Accumulate-Repeat-Accumulate-Accumulate-Codes

Dariush Divsalar  
Jet Propulsion Laboratory  
California Institute of Technology  
4800 Oak Grove Drive  
Pasadena, CA 91109-8099  
Email: Dariush.Divsalar@jpl.nasa.gov

Sam Dolinar  
Jet Propulsion Laboratory  
California Institute of Technology  
4800 Oak Grove Drive  
Pasadena, CA 91109-8099  
Email: sam@shannon.jpl.nasa.gov

Jeremy Thorpe  
California Institute of Technology  
136-93  
1200 E. California Blvd  
Pasadena, CA 91125  
Email: thorpe@its.caltech.edu

Abstract—Inspired by recently proposed Accumulate-Repeat-Accumulate (ARA) codes [15], in this paper we propose a channel coding scheme called Accumulate-Repeat-Accumulate-Accumulate (ARAA) codes. These codes can be seen 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 a high-speed iterative decoder implementation using belief propagation. An ARAA code can be viewed as a precoded Repeat-and-Accumulate (RA) code with puncturing in concatenation with another accumulator, where simply an accumulator is chosen as the precoder; thus ARA codes have a very fast encoder structure. Using density evolution on their associated protographs, we find examples of rate-1/2 ARA codes with maximum variable node degree 4 for which a minimum bit-SNR as low as 0.21 dB from the channel capacity limit can be achieved as the block size goes to infinity. Such a low threshold cannot be achieved by RA or Irregular RA (IRA) or unstructured irregular LDPC codes with the same constraint on the maximum variable node degree. Furthermore by puncturing the accumulators 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 show comparable performance with the best-known LDPC codes but with very low error floor even at moderate block sizes.

I. INTRODUCTION

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 LDPC codes, and extended the work of Gallager, see for example [9], [12] and references there. Recently Repeat-and-Accumulate (RA) [4], Irregular Repeat-and-Accumulate (IRA) [6] and Accumulate-Repeat-Accumulate-Accumulate (ARAA) [15] codes were proposed as simple subclasses of LDPC codes with fast encoder structures. Classical RA codes in addition to simplicity have reasonably good performance for rates less than or equal to 1/3. RA codes use a fixed repetition of input bits. On the other hand, IRA codes inspired by RA and irregular LDPC [3] codes have irregular repetitions of input bits. To achieve very low decoding thresholds for IRA and irregular LDPC codes, the maximum variable node degree for some portion of the input bits must be very high. ARA codes, on the other hand, can achieve a very low threshold within 0.07 dB of the capacity limit for rate-1/2 codes with a low maximum variable node degree of 5 [15]. These recent results on RA, IRA, ARA codes, and serial concatenation of rate-1 codes by Pfister and Siegel [5], motivated us to define a new subclass, Accumulate-Repeat-Accumulate-Accumulate-Accumulate (ARAA) codes. Our objective was to enhance the error-floor performance of ARA codes, while maintaining a simple encoding structure and low maximum variable node degree. As with ARA codes, ARAA codes also have a projected graph [8] or protograph [7] representation, which makes high-speed implementation of the decoder more feasible. A protograph [7] is a Tanner graph with a relatively small number of nodes. As a simple example, we consider the protograph shown in Fig. 1(b). This graph consists of 4 variable nodes and 3 check nodes, connected by 9 edges. By itself, this graph is recognized as the Tanner graph of an (n = 3, k = 1) LDPC code, in this case, the rate-1/3 RA code shown in Fig. 1(a). In Fig. 1(b), the variable nodes connected to the channel are denoted by dark filled circles. Blank circles represent variable nodes not connected to the channel (i.e., punctured). Check nodes are denoted by circles with a plus sign inside. Thresholds for infinite block size are computed using density evolution on the protograph; the minimum Eb/N0 threshold is 0.502 dB for the rate-1/3 RA code with iterative decoding. We can obtain a larger graph by

Fig. 1. (a) A rate-1/3 RA code with repetition 3, and (b) its protograph

a "copy-and-permute" operation as shown in Fig. 2.

