Accumulate-Repeat-Accumulate Codes

Aliazam Abbasfar, *Member, IEEE*, Dariush Divsalar, *Fellow, IEEE*, and Kung Yao, *Life Fellow, IEEE*

***Abstract*—In this paper, we propose an innovative channel coding scheme called accumulate-repeat-accumulate (ARA) codes. This class of codes can be viewed as serial turbo-like codes or as a subclass of low-density parity check (LDPC) codes, and they have a projected graph or protograph representation; this allows for high-speed iterative decoding implementation using belief propagation. An ARA code can be viewed as precoded repeat accumulate (RA) code with puncturing or as precoded irregular repeat accumulate (IRA) code, where simply an accumulator is chosen as the precoder. The amount of performance improvement due to the precoder will be called precoding gain. Using density evolution on their associated protographs, we find some rate-1/2 ARA codes, with a maximum variable node degree of 5 for which a minimum bit SNR as low as 0.08 dB from channel capacity threshold is achieved as the block size goes to infinity. Such a low threshold cannot be achieved by RA, IRA, or unstructured irregular LDPC codes with the same constraint on the maximum variable node degree. Furthermore, by puncturing the inner accumulator, we can construct families of higher rate ARA codes with thresholds that stay close to their respective channel capacity thresholds uniformly. Iterative decoding simulation results are provided and compared with turbo codes. In addition to iterative decoding analysis, we analyzed the performance of ARA codes with maximum-likelihood (ML) decoding. By obtaining the weight distribution of these codes and through existing tightest bounds we have shown that the ML SNR threshold of ARA codes also approaches very closely to that of random codes. These codes have better interleaving gain than turbo codes.**

***Index Terms*—Error bounds, graphs, low-density parity-check (LDPC) codes, thresholds, turbo-like codes, weight distribution.**

I. Introduction and Motivation of the Study

LOW-DENSITY parity-check (LDPC) codes were proposed by Gallager [1] in 1962. After the introduction of turbo codes by Berrou *et al.* [2] in 1993, researchers revisited the LDPC codes and extended the work of Gallager. During 1962 to 1993, only a few people, notably Tanner in 1981 [3], paid attention to the work of Gallager and made some contributions. After 1993, over 500 contributions have been made to LDPC codes; for example, see [5], [12], [13], [16], [19], and [20], and references therein.

Paper approved by A. H. Banihashemi, the Editor for Coding and Communication Theory of the IEEE Communications Society. Manuscript received March 29, 2004; revised November 15, 2005 and August 7, 2006. The work of D. Divsalar was supported by a contract from NASA. The work of K. Yao was supported by a contract from the Jet Propulsion Laboratory. This paper was presented in part at the IEEE International Symposium on Information Theory, 2004, and in part at IEEE Globecom, 2004 .

A. Abbasfar is with Rambus Inc., Los Altos CA 94022 USA (e-mail: aliazam@rambus.com).

D. Divsalar is with the Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA 91109 USA (e-mail: Dariush.Divsalar@jpl.nasa.gov).

K. Yao is with the Electrical Engineering Department, University of California, Los Angeles, CA 90095-1594 USA (e-mail: yao@ee.ucla.edu).

Digital Object Identifier 10.1109/TCOMM.2007.894109

Recently, repeat accumulate (RA) and irregular repeat accumulate (IRA) codes, as simple subclasses of LDPC codes with fast encoder structure, were proposed [6], [7]. RA and IRA codes can also be considered as serial concatenated codes [22]. In addition to simplicity, RA codes have fairly good performance. For rates less than or equal to 1/3, their thresholds with iterative decoding are within 1 dB of the capacity. RA codes use fixed repetition for input bits. On the contrary, IRA codes inspired by RA and irregular LDPC [4] codes have irregular repetition for input bits. In fact, node degree distribution can be optimized to achieve low threshold performance. To achieve very low threshold for IRA, as for LDPC codes, maximum repetition for some portion of input bits can be very high.

These recent results for RA and IRA codes motivated us to find very simple codes with enhanced performance. Researchers in [9] tried to improve the input–output extrinsic signal-to-noise ratio (SNR) behavior of the outer convolutional codes in serial concatenation in low extrinsic SNR region to lower SNR threshold of serial concatenation by using repetition of certain bits of the outer code. On the other hand, if a repetition code is used as an outer code, e.g., as in RA codes, one should try to improve the input–output extrinsic SNR behavior at high extrinsic SNR region, since the input–output extrinsic SNR behavior of repetition codes are excellent in the low extrinsic SNR region. We discovered that an accumulator as a rate-1 precoder applied before the repetition code will improve the performance.

Before elaborating on the role of accumulator as a precoder for RA codes and graph representation of ARA codes, we use the definition of protograph introduced by Thorpe in [8]. A similar definition called projected graph was introduced by Richardson *et al.* in [10] for implementation of the decoder for LDPC codes. They show that, if an LDPC code can be represented by the smallest base-graph or projected graph, then high-speed implementation of the decoder will be more feasible. Protograph definition also facilitates the minimal graph representation for the overall graph of an LDPC code. We will show that ARA codes have such a protograph or projected graph representation, which is another advantage. Projected graph definition also has been extended to turbo-like codes in [18].

A protograph [8] is a Tanner graph with a relatively small number of nodes. A protograph $G = (V, C, E)$ consists of a set of variable nodes V, a set of check nodes C, and a set of edges E. Each edge $e \in E$ connects a variable node $v_e \in V$ to a check node $c_e \in C$. Parallel edges are permitted, so the mapping $e \rightarrow (v_e, c_e) \in V \times C$ is not necessarily 1:1. As a simple example, we consider the protograph shown in Fig. 1. This graph consists of $|V| = 4$ variable nodes and $|C| = 3$ check nodes, connected by $|E| = 9$ edges. The four variable nodes in the protograph are denoted by "0,1,2,3" and the three check nodes by "0,1,2." By itself, this graph may be recognized as the Tanner

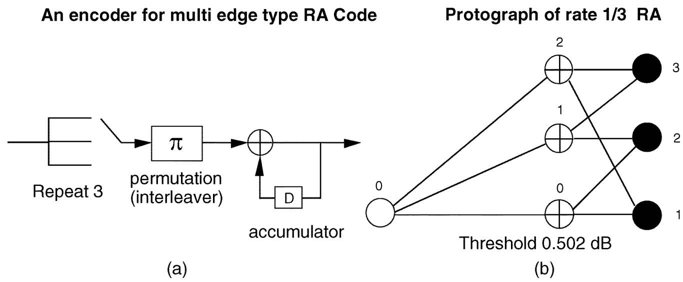


Fig. 1. Rate–1/3 RA code with repetition 3, multi-edge, and protograph versions.

graph of an LDPC code with ($n = 3$, $k = 1$). In Fig. 1, the variable nodes connected to the channel are shown with dark circles i.e., transmitted nodes. Blank circles are those variable nodes not transmitted through the channel, i.e., punctured nodes. Check nodes are circles with plus sign inside. Under certain conditions on the corresponding parity check matrix, i.e., full rank, the code rate for the protograph code can be defined as $R_c = (|V| - |C|)/|V_t|$, where V_t represent a set of transmitted nodes in the protograph. There are $|E| = 9$ types of edges in the protograph representation of RA code. In fact, the protograph LDPC codes are subclass of multi-edge-type LDPC codes introduced in [11]. In multi-edge-type LDPC code, a group of edges may represent a type, where in the protograph LDPC codes each edge is a type.

For a given protograph, we can obtain a larger graph by a copy-and-permute operation. For more details on protographs, see [8]. The resulting larger graph is called the derived graph and the corresponding LDPC code is a protograph code. In general, we can apply the copy-and-permute operation to any protograph to obtain derived graphs of different sizes. This operation consists of first making T copies of the protograph and then permuting the endpoints of each edge among the T variable and T check nodes connected to the set of T edges copied from the same edge in the protograph. Equivalently, the derived graph can be viewed as a multi-edge-type LDPC code in which each edge in the protograph is a distinct type. Thus, the derived graph can be obtained by replacing each edge of the protograph with a permutation of size T. In other words, LDPC codes with a protograph are multi-edge codes with an equal number (T) of edges for each type. In our examples, we will consider both multi-edge-type LDPC codes and protograph LDPC codes. The difference is mainly in the use of permutation on multiple edges or on single edges. Multi-edge-type LDPC codes with rational degree distribution can also have a projected graph description. In Fig. 1(a), the encoder with single permutation (interleaver) may represent a multi-edge-type RA code. As follows we assume tail biting is used for accumulators. In Fig. 1(b), protograph-based RA code is shown. In the figure, the minimum E_b/N_o iterative decoding threshold of protograph RA code is also shown.

