

Protograph-based q -ary LDPC Codes for Higher-Order Modulation

Andrea Marinoni¹, Pietro Savazzi¹, Richard D. Wesel²

¹ Università degli Studi di Pavia, Dipartimento di Elettronica, Via Ferrata 1, 27100, Pavia, Italy.

Email: {andrea.marinoni, pietro.savazzi}@unipv.it

² University of California, Los Angeles, Electrical Engineering Department, Los Angeles, CA 90095-1594, USA.

Email: wesel@ee.ucla.edu

Abstract— This paper introduces a protograph-based method for designing q -ary LDPC codes for use with modulations larger than QPSK. Simulations focus on a $GF(16)$, 16-QAM example. The proposed construction method achieves the maximum gain when the average column weight is chosen so that the linear minimum distance growth property is satisfied. In this region, the benefit of a protograph-based design over a standard PEG approach was 0.3 dB. We found that a careful field-element selection algorithm provides about 0.1 dB of improvement over random field-element selection. Overall, the proposed improvements yielded 0.4 dB of gain over a PEG-based $GF(16)$ code with randomly selected Galois field elements. The performance of this baseline $GF(16)$ code was comparable to the best known binary LDPC code for 16-QAM, so that the proposed improvements allow the $GF(16)$ LDPC code to outperform known binary approaches.

I. INTRODUCTION

In order to achieve the best possible error-rate performance, capacity-approaching codes such as Turbo-Codes (TC) [2] and Low-Density Parity-Check (LDPC) [1] codes have been adopted by a multitude of systems - with applications ranging from storage to optical communications.

LDPC codes [1] are algebraic codes characterized by a sparse parity-check (PC) matrix, H , having M rows and N columns. An LDPC code is either regular or irregular depending on their row and column degree-distributions. Regular LDPC codes have a PC matrix in which all rows (and columns) have equal weight, while irregular LDPC codes do not exhibit this property.

Non-binary (or q -ary) LDPC codes have codewords (and also a PC matrix) whose symbols are elements of the finite field $GF(q)$, with $q > 2$. These q -ary LDPC codes typically have steeper bit-error-rate waterfall curves, however the decoding complexity is $O(Ntq^2)$, where N is the blocklength, t is the average column weight, and q is the alphabet size [3], [4].

Using their bipartite graph representation, [5] and [12] showed that LDPC codes may perform very close to capacity on Additive White Gaussian Noise (AWGN) channels and achieve capacity on binary erasure channels. Therefore, it is natural to ask if LDPC codes can improve the bit-error-rate performance of a code in a communication system that has several requirements from high bandwidth efficiency to high coding rate.

This paper introduces a new method to optimize the performance of q -ary LDPC codes and applies it to high-bandwidth-efficiency transmission. The new method combines the protograph construction proposed in [13] for binary LDPC codes and the selection of non-zero entries provided in [20] and [21]. Specifically, the method builds PC matrices suitable to preserve the linear minimum distance growth property [13] and to maximize the row minimum distance as well.

For a variety of rates and average column weights, comparisons with q -ary LDPC codes constructed by using design algorithms from the current literature demonstrate the performance advantage of the proposed approach on the AWGN channel. The performance improvement varies as a function of average column weight, but can be as large as 0.4 dB.

The paper is organized as follows: Section II describes the system model. Section III introduces the new protograph-based q -ary LDPC code construction method. Section IV presents the simulation results. This section also discusses the practical aspects of the code construction and decoding. Section V delivers the conclusions.

II. SYSTEM MODEL

This paper uses LDPC codes over $GF(q)$ with q -ary modulation to achieve bandwidth-efficient transmission. For a chosen code rate R , and blocklength N , the LDPC code requires a PC matrix, $H = \{h_{ij}\}_{i=1,\dots,M,j=1,\dots,N}$, where $h_{ij} \in GF(q)$ and $R = 1 - \frac{M}{N}$.

In this manner, the $K = NR$ information symbols and the M parity symbols are encoded into a q -ary vector $\mathbf{x} \in GF(q)^N$. After q -ary LDPC encoding, the N elements of \mathbf{x} are mapped into the modulated sequence $\mathbf{s} = \{s_j\}_{j=1,\dots,N}$. This sequence of modulation symbols depends on the sequence of addresses given by $\mathbf{x}_b = \{\bar{x}_b^j\}_{j=1,\dots,N}$, where $\bar{x}_b^j = \{x_{bk}^j\}_{k=0,\dots,p-1}$ is the binary vector representation of the non-binary codeword symbol x_j . Therefore, for field size $q = 2^p$ with p an integer, the bandwidth efficiency of this structure is equal to Rp bits per channel use.