Fig. 2. Copy and permute operation to generate larger graphs
II. ACCUMULATE-REPEAT-ACCUMULATE (ARA) CODES

A rate-1/2 classical RA code has a high threshold of 3.01 dB. ARA codes can be viewed as precoded RA codes but with puncturing where an accumulator is used as a precoder. The simplest example of a rate-1/2 ARA code, its encoder, its protograph, and the corresponding decoding threshold of 0.516 dB are shown in Fig. 3. Other rate-1/2 ARA protographs with maximum variable degree 5 have thresholds as low as 0.26 dB, compared to the Shannon capacity limit of 0.19 dB [15].

III. ACCUMULATE-REPEAT-ACCUMULATE-ACCUMULATE (ARAA) CODES

A simple example of a rate-1/2 ARAA code, its protograph, and the corresponding threshold of 0.654 dB are shown in Fig. 4. This ARAA protograph is similar to that of the ARA protograph in Fig. 3, except for its extra accumulator stage and fewer parallel edges. The ARAA protograph in Fig. 4 only has maximum variable node degree 4 but has more total nodes than the ARA protograph in Fig. 3. Fig. 5 shows simulation results comparing the performance of the ARA and the ARAA codes in Fig. 3 and Fig. 4, respectively.

Other rate-1/2 ARAA examples with maximum variable node degree 4 (but with larger protographs) can reduce the threshold further. For comparison, the minimum SNR threshold reported in [3] for the best rate-1/2 unstructured irregular LDPC code with maximum variable node degree 4 is 0.8 dB. An example of a rate-1/2 ARAA code with threshold 0.562 dB is shown in Fig. 6. Here two-thirds of the input bits are precoded by the outer accumulator. The threshold can be further reduced to 0.398 dB (only 0.21 dB from the capacity limit) if thirteen-fourteenths of the input bits are precoded by the outer accumulator. In these examples the gain due to precoding is about 0.5 dB.

ARAA codes with higher code rates are obtained by puncturing the output of the middle accumulator. For example, we obtained thresholds of 1.46 dB and 2.00 dB for rates 2/3 and 3/4, respectively, for punctured versions of the code shown in Fig. 4. The rate-2/3 and rate-3/4 ARAA codes are shown in Fig. 7 and Fig. 8, respectively.

In most wireless standards, either a convolutional or turbo code with puncturing for various code rates is included. The proposed ARAA codes have fast encoders, puncturing can produce various code rates, and a single fast decoder using belief propagation with depuncturing can be implemented to handle the different code rates. Thus ARAA codes have all the required practical features, and they achieve excellent performance, and low error floor even for moderately short blocks.

IV. ENSEMBLE WEIGHT DISTRIBUTION OF ARAA CODES

We use the concept of a uniform interleaver [13] to compute the overall input-output weight enumerator (IOWE) of ARAA
The IOWE of an accumulator without puncturing is [4]:
\[ A_{w,d}^{(2)} = \binom{N - d}{\lfloor \frac{d}{2} \rfloor} \right( \binom{d - 1}{\lfloor \frac{d}{2} \rfloor} - 1 \right) \]

To compute the IOWE of an accumulator with puncturing, we use the equivalent graph depicted in Fig. 9 [15]. The code for computing the IOWE is a concatenation of a regular check code and an accumulator [15], which is shown in Fig. 10. The check code takes \( p \) input bits and produces 1 overall parity bit as output. The first step is to compute the IOWE \( A_{w,d}^{(p)} \) of the check code. This IOWE can be expressed in closed form if we use the two dimensional Z-transform \( A_{w,d}^{(p)}(W,D) \). Then the IOWE \( A_{w,d}^{(p)} \) is obtained from the inverse Z-transform.