II. Punctured RA Codes

Classical rate-1/2 RA code, with repetition 2, has a high threshold of 3.01 dB. Rate-1/2 RA code with lower threshold can be obtained by puncturing the lower rate RA codes that use repetition 3 or higher, provided that the systematic bits are transmitted through the channel, i.e., systematic punctured RA codes. Instead of puncturing, equivalently, we can use a single parity check ($\mathrm{SPC}(n)$) which adds n input bits and outputs one parity, followed by an accumulator without puncturing. Based on the equivalent graph of a punctured accumulator, we obtain the protograph of a systematic punctured RA code. The iterative decoding threshold for this code with protograph representation is 1.116 dB, which is an improvement of close to 2 dB. The systematic punctured multi-edge-type RA code is shown in Fig. 2(a), and the protograph RA is shown in Fig. 2(b). An irregular punctured RA code (which also can be viewed as IRA code) can be constructed by using irregular repetition. Referring to Fig. 2, if 2/3 of the input nodes use repetition 3, and 1/3 use repetition 4, then the corresponding protograph will be the same as in the figure except for additional edge from variable node 1 to check node 1. The iterative decoding threshold for this rate-1/2 irregular punctured protograph RA code is 0.990 dB.

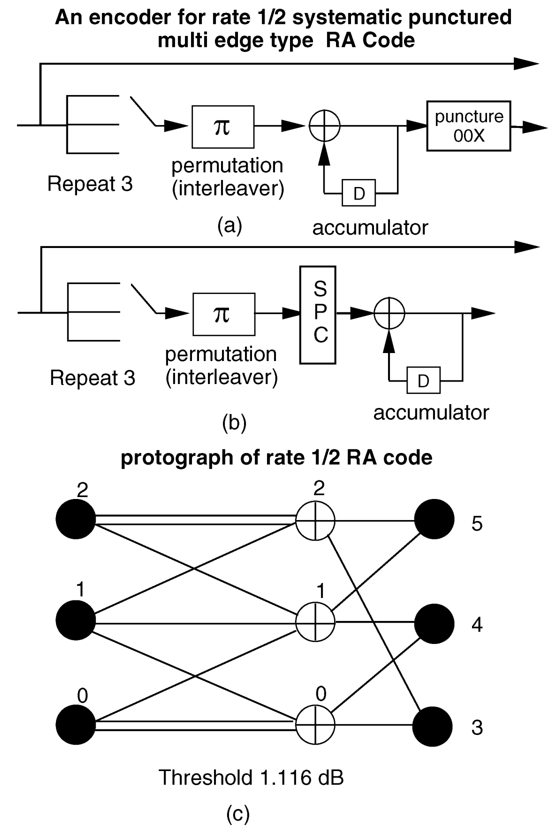


Fig. 2. Systematic RA code. (a) Encoder for multi-edge-type using puncturing. (b) Equivalent encoder using SPC. (c) Rate-1/2 protograph of systematic RA.

III. Accumulate-Repeat-Accumulate Codes

Let us consider a rate-1/3 serial concatenated code where the outer code is a repetition 3 code. Assume that the systematic bits are transmitted to the channel. Alternatively consider the same outer code but the repetition 3 is precoded by an accumulator. Let us compare the extrinsic SNR behavior of these two outer codes using Gaussian density evolution as shown in Fig. 3. As the Gaussian density evolution analysis shows, the use of a

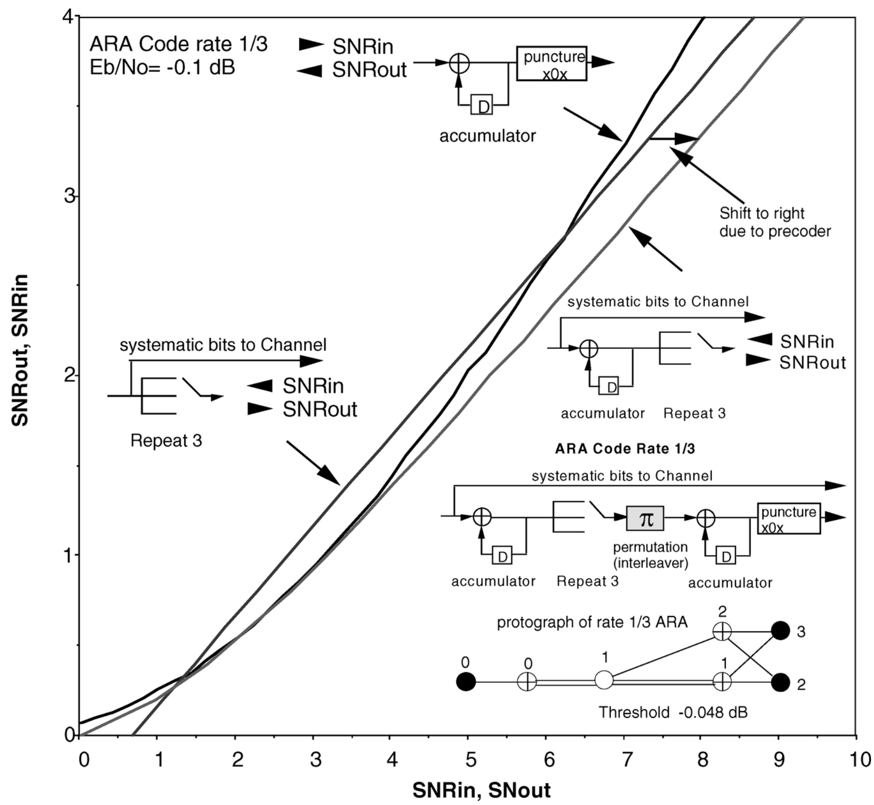


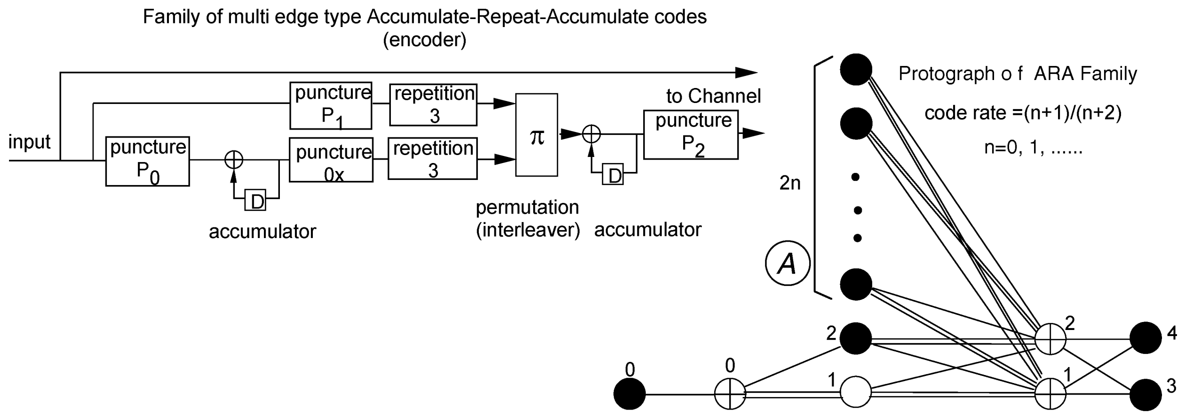
Fig. 3. Gaussian density evolution for rate-1/3 ARA and RA codes showing the improvement due to the precoder.

rate-1 accumulator dramatically improves the extrinsic SNR behavior of repetition 3 at a high extrinsic SNR region. However, it slightly deteriorates the behavior of repetition 3 code at a very low extrinsic SNR region.

Now let us use a punctured accumulator as the inner code. The periodic puncturing pattern in this example is X0X, where 0's indicate the puncturing positions. Since the serial concatenation consists of outer accumulator, middle repetition, and inner accumulator, we call it Accumulate-Repeat- Accumulate (ARA) code. The rate-1/3 ARA code, the corresponding protograph, and the extrinsic input–output SNR curves using Gaussian density evolution are shown in Fig. 3. Except for this example in which we used Gaussian density evolution to show the advantage of precoding, in this paper, actual density evolution on protographs will be used. Using density evolution, we obtain the exact iterative decoding threshold of -0.048 dB for the protograph shown in Fig. 3. If we remove the precoder and transmit node 1 in Fig. 3, the threshold will be 0.73 dB. Thus, the precoder improves the iterative decoding threshold by 0.7 dB. We call such performance improvement due to the use of a precoder as " **Precoding gain**."

