Construction of Regular Quasi-Cyclic Protograph LDPC codes based on Vandermonde Matrices

Nicholas Bonello, Sheng Chen, and Lajos Hanzo
School of ECS, University of Southampton, SO17 1BJ, United Kingdom.
Tel: +44-23-8059 3125, Fax: +44-23-8059 4508
Email: {nb06r,sqc,lh}@ecs.soton.ac.uk, http://www-mobile.ecs.soton.ac.uk

***Abstract*— In this contribution, we investigate the attainable performance of quasi-cyclic (QC) protograph Low-Density Parity-Check (LDPC) codes for transmission over both Additive White Gaussian Noise (AWGN) and uncorrelated Rayleigh channels. The codes presented are constructed using the Vandermonde matrix and benefit from both low-complexity encoding and decoding, low memory requirements as well as hardware-friendly implementations. Our simulation results demonstrate that the advantages offered by this family of QC protograph LDPC codes accrue without any compromise in the attainable Bit Error Ratio (BER) and Block Error Ratio (BER) performance. In fact, it is also shown that despite their implementational benefits, the proposed codes exhibit slight BER/BLER gains when compared to some of their more complex counterparts of the same length.**

I. Introduction

Following more than three decades of neglect, Low Density Parity-Check (LDPC) codes [1], [2] are nowadays in the center of attention of the coding research community. This rekindled interest has been motivated by the outstanding performance demonstrated by turbo codes [3] which employ a similar soft-input soft-output iterative decoding strategy [4].

In the context of LDPC codes, the relationship between the information bits and the redundant parity-check bits is described by a sparse parity-check matrix (PCM) or by the corresponding bipartite Tanner graph [5]. The design of a LDPC code is characterized by a range of contradictory design factors, such as their Bit Error Ratio (BER), their mathematical construction attributes and their hardware complexity. Of prime concern is the BER performance exhibited by the code in both the 'waterfall' and 'error-floor' region. The mathematical construction attributes are related to the specific design of the PCM, which generally speaking, can be constructed in either a pseudo-random [2] or a structured manner [6] (see also the references in [6]). It has been shown that the former method [2], [7] exhibits excellent error-correction capabilities and thus is capable of operating close to the Shannon limit, especially for high codeword lengths. However, such codes typically exhibit complex hardware implementations due to the high-complexity descriptions, and generally, their encoding complexity grows quadratically (or slower [8]) with the block length.

In this paper, we will pursue a more holistic LDPC code design approach, and thus search for good LDPC codes, which strike an attractive tradeoff between the range of contradictory design factors. More explicitly, we investigate novel structured PCMs, which are designed based on Vandermonde-like block matrices [9]. The employment of Vandermonde block matrices was first proposed for classic Reed-Solomon codes and was also adopted for array codes by Fan [9]. Both Yang *et al.* [10] as well as Mittelholzer [11] investigated the minimum distance bounds of array codes, whilst the rank of various LDPC code constructions based on Vandermonde matrices was analytically determined by Gabidulin *et al.* in [12]. In [13], the authors constructed variable rate codes using Vandermonde-matrix based LDPC codes having rates compliant with the DVB-S2 standard.

The above-mentioned construction has the benefit of having a quasi-cyclic (QC) form [14] and thus significantly reduces the non-volatile memory-storage requirements. Additionally, the encoding procedure can be implemented with the aid of shift-registers, thus rendering the encoding complexity linear in the block length [15]. We further reduce the associated decoding complexity by invoking a so-called projected graph construction, which is also referred to as a 'protograph' by Thorpe [16]. As a benefit of imposing a structural regularity, these codes can be decoded by means of a semi-parallel architecture, as suggested by Lee *et al.* in [17], thus facilitating high-speed decoding.

Against this backdrop, the novel contribution of this paper is that we propose a PCM construction, which is based on Vandermonde-like block matrices for the first time in the context of protograph LDPC arrangements. This results in the implementation-related advantages of combining the benefits of having a low-complexity quasi-cyclic encoder structure with a readily parallelizable protograph decoder structure. More explicitly, the resultant quasi-cyclic protograph LDPC codes exhibit a low encoding and decoding complexity as well as reduced memory requirements, while facilitating hardware-friendly parallel implementations. We will compare our performance results to those attained by MacKay's codes [18] and to the codes generated using the Extended Bit-Filling (EBF) [19] as well as to the Progressive Edge-Growth (PEG) [20] algorithms. Simulation results are provided for both AWGN and uncorrelated Rayleigh (UR) channels. It is demonstrated that the achievable performance is comparable to or slightly better than that exhibited by the higher-complexity benchmarker codes of [2], [19], [20] having the same lengths.

