Structured Nonbinary Rate-Compatible Low-Density Parity-Check Codes

Jie Huang, Wei Zhou, and Shengli Zhou, *Member, IEEE*

***Abstract*—While existing works on rate-compatible low-density parity-check (RC-LDPC) codes, either binary or nonbinary, all focus on random-like construction, we in this letter present a novel method to construct *structured* nonbinary RC-LDPC codes. Protograph-based design with structured puncturing is applied while numerical simulation of code performance is adopted for optimization. The resultant codes are qualified as nonbinary quasi-cyclic LDPC (QC-LDPC) codes which are amenable to high-speed implementation of encoder/decoder. Numerical results show that the proposed codes achieve very good performance.**

***Index Terms*—Protograph, rate-compatible, LDPC.**

I. INTRODUCTION

COMPARED with rate-compatible (RC) codes using shortening/appending techniques and multiple-rate codes with constant block length, RC codes using puncturing/extending is more suitable for a type-II hybrid automatic repeat request (HARQ) system which provides an efficient framework for data transmission using ARQ/forward error correction (ARQ/FEC) protocols. This type of RC codes are a nested sequence of codes where the parity bits of higher rate codes are embedded in those of lower rate codes [1]–[3]. The beauty of RC codes is that all the codes in the sequence can be encoded and decoded using a single encoder/decoder pair.

Recently, the design of RC low-density parity-check (RC-LDPC) codes using puncturing/extending has obtained much attention, including both binary [2], [3] and nonbinary [4], [5]. Especially, reference [2] showed that puncturing alone cannot provide a sequence of good RC-LDPC codes across a wide range of rates because too much puncturing will induce noticeable performance loss for high rate codes. To overcome this, both puncturing and extending is used to create a sequence of RC-LDPC codes [2], [3]. More recently, nonbinary RC-LDPC (NB-RC-LDPC) codes over a Galois field GF(q) with $q > 2$ have been shown to significantly outperform its binary counterparts [4] in small to moderate code lengths. This is due to the fact that the selection of the operating Galois field provides the designer with another degree of freedom to achieve better performance and/or simplify the code design (see [6] and the references therein). Specifically, with large q ($q \geq 256$), nonbinary LDPC cycle codes with column weight of 2 perform very good. With small to moderate q ($4 \leq q \leq 64$), nonbinary LDPC codes with mixing column weight of 2, 3, or 4 have very good performance.

However, in all existing design, including both binary and nonbinary codes, random-like construction is adopted. In practice, structured design is highly desirable. In this letter we propose a novel design of *structured* NB-RC-LDPC codes. To the best of our knowledge, this is the first paper to construct *structured* RC-LDPC codes. Two main features of the proposed design distinguish itself from existing works. Firstly, our design is based on protographs [7] and the puncturing pattern is also designed in a structured way. A protograph is a graph with a relatively small number of nodes (parallel edges are permitted). A "copy-and-permute" operation can be applied to the protograph to obtain larger derived graphs of various sizes. With circulant permutation applied to each protograph edge in our design, the constructed codes are qualified as nonbinary *quasi-cyclic* LDPC (QC-LDPC) codes and thus amenable to high-speed implementation of encoder/decoder.

Secondly, in contrast to [5], [8] where decoding threshold obtained from density evolution over the binary erasure channel or approximate infinite-length Monte Carlo simulation over the AWGN channel is used for code optimization, we here adopt a new and more practical criterion for code optimization. In practice with small to moderate code lengths (on the order of hundreds to thousands of bits) the code performance highly depends on the code length. Thus we use the code's decoding performance as criterion for code optimization. This is especially important yet also feasible in finite-length design and in the context of type-II HARQ system because the decoding performance above block-error-rate of 10^{-2} is more important in a type-II HARQ system. Further, in stead of using only one value (the threshold) as criterion, one can evaluate more than one signal-to-noise ratio (SNR) points of the code's decoding performance for code optimization.