These comparisons are fair if we fix the maximum variable node degree. Shortly, we will show such comparisons with rate-1/2 unstructured irregular LDPC codes. In a similar way, we can construct rate-1/2 ARA codes. However, due to more puncturing of the inner accumulator, some portion of the input bits should not be passed through the precoder in order to allow the iterative decoding to start, i.e., " **Doping** " [29] is required for the iterative decoder to start. An example of a simple rate-1/2 ARA code, its protograph, and the corresponding threshold are shown in Fig. 4 when $n = 0$. We also propose a constructive method to design codes with higher rates from a rate-1/2 ARA code and its protograph by further puncturing of the rate-1/2 ARA code. $2n$ additional variable nodes each with degree 3 are added to the protograph. This is like adding $2n$ repeat 3 RA codes to a single rate-1/2 ARA code. In this case, the addition is done at the inner check nodes of rate-1/2 ARA, and one common inner accumulator is used. Similar high rate codes can be obtained using the protograph of the other rate-1/2 ARA codes that will be discussed in this paper. One example of such a family of ARA codes for various code rates is shown in Fig. 4. For single decoder implementation that can handle various code rates, one should consider the protograph for the highest desired code rate. For encoding of lower rates, one should insert zeros for those input variable node bits that corresponds to the higher code rate generation (see variable nodes marked with letter A in Fig. 4). These bits should not be sent to the channel, i.e., to be punctured. Thus, this method can be considered a solution to the problem of decoding various code rates using a single protograph.

Another simple example of a rate-1/2 ARA code with regular repetition is the precoded version of the punctured RA code in Fig. 2. The precoded version is shown in Fig. 5. This code has a threshold of 0.400 dB, compared with 1.116 dB for the systematic punctured RA code in Fig. 2. Thus, there is a 0.7-dB precoding gain due to the use of a rate-1 accumulator precoder. Fig. 5 shows an alternative implementation of an encoder using a differentiator that has the same protograph. In this case, we call the coding scheme Differentiate-Repeat-Accumulate (DRA) code, which is a nonsystematic code. We can also use



Code Rate	Protograph Threshold	Capacity	Difference	Puncturing Patterns		
				P_0	P_1	P_2
1/2	0.516	0.187	0.329	XX	X0	00X
2/3	1.288	1.059	0.229	00XX	XXX0	00000X
3/4	1.848	1.626	0.222	0000XX	XXXXX0	00000000X
4/5	2.277	2.040	0.237	000000XX	XXXXXXX0	00000000000X
5/6	2.620	2.362	0.258	00000000XX	XXXXXXXXX0	00000000000000X
6/7	2.897	2.625	0.272	0000000000XX	XXXXXXXXXXX0	00000000000000000X
7/8	3.129	2.845	0.284	000000000000XX	XXXXXXXXXXXXX0	00000000000000000000X

Fig. 4. ARA code family for rates 1/2 to 7/8, encoder for multi-edge-type , protographs, and protograph thresholds in decibels.

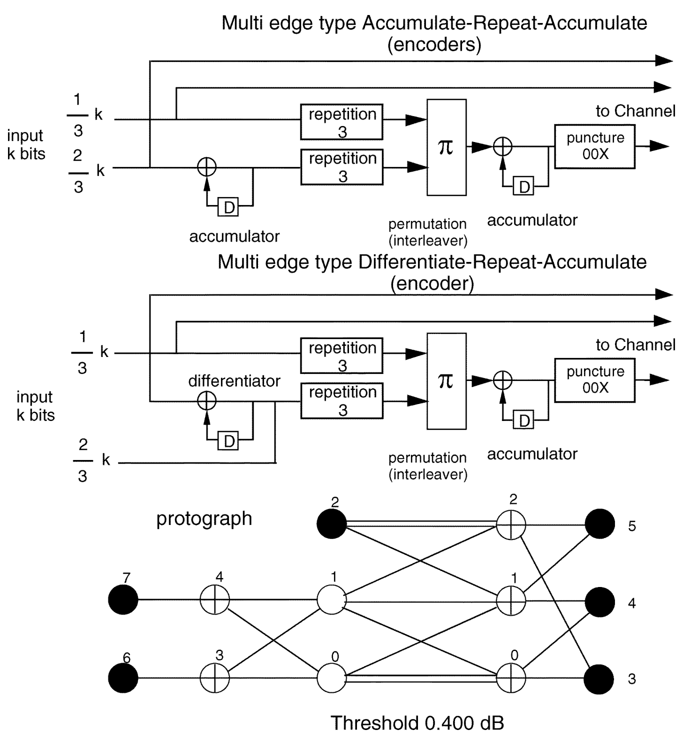


Fig. 5. Rate-1/2 ARA, DRA, and the corresponding protograph representation.

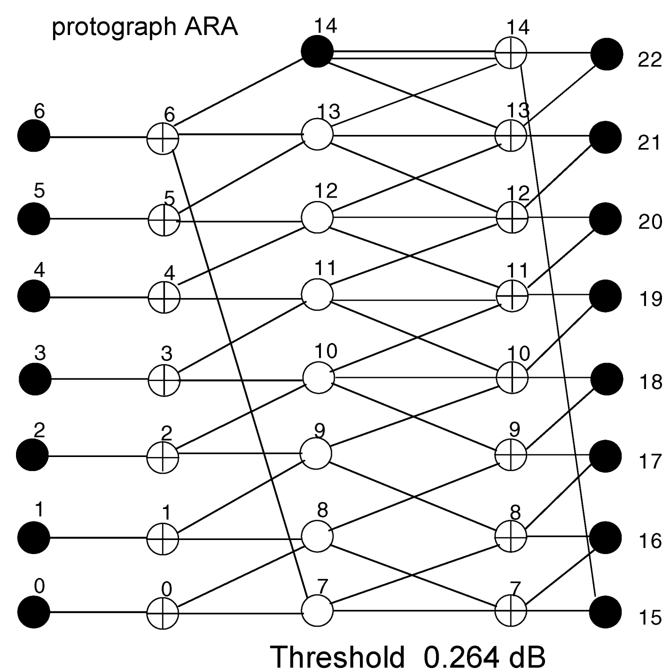


Fig. 6. Rate-1/2 low-threshold ARA protograph.

irregular repetition instead of regular repetition in ARA codes, which results in irregular ARA (IARA) codes.

A simple example of a rate-1/2 IARA code is the precoded version of the IRA example discussed previously. The threshold for that IRA example was 0.990 dB. The precoded version has a threshold of 0.364 dB. A family of higher rate ARA codes can be obtained simply by puncturing the ARA code or IARA code in the previous two examples. A low-threshold (0.264 dB) rate-1/2 ARA code is shown in Fig. 6. The protograph has a maximum degree of 5. The best rate-1/2 unstructured irregular LDPC code with a maximum degree of 5 in [4] has a threshold of 0.72 dB. There are few reasons for such difference. In [4], the