The structure of this paper is as follows. Sections II and III introduce the basic principles of LDPC codes and the protograph codes' construction. Our discourse continues with a description of the Vandermonde matrix construction. The original PEG algorithm of [20] is then further developed in Section IV. Our simulation results are presented in Section V. Finally, Section VI is devoted to our conclusions.

II. Preliminaries

We consider a binary LDPC code defined by the null space of a low-density PCM matrix $\mathbf{H}$ constructed over GF(2). Then, assuming a full-rank PCM composed of M rows and N columns, the rate of this code becomes $R = 1 - M/N$. This can also be represented by means of a bipartite Tanner graph [5] consisting of M check nodes and N variable nodes. More explicitly, we consider a regular construction code having a uniform degree of edges emerging from each check and variable node. The variable and check nodes' degrees will be denoted by γ and ρ, which also correspond to the row and column weight of the PCM, respectively.

LDPC codes are typically decoded using the sum-product algorithm (SPA) [21], where messages are exchanged between the nodes residing at both sides of the graph. The independence of these messages is characterized by the length of the shortest cycle on the graph,

The financial support of the European Union under the auspices of the Newcom and Phoenix projects, as well as that of the EPSRC UK is gratefully acknowledged.

978-1-4244-1722-3/08/$25.00 ©2008 IEEE.

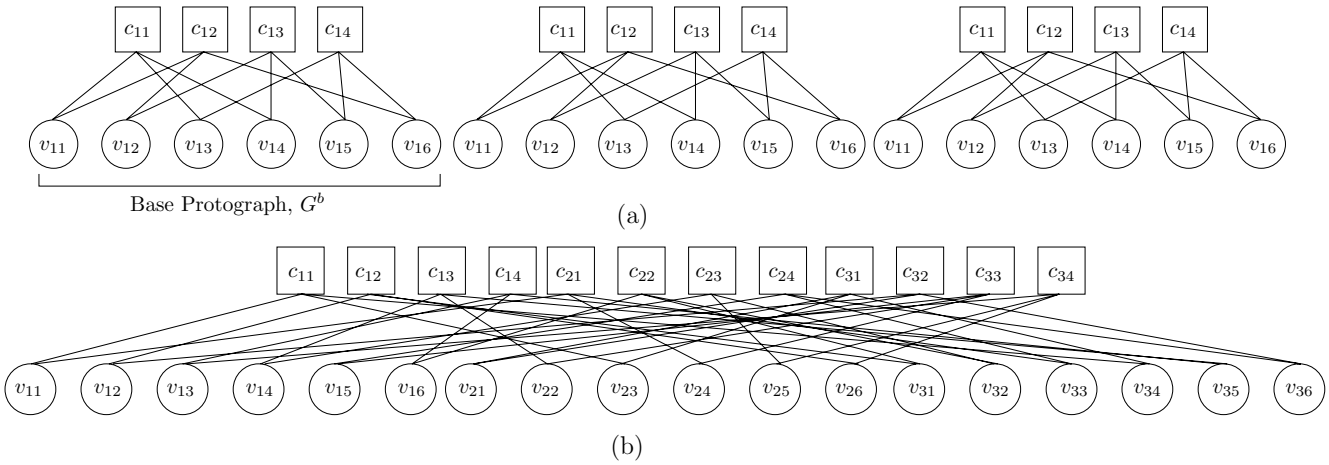


Fig. 1. (a) The base protograph is replicated by a factor J, in this case J = 3. (b) The construction of the derived graph is obtained by permuting the edges between the check and variable nodes of the J copies of the base protograph. The permutations are performed in a way so as to maximize the girth, whilst exhibiting a quasi-cyclic construction constrained by the Vandermonde-matrix based protograph.

which is typically referred to as the girth g. Specifically, Gallager demonstrated in [1] that the number of independent iterations T, i.e. the iterations that provide valuable extrinsic information and hence a useful iteration gain, is bounded by $T < g/4 \leq T + 1$. Clearly, for the girth to be high, the block length also has to be sufficiently high[1]. Furthermore, we only consider codes having $\gamma \geq 3$ and hence the resultant minimum distance grows linearly, instead of logarithmically, with the block length [1].

