

# Practical Interleavers for Repeat–Accumulate Codes

Sarah J. Johnson *Member IEEE* & Steven R. Weller *Member IEEE*

**Abstract**—In this paper we design practical interleavers for regular, systematic repeat-accumulate (RA) codes. The new interleavers, which we call *L*-type and modified *L*-type interleavers, are deterministic, described by a single parameter, and straightforward to implement. Despite their simple description, the new interleavers are shown to perform equally as well as, or better than, traditional interleavers over a wide range of code lengths and rates.

## I. INTRODUCTION

Repeat-accumulate (RA) codes, introduced in [1], are simultaneously a class of simple “turbo-like” codes and a class of low-density parity-check (LDPC) codes. This dual representation of RA codes allows the flexibility of using a turbo code representation for the encoding and an LDPC code representation for the decoding, thereby gaining the benefits of both schemes.

When viewed as a serially concatenated 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 an interleaver between them. In some cases, a combiner, which performs modulo-2 addition on sets of  $a$  bits, can be added between the interleaver and the accumulator. For fixed  $q$  and  $a$  we call the RA code  $(q, a)$ -regular. The input to the repetition code is the  $K$ -bit message and the output of the accumulator is the  $M = Kq/a$ -bit parity string. We consider systematic codes, so that the length  $N = K + M$  codeword contains the message and parity bits as contiguous substrings.

When viewed as an LDPC code, the accumulator and interleaver determine the structure of the parity-check matrix  $H = [H_1, H_2]$ . Here  $H_1$  is an  $M \times K$ , column weight  $q$ , row weight  $a$ , matrix specified by the interleaver, and  $H_2$  is an  $M \times M$  dual diagonal matrix representing the accumulator (see e.g. [2]). LDPC codes with a dual diagonal submatrix in  $H$ , as in [2], [3], can be interpreted as RA codes.

RA codes are decoded by sum-product decoding on the code’s Tanner graph in exactly the same way as for LDPC codes (see e.g. [4] 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.

While the RA interleaver can be chosen randomly, and indeed for long codes this produces very good performance results, random interleavers present significant challenges to implementation. Traditional turbo interleavers which have been designed for other constituent codes do not work well with the repetition and accumulator component codes in RA

codes. Further, since repeat-accumulate codes are decoded as LDPC codes, the interleaver must be designed to control the properties of the resulting Tanner graph, and so even general turbo interleavers such as the row-column (or block) interleaver perform poorly as RA interleavers. Moreover, an arbitrarily chosen interleaver is not even guaranteed to be valid for an RA code, since a Tanner graph with repeated edges will result if the repeated bits at the output of the repetition code are too close at the output of the interleaver.

The aim of this paper is to design simple and effective RA code interleavers by considering the resulting LDPC code structure. Section II details our proposed interleavers, the performance of the new interleavers is compared with some traditional interleavers in Section III and we conclude in Section IV.

## II. RA INTERLEAVERS

Designing a  $(q, a)$ -regular RA code requires that we design an interleaver,  $\Pi = [\pi_1, \pi_2, \dots, \pi_n]$ , such that

$$\pi_i \in \{1, 2, \dots, n\}, \pi_i \neq \pi_j \forall i \neq j, \text{ and } n = Kq.$$

The  $i$ -th entry,  $\pi_i$ , specifies that the  $\lceil i/a \rceil$ -th row of  $H$  has a one in the  $\lceil \pi_i/q \rceil$ -th column, where  $\lceil x \rceil$  denotes the smallest integer greater than or equal to  $x$ .

### A. Desired RA interleaver properties

Since  $H$  is binary, repeated entries in  $H$  are not permitted, which is equivalent to requiring that

$$\lceil \pi_i/q \rceil \neq \lceil \pi_j/q \rceil \quad \forall i \neq j \text{ such that } \lceil i/a \rceil = \lceil j/a \rceil. \quad (1)$$

Since we are decoding the code using sum-product decoding rather than by turbo decoding, the performance of the code is determined by the properties of the Tanner graph, as for an LDPC code. In particular, the decoder performance is improved if short cycles in the code’s Tanner graph are avoided. A cycle of length  $l$ , called an  $l$ -cycle, is a path through the Tanner graph that includes  $l$  edges and closes back on itself. The girth of the graph is the length of its shortest cycle, and is a minimum of 4. For an RA code we define two classes of 4-cycles. A *type-1 4-cycle* is said to occur if a column in  $H_1$  contains two consecutive ones (with the second pair of ones comprising the 4-cycle occurring in  $H_2$ ). A *type-2 4-cycle* is formed if two columns of  $H_1$  contain two entries in common.