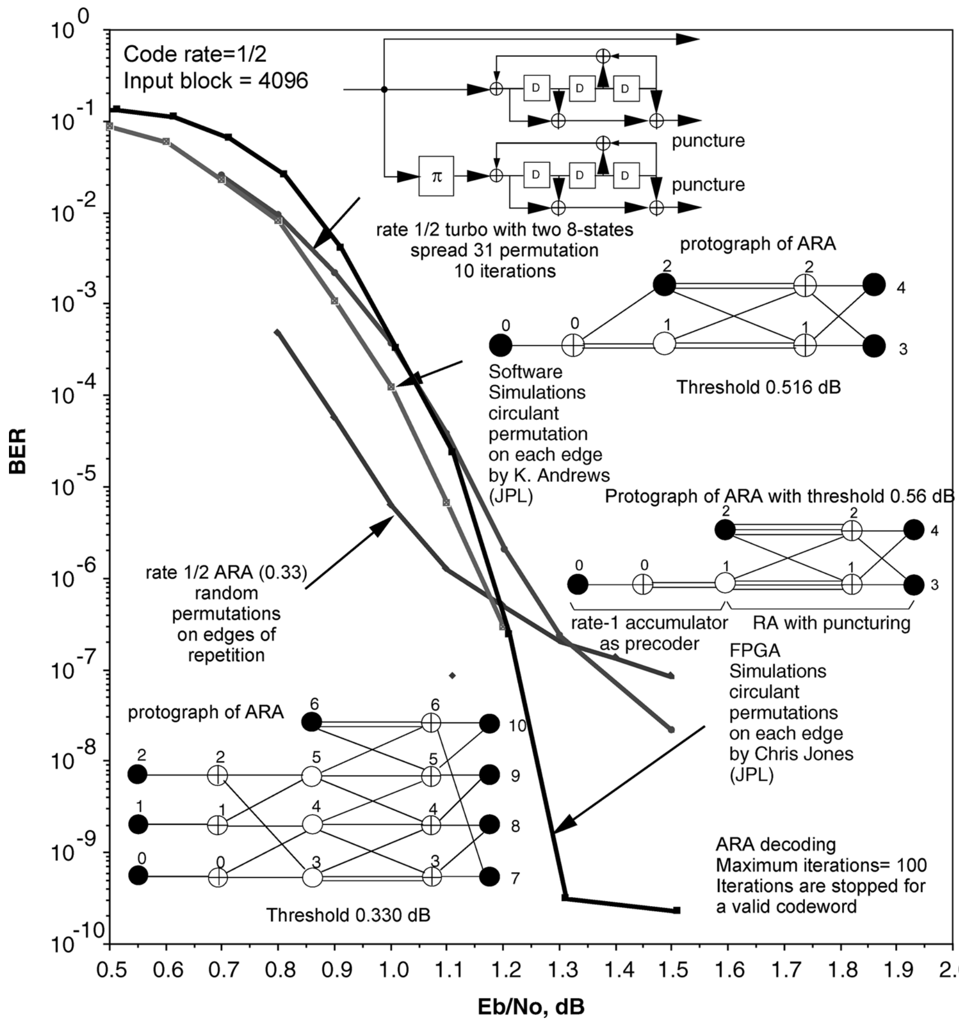


Fig. 7. Simulation results for few examples of rate-1/2 ARAs, and the rate-1/2 wireless standard turbo code.

degree of variable nodes are greater or equal to 2, and punctured variable nodes were not allowed. If we look at the protographs of ARA codes, they contain degree-1 variable nodes and punctured variable nodes. In fact, later, Richardson and Urbanke [11], [13] mentioned that the degree-1 variable nodes and punctured variable nodes also can be used in the multi-edge-type LDPC code design. As we mentioned earlier, protograph-based ARA codes use permutations per each edge of protograph. When we have accumulators, a protograph-based ARA can be constructed if the structure of the interleaver in ARA codes is based on edge connections in the ARA protograph between middle variable nodes and inner check nodes (right most check nodes), i.e., between repetition and accumulator. However a multi-edge-type ARA code can be defined when permutations are used per group of edges, e.g., a single random permutation is used for repetition.

Consider the rate-1/2 ARA code with threshold 0.516 dB that was shown in Fig. 4, for $n = 0$ (also shown in Fig. 7). For this example, we can use six interleavers each of size $N/2$, where N is the input block size. Each interleaver corresponds to one of the six edges between the repetition 3 and the punctured accumulator; i.e., the edges between variable nodes "1,2" and check nodes "1,2" in the ARA protograph in Fig. 7. Simulation results for random type interleavers and circulant permutations per each edge of protograph are shown in Fig. 7 for three examples of rate-1/2 ARA codes with a protograph structure and compared with a rate-1/2 turbo code with well-optimized spread interleaver. Use of circulant permutations instead of random permutations is preferred for implementation of an encoder and decoder. For decoding ARA codes, the message passing algorithm was used. For iterative decoding of turbo code the standard BCJR algorithm is used. For performance simulation results of protograph ARA codes for various code rates, see [23] and [24]. For encoders for protograph-based ARA codes with circulant permutations, see [27].

IV. Maximum-Likelihood (ML) Decoding Analysis of ARA Codes

Here, we intend to prove that the ensemble weight distribution of RA codes can be improved by using precoding. Also, if ML decoding is used, there will be roughly the same amount of precoding gain as in the iterative decoding. First, we analyze the ML decoding performance of multi-edge-type RA codes with single random permutation and with regular puncturing. We show that, with puncturing, we can construct better codes as far as the performance is concerned. Then, we extend the results to multi-edge-type ARA codes with single random permutation.

Since there is no practical ML decoding algorithm available for block codes with large block size, we use the performance bounds to obtain some insight on codes' behavior. In [14], Divsalar provides a tight upper bound on frame (word) error rate (FER) and bit error rate (BER) for a (n,k) linear block code with code rate $R_c = k/n$ and distance spectrum A_d (number of codewords with weight d), decoded by ML criterion over an additive white Gaussian noise (AWGN) channel. It also provides a minimum E_b/N_o threshold with closed-form expression. We use this bound throughout the paper. Defining normalized distance as $\delta = d/n$ and normalized distance spectrum as $r(\delta) = \ln A_d/n$, the FER bound in [14] or [15] can be expressed as

$$P_e \le \sum_{d=d_{\min}}^{d_{\max}} e^{-nE(\delta,\beta,E_c/N_o)} \quad (1)$$

where

$$E\left(\delta,\beta,\frac{E_c}{N_o}\right) = -\frac{1}{2}\ln(1-\beta+\beta e^{-2r(\delta)}) + \frac{\beta\delta}{1-(1-\beta)\delta}\frac{E_c}{N_o} \quad (2)$$

and

$$\beta = \frac{1-\delta}{\delta} \times \left[\sqrt{\frac{E_c}{N_o}\frac{\delta}{1-\delta}\frac{2}{1-e^{-2r(\delta)}} + \left(1+\frac{E_c}{N_o}\right)^2 - 1} - \left(1+\frac{E_c}{N_o}\right)\right] \quad (3)$$

where $E_c/N_o = R_c E_b/N_o$, and $0 < \beta < 1$. When $\beta = 1$, the bound reduces to union bound. To compute the BER bound, replace A_d with $\sum_{w=1}^{R_c n}(w/R_c n)A_{w,d}$ in the FER bound, where $A_{w,d}$ is the number of codewords with input weight w and output weight d. An important result of this bound is the tightest minimum E_b/N_o that can be written in closed form as

$$\left(\frac{E_b}{N_o}\right)_{\min} \le \frac{1}{R_c}\max_{\delta}(1-e^{-2r(\delta)})\frac{1-\delta}{2\delta}. \quad (4)$$

The threshold will be the minimum E_b/N_o when $n \to \infty$, thus we need to obtain the asymptotic distance spectrum, i.e., $r(\delta)$ as $n \to \infty$.

A. *Weight Distribution of Ensemble of RA Codes With Puncturing*

RA codes are the simplest codes among turbo-like codes, which make them very attractive for analysis. In an RA code, an information block of length N is repeated q times and interleaved to make a block of size qN, and then followed by a rate-1 accumulator (see Fig. 1 for $q = 3$). We use the concept of uniform interleaver [21] to compute the overall input–output weight enumerator (IOWE). The final derived IOWE should be considered as the averaged IOWE over all interleavers between repetition code and the inner accumulator with puncturing. Therefore, we need to compute the IOWE of both repetition code and the accumulator. For repetition code, it is [6]

$$A_{w,d}^{\mathrm{rep}(q)} = \binom{N}{w}\delta_{d,qw} \quad (5)$$

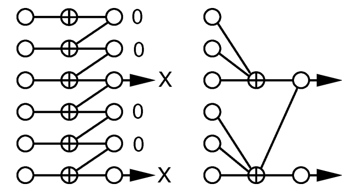


Fig. 8. Accumulator with period $\mathrm{p} = 3$ puncturing 00X and its equivalent graph.

where $\delta_{i,j}$ is Kronecker delta function. The IOWE of the accumulator is

$$A_{w,d}^{\mathrm{acc}} = \binom{N-d}{\lfloor \frac{w}{2} \rfloor}\binom{d-1}{\lceil \frac{w}{2} \rceil - 1}. \quad (6)$$

RA codes are usually nonsystematic codes, i.e., the information block is not sent along with the output of the accumulator. However, the RA codes with puncturing should be systematic in order to be decodable by iterative decoding. To compute the IOWE of the RA codes with puncturing, we use the equivalent graph depicted in Fig. 8 instead of the accumulator with puncturing. Puncturing uses a periodic pattern $0\ldots0X$ with period p, where zeros indicate the puncturing positions. For computation of IOWE, the equivalent code is a concatenated code of a regular check code (a rate p code with p inputs and one parity check as output), and an accumulator. Since the check code is regular and memoryless, the presence of any interleaver between two codes does not change the IOWE of the overall code. This is a key observation in obtaining the IOWE. In order to compute the IOWE for this code, we insert a uniform interleaver between two codes. The next step is to compute the IOWE of the check code. The IOWE of the check code can be expressed by a simple closed-form formula if we use the two-dimensional (2-D) Z-transform of IOWE denoted by $A^c(W,D)$. The inverse Z-transform results in $A_{w,d}^c$. Starting with $N = 1$, i.e., one check code with p inputs and one output, $A^c(W,D)$ the of check code can be expressed as

$$A^c(W,D) = E_p(W) + O_p(W)D \quad (7)$$

where $E_p(W) = \mathrm{Even}[(1+W)^p] = (((1+W)^p + (1-W)^p)/2)$, and $O_p(W) = \mathrm{Odd}[(1+W)^p] = (((1+W)^p - (1-W)^p)/2)$. Since there are N independent check nodes in the code the IOWE can be written in Z-transform as

$$A^c(W,D) = [E_p(W) + O_p(W)D]^N = \sum_{d=0}^{N}\binom{N}{d}[E_p(W)]^{N-d}[O_p(W)]^d D^d. \quad (8)$$

The IOWE is obtained by taking the inverse Z-transform. The closed-form expression for $A_{w,d}$ for arbitrary p can be derived, but the expression is complicated. Instead, in this paper, we derive the IOWE for $p = 2$, 3, and 4, which are practically most useful for the codes that we already discussed in previous sections.