III. Protograph LDPC Code Construction

The construction of a protograph code, illustrated in Fig. 1, can be described in two main steps [16]:

1) Determine the base protograph, typically a graph with a relatively low number of nodes, and replicate this graph J times.
2) Permute the edges of the nodes in the J replicas of the base protograph in order to obtain the resultant graph.

Consider the base protograph, G^b, described by the set of check nodes $C^b = \left\{c_{ji} : j = 1; i = 1, \ldots, M^b\right\}$, the set of variable nodes $V^b = \left\{v_{ji} : j = 1; i = 1, \ldots, N^b\right\}$ and the set of edges E^b, where $\left|E^b\right| = M^b\rho = N^b\gamma$. We denote the number of check and variable nodes on the base protograph by M^b and V^b, respectively. The value of j = 1 refers to the base protograph. The base protograph will therefore have the corresponding base PCM of size $(M^b \times V^b)$. After replicating G^b by J times, we obtain the resultant graph of the protograph code, G', defined by the sets C', V' and E', where each set has a size, which is J times larger than the corresponding sets in the base protograph. The permutations of the nodes' edges in the graph derived obey certain constraints, which will be discussed in more detail in Section IV.

A. Vandermonde Matrix Based LDPC Code Construction

Since we want to impose a quasi-cyclic (QC) structure on our protograph code, we opt for constructing the QC base protograph from the Vandermonde matrix (VM) [9] construction. Let $\mathbf{I}_q$ represent a $(q \times q)$ identity matrix where q is either larger than the row as well as the column weight and it is a relative prime with respect to all the numbers less than ρ, or else obeys $q > (\rho - 1)(\gamma - 1)$. We also construct the permutation matrix $\mathbf{P}_q$, having elements of p_{mn}, $0 \leq m < q$ and $0 \leq n < q$, which is defined by [22]:

$$p_{mn} = \begin{cases} 1 & \text{if } m = (n-1) \bmod q, \\ 0 & \text{otherwise,} \end{cases} \tag{1}$$

[1]The loose lower bound on the required N was given by Gallager in [1].

where a mod b represents the modulus after division of a by b. For the sake of simplifying our analysis, we consider the example of q= 4, where the permutation matrices $\mathbf{P}_q$, $\mathbf{P}_q^2$ and $\mathbf{P}_q^3$ are given by:

$$\begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

and

$$\mathbf{P}_q^x = \begin{cases} \mathbf{I}_q & \text{if } x \bmod q = 0, \\ \mathbf{P}_q^{x \bmod q} & \text{otherwise.} \end{cases} \tag{2}$$

Then, the VM-based sparse PCM constructed for the base protograph is formulated by [22]:

$$\mathbf{H}^b = \begin{pmatrix} \mathbf{I}_q & \mathbf{I}_q & \mathbf{I}_q & \cdots & \mathbf{I}_q \\ \mathbf{I}_q & \mathbf{P}_q & \mathbf{P}_q^2 & \cdots & \mathbf{P}_q^{\rho-1} \\ \mathbf{I}_q & \mathbf{P}_q^2 & \mathbf{P}_q^4 & \cdots & \mathbf{P}_q^{2(\rho-1)} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{I}_q & \mathbf{P}_q^{(\gamma-1)} & \mathbf{P}_q^{2(\gamma-1)} & \cdots & \mathbf{P}_q^{(\gamma-1)(\rho-1)} \end{pmatrix}. \tag{3}$$

The PCM $\mathbf{H^b}$ of size $(\gamma q \times \rho q)$ will describe the null space for a base protograph LDPC code defined by the block length $N^b = \rho \times q$ and rate $R \geq 1 - \gamma/\rho$. The permutation matrix $\mathbf{P}_q$ is essentially constructed from an appropriate cyclic shift of the identity matrix $\mathbf{I}_q$. The restrictions imposed on the parameters q, ρ and γ ensure that no permutation matrix $\mathbf{P}_q^x$, $0 \leq x \leq (\gamma - 1)(\rho - 1)$, is repeated in the same row or column of the permutation matrices.