To avoid type-1 4-cycles requires that

$$\lceil \pi_i/q \rceil \neq \lceil \pi_j/q \rceil \quad \forall i \neq j \text{ such that } \lceil i/a \rceil = \lceil j/a \rceil \pm 1,$$

while to avoid type-2 4-cycles requires that

$$\lceil \pi_j/q \rceil \neq \lceil \pi_i/q \rceil \quad \forall i \neq j \text{ such that } \exists k, l$$

where  $\lceil l/a \rceil = \lceil j/a \rceil$ ,  $\lceil k/a \rceil = \lceil i/a \rceil$ ,  $\lceil \pi_l/q \rceil = \lceil \pi_k/q \rceil$ .

This work was supported by the Australian Research Council under grants DP0449627 and DP0665742. An earlier version of this paper was presented at the IEEE 63rd Vehicular Technology Conference, Melbourne, Australia, May 2006. The authors are with the School of Electrical Engineering and Computer Science, The University of Newcastle, Callaghan 2308, NSW Australia. (email: {sarah.johnson, steven.weller}@newcastle.edu.au)

Cycles cannot be formed solely within the columns of  $H_2$  and so type-1 and type-2 4-cycles cover all possible 4-cycles in an RA code.

Type-1 4-cycles can be avoided by using a popular turbo interleaver called an  $S$ -type interleaver [5]. In an  $S$ -type interleaver no two entries of  $\Pi$  within  $S$  of each other have a value within  $S$  of each other. Pseudo-randomly constructed  $S$ -type interleavers are called  $S$ -random interleavers. While type-1 4-cycles can be avoided by specifying that  $S \geq \max(q-1, 2a-1)$ , this can be a more stringent requirement than what is actually required to avoid type-1 4-cycles. More importantly though, an  $S$ -type interleaver only acts within the  $S$  adjacent entries of the interleaver and so cannot be used to avoid type-2 4-cycles which can be formed by interleaver entries spaced widely apart.

Another popular turbo interleaver is the row-column (or block) interleaver which, unlike  $S$ -random interleavers, has a very simply algorithmic implementation. A length- $n$  row-column interleaver is achieved by writing the bits 1 through  $n$  into a square matrix row-wise and reading them out column-wise. However, the row-column interleaver is particularly bad for RA codes since it can add a large number of 4-cycles when  $a > 1$ .

*Example 1:* The parity-check matrix of a length-16, rate- $1/2$ ,  $(2, 2)$ -regular RA code with a row-column interleaver is:

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

There are four type-2 4-cycles in  $H$ , one of which is shown in bold.

We require for RA codes an interleaver which, like the row-column interleaver, is easy to describe, and thus implement, but which avoids 4-cycles in the code. The first interleaver we propose for RA codes, and call an  $L$ -type interleaver, is presented in the following.

### B. $L$ -type interleavers

Given  $K$ ,  $q$  and the integer-valued interleaver parameter  $L$ , the  $L$ -type interleaver is defined in (2). The repetition code and interleaver can together be thought of as selecting from the  $K$  message bits  $q$  times. First the  $K$  message bits are selected in order in  $\Pi_1$ . Then selecting from the message bits in  $\Pi_1$ ,  $\Pi_2$  is produced by skipping  $L$  bits ahead after each selection and starting again with the first unselected bit each time until all  $K$  message bits in  $\Pi_1$  have been selected. This process is repeated  $q-1$  times, selecting from  $\Pi_{i-1}$  to form  $\Pi_i$ . In practice the bits  $iK+1$  to  $(i+1)K$  of  $\Pi$ , which we denote by  $\Pi_{i+1}$ , are a type of row-column permutation of bits  $(i-1)K+1$  to  $iK$  of  $\Pi$ , denoted by  $\Pi_i$ , where the bits are written row-wise into a matrix with  $L$  columns and read out column wise. Thus the  $L$ -type interleaver can be implemented as the concatenation of  $q-1$  of these matrix row-write, column-read, operations. See Fig. 1.

