Multilevel Structured Low-Density Parity-Check Codes

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— **Low-Density Parity-Check (LDPC) codes are typically characterized by a relatively high-complexity description, since a considerable amount of memory is required in order to store their code description, which can be represented either by the connections of the edges in their Tanner graph or by the non-zero entries in their parity-check matrix (PCM). This problem becomes more pronounced for pseudo-random LDPC codes, where literally each non-zero entry of their PCM has to be enumerated, and stored in a look-up table. Therefore, they become inadequate for employment in memory-constrained transceivers. Motivated by this, we are proposing a novel family of structured LDPC codes, termed as Multilevel Structured (MLS) LDPC codes, which benefit from reduced storage requirements, hardware-friendly implementations as well as from low-complexity encoding and decoding. Our simulation results demonstrate that these advantages accrue without any compromise in their attainable Bit Error Ratio (BER) performance, when compared to their previously proposed more complex counterparts of the same code-length. In particular, we characterize a half-rate quasi-cyclic (QC) MLS LDPC code having a block length of 8064 that can be uniquely and unambiguously described by as few as 144 edges, despite exhibiting an identical BER performance over both Additive White Gaussian Noise (AWGN) and uncorrelated Rayleigh (UR) channels, when compared to a pseudo-random construction, which requires the enumeration of a significantly higher number of 24,192 edges.**

I. Introduction

Low Density Parity-Check (LDPC) codes [1], [2] have attracted substantial interest in the coding research community. It is widely recognized that their soft-input soft-output iterative decoding strategy, is capable of exhibiting a performance close to the Shannon limit [3], when sufficiently high codeword lengths are considered. Moreover, the sparseness of their parity-check matrix (PCM) ensures that this performance is achieved at an acceptable decoder complexity.

The pseudo-random allocation of the logical one values in the PCM was considered to be an important feature in LDPC design, since it was demonstrated in [2], [3] that these codes exhibit excellent error correction capabilities. Other algorithmic constructions tend to focus on a particular attribute of the associated graph such as the girth [4] or the employment of cycle-conditioning [5]. However, the resultant PCM structure remains random and therefore do not possess any compact description that would facilitate their efficient implementation. For this reason, various structured constructions have been investigated, such as those using geometric approaches or combinatorial designs (see [6] and references therein). Most of these structured constructions are cyclic or quasi-cyclic (QC) [7], and therefore their encoding can be implemented with the aid of linear shift-registers [8], thus rendering the encoding complexity a linear function of the block length.

The iterative decoder of LDPC codes can be regarded as a serial concatenation of two constituent decoders separated by an edge interleaver, which defines the edge interconnections between the nodes involved in the parity-check equations, as governed by the code's PCM or by the corresponding bipartite Tanner graph. This effectively means that each non-zero position in the PCM or equivalently, each edge of the Tanner graph represents an entry either in a large look-up table (LUT) or in a large-area hardwired mesh of interconnections on a chip. The complexity of the code's description tends to increase linearly with the block length and again, it is essentially determined by the specific design of the PCM.

Multilevel Structured (MLS) LDPC codes attempt to strike a balance between two contradictory factors in the design of LDPC codes, i.e. that of having a pseudo-random versus structured PCM. In actual fact, MLS LDPC codes are capable of favoring either of these factors, however, we are particularly interested in how far the pseudo-random structure of the PCM can be restricted in favor of becoming more structured, without adversely affecting the BER performance. The novel contribution of this paper is that we propose a class of structured PCMs for LDPC codes, which benefit from reduced storage requirements, hardware-friendly implementations as well as from low-complexity encoding and decoding. Our simulation results provided for both Additive White Gaussian Noise (AWGN) and uncorrelated Rayleigh (UR) channels demonstrate that these advantages accrue without compromising the attainable BER performance, when compared to their previously proposed more complex counterparts of the same length.

The structure of this paper is as follows. Sections II and III introduce the basic principles of LDPC codes and the general construction as well as the necessary constraints of MLS codes. Our discourse continues with the characterization of the code's complexity and its internal structure in Section IV. The external structure of MLS codes is then detailed in Section V. Section VI describes the additional constraints, which were introduced in order to aid the efficient hardware implementation of MLS codes even further. Then, in Section VII, we present an efficient search method designed for graphs having a large girth, which is based on exploiting the isomorphism of edge-colored bipartite graphs. The corresponding simulation results are then detailed in Section VIII, before offering our conclusions in Section IX.