IV. Modifications of the Progressive Edge Growth Algorithm

The permutation pattern of the node's edges in the derived graph was determined using a modified version of the PEG algorithm. Whilst we still maintain the elegant characteristics of the PEG as regards to maximizing the girth of the graph and the minimum distance of the code [20], we impose two additional constraints. The first constraint ensures that the derived graph has the same structure as the base protograph whilst the second ascertains that the derived graph is also QC. The procedure that was used is summarized in Algorithm 1.

It can be observed from Fig. 1(b) that the permutations of the nodes' edges follow a particular pattern, which is governed by the PCM of the base protograph. For example, the edges emerging from the variable nodes v_{j1}, $j = 1, \ldots, 3$, are only connected to the check

Algorithm 1: The modified PEG.

input : M^b, N^b, J, q, γ
output: $C^{'ji}$ for j = 1,...,J and $i = 1, \ldots, M^b$, G'

```
Lines 2 - 21 determine the forbidden set of
check nodes based on the VM PCM of the base
protograph (Constraint 1).
for k^th variable node ← 1 to N^b J do
    k ← (k^th variable node) mod N^b, n ← 0, C'_tmp3 = ∅
    if k ≤ q then
        C'^ji = {c_ji : j = 1, ..., J; i = k, k + q, k + 2q, ..., M^b}
    else
        x ← (integer value of) [(k − 1)/q], r ← 1
        C'_tmp1 = {c_ji : j = 1, ..., J; i = n + 1}
        for y ← x to x(γ − 1),
        (step : y ← 2 × previous value of y) do
            C'_tmp2 =
            {c_ji : j = 1, ..., J; i = (rq + 1) + (n − y) mod q}
            C'_tmp3 = C'_tmp2 ∪ (previous C'_tmp3)
            r ← r + 1
        end
        C'^ji = C'_tmp1 ∪ C'_tmp3, conj(C'^ji) = C' \C'^ji
        if x > previous value of x then
            n ← 0
        else
            n ← n + 1
        end
        foreach c_ji ∈ conj(C'^ji) do Store the number of connections
        under the current graph construction and then set their
        number of connections to ρ
    end
    if j > 1 then
        Set the number of connections of the check nodes
        connected with variable nodes v_ji, with 1 ≥ j ≤ (current j)
        - 1 and i = k to ρ
    end
    Starting the modified PEG algorithm.
    for connection ← 1 to γ do
        if connection = 1 then
            Similar to PEG [20] with the chosen c_ji ∈ C'^ji
        else
            Similar to PEG [20] but the chosen c_ji ∈ C'^ji must
            have the lowest degree (under the current graph
            construction) and be the nearest to the selected c_j(i−1)
            for the same connection (Constraint 2).
        end
    end
    foreach c_ji ∈ C' do Restore the original number of
    connections.
end
```

nodes c_{ji} associated with $i = 1, 2$ and $j = 1, \ldots, 3$. This effectively imposes the structure of the base protograph on the graph derived. For each variable node v_{ji}, $j = 1, \ldots, J$ and $i = 1, \ldots, N^b$, we define the set of "allowed" checks $C^{'ji}$ and the set of "forbidden" checks by the complementary set $\overline{C^{'ji}} = C^{'} \backslash C^{'ji}$, i.e. the set of elements in $C^{'}$ but not in $C^{'ji}$. It is only necessary to calculate N^b different sets, since the sets repeat every N^b variable nodes. Then, for each v_{ji}, the algorithm selects that check node in the specific $C^{'ji}$ set having the lowest number of edges emerging from it under the current graph construction. On the other hand, we set the number of edges of every check node in $\overline{C^{'ji}}$ equal to ρ, which corresponds to the maximum number of connections a check node is allowed to have. In such manner, it is guaranteed that no connection between a variable node and a check node in the corresponding set $\overline{C^{'ji}}$ will be established.

However, by imposing only this constraint on the original PEG, the resultant graph will be acyclic (AC). This is due to the fact that the PEG [20] will *randomly* select the check nodes, if multiple choices are available. Therefore, we further restrict the algorithm to choose a check node $c_{ji} \in C^{'ji}$, which is the nearest to the previously selected $c_{j(i-1)}$, for the same connection. Since the base protograph was chosen to be QC, the algorithm is always capable of choosing that check node, which still retains the structural characteristics of the base, and so, the resulting protograph code will also be QC. This modification will lead to similar results to those attained by the QC-PEG proposed by Li *et al.* in [23], where in our case the 'QC-constraint' [23] is imposed by the base protograph PCM. When compared to the PEG algorithm, as originally proposed by Hu *et al.* [20], the modified algorithm is capable of reducing the size of the set of allowed checks from being governed by the binomial coefficient $\binom{N}{\gamma}$, $N = JM^b$, to $\binom{J\gamma}{\gamma}$.