*Example 2:* The parity-check matrix of a length-16, rate- $1/2$ ,  $(2, 2)$ -regular RA code with the  $L$ -type ( $L=2$ ) interleaver  $\Pi = [1, 3, 5, 7, 9, 11, 13, 15, 2, 6, 10, 14, 4, 8, 12, 16]$  is:

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

As well as being simple to describe, a key benefit of the  $L$ -type interleaver is that some girth properties for the codes can be guaranteed. In the following we denote by  $R_i$  the set of  $K$  rows of  $H_1$  corresponding to  $\Pi_i$ .

*Lemma 1:* A  $(3, a)$ -regular RA code can be constructed without 4-cycles whenever  $K > a^3$  by using an  $L$ -type interleaver and setting  $L = a$ .

*Proof:* Firstly we note that the rows in each set,  $R_i$ , are disjoint and so a 4-cycle will only occur across rows from two different sets. Thus a type-1 4-cycle can only be formed between the last row of one set and the first row of the next. This will only occur if the last  $a$  bits of  $\Pi_i$ , which correspond to the last row of  $R_i$ , have entries in common with the first  $a$  bits of  $\Pi_{i+1}$ , which will be selected from to form the first row of  $R_{i+1}$ . The first  $a$  bits of  $\Pi_{i+1}$  are selected from the first  $La$  bits of  $\Pi_i$ . Since  $K > La + a = a^2 + a$  the first  $La$  and last  $a$  bits of  $\Pi_i$  do not overlap and thus type-1 4-cycles are always avoided. Next, setting  $L = a$  ensures that each of the bits in a row in  $R_2$  are chosen from different rows in  $R_1$ , and similarly that each of the bits in a row in  $R_3$  are chosen from different rows in  $R_2$ . Further, since  $K/L > a$  the interleaver cannot wrap back around the  $K$  bits within a single row in  $R_2$ . Setting  $(K/L)/L > a$  i.e.  $K > a^3$  ensures that the interleaver cannot wrap back around the  $K$  bits within a single row in  $R_2$  and so type-2 cycles are also avoided. ■

Furthermore, the same construction can be used to guarantee RA codes without 6-cycles. We define a *type-1 6-cycle* as one formed from two accumulator columns, a *type-2 6-cycle* as one involving one accumulator column, and finally a *type-3 6-cycle* as one formed from three columns solely within  $H_1$ .

*Lemma 2:* A  $(3, a)$ -regular RA code can be constructed without 6-cycles whenever  $K \geq 8a^3$  by using an  $L$ -type interleaver and setting  $L = 2a$ .

*Proof:* Since each set of rows  $R_i$  are disjoint, a 6-cycle must occur across three row sets. Thus a type-1 6-cycle can only be formed between the 2nd last (or last) rows of one set and the first (respectively, second) row of the next. This will only occur if the last  $2a$  entries of  $\Pi_i$ , which correspond to the last two rows of  $R_i$ , are within the first  $2La$  entries of  $\Pi_i$  which will be selected from to form the first two rows of  $R_{i+1}$ . Since  $K > 2La + 2a = 4a^2 + 2a$  this cannot occur and type-1 6-cycles are always avoided. Next setting  $L = 2a$  ensures that each of the bits in a row in  $R_2$  are chosen from rows in  $R_1$  separated by at least one row, and similarly that each of the bits in a row in  $R_3$  are chosen from rows in  $R_2$  separated by at least one row. Further, since  $K/L > 2a$  the interleaver cannot wrap back around the  $K$  bits within two consecutive rows in  $R_2$ . Setting  $(K/L)/L > 2a$  ensures that the interleaver cannot wrap back around the  $K$  bits within within two consecutive rows in  $R_3$  and so type-2 6-cycles are also avoided. Finally a type-3 6-cycle must involve one row from each of the three sets. Given any two overlapping rows in  $R_1$  and  $R_2$  the column entries in the row from  $R_1$  are consecutive while the column

$$\Pi = [\Pi_1(1), \dots, \Pi_1(K)], \dots, [\Pi_{q-1}(1) + (q-2), \dots, \Pi_{q-1}(K) + (q-2)], [\Pi_q(1) + (q-1), \dots, \Pi_q(K) + (q-1)], \quad (2)$$