II. Preliminaries

We consider LDPC codes defined by regular bipartite graphs, $G(\mathbf{H})$, associated with a PCM $\mathbf{H}$, whose rows span the null space of the code constructed over GF(2). Then, the graph $G(\mathbf{H})$ consists of the non-empty sets $\{V(G), C(G), E(G)\}$, where $V(G)$ and $C(G)$ represent the disjoint vertex-sets of the variable nodes and check nodes, whilst $E(G)$ is the set representing the edges. Furthermore, we assume that the degree of the variable nodes $v \in V(G)$ and of the check nodes $c \in C(G)$ is γ and ρ, respectively. If $\mathbf{H}$ is full rank, then it has a dimension of $M \times N$ elements, where $N = |V(G)|$ represents the LDPC block length, and the number of parity bits becomes $M = N - K = |C(G)|$, where K represents the number of original information bits. Consequently, $|E(G)|$ becomes equal to γN or ρM and the code rate becomes $R = K/N = 1 - M/N$.

III. General Construction Methodology

Naturally, every structured code is governed by a set of constraints and the larger the number of constraints satisfied, the more structured

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.

978-1-4244-2075-9/08/$25.00 ©2008 IEEE

the code's construction becomes. We particularly distinguish between two types of constraints, the *necessary* constraints which must be satisfied by every MLS code, and the *additional* constraints. We impose the additional over the necessary constraints, in order to generate code constructions, which facilitate efficient implementations. For the sake of simplifying our discourse, we introduce the following three definitions:

Definition 3.1: The *base matrix*, represented by $\mathbf{H}^b$, which is a sparse matrix defined over GF(2) having $(M^b \times N^b)$ elements, and containing exactly ρ and γ non-zero entries in each of its row and column, respectively.

Definition 3.2: The *constituent matrices*, represented by the set $\mathbf{\Omega} = \{\mathbf{Q}_0, \mathbf{Q}_1, \ldots, \mathbf{Q}_{J-1}\}$, where each *non-zero* constituent matrix $\mathbf{Q}_j$, $j = 0, \ldots, J-1$, is a distinct sparse matrix over GF(2) having the same dimensions as the base matrix. The parameter J denotes what we refer to as the *level* of the MLS code.

Definition 3.3: The *adjacency matrix*, which is a $(J \times J)$-element array matrix represented by $\mathbf{P}_J$, whose row blocks represent a *sharply transitive* set of J permutations within $\mathbf{\Omega}$. This implies that given any pair of constituent matrices $\mathbf{Q}_x, \mathbf{Q}_y \in \mathbf{\Omega}$, there exists a unique bijective mapping function $f : \mathbf{\Omega} \mapsto \mathbf{\Omega}$ in the set described by the row block of $\mathbf{P}_J$ that maps $\mathbf{Q}_x \in \mathbf{\Omega}$ to the image $\mathbf{Q}_y = f(\mathbf{Q}_x) \in \mathbf{\Omega}$.

These definitions enable us to describe the necessary constraints:

- Constraint 1: Each of the sparse constituent matrices $\mathbf{Q}_j \in \mathbf{\Omega}$ must avoid having any non-zero entries in the PCM that are symmetrically repeated in two or more rows (or columns). It may be readily shown that this ensures that the girth of each constituent matrix is at least six.
- Constraint 2: All the non-zero entries of all the sparse constituent matrices $\mathbf{Q}_j \in \mathbf{\Omega}$ must occur in the same position of the base matrix. Furthermore, having a non-zero entry in a particular location $\mathbf{Q}_j \in \mathbf{\Omega}$, implies that the entries in the corresponding locations of $\mathbf{Q}_{i \neq j}$ are zero. This ensures that the number of non-zero elements in $\mathbf{H}^b$ is equal to the sum of those in the J constituent matrices.

It may be readily demonstrated that the first and second constraints are closely related; in fact, any base matrix having a girth of $g > 4$ will produce a set of constituent matrices $\mathbf{Q}_j$, $j = 0, \ldots, J-1$, satisfying the first constraint. Naturally, a girth higher than four requires that the base matrix has a sufficiently large dimension.