Next we present a design example over GF(16). We focus on GF(16) because in practice nonbinary LDPC codes over small to moderate Galois fields (e.g., $q \leq 16$) is more promising from the decoding complexity point of view. However, the proposed approach is quite general and can be generalized to different Galois fields and sequence of code rates.

We will compare our design with that in [4] where the number of information bits is 448. We consider a protograph design with 8 information nodes and code rates $8/10$, $8/11$, $\ldots$, 8/19, then lift it by the progressive edge-growth (PEG) algorithm [9] with a factor of 2 to remove parallel edges. The resulting graph is then lifted by the approximate cycle extrinsic (ACE) algorithm [10] with a factor of 7, resulting in an information length of $8 \times \log_2(16) \times 2 \times 7 = 448$ bits.

II. NONBINARY RATE-COMPATIBLE PROTOGRAPH CODES

A. Structured Symbol-Wise Puncturing

In a nonbinary LDPC code each variable node is represented by multiple bits, thus it can be punctured entirely (symbol-

---

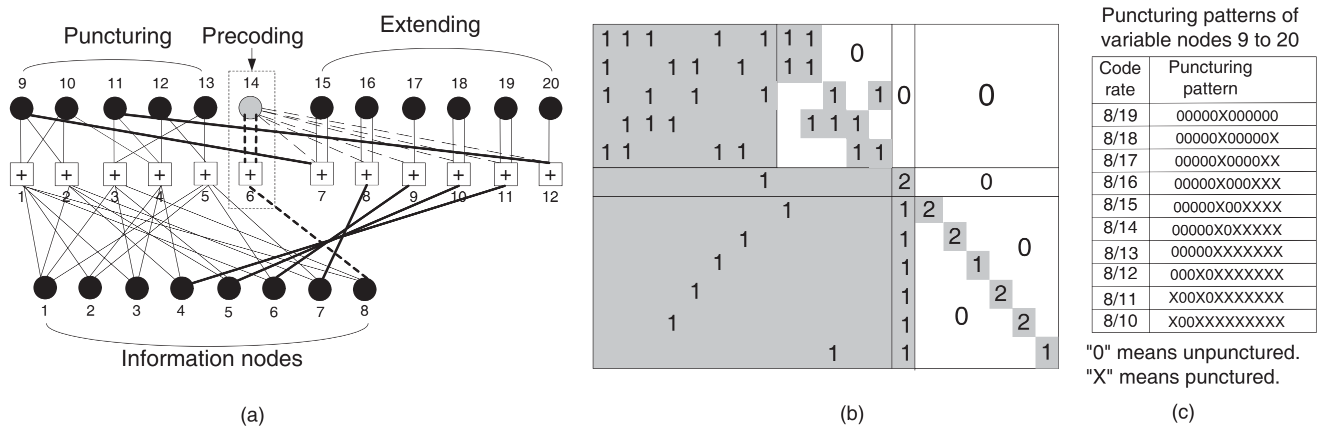
Manuscript received May 11, 2011. The associate editor coordinating the review of this letter and approving it for publication was A. G. i Amat.

This work is supported by the US Office of Naval Research grant N00014-09-1-0704 (PECASE).

J. Huang was with the University of Connecticut, Storrs. He is now with Wireless System R&D, Marvell Semiconductor, Santa Clara (e-mail: jiehuang@marvell.com).

W. Zhou and S. Zhou are with the Department of Electrical and Computer Engineering, University of Connecticut, Storrs (e-mail: {wez10003, shengli}@engr.uconn.edu).

Digital Object Identifier 10.1109/LCOMM.2011.072911.110981



Puncturing patterns of variable nodes 9 to 20

Code rate	Puncturing pattern
8/19	00000X000000
8/18	00000X00000X
8/17	00000X0000XX
8/16	00000X000XXX
8/15	00000X00XXXX
8/14	00000X0XXXXX
8/13	00000XXXXXXX
8/12	000X0XXXXXXX
8/11	X00X0XXXXXXX
8/10	X00XXXXXXXXX

"0" means unpunctured.
"X" means punctured.