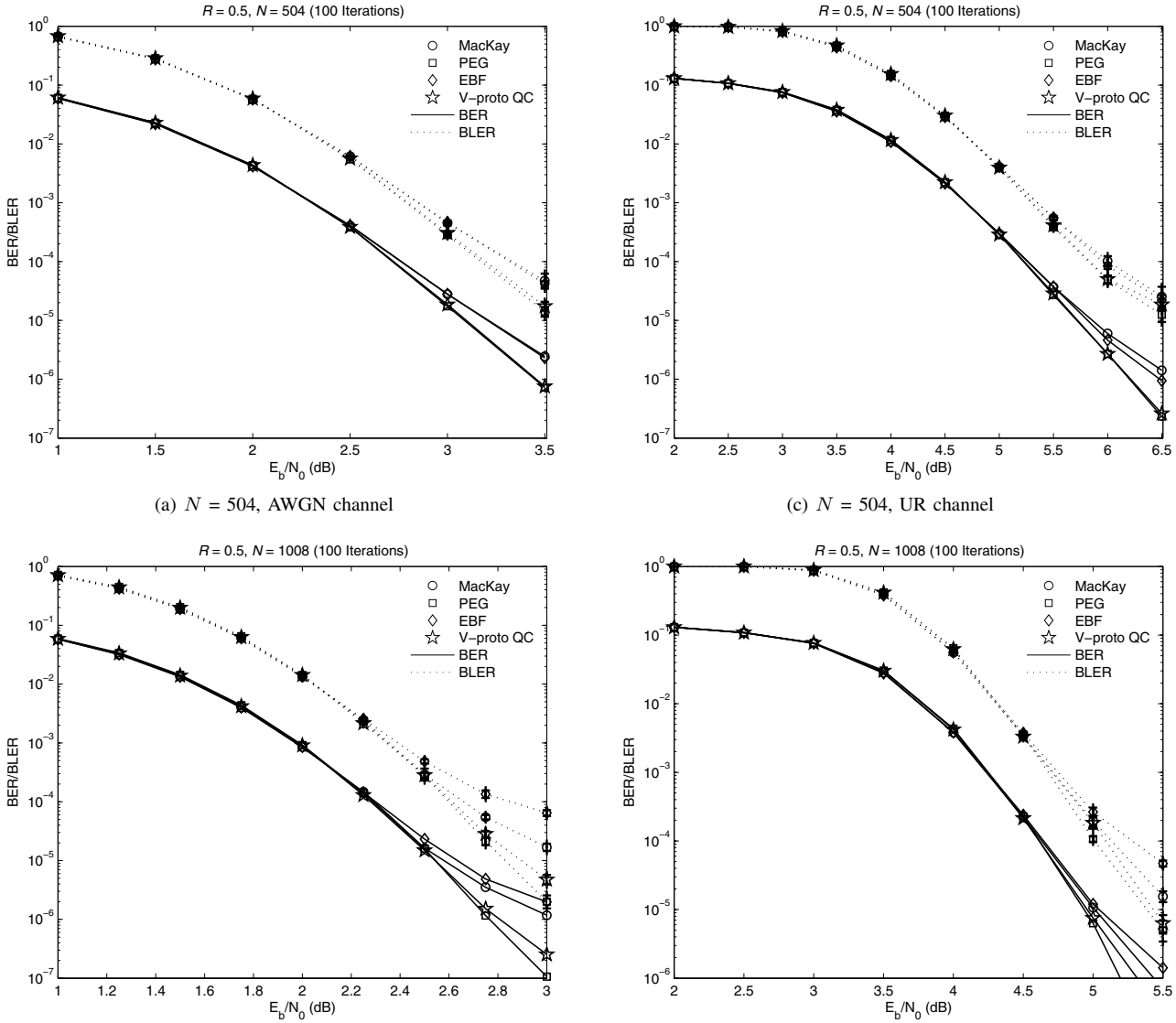
V. Results and Discussion

The results presented in this section were obtained using Binary Phase Shift Keying (BPSK) modulation, when transmitting over the AWGN and uncorrelated Rayleigh channel and using a maximum of I = 100 decoding iterations of the SPA. We will consider codes having γ = 3, a block length N ranging from 200 to 3060 and code rates R spanning from 0.4 to 0.8[2]. We compare both the achievable Block Error Ratio (BLER) and the BER performance for transmission over both AWGN and UR channels for four different code constructions, namely those of MacKay [18], the EBF [19], of the PEG [20] as well as of the proposed QC protograph codes. We will appropriately distinguish between the codes using the notation (N, K), where K represents the number of original information bits. The error bars shown on the BLER curves are associated with a 95% confidence level, and it was ensured that at least 100 block errors were collected at each point on the simulation curve.

The BLER and BER performance results over the AWGN channel recorded for the (504, 252) and (1008, 504) codes are illustrated in Figs. 2(a) and 2(b), respectively. The (504, 252) protograph codes were constructed from 12 replicas of Vandermonde matrix based protographs using q = 7. In a similar manner, 14 replicas of Vandermonde matrix based protographs having permutation matrix of size (12 × 12) were used for the protograph LDPC codes having a length of N = 1008.

It can be observed from Figure 2(b) that the proposed half-rate code having a block length of $N = 1008$, as well as parameters of $\gamma = 3$ and $\rho = 6$ attains a BER of 10^{-6} at 2.831 dB with a maximum of 100 decoder iterations, and therefore is only 2.643 dB away from the Shannon limit of 0.188 dB. At this BER, the performance of the QC protograph code is superior to that of the randomly generated MacKay code by about 0.2 dB. There is only 0.06 dB loss in the performance of the QC protograph code when compared to the significantly more complex, unstructured PEG code, which is deemed to have the best performance for transmission of short blocks over the AWGN channel, at the time of writing. This QC protograph code also exhibits a gain of about 0.157 dB over the corresponding QC half-rate code based on the Euclidean sub-geometry EG*(2,2^4) (*cf.* Table I in [14]) having a block length of N = 1020, and parameters $\gamma = 4$ and $\rho = 8$. Furthermore, this superior error correction performance is achieved at a lower decoding complexity due to the lower logic depth. The logic depth is defined by the value of ρ as well as γ and is directly related to the depth of the graph tree spreading from a variable node v_{ji}, $j = 1, \ldots, J$ and $i = 1, \ldots, N^b$.