At the receiver, the output of the AWGN channel may be expressed as $y_\kappa = s_\kappa + n_\kappa = (s_{\kappa_I} + j s_{\kappa_Q}) + (n_{\kappa_I} + j n_{\kappa_Q}) = y_{\kappa_I} + j y_{\kappa_Q}$, where subscripts I and Q correspond to

the in-phase and quadrature components and $\kappa = 1, \dots, N$. The in-phase and quadrature noise components $n_{\kappa I}, n_{\kappa Q}$ are independent with the same variance σ^2 .

Applying Bayes' theorem [10] to $P(y_\kappa|s_\kappa)$, the *a posteriori* probability satisfies

$$P(s_\kappa|y_\kappa) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{(y_{\kappa I} - s_{\kappa I})^2 + (y_{\kappa Q} - s_{\kappa Q})^2}{2\sigma^2}\right). \quad (1)$$

These probabilities are used to initialize the message passing algorithm in the decoder [3]. The computational complexity of the algorithm provided by [3] may be reduced by employing the Fast Fourier Transform (FFT) or the Fast Hadamard Transform (FHT) approach [10].

III. PROTOGRAPH q -ARY LDPC CODES

This section presents a protograph-based design approach for LDPC codes using modulation larger than QPSK. Several papers [3], [4], [10], [14], [20] - [22] have investigated the optimization of q -ary LDPC codes. Among these papers, the primary goals have been the following: (i) improving the waterfall region; (ii) lowering the error floor; (iii) controlling the trade off between decoding complexity and error-rate performance.

For Galois fields with more than two elements, there has been limited work on optimizing the left and right LDPC degree distributions in the context of a code ensemble analysis, such as density evolution [18] has done for $GF(2)$. Most notable in this regard is the recent work of Urbanke [14] that provides degree distributions for field sizes up to 8.

The common design approach has begun with the design of a binary mother PC matrix using a binary degree distribution obtained through density evolution and a specific PC matrix satisfying that distribution obtained using well known algorithms such as [8] or [9]. This design approach then replaces each binary one in that matrix with a nonzero $GF(q)$ element and each binary zero with a $GF(q)$ zero.

A key issue is the choice of the distribution of the nonzero $GF(q)$ elements within a row. The $GF(q)$ elements chosen to replace each non-zero entry in the binary mother matrix typically have to fulfill the row minimum distance condition [20], [21]. Specifically, in [20] MacKay found several valid sequences for 16-ary and 64-ary LDPC codes using a Monte Carlo approach, choosing $GF(q)$ values in order to maximize the marginal entropy of the syndrome related to the output sequences. In [21] the authors proposed a method based on the binary images of each element living in the Galois Field $GF(q)$, considering therefore the binary row minimum distance to be optimized and providing results for 64-ary and 256-ary LDPC codes.

In this paper, we provide a new design technique for q -ary LDPC codes that maximizes the row minimum distance by using proper sequences of q -ary elements in a protograph-based binary PC mother matrix [13], [16]. Our method aims to improve the waterfall region for q -ary LDPC codes. It does not address the decoding complexity of q -ary LDPC codes. The

next subsection introduces the protograph-based construction scheme. Following that, the procedure to properly choose the non-zero entries in the PC matrix is described.

A. Protograph LDPC codes

Several papers have investigated the design of LDPC codes with imposed substructures, from multi-edge-type codes [19] to quasi-cyclic (QC) codes [17]. Protograph-based codes are structured codes as well [16], [13].

A protograph is a Tanner graph with a relatively small number of nodes [16]. Given a set of variable nodes $V = \{v_i\}_{i=1, \dots, |V|}$ and a set of check nodes $C = \{c_j\}_{j=1, \dots, |C|}$, each edge of the protograph has to connect a variable node v_i to a check node c_j ; parallel edges are permitted [16].

In order to obtain larger derived graphs of various sizes, a "copy and permute" operation can be applied to the protograph. This operation consists of first making G copies of the protograph and then permuting the endpoints of each edge among the G variable nodes and the G check nodes connected to the set of G edges copied from the same edge in the protograph. The derived graph is the graph of a code G times as large as the code corresponding to the protograph, with the same rate $R_P = 1 - \frac{|C|}{|V|}$ and the same distribution of variable and check node degrees as the protograph. In fact, the derived graph contains $G \cdot N_P$ transmitted nodes, and $G \cdot M_P$ check nodes.