In this paper we work out the details for \( p = 2 \). For other values of \( p \), we can use a similar method. In the case of \( p = 2 \), the Z-transform of the IOWE for a single check code with 2 inputs and one output can be obtained as:
\[ A_{w,d}^{(2)}(W,D) = 1 + W^2 + 2WD \]

For a code with \( N \) checks, the Z-transform is simply:
\[ A_{w,d}^{(2)}(W,D) = \left( A_{w,d}^{(2)}(W,D) \right)^N = (1 + W^2 + 2WD)^N \]

This can be expanded as:
\[ A_{w,d}^{(2)}(W,D) = \sum_{d=0}^{N} \sum_{w=0}^{2N} \left( \binom{N - d}{\lfloor \frac{d}{2} \rfloor} \right) \right( \binom{N - d}{\lfloor \frac{d}{2} \rfloor} - 1 \right) \]
\[ \times 2^d \delta_{w,2j+d} W^w D^d \]

The IOWE is obtained from the inverse Z-transform as:
\[ A_{w,d}^{(2)} = \sum_{j=0}^{N-d} \binom{N-d}{j} 2^d \delta_{w,2j+d} \]

Next we compute the IOWE of the accumulator with puncturing shown in Fig. 10. Since the check code is regular and memoryless, any interleaver between the two constituent codes will not change the IOWE of the overall code [15]. In order to compute the IOWE for this code we insert a uniform interleaver between the two constituents, as shown on the right side of Fig. 10. Then we use the uniform interleaver formula [14] to compute the IOWE of the accumulator with puncturing. We have in general
\[ A_{w,d}^{(p)} = \sum_{h=0}^{N-h} A_{w,h}^{(p)} A_{h,d}^{(2)} \]

For the case \( p = 2 \) this is explicitly evaluated as
\[ A_{w,d}^{(2)} = \sum_{h=0}^{N-h} \sum_{j=0}^{N - h} \left( \binom{N-h}{j} \right) \left( \binom{N-d}{\lfloor \frac{d}{2} \rfloor} \right) \right( \binom{d-1}{\lfloor \frac{d}{2} \rfloor} - 1 \right) \]
\[ \times 2^h \delta_{w,2j+h} \]

It should be noted that despite the fact that we use a uniform interleaver to obtain the IOWE, we obtain the exact IOWE for an accumulator with puncturing [15].
Before obtaining the IOWE of the full ARAA code, we first find the IOWE of the systematic RAA code with puncturing, which can be evaluated in the case of a uniform interleaver after repetition as

\[
A_{w,d}^{\text{rep}(q)-\text{acc}(p)-\text{acc}} = \sum_{i=0}^{qN} \sum_{l=0}^{N} \sum_{h=0}^{N-h} \frac{(N)^{i}}{(N)^{l}} \binom{N-h}{j}
\]

(9)

For the systematic punctured RAA code with repetition \(q = 2\) and puncturing \(p = 2\) we obtain:

\[
A_{w,d}^{\text{rep}(2)-\text{acc}(2)-\text{acc}} = \frac{(N)^{i}}{(2w)^{i}} \sum_{i=0}^{N} \sum_{h=0}^{N-h} \sum_{j=0}^{j} \binom{N-h}{j}
\]

\[
\times \binom{N-i}{i-1} \binom{[h/2]}{[h/2]-1}
\]

\[
\times \binom{N-d+w}{d-w-1} \binom{[i/2]}{[i/2]-1}
\]

\[
\times 2^{(d-w+h)}
\]

(10)