1) Case $p = 2$: Using the general formula in Z-transform, we have

$$A^{c(2)}(W,D) = (1 + W^2 + 2WD)^N. \tag{9}$$

This can be expanded as

$$A^{c(2)}(W,D)=\sum_{d=0}^{N}\sum_{w=0}^{2N}\binom{N}{d}\sum_{j=0}^{N-d}\binom{N-d}{j}2^d\delta_{w,2j+d}W^wD^d. \tag{10}$$

Therefore, the IOWE can be expressed as

$$A_{w,d}^{c(2)} = \binom{N}{d}\sum_{j=0}^{N-d}\binom{N-d}{j}2^d\delta_{w,2j+d}. \tag{11}$$

2) Case $p = 3$: Starting from the general formula in the Z-transform, we have

$$A^{c(3)}(W,D) = [1 + 3W^2 + (3W + W^3)D]^N. \tag{12}$$

This can be expanded as

$$A^{c(3)}(W,D) = \sum_{d=0}^{N}\binom{N}{d}\left[\left(\sum_{k=0}^{N-d}\binom{N-d}{k}3^kW^{2k}\right)\times\left(\sum_{i=0}^{d}\binom{d}{i}3^{(d-i)}W^{2i}\right)W^d\right]D^d. \tag{13}$$

Letting $i + k = j$, we then have

$$A^{c(3)}(W,D) = \sum_{d=0}^{N}\binom{N}{d}\times\left(\sum_{j=0}^{N}\sum_{i=\max(0,j-N+d)}^{\min(j,d)}\binom{d}{i}\times\binom{N-d}{j-i}3^{(d+j-2i)}W^{(d+2j)}\right)D^d. \tag{14}$$

Therefore, the IOWE is

$$A_{w,d}^{c(3)} = \binom{N}{d}\sum_{j=0}^{N}\sum_{i=\max(0,j-N+d)}^{\min(j,d)}\binom{d}{i}\binom{N-d}{j-i}3^{d+j-2i}\delta_{w,2j+d}. \tag{15}$$

3) Case $p = 4$: The check code for this case can be viewed as a concatenation of two check codes each with $p = 2$. Because the check code is regular and memoryless, we can put any interleaver between the check codes without changing the IOWE of the overall check code with $p = 4$. By using a uniform interleaver and the results found for case $p = 2$, the IOWE can be written as

$$A_{w,d}^{c(4)} = \sum_{h=0}^{2N}\frac{A_{w,h}^{c(2)}A_{h,d}^{c(2)}}{\binom{2N}{h}}. \tag{16}$$

Using the result for $A_{w,d}^{c(2)}$, we obtain

$$A_{w,d}^{c(4)} = \binom{N}{d}\sum_{j=0}^{N-d}\sum_{i=0}^{2N-d-2j}\times\binom{2N-d-2j}{i}\binom{N-d}{j}2^{2d+2j}\delta_{w,d+2i+2j}. \tag{17}$$

This method can be applied for any p that can be decomposed into two smaller numbers. Having computed the IOWE of the check code, we can use the uniform interleaver formula to compute the IOWE of the accumulator with puncturing as

$$A_{w,d}^{\mathrm{acc}(p)} = \sum_{h=0}^{N}\frac{A_{w,h}^{c(p)}A_{h,d}^{\mathrm{acc}}}{\binom{N}{h}}. \tag{18}$$

Thus, for $p = 3$, and $p = 4$, we obtain

$$A_{w,d}^{acc(3)} = \sum_{h=0}^{N}\sum_{j=0}^{N}\sum_{i=\max(0,j-N+h)}^{\min(j,h)}\times\binom{h}{i}\binom{N-h}{j-i}\binom{N-d}{\lfloor\frac{h}{2}\rfloor}\times\binom{d-1}{\lceil\frac{h}{2}\rceil-1}3^{h+j-2i}\delta_{w,2j+h} \tag{19}$$

$$A_{w,d}^{\mathrm{acc}(4)} = \sum_{h=0}^{N}\sum_{j=0}^{N-h}\sum_{i=0}^{2N-h-2j}\times\binom{2N-h-2j}{i}\binom{N-h}{j}\binom{N-d}{\lfloor\frac{h}{2}\rfloor}\times\binom{d-1}{\lceil\frac{h}{2}\rceil-1}2^{2h+2j}\delta_{w,h+2i+2j} \tag{20}$$

It should be noted that, despite the fact that we used a uniform interleaver to obtain the IOWE, we come up with the exact (not averaged) IOWE for accumulator with puncturing. The next step is to find the IOWE of the systematic RA code with puncturing, which is derived in case of a uniform interleaver after repetition as

$$A_{w,d}^{\mathrm{rep(q)-acc(p)}} = \sum_{l=0}^{qN}\frac{A_{w,l}^{\mathrm{rep(q)}}A_{l,d-w}^{\mathrm{acc(p)}}}{\binom{qN}{l}}. \tag{21}$$

For systematic punctured RA (q = 3, p = 3), which will be denoted by RA(3,3), we get

$$A_{w,d}^{\mathrm{rep}(3)-\mathrm{acc}(3)} = \frac{\binom{N}{w}}{\binom{3N}{3w}}\sum_{h=0}^{N}\sum_{j=0}^{N}\sum_{i=\max(0,j-N+h)}^{\min(j,h)}\times\binom{h}{i}\binom{N-h}{j-i}\times\binom{N-d+w}{\lfloor\frac{h}{2}\rfloor}\times\binom{d-w-1}{\lceil\frac{h}{2}\rceil-1}3^{h+j-2i}\delta_{3w,2j+h} \tag{22}$$

Now, we obtain the asymptotic expression of $r(\delta)$ for RA(3,3) as the block size goes to infinity. To do so, we sum (22) over w to get $A_d^{\mathrm{rep}(3)-\mathrm{acc}(3)}$, and use $\lim_{n\to\infty}(1/n)\ln\binom{n}{k} = H(k/n)$, where $H(\cdot)$ is the binary (natural) entropy function, $H(x) =$

$-x\ln x-(1-x)\ln(1-x)$. Let $\delta \triangleq d/2N$ for $0<\delta<1$, $\eta \triangleq h/2N$ for $0<\eta<1/2$, $\rho_1 \triangleq i/2N$ for $\max(0,\rho_2+\eta-1/2)<\rho_1<\min(\rho_2,\eta)$, and $\rho_2 \triangleq j/2N$ for $0<\rho_2<1/2$, with $(1/3)(2\rho_2+\eta)<\min(0.5,\delta)$, to yield

$$\begin{aligned} r(\delta) = \max_{\eta,\rho_1,\rho_2} \Bigg\{ & -H\left(\frac{4\rho_2+2\eta}{3}\right)+\eta H\left(\frac{\rho_1}{\eta}\right) \\ & +\left(\frac{1}{2}-\eta\right) H\left(\frac{\rho_2-\rho_1}{\frac{1}{2}-\eta}\right) \\ & +\left(\frac{1}{2}-\delta+\frac{2\rho_2+\eta}{3}\right) H\left(\frac{\frac{\eta}{2}}{\frac{1}{2}-\delta+\frac{2\rho_2+\eta}{3}}\right) \\ & +\left(\delta-\frac{2\rho_2+\eta}{3}\right) H\left(\frac{\frac{\eta}{2}}{\delta-\frac{2\rho_2+\eta}{3}}\right) \\ & +(\eta+\rho_2-2\rho_1)\ln 3 \Bigg\}. \end{aligned} \tag{23}$$

Using (23) in (4), we can obtain the minimum E_b/N_o threshold for RA(3,3).