The local neighborhood of a node in the derived graph is completely determined by the protograph [16]. The local neighborhood to depth d consists of all nodes and edges connected to a given node by a path of length d or less. This neighborhood is a tree if there is at most one path of length d or less to any other node. If the neighborhood of a variable node in the derived graph is a tree, the connections among the nodes are still determined by the protograph structure. As a result, density evolution analysis [18] can be applied on the protograph to determine whether or not decoding will yield arbitrarily small bit-error probability on a large derived graph.

In [13], the authors discuss protograph codes that benefit from both capacity-approaching thresholds and linear minimum distance growth. They provide methods to compute iterative decoding thresholds and asymptotic ensemble weight enumerators for protograph-based LDPC codes as well.

The analysis they provide aims to compute the ensemble weight enumerator for an LDPC code ensemble built from a protograph. The normalized weight distribution is used to obtain an upper bound on the threshold of E_b/N_0 when the code ensemble is used on an AWGN channel with maximum-likelihood (ML) decoding and to determine whether or not the minimum distance of typical codes in the ensemble increases linearly with the code length. Thus, the asymptotic ensemble weight enumerator is used to determine whether the code ensemble achieves linear minimum distance growth, i.e. whether the minimum distance of most codes in the ensemble increases linearly with the blocklength.

Also in [13], the authors provide design methods that guarantee code ensembles constructed from certain protographs

have linear minimum distance growth. This property holds for protographs having all variable node degrees equal to 3 or higher. However, it is known from the analysis in [11] and the results provided in [10] that good iterative decoding thresholds for LDPC codes can require a substantial fraction of degree-2 variable nodes. To resolve this conflict, methods in [13] allow the addition of degree-2 and degree-1 nodes to improve the iterative decoding threshold while preserving the linear minimum distance growth property. Specifically, the “check node splitting technique” in [13] accomplishes this.

In this paper, we use the analysis provided in [13] to construct protograph LDPC codes that have the linear minimum distance growth property. We derive a binary mother matrix from a protograph having the linear minimum distance growth property by using the “copy and permute” operation.

Let P be a protograph containing only transmitted variable nodes having degree equal to 2 or higher and let S_P be the subgraph of P containing only its degree-2 variable nodes and their attached edges and checks. In case the subgraph is not connected, decompose it into its disjoint connected pieces $S_P(j)$. Thus, we can write $S_P = \bigcup_j S_P(j)$. Each connected subgraph $S_P(j)$ has n_j degree-2 variable nodes. The protograph P satisfies the check node splitting condition (that is, has the minimum distance that grows linearly with the blocklength) if each subgraph $S_P(j)$ involves at least $n_j + 1$ check nodes. Verifying this condition provides binary mother matrices suitable to be optimized with a proper choice of the q -ary elements, that will be introduced in the next subsection.

B. Selecting the q -ary elements in the PC matrix

In this subsection we consider a method to optimize the error-rate performance of a q -ary LDPC code having a binary mother PC matrix constructed as in subsection III-A. The optimization of q -ary LDPC codes has been studied since their introduction in the late 90’s [3], [4]. MacKay [20] and Poulliat et al. [21] have addressed the problem of the proper selection of specific $GF(q)$ elements to replace the ones in a given binary mother PC matrix.

In [20], the author considered the PC matrix as constructed using the algorithm presented in [8]. He examined the marginal entropy of a single element of the syndrome vector for each choice of the κ non-zero entries in a row of the PC matrix. He chose this metric because if the entropy of the syndrome increases, then an optimal decoder can get closer to the Shannon limit.

For a specified number κ of ones in a row of the binary mother PC matrix, the entropy-maximizing algorithm of [20] provides a set of κ distinct field elements that can be used in any order to replace the κ ones in that row. In [20] results are shown for LDPC codes over $GF(16)$ and $GF(64)$ having 4 and 5 non-zero elements per row.

In [21], the authors propose a method for selecting the specific $GF(q)$ elements to replace the ones in a binary mother PC matrix constructed by using either progressive edge growth (PEG) [8] or the ACE algorithm [9]. This optimization scheme is based on the binary image representation of the code and it

aims to improve the waterfall region. It intends to lower the computational cost of the method proposed in [20] for high order fields.

