

Constructing LDPC Codes from Simple Loop-Free Encoding Modules

Dariusz Divsalar, Sam Dolinar, Jeremy Thorpe, and Christopher Jones

Jet Propulsion Laboratory
California Institute of Technology
Pasadena, CA 91109-8099

e-mails: Dariusz.Divsalar@jpl.nasa.gov, sam@shannon.jpl.nasa.gov,
thorpe@its.caltech.edu, christop@jpl.nasa.gov

Abstract—Inspired by recently proposed Accumulate-Repeat-Accumulate (ARA) codes, in this paper we propose a construction method for LDPC codes using simple loop-free encoding modules. Such codes can be viewed as serial/parallel concatenations of simple modules such as accumulators, repetition codes, differentiators, and punctured single parity check codes. Examples are Accumulate-Repeat-Accumulate (ARA) codes, Accumulate-Repeat-Accumulate-Accumulate (ARAA) codes and Accumulate-Repeat-Check-Accumulate codes, and other variations. These codes constitute a subclass of LDPC codes with very fast encoder structure. They also have a projected graph or protograph representation that allows for high-speed decoder implementation. Based on density evolution, we show through some examples that low iterative decoding thresholds close to the channel capacity limits can be achieved with low maximum variable node degrees, as the block size goes to infinity. The decoding threshold in many examples outperforms that of the best known unstructured irregular LDPC codes constrained to have the same maximum node degree. Furthermore, by puncturing the accumulator modules, any desired higher rate codes can be obtained with thresholds that stay close to their respective channel capacity thresholds uniformly.

I. INTRODUCTION

Low-density parity-check (LDPC) codes were proposed by Gallager [1] in 1962. After introduction of turbo codes by Berrou et al [2] in 1993, researchers revisited LDPC codes, and extended the work of Gallager using the code graphs introduced by Tanner [3] in 1981. After 1993 there have been many contributions to the design and analysis of LDPC codes; see for example [12], [14], [15], [5], [16], [17], [20] and references there. Recently there has been a flurry of work on designing LDPC codes with imposed sub-structure, starting with the introduction of multi-edge type codes in [11] and [13].

Repeat-Accumulate (RA) [6], Irregular Repeat-Accumulate (IRA) [7] and recently Accumulate-Repeat-Accumulate (ARA) [19] codes were proposed as simple subclasses of LDPC codes with fast encoder structures. For high-speed decoding, it is advantageous for an LDPC code to be constructed from a protograph [8] or a projected graph [10]. A protograph is a Tanner graph with a relatively small number of nodes. A "copy-and-permute" operation [8] can be applied to the protograph to obtain larger derived graphs of various sizes. This operation consists of first making T copies of the protograph, and then permuting the endpoints of each edge

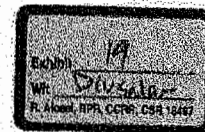
among the T variable and T check nodes connected to the set of T edges copied from the same edge in the protograph. The derived graph is the graph of a code T times as large as the code corresponding to the protograph, with the same rate and the same distribution of variable and check node degrees.

RA, IRA, and ARA codes, with suitable definitions of their interleavers, all have simple protograph representations and thus are amenable to both high-speed encoding and decoding. In this paper we define further extensions of RA, IRA, and ARA codes, all constructed from simple loop-free encoding modules. These extensions provide greater flexibility to construct codes with lower decoding thresholds and lower error floors.

II. REPEAT-ACCUMULATE (RA) AND IRREGULAR REPEAT-ACCUMULATE (IRA) CODES

Classical RA codes, in addition to simplicity, have reasonably good performance, with iterative decoding thresholds within 1 dB from capacity for rates less than or equal to 1/3. RA codes use fixed repetition for the input bits. As a simple example, we consider the rate-1/3 Repeat-Accumulate (RA) code depicted in Fig. 1(a). For this code the minimum E_b/N_0 threshold with iterative decoding is 0.502 dB. This code has a protograph representation shown in Fig. 1(b), as long as the interleaver π is chosen to be decomposable into permutations along each edge of the protograph. The iterative decoding threshold is unchanged despite this constraint imposed by the protograph. The protograph consists of 4 variable nodes and 3 check nodes, connected by 9 edges. Three variable nodes are connected to the channel and are shown as dark filled circles. One variable node is not connected to the channel (i.e., it is punctured) and is depicted by a blank circle. The three check nodes are depicted by circles with a plus sign inside.