Finally we obtain the ensemble weight distribution of an ARAA code as a precoded RAA code with puncturing, using an accumulator as the precoder. As we have seen, in ARAA codes, only a portion of the information block goes to the outer accumulator. In other words, \(M\) bits are passed through without any change and the remaining \(N-M\) bits go through the outer accumulator. Then the overall output bits are applied to the punctured RAA code. The technique of letting \(M\) bits bypass the precoder can be called doping after ten Brink [16]. The use of these \(M\) bits is essential for the iterative decoding to start the message-passing algorithm; \(M\) is considered a parameter in the code design. Fig. 11 shows a block diagram of the precoded RAA with puncturing, which represents the general form of an ARAA code. In order to find the IOWE explicitly

\[
A_{w,d}^{\text{pre-rep}(2)-\text{acc}(2)-\text{acc}} = \sum_{m=0}^{M} \sum_{n=0}^{N} \sum_{h=0}^{N} \sum_{j=0}^{j} \frac{(N)^{m}}{(2N)^{m}} \binom{N-h}{j}
\]

\[
\times \binom{N-i}{i-1} \binom{[h/2]}{[h/2]-1}
\]

\[
\times \binom{N-d+w}{d-w-1} \binom{[i/2]}{[i/2]-1}
\]

\[
\times 2^{(d-w+h)}
\]

(12)

V. MAXIMUM LIKELIHOOD DECODING ANALYSIS OF ARAA CODES

Reference [10] provides a tight upper bound on frame error rate (FER) and bit error rate (BER) for an \((n, k)\) linear block code with code rate \(R = k/n\) and distance spectrum \(A_d\) (number of codewords with weight \(d\)), decoded by the Maximum Likelihood (ML) criterion over an additive white Gaussian noise (AWGN) channel. It also provides a closed-form expression for (a tight bound on) the minimum required bit signal-to-noise ratio (SNR) \(E_b/N_0\). We can use this bound to characterize the ML performance of ARAA codes.

Define a normalized distance as \(\delta = d/n\), and a normalized distance spectrum (also called the rate distance spectrum) as \(r(\delta) = A_{0}\). Then the FER bound in [10] or [11] can be expressed as:

\[
P_e \leq \sum_{d=d_{\text{min}}}^{d_{\text{max}}} e^{-nE(\delta, \beta \frac{E_b}{N_0})}
\]

(14)

where

\[
E(\delta, \beta \frac{E_b}{N_0}) = -\frac{1}{2} \ln(1 - \beta + \beta e^{-2r(\delta)}) + \frac{\beta \delta}{1 - (1 - \beta) \delta} \frac{E_b}{N_0}
\]

and

\[
\beta = \frac{1 - \delta}{\delta} \left[ \frac{E_c}{N_0} \left( 1 - \frac{2}{\delta} - e^{-2r(\delta)} \right) + \left( 1 + \frac{E_c}{N_0} \right)^2 - 1 \right]
\]

(16)

(15)

where \(\beta = R_c \frac{E_b}{N_0}\), and \(0 < \beta < 1\). When \(\beta = 1\), this bound reduces to the union bound. To compute the corresponding bound on BER, just replace \(A_d\) with \(A_{w,d}\) 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 closed-form threshold on minimum \(E_b/N_0\) that can be written as

\[
\left( \frac{E_b}{N_0} \right)_{\text{min}} \leq \frac{1}{R_c} \max(1 - e^{-2r(\delta)}) \frac{1 - \delta}{2\delta}
\]

(17)
In order to use this bound to evaluate the $\frac{\delta}{2N}$ threshold for an ARAA code in the limit of large $N$, we require an asymptotic expression (as $N \to \infty$) for $r(\delta)$ for the full ARAA $(q, p)$ code. We illustrate this procedure for $q = 2 , p = 2$ after summing (10) over $\omega$. Let $\delta = \frac{\delta}{2N}$ for $0 < \delta < 1$, $\alpha = \frac{\delta}{2N}$ for $0 < \delta < 1/2$, $\delta = \frac{\delta}{2N}$ for $0 < \delta < 1/2$, and $\frac{\delta}{2N}$ for $0 < \delta < 1/2$. Then, using asymptotic expressions for the binomial coefficients in (10), we have:

$$r(\delta) = \max_{\epsilon_1, \epsilon_2, \rho_1, \rho_2} \left\{ \frac{1}{2} H(2\rho_2 + \eta) - \frac{1}{2} H(2\rho_1) + \frac{1}{2} - \eta \right\} H\left(\frac{\rho_2}{2 - \eta}\right) + \frac{1}{2} - \rho_1 \right\} H\left(\frac{\eta/2}{2 - \rho_1}\right) + \rho_1 H\left(\eta/2\right) + \frac{1}{2} - \delta + \rho_2 + \frac{\eta}{2} H\left(\frac{\rho_1/2}{2 - \rho_2 - \eta}\right) + \frac{\delta}{2N} - \rho_2 \right\} H\left(\frac{\rho_1/2}{2 - \rho_2 - \eta}\right) + \eta \ln 2 \right\}$$

(18)

where $H(\cdot)$ is the binary entropy function, $H(x) = -x \ln x - (1-x) \ln(1-x)$. Using (18) in (17), we can obtain the minimum $E_0/N_0$ threshold for a punctured RAA code with $q = 2 , p = 2$ as 1.03 dB.

Finally we obtain the asymptotic expression for $r(\delta) = \frac{\ln A^2}{2N}$ for an ARAA code with $q = 2 , p = 2$ after summing (10) over $\omega$. Let $\alpha = \frac{\delta}{2N}$ for $0 < \alpha < 1/2$, $\epsilon_1 = \frac{\alpha}{2N}$ for $0 < \epsilon_1 < \alpha$, $\epsilon_2 = \frac{\alpha - \epsilon_1}{2N}$ for $0 < \epsilon_2 < 1 - \alpha$, $\delta = \frac{\delta}{2N}$ for $0 < \delta < 1$, $\eta = \frac{\eta}{2N}$ for $0 < \eta < 1/2$, $\rho_1 = \frac{\rho_1}{2N}$ for $0 < \rho_1 < 1/2$, $\rho_2 = \frac{\rho_2}{2N}$ for $0 < \rho_2 < 1/2 - \eta$.

$$r(\delta) = \max_{\epsilon_1, \epsilon_2, \rho_1, \rho_2} \left\{ \alpha \left\{ H\left(\frac{\epsilon_1}{\alpha}\right) - H(2\rho_2 + \eta) - \frac{1}{2} H(2\rho_1) \right\} + \frac{1}{2} - \eta \right\} H\left(\frac{\rho_2}{2 - \eta}\right) + \frac{1}{2} - \rho_1 \right\} H\left(\frac{\eta/2}{2 - \rho_1}\right) + \rho_1 H\left(\eta/2\right) + \frac{1}{2} - \delta + \epsilon_1 + \epsilon_2 \right\} H\left(\frac{\rho_1/2}{2 - \delta + \epsilon_1 + \epsilon_2}\right) + \frac{\delta}{2N} - \epsilon_1 \right\} H\left(\frac{\rho_1/2}{2 - \delta + \epsilon_1 + \epsilon_2}\right) + \eta \ln 2 \right\}$$

(19)

Using (19) in (17), we can obtain the minimum $E_0/N_0$ threshold for an ARAA code with $q = 2 , p = 2$ as 0.31 dB when $\alpha \rightarrow 0$. Thus, the precoding gain using ML decoding is about 0.7 dB. This is similar to the observation we made earlier using iterative decoding.

VI. CONCLUSION

In this paper we propose a new class of channel codes called ARAA codes. These codes are a subclass of Low Density Parity Check (LDPC) codes with a fast encoder structure. Puncturing can produce families of good codes of various rates. ARAA codes have a projected graph or protograph representation, which allows for high-speed decoder implementation. Iterative decoding simulations verify excellent performance and low error floor.

ACKNOWLEDGMENT

This research was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with NASA.

REFERENCES