where

$$\Pi_1 = [1, 1 + q, 1 + 2q, 1 + 3q, \dots, 1 + (K-1)q],$$

$$\Pi_2 = [\Pi_1(1), \Pi_1(1+L), \Pi_1(1+2L), \dots, \Pi_1(1+K-L)], [\Pi_1(2), \Pi_1(2+L), \Pi_1(2+2L), \dots, \Pi_1(2+K-L)], \dots, [\Pi_1(L), \Pi_1(L+L), \Pi_1(L+2L), \dots, \Pi_1(L+K-L)],$$

$$\Pi_q = [\Pi_{q-1}(1), \Pi_{q-1}(1+L), \Pi_{q-1}(1+2L), \dots, \Pi_{q-1}(1+K-L)], [\Pi_{q-1}(2), \Pi_{q-1}(2+L), \Pi_{q-1}(2+2L), \dots, \Pi_{q-1}(2+K-L)], \dots, [\Pi_{q-1}(L), \Pi_{q-1}(L+L), \Pi_{q-1}(L+2L), \dots, \Pi_{q-1}(L+K-L)].$$

and  $\Pi_i(j)$  is the  $j$ -th element of  $\Pi_i$

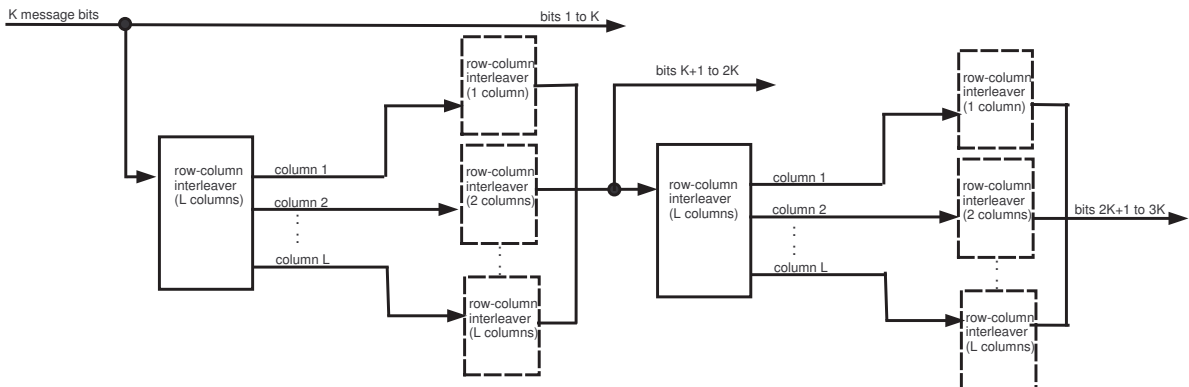


Fig. 1. Block diagram of the encoding circuit for a combined  $q = 3$  repetition code and modified  $L$ -type interleaver. Removing the dashed row-column interleaver blocks gives an  $L$ -type interleaver.

entries in the row in  $R_2$  are spaced  $L$  apart. Thus any two column entries from the row in  $R_1$  and the row in  $R_2$  are at most  $(a-1) + L(a-1) = 2a^2 - a - 1$  columns apart. However the column entries in the rows in  $R_3$  are at least  $L^2 = 4a^2$  columns apart and so we cannot find a row in  $R_3$  which overlaps both of the two rows from  $R_1$  and  $R_2$ . Thus type-3 6-cycles are always avoided. ■

For the  $a = 1$  case we can do even better, and prove that girth  $\geq 10$  codes are produced for  $L = 2$  and  $K$  odd,  $K \geq 7$ , and girth  $\geq 12$  codes for  $L = 3$  and  $K = 1, 2 \pmod{3}$ ,  $K \geq 21$ . Unfortunately, however, when  $a > 1$  the  $L$ -type interleaver always adds cycles of size 8. To see this consider that the first row of  $R_2$  contains ones in columns 1 and  $L+1$ , while a later row in  $R_2$  will contain ones in columns 2 and  $2+L$ . The columns 1 and 2 share a row in  $R_1$ , as do the columns  $L$  and  $L+1$ . These four rows always make up an 8-cycle, and unfortunately there are many 8-cycles formed similarly. Thus while the  $L$ -type interleaver will produce good codes for shorter lengths, where a girth of eight is beneficial, it will not be a good choice for long codes where codes with large girth can easily be constructed randomly. For longer codes we propose a modified  $L$ -type interleaver in the following which breaks these 8-cycles.