If both the first and the second constraints are satisfied, then the girth of the graph $G(\mathbf{H})$ associated with the PCM of the MLS code is definitely larger than $g = 4$, since the adjacency matrix will avoid positioning any constituent matrix in the same row or column block. The PCM $\mathbf{H}$ of a J-level MLS code will then have $(JM^b \times JN^b)$ elements. For example, given a particular adjacency matrix $\mathbf{P}_J$, the PCM $\mathbf{H}$ of a J-level MLS code may have the following form:

$$\mathbf{H} = \begin{pmatrix} \mathbf{Q}_0 & \mathbf{Q}_1 & \mathbf{Q}_2 & \cdots & \mathbf{Q}_{J-1} \\ \mathbf{Q}_{J-1} & \mathbf{Q}_0 & \mathbf{Q}_1 & \cdots & \mathbf{Q}_{J-2} \\ \mathbf{Q}_{J-2} & \mathbf{Q}_{J-1} & \mathbf{Q}_0 & \cdots & \mathbf{Q}_{J-3} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \mathbf{Q}_1 & \mathbf{Q}_2 & \cdots & \mathbf{Q}_{J-1} & \mathbf{Q}_0 \end{pmatrix}, \tag{1}$$

which is also sparse and its null space represents an LDPC code having a rate of $R \geq 1 - M^b/N^b$.

Previously, we have mentioned that the PCM construction of MLS codes simultaneously exhibits both pseudo-random as well as deterministic structural characteristics. The pseudo-random PCM structure of MLS codes is attributed to the fact that no constraints are imposed on the actual base matrix selected, and therefore any previously proposed pseudo-random PCM construction can be utilized as a base matrix. The position of the non-zero entries in each of the constituent matrices $\mathbf{Q}_j$ in $\mathbf{\Omega}$ can also be chosen at random, while obeying the previously described first and second constraints.

In our work, we have assumed both randomly and uniformly distributed positions for the non-zero entries in the constituent matrices $\mathbf{Q}_j$ of the set $\mathbf{\Omega} = \{\mathbf{Q}_0, \mathbf{Q}_1, \ldots, \mathbf{Q}_{J-1}\}$. For the case of uniformly distributed positions, we have introduced *additional* constraints, which enhance the code's structure and thus improve the associated implementational aspects even further. The additional constraints will be discussed in Section VI.

IV. Complexity of the Code Description and the Protograph-based Internal Structure of MLS Codes

It is quite easy to recognize the reduced code description complexity, that accrues from having a PCM obeying Equation (1). Increasing the number of levels J will automatically imply that the size of both the base matrix as well as of the constituent matrices $\mathbf{Q}_j \in \mathbf{\Omega}$ will be decreased, and consequently, the grade of randomness in the resultant MLS code's construction will become less pronounced. Following this argument, we formulate the following two conjectures:

Conjecture 1: The complexity of an MLS code's description is reduced by a factor, which is proportional to the number of levels J, when compared to other pseudo-random codes.

The complete bipartite graph represented by the PCM $\mathbf{H}$ of an MLS code can be interpreted as a specific partition of an edge set $E(\mathbf{H})$ constituted by the following union:

$$E(\mathbf{H}) = E(\mathbf{H}_0) \cup E(\mathbf{H}_1) \cup E(\mathbf{H}_2) \ldots E(\mathbf{H}_{J-1}), \tag{2}$$

where $E(\mathbf{H}_j)$, $j = 0, \ldots, J-1$, are all disjoint (as required by the second constraint), non-empty sets, each representing the edges that uniquely and unambiguously describe the connections between the check nodes c_{mj}, $m = 1, \ldots, M^b$ and variable nodes v_{nj}, $n = 1, \ldots, N^b$. Consequently, the number of LUT entries required to store the PCM description of a J-level MLS code is equal to $|E(\mathbf{H})|$, which by the second constraint is identical to $|E(\mathbf{H}^b)| = N^b\gamma$. On the other hand, an LUT that is storing a pseudo-random PCM description must enumerate $N\gamma$ edges, where $N = N^bJ$. Therefore, it is only necessary to enumerate the edges present in each constituent matrix $\mathbf{Q}_j$ in $\mathbf{\Omega}$ in order to describe an entire MLS code.

Conjecture 2: MLS codes constitute a subclass of protograph codes, which were defined in [9].