The idea behind that scheme is as follows: the higher the minimum distance computed on the binary image, the more distinguishable (that is, reliable) the messages involved in the Message Passing (MP) algorithm. Therefore, the algorithm seeks to maximize the minimum distance of the associated binary code, selecting the κ -tuples with the maximum minimum distance and, among them, those with the smallest weight enumerator coefficient [21], [22].

By comparing the results of both the aforesaid methods, the authors in [21] observed that the sets they obtained may include the sets given in [20]. Specifically, they found that the method proposed in [20] could provide the set of κ -tuples with the smallest weight enumerator coefficient for $GF(16)$ and for $GF(64)$ too, if κ is set to 5. On the other hand, the method of Poulliat et al. could optimize rows, while MacKay’s is too computationally intensive for larger field sizes.

We propose to select the non-zero entries as follows for a given binary mother PC matrix derived from a protograph as in subsection III-A. Let H_b be the binary mother PC matrix derived from a protograph by using the “copy and permute” operation. H_b has N columns and M rows: each element of H_b lives in $GF(2)$. Let $\Omega(\kappa|q)$ be the set of length- κ sequences $\Omega_j(\kappa|q)$ (where $j = 1, \dots, |\Omega(\kappa|q)|$) having maximum minimum distance computed on the binary image of the related q -ary LDPC code.

To obtain the optimized PC matrix of the q -ary LDPC code, we consider each row of H_b . Then, we replace the sequence of d_{c_i} non-zero entries of H_b with a d_{c_i} -tuple picked from the set $\Omega(d_{c_i}|q)$ provided in [20] and [21], where d_{c_i} represents the degree of the i -th check node, $i = 1, \dots, M$.

In case the weight of a given row did not match the length of the provided sequences, we use the following procedure. As we previously mentioned in this subsection, the i -th row of the PC matrix of the q -ary LDPC code can be a permutation or multiplication by a constant of d_{c_i} -tuples of the set $\Omega(d_{c_i}|q)$. Therefore, each element $\Omega_j(d_{c_i}|q) \in \Omega(d_{c_i}|q)$ can be written as an ordered sequence as $\Omega_j(d_{c_i}|q) = \alpha_r^v = [\alpha^r, \alpha^s, \dots, \alpha^u, \alpha^v]$ where α is the primitive element of the Galois field $GF(q)$, $|r - v| = d_{c_i}$, $r < s < \dots < u < v$ and $\{r < s < \dots < u < v\} \in \{0, \dots, q - 2\}$.

Then, we can derive a $(d_{c_i} + 1)$ -tuple from $\Omega(d_{c_i}|q)$ by adding an element [21] as $\Omega_l(d_{c_i} + 1|q) = [\Omega_j(d_{c_i}|q), \alpha^a] = [\alpha_r^v, \alpha^a]$ where $v < a$, $a \in \{0, \dots, q - 2\}$, $j = 1, \dots, |\Omega(d_{c_i}|q)|$, $l = 1, \dots, |\Omega(d_{c_i} + 1|q)|$. We call the matrix we get after this operation H_r .

To completely define the PC matrix of the q -ary LDPC code, we apply a law that multiplies every non-zero element in each row of H_r by a constant. We apply the law $L(\gamma_i)$ that multiplies each element in the i -th row for a factor α^{γ_i} , as follows: $L(\gamma_i) : \alpha^{u_j} \mapsto \alpha^{u_j} \cdot \alpha^{\gamma_i}$ where $\{u_j, \gamma_i\} \in \{0, \dots, q - 2\}$, $j = 1, \dots, d_{c_i}$ and α represents the primitive element of $GF(q)$.

In order to avoid the occurrence of cycles in the PC matrix,

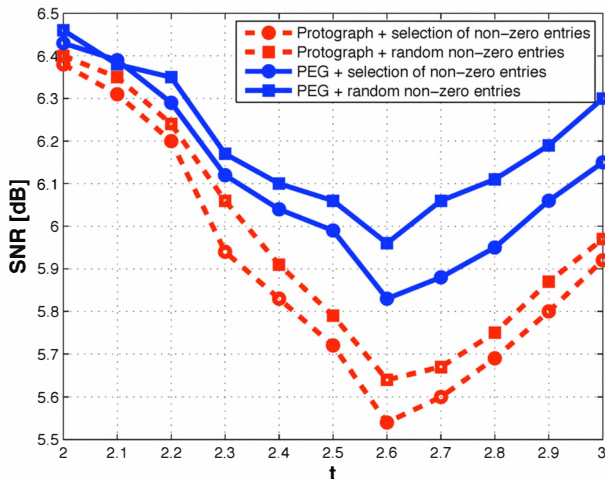


Fig. 1. SNR at which each considered q -ary LDPC code has $\text{BER}=10^{-5}$ for different values of the average column weight t . The codes taken into account are PEG-based and protograph-based, with random or careful selection of the non-zero entries in the PC matrix.