### C. Modified $L$ -type interleavers

In this interleaver  $\Pi_1$  is unchanged and  $\Pi_i$  is formed starting with the same process as previously, i.e. the bits of  $\Pi_{i-1}$  are written row-wise into a matrix,  $M_i$  with  $L$  columns and read out column-wise. Now, additionally, the bits from each column are written row-wise into another matrix,  $A_j$ , and read out column-wise. The  $j$ -th column of  $M_i$  is written into a matrix  $A_j$  having  $j$  columns.

The implementation of this second RA interleaver now requires  $(q-1)L$  matrix write-read operations instead of  $q-1$ . However the modified interleaver is still extremely simple to specify, requiring just three integer-valued parameters:  $L$ ,  $q$  and  $K$ . Fig. 1 shows the encoding system for a combined repetition code and modified  $L$ -type interleaver. Decoding is performed on the code Tanner graph and so interleave and de-interleave operations need not be used at the decoder. However, the simple description of the interleaver means that the Tanner graph may not need to be stored but rather specified by the parameters  $K$ ,  $L$  and  $q$  and constructed as needed.

While this new construction method breaks the 8-cycles, and, as we will see from performance results, works very well, it can add 4-cycles in the process. Consequently, we cannot present analogous results to Lemmas 1 and 2 for the modified  $L$ -type interleavers. However, this happens so rarely that the decoding performance of the codes is not appreciably affected.

## III. DECODING PERFORMANCE

In this section we consider the decoding performance of RA codes with the proposed interleavers on the binary-input additive white Gaussian noise channel when decoded with sum-product decoding.

Figs. 2–4 show the performance of the proposed  $L$ -type and modified  $L$ -type interleavers over a range of code rates and lengths compared with the performance of existing pseudo-random and deterministic interleavers. These interleavers are: row-column interleavers; randomly constructed interleavers subject to (1);  $S$ -random interleavers which achieve the largest possible  $S$  value;  $\pi$ -rotation LDPC codes; and combinatorial LDPC codes. The  $\pi$ -rotation LDPC codes [3] correspond to

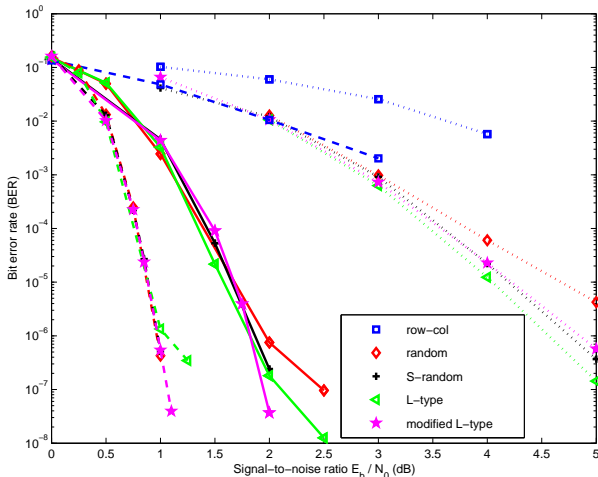


Fig. 2. The performance of rate-1/4,  $q = 6$ ,  $a = 2$ , RA codes of length-200 with max. 100 iterations (dotted lines), length-2000 with max. 1000 iterations (solid lines) and length-10,000 with max. 1000 iterations (dashed lines). The  $L$ -type codes are constructed with  $L = 8$  while the modified  $L$ -type interleavers use  $L = 3, 20$  and  $20$  respectively.