Jin et al [7] generalized the notion of RA codes by allowing irregular repetition of the input bits. An Irregular RA (IRA) code can be viewed as a serial concatenation of a simple low density generator matrix (LDGM) code with different-degree variable nodes (irregular repetition) as an outer code, and an accumulator as an inner code. The encoder can be implemented by repetition codes, exclusive-OR's, and an accumulator as shown in Fig. 2.



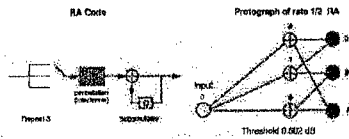


Fig. 1. (a) A rate-1/3 RA code with repetition 3, and (b) its corresponding protograph.

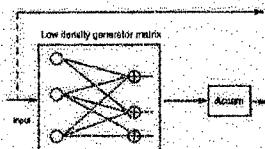


Fig. 2. Structure of an IRA code.

III. PUNCTURED RA CODES

A rate-1/2 classical RA code with a repetition-2 outer code has a high iterative decoding threshold of 3.01 dB. A much lower threshold of 1.116 dB was obtained in [19] for a rate-1/2 code using a modified RA construction as shown in Fig. 3. Here the outer code has repetition 3 or 4, the systematic bits are transmitted, and the accumulator code is punctured to make the overall rate 1/2. With suitable definitions of the interleaver π , the systematic punctured RA code can be represented by various protographs that yield the same threshold, as illustrated in the figure.

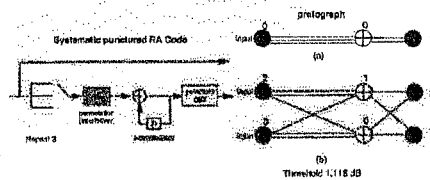


Fig. 3. A systematic punctured RA code and two possible representations by protographs.

IV. ACCUMULATE-REPEAT-ACCUMULATE (ARA) CODES

For IRA codes the node degree distribution can be optimized to achieve low thresholds. However, to achieve a very low threshold, the maximum repetition for some portion of the input bits can be very high. Similar requirements on the maximum variable node degree were noted for a general irregular LDPC code [4] to achieve a very low threshold. ARA codes, on the other hand, can achieve a very low threshold (e.g., within 0.08 dB from the capacity limit for rate-1/2 codes) with variable and check nodes of low maximum degree.

Consider the rate-1/2 systematic punctured RA code with repetition 3, and puncturing period 3, shown in Fig. 3. In [19]

it was shown that the threshold can be further improved by precoding the repetition code by an accumulator. The design of the precoder in [19] was guided by an analysis of the extrinsic SNR behavior of repetition codes and punctured accumulator codes using Gaussian density evolution.¹ The use of a rate-1 accumulator as a precoder dramatically improves the extrinsic SNR behavior of a repetition 3 outer code in the high extrinsic SNR region, and hence improves the threshold, as shown by comparison of the two outer code extrinsic SNR curves in Fig. 4. An RA code with an accumulator precoder is called

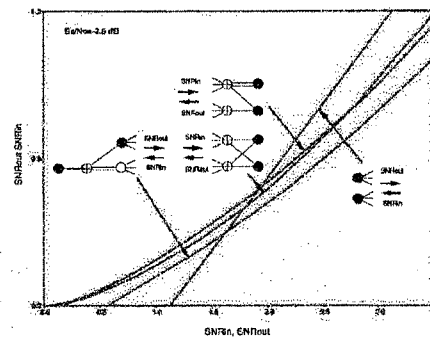


Fig. 4. Extrinsic SNR curves obtained by Gaussian density evolution comparing: outer codes using repetition-3 with or without precoder, and inner codes using an accumulator or check plus accumulator.

an Accumulate-Repeat-Accumulate (ARA) code [19].

An example of a simple rate-1/2 ARA code, its protograph, and the corresponding threshold are shown in Fig. 5. The ARA encoder in Fig. 5 uses a punctured accumulator as the precoder. A parallel decoder architecture based on decoding one copy

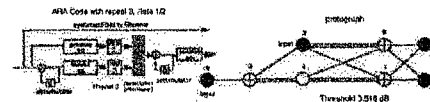


Fig. 5. A rate-1/2 ARA code and its protograph.

of the protograph (one page) per clock cycle per half-iteration is shown in Fig. 6.

V. ACCUMULATE-REPEAT-CHECK-ACCUMULATE (ARCA) CODES

The thresholds of ARA codes can be lowered further if we also modify the inner punctured accumulator. We can improve the extrinsic SNR behavior of the inner accumulator by using simple single-parity-check (SPC) codes prior to accumulation, as illustrated by comparison of the two inner code extrinsic

¹All other iterative decoding thresholds in this paper (except for the analysis in Fig. 4) were obtained using actual density evolution for LDPC codes [15].