$L(\gamma_i)$ has to satisfy the Full Rank Condition (FRC) [21]. Therefore, the value of γ_i for each $L(\gamma_i)$ has to be chosen such that, given two rows k and l having the same degree $d_{c_k} = d_{c_l}$, $\gamma_k \neq \gamma_l$. Finally, once $L(\gamma_i)$ has been applied to all the rows, we have the PC matrix H for our q -ary LDPC code: $H = \{h_{ij}\}_{i=1,\dots,M;j=1,\dots,N}$, $h_{ij} \in GF(q)$.

IV. SIMULATION RESULTS

This section presents simulation results obtained by implementing 16-ary modulation using 16-ary LDPC codes. In each implementation, we use a symmetric ultracomposite [6] Gray-labeled 16-QAM modulation with a bandwidth efficiency of 2 bits/symbol (i.e. a coding rate R equal to 0.5). For each 16-ary LDPC code we considered, the blocklength N was set to 2500 symbols (10000 bits). In this paper we consider Quasi-Regular (QR) LDPC codes [10], [12].

We designed and simulated codes using four approaches for a variety of average column weights. First, we construct two binary mother PC matrices: one by using the Progressive Edge-Growth (PEG) algorithm [8] and one by using the protograph-based algorithm introduced in subsection III-A. For the protograph-based PC matrix, the number of transmitted variable nodes N_P of the protograph has been set to 10, while the number of check nodes M_P of the protograph has been set to 5. The protograph is copied and permuted 250 times to produce the 2500-symbol LDPC code. For both the PEG and the protograph-based PC matrices we chose the specific $GF(q)$ elements to replace the ones in the binary mother PC matrix both randomly and by using the selection method introduced in subsection III-B. The primitive polynomial of the considered 16-ary Galois Field is $p(x) = 1 + x + x^4$.

Figure 1 plots the SNR required to achieve a $\text{BER}=10^{-5}$ for codes with a variety of average column weights t . Performance of the codes whose binary mother PC matrix was

constructed using the PEG algorithm are plotted as blue solid lines, while performance of those constructed by using the protograph-based algorithm introduced in subsection III-A are plotted as red dashed lines. Performance of codes produced by the random selection of $GF(q)$ elements are identified by a square marker, and performance of codes produced by careful selection of the non-zero entries according to III-B are identified by a circle marker.

This figure shows how the linear minimum distance growth property, which holds for $t > 2.6$, influences the performance of the protograph-based codes. As t decreases from 2.6 to 2 the benefit provided by the protograph-based approach over the PEG approach for the same (either random or selected) $GF(q)$ element selection algorithm decreases from about 0.3 dB to less than 0.05 dB. For values of t above 2.6 the benefit provided by the protograph-based design is relatively constant. Also, Figure 1 clearly shows that the performance is best for $t = 2.6$, when it is just large enough to provide the linear minimum distance growth [13].

Let us look more closely at the $t = 2.6$ design, which has the variable-node degree distribution [11] $\lambda_2 = 0.4$ and $\lambda_3 = 0.6$. In that case, the binary adjacency matrix H_P of the protograph is as follows:

$$H_P = \begin{bmatrix} 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \end{bmatrix} \quad (2)$$

Figures 2 and 3 show the bit-error-rate (BER) and frame-error-rate (FER) performance respectively from our simulation results for $t = 2.6$ codes. The proposed q -ary LDPC code constructed by the combination of the protograph-based mother code construction of subsection III-A and the $GF(q)$ selection algorithm of section III-B outperforms the other codes. By itself, the selection of the non-zero entries according to III-B provides about 0.1 dB of gain in terms of Signal-to-Noise Ratio (SNR) w.r.t. the random selection for both the PEG-based and the protograph-based LDPC codes. By itself, the protograph-based mother code construction of subsection III-A outperforms the PEG-based non-protograph construction algorithm by about 0.3 dB for either $GF(q)$ selection algorithm. Overall, the benefit of a protograph-based design and the $GF(q)$ selection algorithm of section III-B provides a 0.4 dB benefit.

