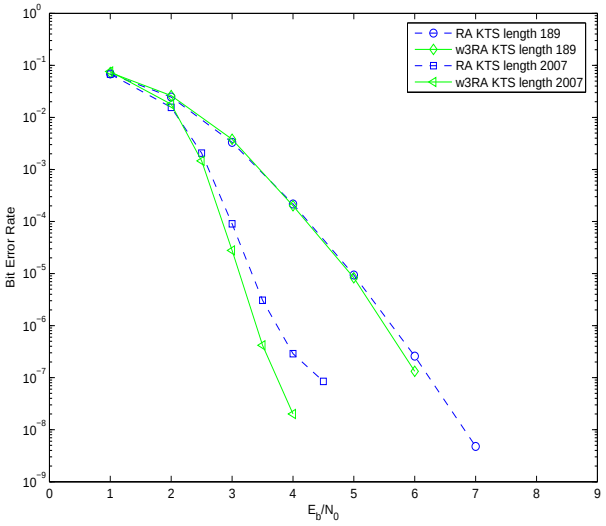

Fig. 10. The error correction performance on an AWGN channel of rate-$2/3$ RA codes with weight two (RA) and weight three (w3RA) accumulators.

REFERENCES

[1] D. Divsalar, H. Jin, and R. J. McEliece, "Coding theorems for "turbo-like" codes," in *Proc. 36th Allerton Conf. on Communications, Control, and Computing*, Allerton, Illinois, September 1998, pp. 201–210.

[2] M. Yang, W. E. Ryan, and Y. Li, "Design of efficiently encodable moderate-length high-rate irregular LDPC codes," *IEEE Trans. Comms.*, vol. 52, no. 4, pp. 564–571, April 2004.

[3] H. Jin, D. Khandekar, and R. J. McEliece, "Irregular repeat-accumulate codes," in *Proc. of the Second International Symposium on Turbo Codes and Related Topics*, Brest, France, September 2000, pp. 1–8.

[4] T. J. Richardson, M. A. Shokrollahi, and R. L. Urbanke, "Design of capacity-approaching irregular low-density parity-check codes," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 619–637, February 2001.

[5] A. Roumy, S. Guemghar, G. Caire, and S. Verdú, "Design methods for irregular repeat-accumulate codes," *IEEE Trans. Inform. Theory*, vol. 50, no. 8, pp. 1711–1727, August 2004.

[6] I. Sason and R. Urbanke, "Complexity versus performance of capacity-achieving irregular repeat-accumulate codes on the binary erasure channel," *IEEE Trans. Inform. Theory*, vol. 50, no. 6, pp. 1247–1256, June 2004.

[7] D. J. C. MacKay, "Good error-correcting codes based on very sparse matrices," *IEEE Trans. Inform. Theory*, vol. 45, no. 2, pp. 399–431, March 1999.

[8] R. Lucas, M. P. C. Fossorier, Y. Kou, and S. Lin, "Iterative decoding of one-step majority logic decodable codes based on belief propagation," *IEEE Trans. Commun.*, vol. 48, no. 6, pp. 931–937, June 2000.

[9] S. J. Johnson and S. R. Weller, "Resolvable 2-designs for regular low-density parity-check codes," *IEEE Trans. Commun.*, vol. 51, no. 9, pp. 1413–1419, September 2003.

[10] C. J. Colbourn and J. Dinitz (Eds.), *The CRC Handbook of Combinatorial Designs*. Boca Raton: CRC Press, 1996.