Let a base protograph, $G(\mathbf{H}_p)$, be described by the set of check nodes $C(\mathbf{H}_p) = \{c_{mj} : m = 1, \ldots, M^p; j = 0\}$ and variable nodes $V(\mathbf{H}_p) = \{v_{nj} : n = 1, \ldots, N^p; j = 0\}$ as well as by the set of edges $E(\mathbf{H}_p)$, where $\mathbf{H}_p$ represents the PCM of the base protograph having $(M^p \times N^p)$ elements. The index $j = 0$ is being assigned to the base protograph. After replicating $G(\mathbf{H}_p)$ J_p times, we obtain the Tanner graph $G(\mathbf{H}_p^{'})$ of the protograph code, defined by the sets $C(\mathbf{H}_p^{'})$, $V(\mathbf{H}_p^{'})$ and $E(\mathbf{H}_p^{'})$, where each set has a cardinality, which is J_p times higher than that of the corresponding set in the base protograph. The matrix $\mathbf{H}_p^{'}$ denotes the PCM of the graph derived, which has $(J_pM^p \times J_pN^p)$ elements. An LDPC code is considered to be a protograph code if and only if the connection of the edges in each of the J_p replicas obey the constraints governed by the base protograph, i.e. the interconnections between the nodes on both sides of the graph derived follow the same specific permutation pattern of the base protograph [9]. This is also valid for the case of an MLS code having $J = J_p$ levels, since their adjacency matrix ensures that the permutations of edges incident to every N^b variable nodes at each level of the graph $G(\mathbf{H})$ are determined using the same J constituent matrices (please refer to the column blocks in the example shown in (1)), where the latter have non-zero entries occurring in the same position of the base matrix (by the second constraint). Developing this analogy slightly further, the base matrix of an MLS code will correspond to a PCM representation of a base protograph.

It is important to note that whilst all MLS codes constitute protograph codes, the reverse is not necessarily true. The reason for this lies in the specific technique used for the construction of protograph codes. Typically, protograph LDPC codes are constructed using a variant of the Progressive Edge Growth (PEG) [4] algorithm, which randomly (subject to maximizing the local girth) connects every variable node v_{nj}, $n = 1, \dots, N^p$ to a check node that satisfies the constraints governed by the base protograph. This "randomness" introduced by the PEG algorithm will make the resultant PCM, $\mathbf{H}_p^{'}$ also random. Andrews *et al.* argue in [10] that although protograph codes do have some internal structure, they still suffer from a high-complexity description due to the random PEG permutations and thus they still require a considerable amount of memory to store the addresses to which each input bit is mapped. On the other hand, the "randomness" of an MLS code is restricted to only a single level, whilst the remaining $(J - 1)$ levels are essentially permutations of the above-mentioned single pseudo-randomly generated level. In this light, we may also interpret MLS codes as specific protograph codes having more compact descriptions. Despite the above-mentioned construction-constraints, MLS codes still benefit from inheriting implementationally attractive semi-parallel architectures, such as those suggested by Lee *et al.* in [11].

V. External Structure of MLS Codes

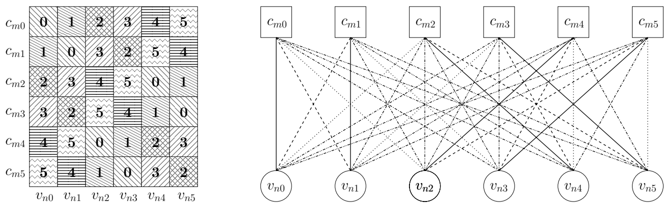
MLS codes possess both an internal and an external structure, where the latter is based on the adjacency matrix $\mathbf{P}_J$ that is chosen for implementation, which is essentially what makes them different from the protograph codes originally proposed by Thorpe [9]. The adjacency matrix appropriately positions each (internally structured) constituent matrix $\mathbf{Q}_j \in \mathbf{\Omega}$ with respect to the (externally structured) PCM of the MLS code, $\mathbf{H}$. This implies that the adjacency matrix must also be stored and therefore it is equally desirable that it has a compact description. Hence, we may identify two classes of MLS codes, which are distinguished by their adjacency matrices and by the complexity of their descriptions, as described in the following subsections.

A. Class I MLS Codes based on a Homogeneous Coherent Configuration

We define the family of Class I MLS codes as follows:

Definition 5.1: A Homogeneous Coherent Configuration (HCC) is identified by the set of binary matrices $\mathbf{A} = \mathbf{A}_0, \dots, \mathbf{A}_{J-1}$ having the sum equal to the all-one matrix, and which is closed under transposition. In addition, the set $\mathbf{A}$ has the property that one of the matrices is the identity matrix and that the product of any two matrices is a linear combination of the matrices in the set.