For systematic punctured RA with (q = 4, p = 4), which will be denoted by RA(4,4), we have

$$\begin{aligned} A_{w,d}^{\text{rep}(4)-\text{acc}(4)} = & \frac{\binom{N}{w}}{\binom{4N}{4w}} \sum_{h=0}^{N} \sum_{j=0}^{N-h} \sum_{i=0}^{2N-h-2j} \binom{2N-h-2j}{i} \\ & \times \binom{N-h}{j} \binom{N-d+w}{\lfloor \frac{h}{2} \rfloor} \\ & \times \binom{d-w-1}{\lceil \frac{h}{2} \rceil -1} 2^{2h+2j} \delta_{4w,h+2i+2j}. \end{aligned} \tag{24}$$

Now, we obtain the asymptotic expression of $r(\delta)$ for RA(4,4), after summing (24) over w. Let $\delta \triangleq d/2N$ for $0<\delta<1$, $\eta \triangleq h/2N$ for $0<\eta<1/2$, $\rho_1 \triangleq i/2N$ for $0<\rho_1<1-\eta-2\rho_2$, and $\rho_2 \triangleq j/2N$ for $0<\rho_2<1/2-\eta$, with $(1/4)(\eta+2\rho_1+2\rho_2)<\min(0.5,\delta)$, to yield

$$\begin{aligned} r(\delta) = \max_{\eta,\rho_1,\rho_2} \Bigg\{ & -\frac{3}{2} H\left(\frac{\eta+2\rho_1+2\rho_2}{2}\right)+(1-\eta-2\rho_2) \\ & \times H\left(\frac{\rho_1}{1-\eta-2\rho_2}\right)+\left(\frac{1}{2}-\eta\right) H\left(\frac{\rho_2}{\frac{1}{2}-\eta}\right) \\ & +\left(\frac{1}{2}-\delta+\frac{\eta+2\rho_1+2\rho_2}{4}\right) \\ & \times H\left(\frac{\frac{\eta}{2}}{\frac{1}{2}-\delta+\frac{\eta+2\rho_1+2\rho_2}{4}}\right)+\left(\delta-\frac{\eta+2\rho_1+2\rho_2}{4}\right) \\ & \times H\left(\frac{\frac{\eta}{2}}{\delta-\frac{\eta+2\rho_1+2\rho_2}{4}}\right)+(2\eta+2\rho_2)\ln 2 \Bigg\}. \end{aligned} \tag{25}$$

Using (25) in (4), we can obtain the minimum E_b/N_o threshold for RA(4,4).

B. Weight Distribution of Ensemble of ARA Codes

Here, we obtain the ML performance of ARA codes, as a precoded RA code with puncturing, using an accumulator as a precoder. As we have seen, in ARA codes, that a portion of the information block goes to the accumulator. In other words, M bits are passed through without any change and the remaining N-M bits go through an accumulator. The overall output bits are applied to the punctured RA code. The use of these M bits is essential for the iterative decoding to start the message-passing algorithm. M is considered a parameter in code design. The effect of this parameter is studied in ML decoding. Fig. 9 shows the ARA code and the block diagram of the precoder.

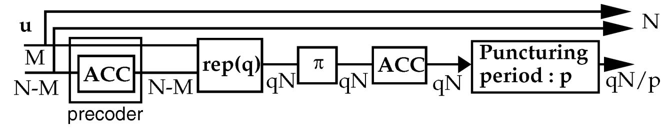


Fig. 9. Block diagram of the ARA code.

In order to find the performance of the code, we need to compute the IOWE of the precoder. It is computed using the IOWE of the accumulator code as follows:

$$A_{w,d}^{\text{pre}} = \sum_{m=0}^{M} \binom{M}{m} A_{w-m,d-m}^{\text{acc}}. \tag{26}$$

Therefore, the IOWE of the overall systematic ARA (q, p) code can be written as

$$A_{w,d}^{\text{pre}-\text{rep(q)}-\text{acc(p)}} = \sum_{l=0}^{N} \frac{A_{w,l}^{\text{pre}} A_{l,d+l-w}^{\text{rep(q)}-\text{acc(p)}}}{\binom{N}{l}}. \tag{27}$$

For systematic punctured ARA with (q = 3, p = 3), which will be denoted by ARA(3,3), we obtain

$$\begin{aligned} A_{w,d}^{\text{pre}-\text{rep}(3)-\text{acc}(3)} = & \sum_{m=0}^{M} \sum_{l=0}^{N} \sum_{h=0}^{N} \sum_{j=0}^{N} \sum_{i=\max(0,j-N+h)}^{\min(j,h)} \\ & \times \frac{\binom{M}{m}}{\binom{3N}{3l}} \binom{h}{i} \binom{N-h}{j-i} \binom{N-d+w}{\lfloor \frac{h}{2} \rfloor} \\ & \times \binom{d-w-1}{\lceil \frac{h}{2} \rceil -1} \binom{N-M-l+m}{\lfloor \frac{(w-m)}{2} \rfloor} \\ & \times \binom{l-m-1}{\lceil \frac{(w-m)}{2} \rceil -1} \times 3^{h+j-2i} \delta_{3l,2j+h}. \end{aligned} \tag{28}$$

Now, we obtain the asymptotic expression of $r(\delta)$ for ARA(3,3) using (28). Let $\alpha \triangleq M/2N$ for $0<\alpha<1/2$; $\epsilon_1 \triangleq M/2N$ for $0<\epsilon_1<\alpha$, $\epsilon_2 \triangleq (w-m)/2N$ for $0<\epsilon_2<1/2-\alpha$, $\delta \triangleq d/2N$ for $0<\delta<1$, $\eta \triangleq h/2N$ for $0<\eta<1/2$, $\rho_1 \triangleq i/2N$ for $\max(0,\rho_2+\eta-1/2)<\rho_1<\min(\rho_2,\eta)$, and $\rho_2 \triangleq j/2N$ for $0<\rho_2<1/2$ to yield

$$\begin{aligned} r(\delta) = \max_{\epsilon_1,\epsilon_2,\eta,\rho_1,\rho_2} \Bigg\{ & \alpha H\left(\frac{\epsilon_1}{\alpha}\right)-\frac{3}{2} H\left(\frac{4\rho_2+2\eta}{3}\right) \\ & +\eta H\left(\frac{\rho_1}{\eta}\right)+\left(\frac{1}{2}-\eta\right) H\left(\frac{\rho_2-\rho_1}{\frac{1}{2}-\eta}\right) \\ & +\left(\frac{1}{2}-\delta+\epsilon_1+\epsilon_2\right) H\left(\frac{\frac{\eta}{2}}{\frac{1}{2}-\delta+\epsilon_1+\epsilon_2}\right) \\ & +(\delta-\epsilon_1-\epsilon_2) H\left(\frac{\frac{\eta}{2}}{\delta-\epsilon_1-\epsilon_2}\right) \end{aligned}$$

$$+\left(\frac{1}{2}-\alpha-\frac{1}{3}(2\rho_2+\eta)+\epsilon_1\right) \times H\left(\frac{\frac{\epsilon_2}{2}}{\frac{1}{2}-\alpha-\frac{1}{3}(2\rho_2+\eta)+\epsilon_1}\right) +\left(\frac{1}{3}(2\rho_2+\eta)-\epsilon_1\right)H\left(\frac{\frac{\epsilon_2}{2}}{\frac{1}{3}(2\rho_2+\eta)-\epsilon_1}\right) +(\eta+\rho_2-2\rho_1)\ln 3\Bigg\}. \tag{29}$$

Using (29) in (4), we can obtain the minimum E_b/N_o threshold for ARA(3,3).

For systematic punctured ARA with (q = 4, p = 4), which will be denoted by ARA(4,4), we obtain

$$A_{w,d}^{\text{pre-rep}(4)-\text{acc}(4)} = \sum_{m=0}^{M}\sum_{l=0}^{N}\sum_{h=0}^{N}\sum_{j=0}^{N-h}\sum_{i=0}^{2N-h-2j} \times \frac{\binom{M}{m}}{\binom{4N}{4l}}\binom{2N-h-2j}{i}\binom{N-h}{j} \times \binom{N-d+w}{\lfloor\frac{h}{2}\rfloor} \times \binom{d-w-1}{\lceil\frac{h}{2}\rceil-1}\binom{N-M-l+m}{\lfloor\frac{(w-m)}{2}\rfloor} \times \binom{l-m-1}{\lceil\frac{(w-m)}{2}\rceil-1}2^{2h+2j}\delta_{4l,h+2i+2j}. \tag{30}$$