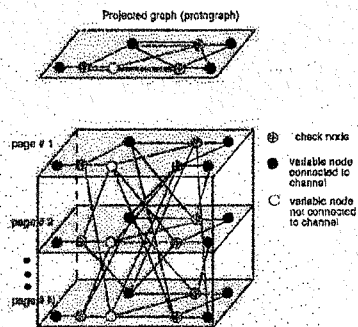


Fig. 6. Parallel LDPC decoder architecture for the rate-1/2 ARA protograph code in Fig. 5.

SNR curves in Fig. 4. We call such codes Accumulate-Repeat-Check-Accumulate code (ARCA) codes. These codes are an improved version of ARA codes. An example of a simple rate-1/2 ARCA code, its protograph, and the corresponding threshold are shown in Fig. 7.

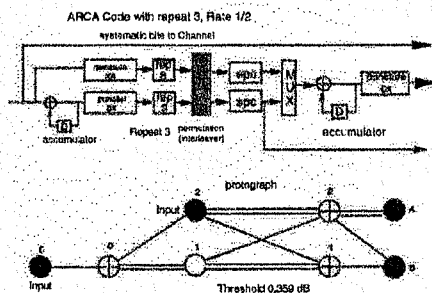


Fig. 7. A rate-1/2 ARCA code and its protograph.

VI. REPEAT-ACCUMULATE-ACCUMULATE (RAA) CODES WITH PUNCTURING

A rate-1/3 Repeat-Accumulate-Accumulate (RAA)² code without puncturing was proposed in [9] to lower the error floor. Its iterative decoding threshold is 0.456 dB. A rate-1/2 RAA code obtained by puncturing the systematic bits from a rate-1/3 RAA code has a threshold of 1.234 dB. A lower threshold of 0.868 dB for a rate-1/2 code can be obtained if we puncture the middle accumulator, and not the systematic bits. This type of code is a systematic punctured RAA code, shown in Fig. 8 along with its corresponding protograph.

²This code was called a Repeat & Double Accumulate (RDA) codes in [9].

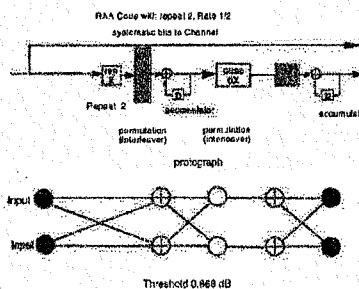


Fig. 8. A rate-1/2 systematic punctured RAA code and its protograph.

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

While RAA codes can yield a very low error floor, we also desire the somewhat lower decoding thresholds of ARA codes. Hence we were motivated to improve the threshold of RAA codes by using an accumulator precoder as in ARA codes. This combination is called an Accumulate-Repeat-Accumulate-Accumulate (ARAA) code. A simple example of a rate-1/2 ARAA code, its protograph, and the corresponding threshold are shown in Fig. 9. A second example is shown in

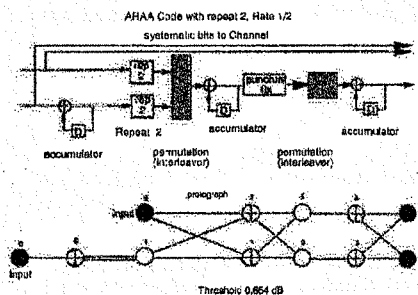


Fig. 9. A rate-1/2 ARAA code and its protograph.

Fig. 10.

VIII. RATE-COMPATIBLE CODE FAMILIES

The codes defined in this paper also allow the construction of code families derived from rate-compatible protographs. For example, ARA codes with repetition 4 (AR4A) are shown in Fig. 11 for rates 1/2 and higher. For this example we chose repetition 4 rather than 3 to achieve lower error floor. This AR4A code family uses an accumulator without puncturing as the precoder. Also shown in Fig. 11 are rate-compatible protograph families for ARCA and ARAA codes of rates 1/2 and higher. The higher code rates are constructed by using

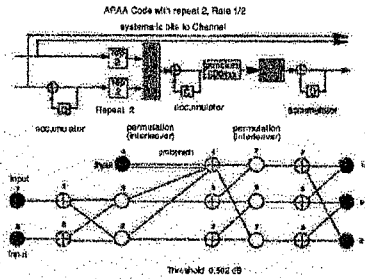


Fig. 10. Another rate-1/2 ARAA code and its protograph.

repetition codes for a portion of the input bits and then adding permuted versions of the repetition to the accumulators. The thresholds achieved by these three families compared to the corresponding capacity limits are shown in Fig. 12.