Class I MLS codes are those codes, whose adjacency matrix describes the adjacency algebra of a HCC. The adjacency matrix of a J-level Class I code is in fact shown in (1), which represents the adjacency matrix of a *non-symmetric association scheme* on J points. Elaborating slightly further, we will use the example of a five-level Class I MLS code having an adjacency matrix $\mathbf{P}_5$ given by:



	v_{n0}	v_{n1}	v_{n2}	v_{n3}	v_{n4}	v_{n5}
c_{m0}	0	1	2	3	4	5
c_{m1}	1	0	3	2	5	4
c_{m2}	2	3	4	5	0	1
c_{m3}	3	2	5	4	1	0
c_{m4}	4	5	0	1	2	3
c_{m5}	5	4	1	0	3	2

Fig. 1. A Latin square (left) representation of the adjacency matrix together with the corresponding edge-colored, complete bipartite graph for a six-level Class II MLS code.

$$\mathbf{P}_5 = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 \\ 4 & 0 & 1 & 2 & 3 \\ 3 & 4 & 0 & 1 & 2 \\ 2 & 3 & 4 & 0 & 1 \\ 1 & 2 & 3 & 4 & 0 \end{pmatrix}, \tag{3}$$

where each element in the matrix corresponds to a subscript and thus defines the position of a constituent matrix $\mathbf{Q}_j \in \mathbf{\Omega}$. The compact description of $\mathbf{P}_J$ can be readily demonstrated by recognizing that a cyclic shift obeying $x \mapsto x + 1$, $x \in \mathbb{Z}_5$, with $\mathbb{Z}$ being the set of integers, is an automorphism of this scheme, and therefore, each of the J zero-one-valued matrices, $\mathbf{A}_j \in \mathbf{A}$, $j \in [0, J-1]$, is a circulant matrix of size J = 5. It can be observed in both (1) as well as (3), that the matrix $\mathbf{A}_0 = \mathbf{I}_J$, where $\mathbf{I}_g$ corresponds to the identity matrix having a size of g, whilst the remaining binary matrices $\mathbf{A}_j$, $j \in [1, J-1]$, have a binary one entry in column $(r + j)$ mod J, where r is the row-index of the circulant matrix, $0 \leq r \leq J - 1$, and $(a$ mod $b)$ represents the modulus after division of a by b.

B. Class II MLS Codes based on Latin Squares

The adjacency matrix $\mathbf{P}_J$ can also be interpreted as a Latin square of order J consisting of row blocks that generate the *symmetric group* $S_{\mathbf{\Omega}}$ on $\mathbf{\Omega}$ and having order $J!$. Figure 1 depicts this representation of an adjacency matrix for a six-level Class II MLS code, where the J rows and columns of the Latin square correspond to the respective *multi-check* node $c_{mj} \subset C(\mathbf{H})$ and to the *multi-variable* node $v_{nj} \subset V(\mathbf{H})$, $|c_{mj}| = |v_{nj}|$, where we have $m = 1, \dots, M^b$, $n = 1, \dots, N^b$ and $j = 0, \dots, J - 1$.

A Latin square is also equivalent to a 1-factorization of a bipartite graph, and hence we can also regard a J-level MLS code as an *edge-coloured*, complete bipartite graph of degree J. Equation (2) shows that the edge set $E(\mathbf{H})$ of the graph $G(\mathbf{H})$ is partitioned into J disjoint, non-empty sets $E(\mathbf{H}_j) \subset E(\mathbf{H})$, $j = 0, \dots, J - 1$. This brings us to the notion of what is known as *coloring* of edges, where $E(\mathbf{H})$ is said to be an edge-coloring of $G(\mathbf{H})$ if any two edges on the graph containing the same vertex have different colors. Correspondingly, each symbol of the Latin square will create a monochromatic 1-factor of the regular bipartite graph. Figure 1 also illustrates the corresponding edge-colored graph for a six-level Class II MLS code having an adjacency matrix represented by a *reduced* Latin square. The different "edge colors" on the Tanner graph of Figure 1 are represented using different line types.

VI. Additional Constraints

We impose the additional constraints over the necessary constraints mentioned in Section III in order to aid the efficient hardware implementation of MLS codes even further. The constraints are described as follows:

- Constraint 3: Starting from any base matrix having $(M^b \times N^b)$ elements, distribute the non-zero entries across the constituent matrices so that each row and column of any $\mathbf{Q}_j \in \mathbf{\Omega}$ contains a single non-zero entry. This constraint can only be applied in the scenario when the number of levels J is at least equal to the column weight ρ of the PCM.