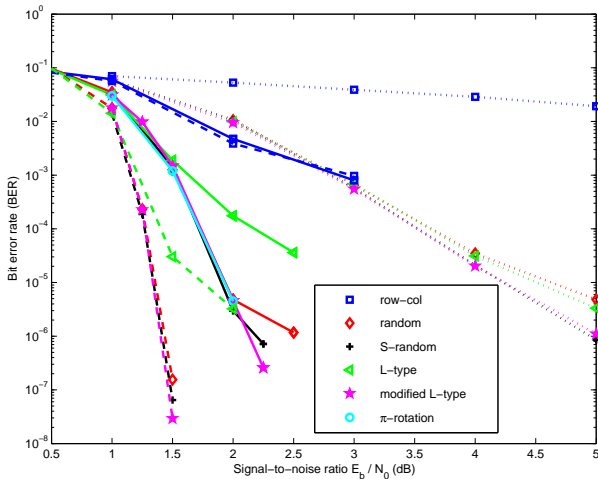


Fig. 3. The performance of rate-1/2,  $q = 4$ ,  $a = 4$ , RA codes of length-200 with max. 100 iterations (dotted lines), length-2000 with max. 1000 iterations (solid lines) and length-10,000 with max. 1000 iterations (dashed lines). The  $L$ -type codes are constructed with  $L = 8, 8$  and  $15$ , while the modified  $L$ -type interleavers use  $L = 5, 20$  and  $20$  respectively.

rate-1/2, regular RA codes with  $q = 4$ . They require a length- $n/8$  permutation vector to be designed and specified for each code rather than the single parameter,  $L$ , required for the  $L$ -type and modified  $L$ -type RA codes. The combinatorial LDPC codes [6] can be described by a limited parameter set, but involve a significantly more complex implementation than  $L$ -type codes, and perform very well at high rates and short to medium code lengths.

As expected, the RA codes with row-column interleavers perform very poorly at all lengths and rates, while the codes with random and  $S$ -random interleavers generally perform well. Despite their simple description the  $L$ -type interleavers perform extremely well for short RA codes, out-performing all but the short high-rate combinatorial code. The modified  $L$ -type interleavers perform extremely well at all lengths and rates considered; even for long codes of around 10,000 bits in length the modified  $L$ -type interleavers perform almost indistinguishably from randomly constructed interleavers.

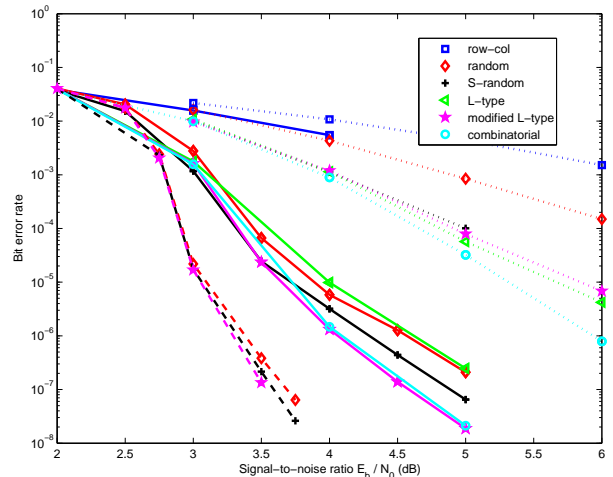


Fig. 4. The performance of rate-4/5,  $q = 3$ ,  $a = 12$ , RA codes of length-195 with max. 10 iterations (dotted lines), length-2085 with max. 100 iterations (solid lines) and length-10,000 with max. 1,000 iterations (dashed lines). The  $L$ -type codes are constructed with  $L = 12$  and  $19$ , while the modified  $L$ -type interleavers use  $L = 12, 19$  and  $30$  respectively.

#### IV. CONCLUSION

In this paper we have presented straightforward deterministic construction methods for practical RA interleavers. These interleavers are shown to give excellent decoding performances for RA codes over a wide range of code lengths and rates. The deterministic nature of the interleaver gives improved performance over random interleavers for short codes, where structure is required to avoid codes with small cycles in their Tanner graphs, without hindering their performance in very long codes. Thus the new interleavers represent an excellent choice over both existing simple interleavers, which they outperform, and randomly constructed interleavers which they match in performance, but with the added advantage of a deterministic description.

#### 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] R. Echard and S.-C. Chang, "The  $\pi$ -rotation low-density parity-check codes," in *Global Telecommunications Conference, San Antonio, TX, 2001*, November 2001, pp. 980–984.
- [4] 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.
- [5] S. Dolinar and D. Divsalar, "Weight distributions for turbo codes using random and non-random permutations," Aug. 1995, JPL TDA Progress Report 42-144.
- [6] S. J. Johnson and S. R. Weller, "Combinatorial interleavers for systematic regular repeat-accumulate codes," to appear *IEEE Transactions on Communications*, 2007.