At this point, we obtain the asymptotic expression of $r(\delta)$ for ARA(4,4), using (30). Let $\alpha \triangleq M/2N$ for $0<\alpha<1/2$; $\epsilon_1 \triangleq M/2N$ for $0<\epsilon_1<\alpha$, $\epsilon_2 \triangleq (w-m)/2N$ for $0<\epsilon_2<1/2-\alpha$, $\delta \triangleq d/2N$ for $0<\delta<1$, $\eta \triangleq h/2N$ for $0<\eta<1/2$, $\rho_1 \triangleq i/2N$ for $0<\rho_1<1-\eta-2\rho_2$, and $\rho_2 \triangleq j/2N$ for $0<\rho_2<1/2-\eta$, to yield

$$r(\delta) = \max_{\epsilon_1,\epsilon_2,\eta,\rho_1,\rho_2}\Bigg\{\alpha H\left(\frac{\epsilon_1}{\alpha}\right)-2H\left(\frac{\eta}{2}+\rho_1+\rho_2\right) +(1-\eta-2\rho_2)H\left(\frac{\rho_1}{1-\eta-2\rho_2}\right) +\left(\frac{1}{2}-\eta\right)H\left(\frac{\rho_2}{\frac{1}{2}-\eta}\right) +\left(\frac{1}{2}-\delta+\epsilon_1+\epsilon_2\right)H\left(\frac{\frac{\eta}{2}}{\frac{1}{2}-\delta+\epsilon_1+\epsilon_2}\right) +(\delta-\epsilon_1-\epsilon_2)H\left(\frac{\frac{\eta}{2}}{\delta-\epsilon_1-\epsilon_2}\right) +\left(\frac{1}{2}-\alpha-\frac{1}{4}(\eta+2\rho_1+2\rho_2)+\epsilon_1\right) \times H\left(\frac{\frac{\epsilon_2}{2}}{\frac{1}{2}-\alpha-\frac{1}{4}(\eta+2\rho_1+2\rho_2)+\epsilon_1}\right) +\left(\frac{1}{4}(\eta+2\rho_1+2\rho_2)-\epsilon_1\right) \times H\left(\frac{\frac{\epsilon_2}{2}}{\frac{1}{4}(\eta+2\rho_1+2\rho_2)-\epsilon_1}\right) +(2\eta+2\rho_2)\ln 2\Bigg\}. \tag{31}$$

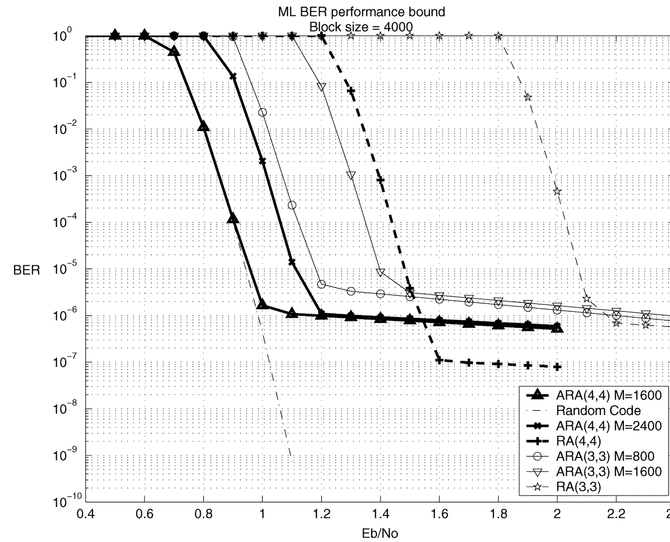


Fig. 10. BER bounds for RA(3,3), RA(4,4), ARA(3,3), ARA(4,4), and random codes for input blocksize of 4000 b.

Using (31) in (4), we can obtain the minimum E_b/N_o threshold for ARA(4,4).

To compare the asymptotic thresholds of RA codes with puncturing and ARA codes with classical RA codes and random codes, we need to know the asymptotic weight distribution of these codes. The asymptotic expression of $r(\delta)$ for RA code with repetition q is [6]

$$r(\delta)=\max_{0<\epsilon<1/q}\left[\frac{1-q}{q}H(q\epsilon)+(1-\delta)H\left(\frac{q\epsilon}{2(1-\delta)}\right)+\delta H\left(\frac{q\epsilon}{2\delta}\right)\right] \tag{32}$$

and the asymptotic expression of $r(\delta)$ for random codes with code rate R_c is (see, for example, [15])

$$r(\delta) = H(\delta) + (R_c - 1)\ln 2. \tag{33}$$

C. ML Performance and ML Thresholds of Ensemble of RA Codes With Puncturing and ARA Codes

Using the derived weight distributions in Section IV-A, the BER performance for rate-1/2 RA(3,3), and RA(4,4) using the bound in (1) are shown in Fig. 10 for input block size of 4000 bi. Similarly, using the derived weight distributions in Section IV-B, the BER performance bound for rate-1/2 ARA(3,3) and ARA(4,4) for different M's are compared with that of random code for the same input block size (4000) in Fig. 10. In the ARA case, it is observed that the larger the number of input bits that pass through the accumulator in the precoder, the better the performance becomes. However, the improvement diminishes past a certain point, which is $M = 1/5\ N$ for ARA(3,3) and $M = 2/5\ N$ for ARA(4,4). When $M = N$, the codes turn into RA with puncturing. It is very interesting that the performance of the ARA(4,4) very closely approaches that of random codes for the same block size in low E_b/N_o region.

The E_b/N_o thresholds of these rate-1/2 codes for infinite block lengths using the derived asymptotic $r(\delta)$ from Sections IV-A, and IV-B and applying to (4) are computed as: 1.497 dB for RA(3,3), 0.871 dB for RA(4,4), 0.51 dB for ARA(3,3), 0.31 dB for ARA(4,4), 0.308 dB for random codes, and 3.364 dB for RA code with $q = 2$. Discrepancy between random code threshold and the Shannon limit is due to the upper bound which is slightly loose for rate-1/2. In the ARA case, we minimized the threshold expression in (4) with respect to α. It is easy to show that the second derivative of the threshold with respect to α is always negative. Thus, for ML decoding, the minimum threshold is achieved when $\alpha \to 0$. However, for all values of $0 < \alpha < 0.1$ in case of ARA(3,3), and $0 < \alpha < 0.2$ in case of ARA(4,4), very small change in threshold was observed. As we expected, based on the BER performance bound for input block of 4000, the E_b/N_o threshold of ARA(4,4) for infinite block is also extremely close to the threshold of random codes, based on the same bound.

To obtain the interleaving gain as defined in [21] and [22], we can use the method in [22] to find the maximum exponent of N in the weight distribution. However, an easier way is to view the ARA codes as serial concatenated codes, where the combined version of outer accumulator (precoder) and repetition 3 or 4 code can be considered as an outer code with a minimum distance of 3 or 4, respectively. The inner code then is the inner accumulator (as a recursive convolutional code) with puncturing. Then, we can use the theorems in [28] to obtain interleaving gains. Note that the ARA codes are systematic; thus, when the output weight of accumulator with puncturing is zero for a nonzero input weight, the ARA codeword weight is nonzero. Then, the interleaving gain for the ARA codes with repetition 3 or 4 for FER is $1/N$ and for BER is $1/N^2$ i.e., averaged word error rate $P_W = O(N^{-1})$, and averaged bit error rate $P_b = O(N^{-2})$. Thus, precoder improves the SNR threshold but the interleaving gain remains unchanged with respect to RA(3,3) or RA(4,4). The ensemble weight distribution of protograph based ARA codes can be obtained using the method proposed in [25] and [26].

V. Conclusion

In this paper, we proposed a new channel coding scheme called Accumulate Repeat Accumulate codes (ARA). This class of codes can be viewed as a subclass of LDPC codes with fast encoder structure, and they have a projected graph or protograph representation, which allows for high-speed iterative decoding implementation using belief propagation. Based on density evolution for protograph-based ARA codes, we have shown that, for maximum variable node degree 5, a minimum bit SNR as low as 0.08 dB from channel capacity for rate-1/2 can be achieved as the block size goes to infinity. Such a low iterative decoding threshold cannot be achieved by RA, IRA, or unstructured irregular LDPC codes with the same constraint on the maximum variable node degree. We constructed families of higher rate both multi-edge-type and protograph-based ARA codes with iterative decoding thresholds that stay close to their respective channel capacity thresholds uniformly. Iterative decoding simulation results for ARA are provided and compared with turbo codes. The weight distribution of some simple multi edge type ARA codes is obtained, and through existing tightest bounds we have shown the ML performance of ARA codes approaches very closely to the performance of random codes.

Acknowledgment

The authors would like to thank J. Thorpe and S. Dolinar of the Jet Propulsion Laboratory for their comments and C. Jones and K. Andrews for providing simulation results for two ARA codes.

References

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

[2] C. Berrou and A. Glavieux, "Near optimum error correcting coding and decoding: Turbo-codes," *IEEE Trans. Commun.*, vol. 44, no. 10, pp. 1261–1271, Oct. 1996.