- Constraint 4: Replace each non-zero entry in each constituent matrix by a circulant matrix of size q from the set $\mathbf{I}_q, \mathbf{I}_q^{(1)}, \mathbf{I}_q^{(2)}, \ldots, \mathbf{I}_q^{(q-1)}$, where $\mathbf{I}_q^{(x)}$ represents a right cyclic shift by x positions for each row of the identity matrix $\mathbf{I}_q$.

The third constraint will facilitate the parallel processing of messages exchanged over the interconnections between the check and variable nodes. Since each non-zero entry in each row or column of the base matrix is positioned in a different constituent matrix, each memory block will only access (read or write) each location once per clock cycle. Furthermore, it becomes possible to simultaneously process the ρ edges incident on each check node by the J memory blocks, if J is at least equal to ρ.

By the fourth constraint, the resultant PCM having $(qJM^b \times qJN^b)$ elements will be composed of only circulant matrices of weight zero or one, and thus the code effectively becomes quasi-cyclic (QC). The amount of memory required to store the code's description is then reduced by factor of $1/qJ$, when compared to other pseudo-random constructions, since memory shifts corresponding to the QC PCM structure can be used to address the messages exchanged between the nodes. The encoding process can then be implemented using simple linear shift-register circuits of length K [8].

VII. Efficient Search for Graphs having a Large Girth

We have selected MLS codes based on the optimization criterion of maximizing the average girth, using an approach similar to that of Mao and Banihashemi in [12]. However, the differentiating feature of our search is that it is now possible to avoid the inspection of isomorphic (edge-colored) graphs based on their corresponding Latin square representation, and hence our search is much more efficient. Formally, we have the following definitions:

Definition 7.1: Two Latin squares S and S' are said to be *isotopic* if there exists a triple (α, β, χ) (referred to as an *isotopy*), where α, β and χ correspond to a respective row, a column and a symbol permutation, which carries the Latin square S to S'. Effectively, this implies that if we consider any particular row and column position of the Latin square specified by the check and variable node (c_{mj}, v_{nj}), containing entry e, where $m = 1, \ldots, M^b$, $n = 1, \ldots, N^b$ and $j = 0, \ldots, J-1$, then the entry at position $(\alpha(c_{mj}), \beta(v_{nj}))$ of the Latin square S' will be equal to $\chi(e)$. Then, an *isotopy class* comprises the set of all the Latin squares isotopic to a given Latin square.

Definition 7.2: Two Latin squares S and S'' are said to be *conjugates* if S'' is obtained from S by simply permuting the "roles" of the rows, columns and symbols of S.

With the aid of the following claim, we can effectively avoid searching through the isomorphic edge-colored graphs.

Claim 7.1 [13]: Two Latin squares S_1 and S_2 will give rise to isomorphic edge-colored complete bipartite graphs if and only if S_1 is isotopic to either S_2 or to $(S_2)^T$, where the superscript T denotes the transpose operation.

The transpose of Latin square S is actually one of its conjugates, which is obtained by exchanging the roles of the columns with that of the rows. Therefore, it is only required to search each isotopy class representative and four of its conjugates.