[2]The row weights of the LDPC codes having rates 0.4, 0.5, 0.625 and 0.8 are 5, 6, 8 and 15, respectively.



(a) N = 504, AWGN channel

(c) N = 504, UR channel

(b) N = 1008, AWGN channel

(d) N = 1008, UR channel

Fig. 2. A BER and BLER performance comparison of R = 0.5 LDPC codes with N = 504 and N = 1008 and a maximum of $I = 100$ decoder iterations when transmitting over the AWGN and UR channels using BPSK modulation. Error bars shown on the BLER curves are associated with a 95% confidence level.

Similar BLER and BER performance trends were observed for the UR channel, as demonstrated in Figs. 2(c) and 2(d). For the sake of completeness, we also investigated the performance of QC protograph codes having rates of 0.4, 0.625 and 0.8 as well as both shorter and longer blocklengths. Our simulation results, which are not shown in this paper owing to space limitations, showed that the performance of the protograph codes is always comparable to that exhibited by the other benchmarker codes. A slight degradation was manifested by the QC protograph codes for high code rates and very short block lengths, because the constraints described in Section III-A could not be satisfied.

A. Encoder and Decoder Complexity

In this sub-section, we provide a more comprehensive comparison of the different code constructions that were considered by taking into account the encoder and decoder complexity. We employ a similar benchmarking technique to that used in [24], where the metrics used for comparison are based on an amalgam of the desirable encoder and decoder characteristics. The former include a low complexity description due to structured row-column connections and simple memory address generation (MAG), the linear dependence of the encoding complexity on the codeword length, and a hardware implementation based on simple components. As regards to attractive decoder characteristics, we are concerned with the reduction of MAG and on-chip wire interconnections, the reduced logic depth and the ability to use parallel decoding architectures for systolic-array type implementations. We also evaluate the decoder's computational complexity expressed in terms of the number of message-passing updates per decoded bit, which is given by $\Delta = \bar{i}|E'|/K$ [24], where $\bar{i}$ represents the average number of iterations required for finding a legitimate codeword at a particular E_b/N_0 value. A summary of these measures recorded for each code considered are summarized in Table I. It can be observed in Table I, that the encoder structure is quite complex for the majority of the five codes considered. Only the PEG and the QC protograph codes have linearly increasing encoding complexity as a function of the codeword length[3]. The QC protograph's encoder can also be implemented using a simple