[3] M. R. Tanner, "A recursive approach to low complexity codes," *IEEE Trans. Inf. Theory*, vol. IT-27, no. 5, pp. 533–547, Sep. 1981.

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

[5] S.-Y. Chung, D. Forney, T. Richardson, and R. Urbanke, "On the design of low-density parity- check codes within 0.0045 dB of the Shannon limit," *IEEE Commun. Lett.*, vol. 5, no. 1, pp. 58–60, Jan. 2001.

[6] D. Divsalar, H. Jin, and R. McEliece, "Coding theorems for turbo-like codes," in *Proc. Allerton Conf.*, 1998, pp. 201–210.

[7] H. Jin, A. Khandekar, and R. McEliece, "Irregular repeat-accumulate codes," in *Proc. 2nd Int. Symp. Turbo Codes*, 2000, pp. 1–8.

[8] J. Thorpe, Low Density Parity Check (LDPC) Codes Constructed from Protographs JPL INP, 2003, Progress Rep. 42-154.

[9] D. Divsalar, S. Dolinar, and F. Pollara, "Iterative turbo decoder analysis based on density evolution," *IEEE J. Sel. Areas Commun.*, vol. 19, no. 5, pp. 891–907, May 2001.

[10] Richardson , "Methods and Apparatus for Decoding LDPC Codes," U.S. Patent 6 633 856, Oct. 14, 2003.

[11] T. Richardson, "Multi-edge type LDPC codes," in *Workshop Honoring Prof. Bob McEliece*, Pasadena, CA, May 24–25, 2002.

[12] D. J. C. MacKay and R. M. Neal, "Near Shannon limit performance of low density parity check codes," *Electron. Lett.*, vol. 32, no. 18, p. 1645, Aug. 1996.

[13] T. Richardson and R. Urbanke, "The renaissance of Gallager's low-density parity-check codes," *IEEE Commun. Mag.*, pp. 126–131, Aug. 2003.

[14] D. Divsalar, A Simple Tight Bound on Error Probability of Block Codes with Application to Turbo Codes Jet Propulsion Laboratory, CA, 1999, JPL TMO Progress Rep. 42-139.

[15] D. Divsalar and E. Biglieri, "Upper bounds to error probabilities of coded systems beyond the cutoff rate," *IEEE Trans. Commun.*, vol. 51, no. 12, pp. 2011–2018, Dec. 2003.

[16] M. Luby, M. Mitzenmacher, A. Shokrollahi, and D. Spielman, "Analysis of low density codes and improved designs using irregular graphs," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 585–598, Feb. 2001.

[17] T. Richardson and R. Urbanke, "The capacity of low-density parity check codes under message- passing decoding," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 599–618, Feb. 2001.

[18] A. Abbasfar, D. Divsalar, and K. Yao, "A class of turbo-like codes with efficient and practical high-speed decoders," in *Proc. IEEE Military Commun. Conf.*, Oct.-Nov. 2004, pp. 245–250.

[19] Y. Kou, S. Lin, and M. P. C. Fossorier, "Low-density parity-check codes based on finite geometries: A rediscovery and new results," *IEEE Trans. Inf. Theory*, vol. 47, no. 11, pp. 2711–2736, Nov. 2001.

[20] F. R. Kschischang, "Codes defined on graphs," *IEEE Commun. Mag.*, vol. 41, no. 8, pp. 118–125, Aug. 2003.

[21] S. Bennedetto and G. Montorsi, "Unveiling turbo codes: Some results on parallel concatenated codes," *IEEE Trans. Inf. Theory*, vol. 42, no. 2, pp. 409–428, Mar. 1996.

[22] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara, "Serial concatenation of interleaved codes: Performance analysis, design, and iterative decoding," *IEEE Trans. Inf. Theory*, vol. 44, no. 5, pp. 909–926, May 1998.

[23] D. Divsalar, S. Dolinar, J. Thorpe, and C. Jones, "Constructing LDPC codes from simple loop-free encoding modules," in *Proc. IEEE ICC*, Seoul, Korea, May 2005, pp. 658–662.
[24] ——, "Low-rate LDPC codes with simple protograph structure," in *Proc. IEEE Int. Symp. Inf. Theory*, Adelaide, Australia, Sep. 2005, pp. 1622–1626.
[25] D. Divsalar, "Finite length weight enumerators for protograph based LDPC code ensembles," presented at the *IEEE Commun. Theory Workshop*, Park City, UT, Jun. 2005.
[26] D. Divsalar, "Ensemble weight enumerators for protograph LDPC codes," in *Proc. IEEE Int. Symp. Inf. Theory*, Seattle, WA, Jul. 2006, pp. 1554–1558.
[27] K. Andrews, S. Dolinar, and J. Thorpe, "Encoders for block-circulant LDPC codes," in *Proc. IEEE Int. Symp. Inf. Theory*, Adelaide, Australia, Sep. 2005, pp. 2300–2304.
[28] H. Jin and R. J. McEliece, "Coding theorems for turbo code ensembles," *IEEE Trans. Inf. Theory*, vol. 48, no. 6, pp. 1451–1461, Jun. 2002.
[29] S. ten Brink, "Code doping for triggering iterative decoding convergence," in *Proc. IEEE Int. Symp. Inf. Theory*, Jul. 2001, p. 235.

Aliazam Abbasfar (S'01–M'05) received the B.Sc. and M.Sc. degrees from the University of Tehran, Tehran, Iran, in 1992 and 1995, respectively, and the Ph.D. degree from the University of California, Los Angeles (UCLA), in 2005, all in electrical engineering.

From 1992 to 1994, he was with the Iran Telecommunication Research Center (ITRC), where he was involved with switching networks for data communications. Between 2001 and 2004, he held positions as a Senior Design Engineer in the areas of communication system design and digital VLSI ASIC design with Innovics Inc., Sequoia Communications, and Jaalaa Inc. Upon graduation from UCLA, he joined Rambus Inc., Los Altos, CA, where he is involved with high-speed data communications on wireline backplane links. His main research interests include wireless communications, equalization, error correcting codes, and VLSI ASICs for digital data communications.

Dariush Divsalar (S'76–M'78–SM'90–F'97) received the Ph.D. degree in electrical engineering from the University of California, Los Angeles (UCLA), in 1978.

Since then, he has been with the Jet Propulsion Laboratory (JPL), California Institute of Technology (Caltech), Pasadena, where he is a Principal Scientist. At JPL, he has been involved with developing state-of-the-art technology for advanced deep space communications systems and future NASA space exploration and Mobile and Satellite Communications. Since 1986, he has taught graduate courses in digital communications, spread spectrum communications, and coding at UCLA and Caltech. He has published more than 150 papers, coauthored a book entitled *An Introduction to Trellis Coded Modulation with Applications* (New York: MacMillan, 1991), contributed to two other books, and holds ten U.S. patents in the above areas.

Dr. Divsalar was the corecipient of the 1986 Prize Paper Award in Communications for the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY. Recently, one of his papers was selected as one of the key research papers published by the IEEE Communications Society during the past five decades. He has received over 30 NASA Tech Brief awards and a NASA Exceptional Engineering Achievement Medal in 1996. He served as Editor and Area Editor in Coding and Communication Theory for the IEEE TRANSACTIONS ON COMMUNICATIONS from 1989 to 1996. He became a Fellow of IEEE in 1997 for contributions to the analysis and design of coding and modulation techniques for satellite, mobile, and deep-space communication systems.

Kung Yao (S'59–M'65–SM'91–F'94–LF'04) received the B.S.E. (Highest Honors), M.A., and Ph.D. degrees in electrical engineering all from Princeton University, Princeton, NJ.

He was a NAS-NRC Post-Doctoral Research Fellow with the University of California, Berkeley. Presently, he is a Professor with the Electrical Engineering Department, University of California, Los Angeles (UCLA). In 1969, he was a Visiting Assistant Professor with the Massachusetts Institute of Technology, Cambirdge. From 1985 to 1988, he served as an Assistant Dean of the School of Engineering and Applied Science at UCLA. His research interests include sensor array system, digital communication theory, wireless radio system, chaos communications, digital and array processing, systolic and VLSI algorithms, and simulation. He has published over 250 journal and conference papers. He was the coeditor of a two-volume series of an IEEE Reprint Book on *High Performance VLSI Signal Processing* (IEEE Press, 1997).

Dr. Yao was the recipient of the IEEE Signal Processing Society's 1993 Senior Award in VLSI Signal Processing. He has served as Associate Editors for the IEEE TRANSACTIONS ON INFORMATION THEORY, IEEE TRANSACTIONS ON SIGNAL PROCESSING, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS, and IEEE COMMUNICATIONS LETTERS, and as a Guest Editors of numerous Special Issues.