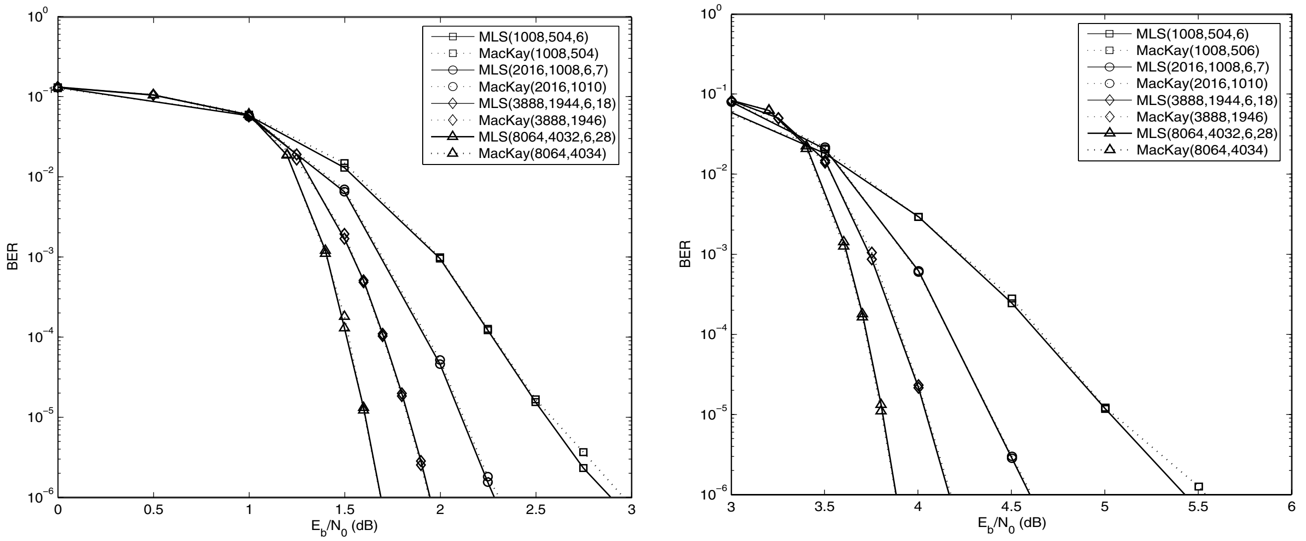
We note that Class I MLS codes are effectively a subclass of Class II MLS codes. It is easy to demonstrate that by permuting the rows, columns or symbols and/or by permuting the roles of the rows, columns and symbols of the Latin square, one can obtain the adjacency matrix of a Class II MLS code from that of a Class I code. As regards to the achievable BER performance, we assume that Class II codes attain a superior BER performance in comparison to Class I codes since the former have to satisfy a lower number of constraints, and possibly attain a higher average girth. In this paper, we will only report the performance of Class I MLS codes, the performance of Class II MLS codes will be left for future work. Nevertheless, Class II MLS codes have the advantage that with aid of the same J constituent matrices, we can represent ζ_J number of different PCMs, where ζ_J depends on the order J of the corresponding Latin square. In [14], we exploited this attractive property and employed Class II MLS codes in a coded Interleave-Division Multiple-Access (IDMA) scenario, where the different users are separated in the code domain using MLS codes having the same constituent matrices but using user-specific adjacency matrices. This considerably reduces the memory requirements of the multiuser receiver, since it is no longer necessary to store a different code description (in this case a different PCM) for each user.

VIII. Results and Discussion

We will appropriately distinguish between MacKay's pseudo-random codes, and the proposed MLS codes satisfying the first three constraints as well as QC MLS codes satisfying all the previously mentioned constraints using the notation (N, K), (N, K, J) and (N, K, J, q), respectively. In Figure 2, we compare the achievable BER performance for transmission over both AWGN and UR channels for both six-level MLS codes and the corresponding MacKay codes having block lengths of 1008, 2016, 3888 and 8064. It can be observed that despite their constrained PCM, the MLS codes exhibit no BER performance loss, when compared to their pseudo-random counterparts, although the MLS codes exhibit substantial implementational benefits.

The MLS(1008,504,6) code was constructed using a (84×168)-element base matrix as well as six constituent matrices, and an adjacency matrix based on a 6-point HCC. This MLS code exhibits an E_b/N_0 gain of about 0.1 dB over the corresponding half-rate QC code based on the Euclidean sub-geometry EG*(2,2^4) (*cf.* Table I in [7]) having a block length of N = 1020, and the parameters of $\gamma = 4$ and $\rho = 8$. Both the QC MLS(2016,1008,6,7) as well as the QC MLS(8064,4032,6,28) codes were constructed using the same (144×288)-element base matrix, but the former was expanded using circulant matrices of size 7, whilst the latter used circulant matrices of size 28. The BER versus E_b/N_0 performance of the QC MLS(2016,1008,6,7) code is approximately 0.4 dB superior in comparison to the the half-rate EG*(3,2^3) code (*cf.* Table I in [7]) having a block length of 2044 as well as parameters of $\gamma = 4$ and $\rho = 8$. Furthermore, we note that the superior error correction performance of both the MLS(1008,504,6) as well as the MLS(2016,1008,6,7) codes is achieved at a lower decoding complexity due to the lower logic depth, since the column and row weight of the PCM of the MLS codes is 3 and 6.