(c)

Fig. 1. (a). Protograph for NB-RC-LDPC codes with rates 8/10, 8/11,. . . ,8/19. (b). The corresponding matrix representation of the designed protograph. (c). Puncturing patterns of variable nodes 9 to 20 for different rates.

wise) or only partially (bit-wise). References [4], [5] showed that generally bit-wise puncturing is better than symbol-wise puncturing, especially when q is large (say $q \geq 64$)). We here take a different methodology. With a small to moderate q, the benefit of bit-wise puncturing would degrade. With this in mind, we consider *structured* symbol-wise puncturing to simplify the design, yet without much performance compromise as shown in simulations.

Our puncturing design stems from the following theorem.

Theorem 1: For a cycle of length $2k$ formed by k degree-2 nodes, as shown in the following matrix

$$\tilde{\mathbf{H}}^c = \begin{bmatrix} \alpha_1 & 0 & 0 & \dots & \beta_k \\ \beta_1 & \alpha_2 & 0 & \dots & 0 \\ 0 & \beta_2 & \alpha_3 & \dots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & \beta_{k-1} & \alpha_k \end{bmatrix}, \qquad (1)$$

where α_is and β_is are from GF(q), when $k-1$ columns are punctured and no other nodes connected to the k check nodes are punctured, then it takes $\lfloor k/2 \rfloor$ steps for all the punctured nodes to obtain useful information in decoding and each step can recover 2 more nodes.

Particularly, when $k=2$ and $k=3$, $k-1$ punctured nodes are all 1 step recoverable (1-SR) [11]. When $k=4$ and three nodes are punctured, two punctured nodes are 1-SR and the other one is 2-SR. In this paper, we apply **Theorem 1** to the code's mother matrix.

B. Base Code and Puncturing

As pointed out in [2], [3] too much puncturing will induce noticeable performance loss at high rates. For this sake we choose $8/13$ as the rate of the base code. The protograph of the base code is shown in Fig. 1 (a) which involves only variable nodes 1 to 13 and check nodes 1 to 5. The base matrix is of size 5×13 and is shown in the upper-left part of Fig. 1 (b) where the last five columns correspond to parity symbols and the first eight columns correspond to information symbols.

We follow the following steps when designing the base matrix. We first design the right-most 5×5 square sub-matrix corresponding to the 5 parity check nodes. We have taken the puncturing pattern of the rate-$8/10$ code into consideration. This sub-matrix is set to be in a block-triangular form, consisting of one length-4 cycle and one length-6 cycle. The rate-$8/10$ code is obtained by puncturing one node from the length-4 cycle (column 9) and two nodes (columns 12 and 13) from the length-6 cycle. Following **Theorem 1**, all the punctured nodes are 1-SR for the rate-$8/10$ code. Also the lower-triangular form of this sub-matrix enables linear-time encoding in a partially parallel manner [12]. Secondly the left 5×8 sub-matrix corresponding to the 8 information nodes is obtained by optimizing the decoding performance of the rate-$8/10$ code and the rate-$8/13$ base code through numerical simulations. The candidate column weight is 2, 3 or 4. For each tentative degree profile, the left 5×8 sub-matrix is obtained by the PEG algorithm.

Then the puncturing patterns for codes of rate $8/11$ and $8/12$ are obtained by optimization using numerical simulations. For example, the puncturing pattern for rate $8/11$ can be represented by three parameters, f_1, f_2, and f_3 which represent the fraction of punctured nodes corresponding to column 9, 12, and 13, respectively and $f_1 + f_2 + f_3 = 2$. With a 2 times of PEG lifting each entry 1 in the base matrix is replaced by a 2×2 matrix. Then f_i can be either 0, 0.5, or 1 (This can be generalized to any size of lifting). After optimization we find that the rate-8/11 code is obtained by puncturing nodes 9 and 12 completely. The same procedure is used for the design of rate-$8/12$ codes, and this code is obtained by puncturing the node 12 completely. Note that all the punctured nodes in these two codes are also 1-SR.