In order to compare the proposed construction method to the existing state of the art of the bandwidth-efficient coded modulation schemes, in Figure 2 the performance of the optimized LDPC coded structure on a 4-PAM modulation exposed in [7] have been plotted as a black dash-dot line. The architecture proposed in [7] is based on a multilevel coding (MLC) approach and it uses a multistage decoding (MSD) system. Further, it has a coding rate equal to 1/2 and an input blocklength fixed to 5000 bits. The protograph-based LDPC codes proposed in this paper outperform that proposed in [7]

by about 0.4 dB. Moreover, the 16-ary PEG-based LDPC code with selected field elements outperforms by about 0.15 dB the structure proposed in [7].

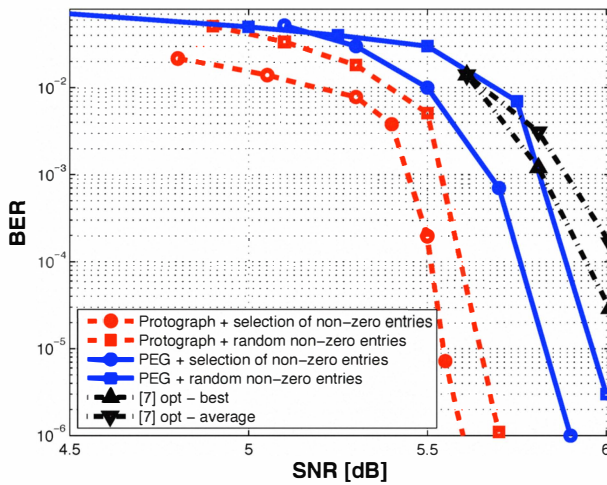


Fig. 2. Bit Error Rate performance of the considered q -ary LDPC codes on the AWGN channel: PEG-based and protograph-based, with random or careful selection of the non-zero entries in the PC matrix. The channel capacity is 4.77 dB. Average and best BER performance of the optimized (“opt”) LDPC coded structure proposed in [7] are provided as well.

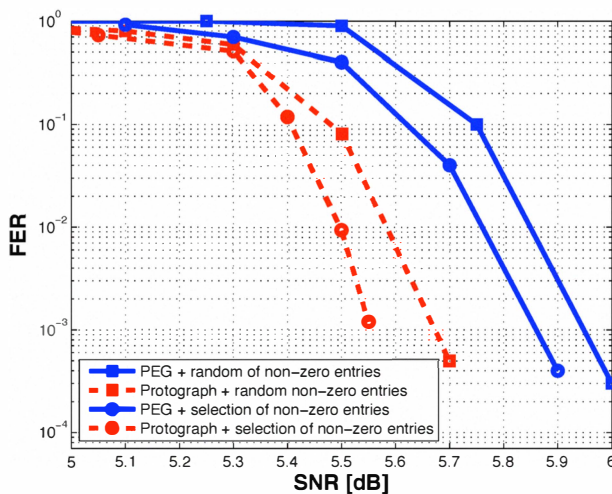


Fig. 3. Frame Error Rate performance of the considered q -ary LDPC codes on the AWGN channel: PEG-based and protograph-based, with random or careful selection of the non-zero entries in the PC matrix. The channel capacity is 4.77 dB.

V. CONCLUSION

This paper introduced a protograph-based method for designing q -ary LDPC codes for use with modulations larger than QPSK. Simulations focused on a $GF(16)$, 16-QAM example. The proposed construction method achieves the maximum gain when the average column weight is chosen so that the linear minimum distance growth property is satisfied. In this region,

the benefit of a protograph-based design over a standard PEG approach was 0.3 dB. For selection of the specific Galois field elements, we found that a careful selection algorithm provides about 0.1 dB of improvement over a random selection algorithm.