IX. SIMPLE SERIAL CODES

The lowest threshold achievable by a rate-1/2 protograph with only 3 variable nodes and 2 check nodes is 0.728 dB, achieved by the protograph shown in Fig. 13. This is just a serial concatenation of an outer accumulator with a punctured inner accumulator, although in this case the input of the outer accumulator is also used as an input to the inner accumulator. The outer code has minimum distance 3. To obtain better interleaving gain, we need to increase the minimum distance of the outer code [18]. The minimum distance of the outer accumulator can be increased to 6 by puncturing its output, and repeating the input of the outer accumulator by 3 as shown in Fig. 14.

X. SIMULATION RESULTS

Performance curves showing bit error rate (BER) and code-word error rate (WER) for codes in the AR4A family are plotted in Fig. 15 for input block size $k = 4096$. Another curve shows the BER performance of Flarion's low-error-floor rate-1/2 code of the same size obtained from their web site (<http://www.flarion.com>). Performance curves for rate-1/2 AR4A and ARAA codes are compared in Fig. 16 for $k = 1024$. Preliminary simulations with $k = 1024$ for other rate-1/2 codes proposed in this paper show comparable performance in the waterfall region but higher error floor. All simulations were performed on a field-programmable gate array (FPGA) implementation of an LDPC decoder developed at JPL.

XI. CONCLUSION

In this paper we designed several new varieties of LDPC codes using simple loop-free encoding modules, allowing fast encoding. Puncturing can produce families of good codes of various rates. These codes have a projected graph or protograph representation, which allows for high-speed decoder implementation. Iterative decoding simulations verify excellent performance.

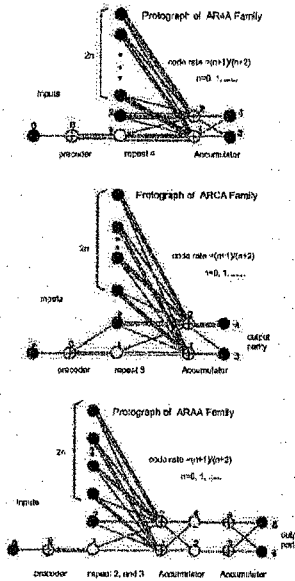


Fig. 11. AR4A, ARCA, and ARAA protographs of rates 1/2 and higher.

| Code Rate | Capacity (dB) | ARCA Threshold | AR4A Threshold | ARAA Threshold |
|-----------|---------------|----------------|----------------|----------------|
| 1/2 | 0.187 | 0.350 | 0.600 | 0.804 |
| 2/3 | 1.060 | 1.248 | 1.434 | 1.602 |
| 3/4 | 1.870 | 1.834 | 1.900 | 2.005 |
| 4/5 | 2.040 | 2.274 | 2.388 | 2.410 |
| 5/6 | 2.302 | 2.610 | 2.717 | 2.720 |
| 5/7 | 2.425 | 2.897 | 2.850 | 2.888 |
| 7/8 | 2.646 | 3.125 | 3.167 | 3.204 |
| 8/9 | 3.033 | 3.024 | 3.085 | 3.081 |

Fig. 12. Iterative decoding thresholds (in dB) for ARCA, AR4A, and ARAA protograph families, compared to the capacity limits.

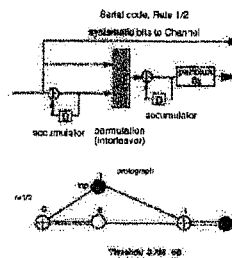


Fig. 13. Serial concatenation of accumulator with punctured accumulator.

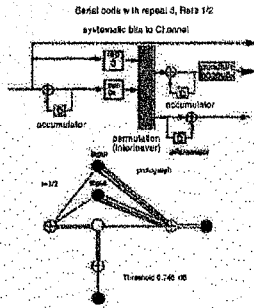


Fig. 14. Concatenation of accumulator, repetition, differentiator, and punctured accumulator.

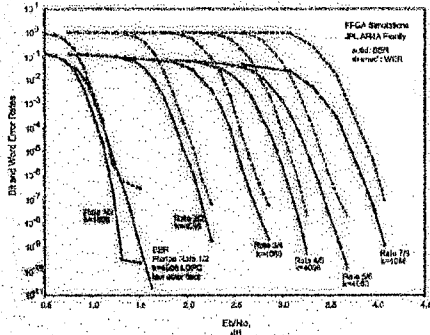


Fig. 15. Performance of AR4A codes of rates 1/2 and higher with input block size $k = 4096$.