With a 2 times of PEG lifting, we are able to further construct a rate-$8/9$ punctured code. Specifically, after PEG lifting the length-4 cycle extends to a length-8 cycle and the length-6 cycle extends to a length-12 cycle. Applying **Theorem 1** to the two extended cycles, based on the rate-$8/10$ punctured code we can further puncture half of the nodes 10 and 11. In this case we get a rate-$8/9$ code.

C. Precoding and Extending

To obtain codes of rates 8/14, . . . , 8/19 from the rate-8/13 base code, six levels of extending are required. For extending we simply use cascading to construct more parity check symbols. To have the parity check symbols of higher-rate

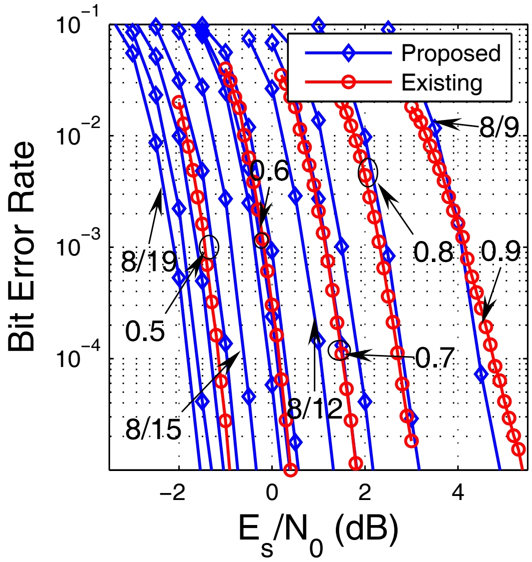


Fig. 2. Performance of structured NB-RC-LDPC codes.

codes embedded in that of extended low-rate codes, extending is done in a way by maintaining the lower-triangular structure (as shown in Fig. 1 (b)). To improve code performance of low rates, the information node 8 is precoded into the variable 14 which is then punctured (shown as dotted lines in Fig. 1 (a)).

At each level of extending one new variable node and one new check node are added to the protograph, and the number of connections between them can be either 1 or 2. Further, for each of the extended parity check nodes, one connection to a variable node of the base code (shown as bold solid lines in Fig. 1 (a)) exists to establish the dependencies. Further each extended parity check node can be connected to the precoded variable node 14 to further complement these dependencies. Each level of extending is optimized by numerical simulations. The optimized protograph is shown in Fig. 1 (a). Its corresponding matrix representation and puncturing patterns are shown in Fig. 1 (b) and Fig. 1 (c), respectively.

Reference [13] showed that for large Galois fields (e.g., $q = 256$), and very low rates (less than 1/3), simple multiplicative repetition is a good choice for extending. In contrast to [13] where block-wise repetition is used, here we use symbol-wise repetition with possible accumulation and one more connection to a precoded symbol can be incorporated to enhance the dependencies.

III. SIMULATION RESULTS

After ACE lifting, the nonzero entries in each circulant permutation sub-matrix of size 7 follow the structure in [12] such that the obtained codes are qualified as nonbinary QC-LDPC codes. Monte Carlo simulations are performed to evaluate the proposed codes. Binary input additive white Gaussian noise (BI-AWGN) channel is used and Q-ary FFT-based sum-product algorithm is used for decoding nonbinary LDPC codes with a maximum number of iterations set to 50.