[3]The PEG codes that were simulated cannot be decoded in linear-time, however, linear-time encoding for PEG codes is possible using "zigzag" [20] connections.

TABLE I

SUMMARY OF THE CHARACTERISTICS OF THE CODES CONSIDERED.

Complexity/Performance Criteria		MacKay	PEG	EBF	Proto QC
Desirable Encoder Characteristics	Simple description and MAG				■
	Complexity linear with N		■		■
	Simple Hardware Implementation				■
Desirable Decoder Characteristics	Reduced Logic Depth	■	■	■	■
	Simple parallel architecture				■
	Simple MAG and on-chip interconn.				■
$\Delta^{\dagger}$	AWGN at E_b/N_0 = 3 dB with I = 50	40	39	41	39
	UR at E_b/N_0 = 4.5 dB with I = 50	58	56	59	57

† The computational decoding complexity Δ (message updates/decoded bit) is measured for the (1008, 504) codes.

linear shift-register circuit of length K and therefore the encoder only requires $r(N-K)$ binary operations, where r is one less than the row weight of the generator matrix. By contrast, the remaining codes must be encoded by means of sparse matrix multiplications which require $(N-K)(2K-1)$ binary operations [25]. As far as the decoder's complexity is concerned, all the five code constructions score at least one point due to their low logic depth which accrues from using small values of ρ and γ. However, the lowest decoding complexity can only be attained using QC protographs codes. All the benchmarker codes suffer from having a high-complexity description due to the pseudo-random permutations. Therefore, their implementation still relies on inflexible hard-wired connections or on lookup tables that require a large amount of memory. By contrast, memory shifts corresponding to the QC PCM structure can be used to address the messages exchanged between the nodes of QC protograph. Several decoders for QC codes have been proposed, in particular that of Chen and Parhi [26], which is capable of doubling the decoding throughput (assuming a dual port memory), when compared to the decoding of randomly constructed codes, by overlapping the variable and check node updates.

VI. SUMMARY AND CONCLUSIONS

In this paper, we have proposed the construction of protograph LDPC codes based on QC Vandermonde matrices. These codes benefit from low-complexity encoding and decoding implementations due to their semi-parallel architectures. We investigated their BLER and BER performance for transmission over both AWGN and UR channels, for various rates and block lengths. Explicitly, our experimental results demonstrate that the performance of these protograph codes is similar to that exhibited by the higher complexity benchmarker codes. Therefore, it can be concluded that the advantages offered by the family of QC protograph LDPC codes accrue without any compromise in the attainable BLER and BER performance.

REFERENCES

[1] R. G. Gallager, "Low-density parity-check codes," *IRE Transactions Information Theory*, vol. 45, no. 2, pp. 21–28, Jan. 1962.

[2] D. J. C. MacKay, "Good error-correcting codes based on very sparse matrices," *IEEE Transactions on Information Theory*, vol. 45, no. 2, pp. 399–431, Mar. 1999.

[3] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding:turbo-codes. 1," in *Proc. International Conference on Communications, Geneva Technical Program*, vol. 2, Geneva, Switzerland, May 23–26, 1993, pp. 1064–1070.

[4] R. J. McEliece, D. J. C. MacKay, and J.-F. Cheng, "Turbo decoding as an instance of Pearl's belief propagation algorithm," *IEEE Journal on Selected Areas in Commununications*, vol. 16, no. 2, pp. 140–152, Feb. 1998.

[5] R. M. Tanner, "A recursive approach to low complexity codes," *IEEE Transactions on Information Theory*, vol. 27, no. 5, pp. 533–547, Sept. 1981.

[6] J. M. F. Moura, J. Lu, and H. Zhang, "Structured low-density parity-check codes," *IEEE Signal Processing Magazine*, vol. 21, no. 1, pp. 42–55, Jan. 2004.