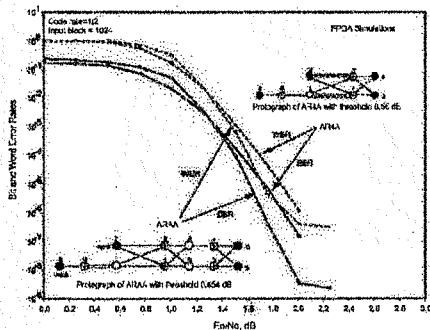


Fig. 16. Performance comparison for examples of rate-1/2 ARA and ARAA codes with input block size $k = 1024$.

ACKNOWLEDGMENT

This research was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under contract with the National Aeronautics and Space Administration.

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, pp. 1261-1271, October 1996.
- [3] M. R. Tanner, "A recursive approach to low complexity codes," *IEEE Trans. Inform. Theory*, vol. 27, pp. 533-547, 1981.
- [4] T. Richardson, A. Shokrollahi, and R. Urbanke, "Design of capacity-approaching irregular low-density parity-check codes," *IEEE Trans. Inform. Theory*, vol. 47, pp. 619-637, 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 Communication Letters*, vol. 5, pp. 58-60, 2001.
- [6] D. Divsalar, H. Jin, and R. McEliece, "Coding theorems for Turbo-like codes," in *Proceedings of the 1998 Allerton Conference*, pp. 201-210, 1998.
- [7] H. Jin, A. Khandekar, and R. McEliece, "Irregular repeat-accumulate codes," in *Proc. 2nd International Symposium on Turbo Codes*, pp. 1-8, 2000.
- [8] Jeremy Thorpe, *Low Density Parity Check (LDPC) Codes Constructed from Protographs*, JPL INP Progress Report 42-154, August 15, 2003.
- [9] D. Divsalar, S. Dolinar, and F. Pollara, "Iterative Turbo Decoder Analysis based on Density Evolution," *IEEE Journal on Select Areas in Communications*, Volume: 19 Issue: 5, May 2001, Page(s): 891-907.
- [10] Richardson, et al., "Methods and apparatus for decoding LDPC codes," United States Patent 6,633,856, October 14, 2003.
- [11] T. Richardson, *Multi-Edge Type LDPC Codes*, presented at the Workshop honoring Prof. Bob McEliece on his 60th birthday, California Institute of Technology, Pasadena, California, May 24-25, 2002.
- [12] D.J.C. MacKay, R.M. Neal, "Near Shannon limit performance of low density parity check codes," *Electronics Letters*, Vol. 32, Issue 18, 29 Aug. 1996, Page(s) 1645.
- [13] T. Richardson and R. Urbanke, "The Renaissance of Gallager's Low-Density Parity-Check Codes," *IEEE Communications Magazine*, pages 126-131, August 2003.
- [14] M. Luby, M. Mitzenmacher, A. Shokrollahi, and D. Spielman, "Analysis of low density codes and improved designs using irregular graphs," *IEEE Trans. Inform. Theory*, vol. 47, pp. 585-598, 2001.
- [15] T. Richardson and R. Urbanke, "The capacity of low-density parity check codes under message-passing decoding," *IEEE Trans. Inform. Theory*, vol. 47, pp. 599-618, 2001.
- [16] Y. Kou, S. Lin, and M.P.C. Fossorier, "Low-density parity-check codes based on finite geometries: a rediscovery and new results," *IEEE Transactions on Information Theory*, Volume: 47 Issue: 7, Nov. 2001, Page(s): 2711-2736.
- [17] F.R. Kschischang, "Codes defined on graphs," *IEEE Communications Magazine*, Vol. 41, Issue 8, Aug. 2003, Pages 118-125.
- [18] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara, "Serial concatenation of interleaved codes: Performance analysis, design, and iterative decoding," *IEEE Trans. Info. Theory*, vol. 44, pp. 909-926, May 1998.
- [19] A. Abbasfar, D. Divsalar, and K. Yao, "Accumulate Repeat Accumulate Codes," *ISIT 2004 and Globecom 2004*.
- [20] J. Xu and S. Lin, "A Combinatoric Superposition Method for Constructing Low Density Parity Check Codes," *Proc. 2003 IEEE Int. Symp. Inform. Theory*, Yokohama, Japan, June 29 - July 4, 2003, p. 30.