REFERENCES

- [1] R.G. Gallager, *Low-Density Parity-Check Codes*, Cambridge, MA, MIT Press, 1963.
- [2] C. Berrou, A. Glavieux, P. Thitimajshima, “Near Shannon limit error-correcting coding and decoding: Turbo codes,” in *proc. IEEE International Conference on Communications (ICC) 1993*, Geneva, CH, pp. 1064-1070.
- [3] M.C. Davey, D. MacKay, “Low-Density Parity-Check Codes over $GF(q)$,” *IEEE Communications Letters*, vol. 2, no. 6, June 1998.
- [4] M.C. Davey, “Error-Correction Using Low-Density Parity-Check Codes,” Ph.D. Thesis, University of Cambridge, UK, December 1999.
- [5] N. Wiberg, “Code and decoding on general graphs,” Dissertation no. 440, Dept. Elec. Eng., Linköping Univ., Linköping, Sweden, 1996.
- [6] R.D. Wesel, X. Liu, C. Komminakis, and J.M. Cioffi, “Constellation Labeling for Linear Encoders,” *IEEE Transactions on Information Theory*, vol. 47, no. 6, Sept. 2001, pp. 1572-1581.
- [7] J. Hou, P.H. Siegel, L.B. Milstein, H.D. Pfister, “Capacity-Approaching Bandwidth-Efficient Coded Modulation Schemes Based on Low-Density Parity-Check Codes,” *IEEE Trans. Inform. Theory*, vol. 49, no. 9, Sept. 2003.
- [8] X.Y. Hu, E. Eleftheriou, D.M. Arnold, “Regular and Irregular Progressive Edge-Growth Tanner Graphs,” *IEEE Trans. on Information Theory*, vol. 51, no.1, Jan. 2005.
- [9] T. Tian, C. Jones, J. Villanor, R.D. Wesel, “Selective avoidance of cycles in irregular LDPC code construction,” *IEEE Trans. on Comm.*, Aug. 2004, vol. 52, no. 8, pp. 1242-1247.
- [10] B. Rong, T. Jiang, X. Li, M.R. Soleymani, “Combine LDPC Codes Over $GF(q)$ With q -ary Modulations for Bandwidth Efficient Transmission,” *IEEE Trans. on Broadcasting*, vol. 54, no. 1, March 2008.
- [11] T. Richardson, A. Shokrollahi, R. Urbanke, “Design of Capacity-Approaching Irregular Low-Density Parity-Check Codes,” *IEEE Trans. Inform. Theory*, vol. 47, no. 2, Feb. 2001.
- [12] T. J. 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, Feb. 2001.
- [13] D. Divsalar, S. Dolinar, C. Jones, K. Andrews, “Capacity-Approaching Protograph Codes,” *IEEE Journal on Selected Areas of Communications*, vol. 27, issue 6, pp. 876-888, Aug. 2009.
- [14] V. Rathi, R. Urbanke, “Density evolution, thresholds and the stability condition for non-binary LDPC codes,” in *IEE Proceedings - Communications*, vol. 152, num. 6, 2005, p. 1069-1074
- [15] X.Y. Hu, M. Fossorier, E. Eleftheriou, “On the Computation of the Minimum Distance of Low-Density Parity-Check Codes,” *IEEE International Conference on Communications 2004*, vol. 2, pp. 767-771, June 2004.
- [16] J. Thorpe, “Low-Density Parity-Check (LDPC) Codes constructed from Protographs,” JPL IPN Progress Report 42-154, Aug. 15, 2003.
- [17] M. Fossorier, “Quasi-Cyclic Low-Density Parity-Check codes from circulant permutation matrices,” *IEEE Trans. on Inf. Theory*, vol. 50, no. 8, pp. 1788-1793, Aug. 2004.
- [18] S.Y. Chung, “On the construction of some capacity-approaching coding schemes,” Ph.D dissertation, Massachusetts Institute of Technology, Cambridge, MA, Sept. 2000.
- [19] T. Richardson, “Multi-Edge Type LDPC codes,” presented at the Workshop honoring prof. R. McEliece on his 60th birthday, California Institute of Technology, Pasadena, CA, May 24-25, 2002.
- [20] D. MacKay, “Optimizing Sparse Graph Codes over $GF(q)$,” available at <http://www.inference.phy.ca.ac.uk/mackay/CodesGallager.html>, Aug. 2003.
- [21] C. Poulliat, M. Fossorier, D. Declercq, “Using binary images of non binary LDPC codes to improve overall performance,” in *proc. 4th International Symposium on Turbo-codes and related topics*, April 2006, Munich, Germany.
- [22] C. Poulliat, M. Fossorier, D. Declercq, “Design of regular $(2, d_c)$ -LDPC codes over $GF(q)$ using their binary images,” *IEEE Trans. Comm.*, vol. 56, pp. 1626-1635, Oct. 2008.