[7] S.-Y. Chung, G. D. J. Forney, T. J. Richardson, and R. Urbanke, "On the design of low-density parity-check codes within 0.0045 db of the Shannon limit," *IEEE Communications Letters*, vol. 5, no. 2, pp. 58–60, Feb. 2001.

[8] T. Richardson and R. Urbanke, "Efficient encoding of low-density parity check codes," *IEEE Transactions on Communications*, vol. 47, no. 6, pp. 808–821, Feb. 2001.

[9] J. L. Fan, "Array codes as low density parity check codes," in *Proc. 2^{nd} International Symposium on Turbo Codes*, vol. 3, Brest, France, 2000, pp. 543–546.

[10] K. Yang and T. Helleseth, "On the minimum distance of array codes as LDPC codes," *IEEE Transactions on Information Theory*, vol. 49, no. 12, pp. 3268–3271, Dec. 2003.

[11] T. Mittelholzer, "Efficient encoding and minimum distance bounds of Reed-solomon-type array codes," in *Proc. IEEE International Symposium on Information Theory*, 2002.

[12] E. M. Gabidulin and M. Bossert, "On the rank of LDPC matrices constructed by Vandermonde matrices and RS codes," in *Proc. IEEE International Symposium on Information Theory*, Seattle, WA, July 2006, pp. 861–865.

[13] N. Pandya and B. Honary, "Variable-rate LDPC codes based on structured matrices for DVB-S2 applications," in *Proc. 8^{th} Internation Symposium on Communication Theory and Applications*, Ambleside, UK, 2005, pp. 368–373.

[14] L. Chen, J. Xu, I. Djurdjevic, and S. Lin, "Near-shannon-limit quasi-cyclic low-density parity-check codes," *IEEE Transactions on Communications*, vol. 52, no. 7, pp. 1038–1042, July 2004.

[15] Z. Li, L. Chen, L. Zeng, S. Lin, and W. H. Fong, "Efficient encoding of quasi-cyclic low-density parity-check codes," *IEEE Transactions on Communications*, vol. 54, no. 1, pp. 71–81, Jan. 2006.

[16] J. Thorpe, "Low-density parity-check LDPC codes constructed from protographs," *IPN Progress Report 42-154*, Aug. 2003.

[17] J. K. S. Lee, B. Lee, J. Thorpe, K. Andrews, S. Dolinar, and J. Hamkins, "A scalable architecture of a structured LDPC decoder," in *Proc. IEEE International Symposium on Information Theory*, June 27–July 2, 2004.

[18] D. MacKay, "Online database of low-density parity-check codes," Available from wol.ra.phy.cam.ac.uk/mackay/codes/data.html.

[19] J. Campello and D. S. Modha, "Extended bit-filling and LDPC code design," in *Proc. IEEE Global Telecommunications Conference*, vol. 2, San Antonio, TX, Nov. 25–29, 2001, pp. 985–989.

[20] X.-Y. Hu, E. Eleftheriou, and D. M. Arnold, "Regular and irregular progressive edge-growth tanner graphs," *IEEE Transactions on Information Theory*, vol. 51, no. 1, pp. 386–398, Jan. 2005.

[21] F. R. Kschischang, B. J. Frey, and H. A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 498–519, Feb. 2001.

[22] B. Ammar, "Error protection and security for data transmission," *PhD thesis, University of Lancaster*, 2004.

[23] Z. Li and B. V. K. V. Kumar, "A class of good quasi-cyclic low-density parity check codes based on progressive edge growth graph," in *Proc. of 38^{th} Asilomar Conference on Signals, Systems and Computers*, vol. 2, Nov. 7–10, 2004, pp. 1990–1994.

[24] D. D. K. Andrews, S. Dolinar and J.Thorpe, "Design of low-density parity-check LDPC codes for deep-space applications," *IPN Progress Report 42-159*, Nov. 2004.

[25] S. J. Johnson and S. R. Weller, "A family of irregular LDPC codes with low encoding complexity," *IEEE Communication Letters*, vol. 7, no. 2, pp. 79–81, Feb. 2003.

[26] Y. Chen and K. K. Parhi, "Overlapped message passing for quasi-cyclic low-density parity check codes," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 51, no. 6, pp. 1106–1113, June 2004.