The QC MLS(3888,1944,6,18) code was then constructed using a base matrix having dimensions (18×36) elements and then expanded by circulant matrices of size 18. Finally, Table I summarizes the girth, together with the distance between the Shannon limit of the exhibited code's performance at a BER of 10^{-6}. We also compared the complexities of the codes' description for the MLS and the corresponding MacKay benchmarker codes, by quantifying the effective number of edges ϵ that must be stored, or equivalently, the number of LUT entries that are needed in order to store the code's



(a) BER, AWGN channel

(b) BER, UR channel

Fig. 2. A BER performance comparison of R = 0.5, MLS and MacKay's [2] LDPC codes with N = 1008-8064 and a maximum of I = 100 decoder iterations when transmitting over the AWGN and uncorrelated Rayleigh (UR) channel using BPSK modulation.

description. It is also important to note that MLS codes having a lower number of levels J, will exhibit a superior BER performance than the corresponding six-level MLS codes characterized here.

Similar trends to those shown in Figure 2 were observed for the Block Error Ratio (BLER) performance. For the sake of completeness, we have also investigated the performance of MLS codes having diverse 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 due to space limitations, demonstrated that the performance of MLS codes can achieve the maximum parallelization factor[1] without suffering from any BER/BLER degradations in comparison to MacKay's pseudo-random codes.

TABLE I

COMPARISON BETWEEN THE MLS AND MACKAY CODES.

			*Shannon Gap**	
Code	*Girth*	$\epsilon^{\dagger}$	*AWGN (dB)*	*UR (dB)*
MLS(1008,504,6)	6	504	2.704	3.590
MacKay(1008,504)	6	3024	2.771	3.702
MLS(2016,1008,6,7)	8	144	2.095	2.762
MacKay(2016,1006)	6	6048	2.120	2.771
MLS(3888,1944,6,18)	8	108	1.759	2.330
MacKay(3888,1942)	8	11664	1.759	2.339
MLS(8064,4032,6,28)	8	144	1.501	2.046
MacKay(8064,4030)	8	24192	1.503	2.046

† The effective number of edges that must be stored, or equivalently, the number of entries in the memory LUT storing the code description.
* The distance between the Shannon limit and the exhibited code's performance at a BER of 10^{-6}. The Shannon limit for the AWGN and the UR channel was assumed to be 0.188 dB and 1.834 dB, respectively.

IX. SUMMARY AND CONCLUSIONS

In this paper, we have proposed the construction of Multilevel Structured (MLS) LDPC codes, which benefit from having a low-complexity description due to the structured row-column connections, whilst having low-complexity encoding and decoding implementations due to their semi-parallel architectures. We investigated their BER performance for transmission over both AWGN and UR channels, for various code rates and block lengths. Explicitly, our experimental results demonstrated that whilst there is no BER/BLER performance loss for the MLS LDPC codes when compared to the corresponding MacKay codes, considerable implementational benefits accrue in terms of the storage memory required for storing the code's description.

REFERENCES

[1] R. G. Gallager, "Low-density parity-check codes," *IRE Transactions Information Theory*, vol. 45, pp. 21–28, Jan. 1962.

[2] D. MacKay, "Good error-correcting codes based on very sparse matrices," *IEEE Transactions on Information Theory*, vol. 45, pp. 399–431, Mar. 1999.

[3] 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, pp. 58–60, Feb. 2001.

[4] X.-Y. Hu, E. Eleftheriou, and D. M. Arnold, "Regular and irregular progressive edge-growth Tanner graphs," *IEEE Transactions on Information Theory*, vol. 51, pp. 386–398, Jan. 2005.

[5] T. Tian, C. R. Jones, J. D. Villasenor, and R. D. Wesel, "Selective avoidance of cycles in irregular LDPC code construction," *IEEE Transactions on Communications*, vol. 52, pp. 1242–1247, Aug. 2004.

[6] J. M. F. Moura, J. Lu, and H. Zhang, "Structured low-density parity-check codes," *IEEE Signal Processing Magazine*, vol. 21, pp. 42–55, Jan. 2004.

[7] 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, pp. 1038–1042, July 2004.

[8] 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, pp. 71–81, Jan. 2006.

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

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

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

[12] Y. Mao and A. H. Banihashemi, "A heuristic search for good ldpc codes at short block lengths," in *Proceedings of the. International Conference on Communications*, vol. 1, (Helsinki, Finland), pp. 41–44, Nov. 2001.

[13] R. A. Bailey and P. J. Cameron, "Latin squares: Equivalents and equivalence," Available online from http://designtheory.org/.

[14] N. Bonello, R. Zhang, S. Chen, and L. Hanzo, "Channel code division multiple access and its multilevel structured LDPC based instantiation," submitted to the IEEE Transactions on Vehicular Technology.

[1] For the case of MLS codes, the maximum parallelization factor of N/ρ, is achieved when we have $J = \rho$.