Fig. 2 shows the simulation results of the proposed construction (denoted as ‘Proposed’). For the purpose of comparison, we have also constructed a rate-0.7 code by reducing the number of punctured symbols in the ACE lifted rate-$8/11$ code by 6 and a rate-0.6 code by adding half of the punctured precoded symbols 14 to the ACE lifted rate-$8/13$ code. In Fig. 3 of [4] NB-RC-LDPC codes over GF(64) of rates 0.5, 0.6, 0.7, 0.8 and 0.9 with the same information length are reported where bit-wise puncturing is adopted for intermediate rates (denoted as ‘Existing’ in Fig. 2). We can see that the proposed codes have similar performance as codes reported in [4]. We want to reiterate that compared with existing design, the proposed design is done in a structured way which can facilitate the implementation of high-speed encoder/decoder.

IV. CONCLUSIONS

In this letter we presented a novel design of *structured* nonbinary RC-LDPC codes using protograph and structured puncturing. Simulation results showed that the proposed codes achieve very good performance. One unique feature of the proposed design is that the obtained codes are qualified as nonbinary *quasi-cyclic* LDPC codes which are amenable to high-speed implementation of encoder/decoder.

REFERENCES

[1] J. Hagenauer, “Rate-compatible punctured convolutional codes (RCPC codes) and their applications,” *IEEE Trans. Commun.*, vol. 36, pp. 389–400, Apr. 1988.

[2] J. Li and K. Narayanan, “Rate-compatible low density parity check codes for capacity-approaching ARQ scheme in packet data communications,” in *Proc. CIIT*, 2002.

[3] M. Yazdani and A. Banihashemi, “On construction of rate-compatible low-density parity-check codes,” *IEEE Commun. Lett.*, vol. 8, no. 3, pp. 159–161, Mar. 2004.

[4] D. Klinc, J. Ha, and S. W. McLaughlin, “On rate-adaptability of nonbinary LDPC codes,” in *Proc. Intern. Conf. on Turbo Codes and Related Topics*, 2008.

[5] M. Gorgoglione, V. Savin, and D. Declercq, “Optimized puncturing distributions for irregular non-binary LDPC codes,” in *Proc. ISITA*, 2010.

[6] J. Huang, S. Zhou, and P. Willett, “Nonbinary LDPC coding for multicarrier underwater acoustic communication,” *IEEE J. Sel. Areas Commun.*, vol. 26, no. 9, pp. 1684–1696, Dec. 2008.

[7] D. Divsalar, S. Dolinar, C. Jones, and K. Andrews, “Capacity-approaching protograph codes,” *IEEE J. Sel. Areas Commun.*, vol. 27, no. 6, pp. 876–888, Aug. 2009.

[8] D. Klinc, J. Ha, and S. W. McLaughlin, “Optimized puncturing and shortening distributions for irregular nonbinary LDPC codes over the binary erasure channel,” in *Proc. Allerton Conf.*, 2008.

[9] X. Hu, E. Eleftheriou, and D. Arnold, “Regular and irregular progressive edge-growth tanner graphs,” *IEEE Trans. Inf. Theory*, vol. 51, no. 1, pp. 386–398, 2005.

[10] T. Tian, C. R. Jones, J. D. Villasenor, and R. D. Wesel, “Selective avoidance of cycles in irregular LDPC code construction,” *IEEE Trans. Commun.*, vol. 52, no. 8, pp. 1242–1247, Aug. 2004.

[11] J. Ha, J. Kim, D. Klinc, and S. W. McLaughlin, “Rate-compatible punctured low-density parity-check codes with short block lengths,” *IEEE Trans. Inf. Theory*, vol. 52, pp. 728–738, Feb. 2006.

[12] J. Huang, S. Zhou, L. Liu, and W. Zhou, “Nonbinary multiple-rate protograph codes,” *IEEE Trans. Commun.*, under review, 2011.

[13] K. Kasai, D. Declercq, C. Poulliat, and K. Sakaniwa, “Rate-compatible non-binary LDPC codes concatenated with multiplicative repetition codes,” in *Proc. ISIT*, 2010.