Low-Rate Non-binary Hybrid LDPC Codes

Lucile Sassatelli, David Declercq and Charly Poulliat
ETIS ENSEA/UCP/CNRS UMR-8051 95014 Cergy, FRANCE
{sassatelli,declercq,poulliat}@ensea.fr

***Abstract*— In this paper, we use the class of non-binary hybrid LDPC codes to design very efficient low rate codes. To this end, we consider both asymptotic and finite length designs. First, we present an asymptotic analysis to design hybrid LDPC codes distributions and we explicit the cases when it is usefull to use it. Then, by optimizing the algebraic properties of some topologies related to the Tanner graph, we design finite length codes. These codes show significant performance improvements both in the waterfall and in the error floor regions, in comparison with existing state-of-the-art low rate coding schemes, like Turbo Hadamard or parallel concatenated Zigzag Hadamard codes, and with no increase of complexity. In particular, the minimum distance of low rate hybrid LDPC codes is by far greater than the ones of previously mentionned codes.**

I. Introduction

The problem addressed in this paper is the design of efficient low rate coding schemes. For communication systems operating in the low signal-to-noise ratio (SNR) regime (e.g., code-spread communication systems and power-limited sensor networks), low-rate coding schemes play a critical role. One important application of low-rate codes is in wideband data communications using code-division multiple-access (CDMA) systems [1], where they are used to replace the spreading code in traditionnal direct-sequence spread spectrum systems.

Although Low-Density Parity-Check (LDPC) codes or Repeat-Accumulate (RA) codes can exhibit capacity-approaching performance for various code rates when the ensemble profiles are optimized [2], in the low-rate region, both RA and LDPC codes suffer from performance loss and extremely slow convergence using iterative decoding. To our knowledge, the most competitive codes at this time are Turbo-Hadamard (TH) [3] and various versions of Zigzag-Hadamard (ZH) codes [4]. All references of various low rate coding schemes can be found in [3][4][5].

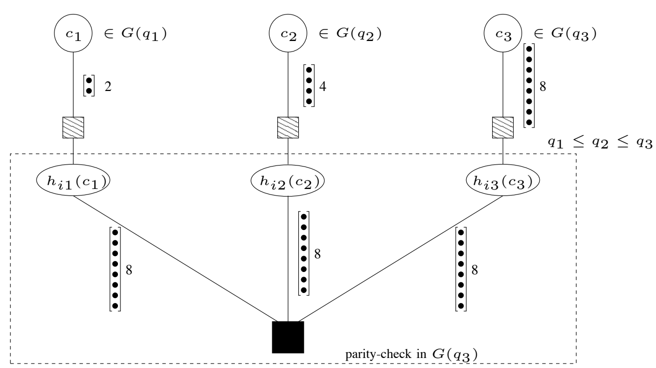
In this paper, we address the profile optimization and finite length design of low-rate hybrid LDPC codes. The hybrid LDPC codes family is a class of non-binary LDPC codes, generalizing existing binary or non-binary classes of LDPC codes. This new class has been first presented in [6], and then a deeper asymptotic analysis has been carried out in [7]. For hybrid LDPC codes, we allow the connectivity profile of the factor graph to be irregular, but also the orders of the symbols in a codeword can be irregular, that is to say, the symbols can belong to finite sets with different orders. We exploit the very rich parameterization of hybrid LDPC codes to find good low-rates codes. The paper is organized as follows. In section II, we review the structure of hybrid LDPC codes and briefly describe the decoding algorithm. We also briefly present the asymptotic analysis of hybrid LDPC codes under belief propagation (BP) decoding and we explain when the optimization method is used for the design of efficient hybrid LDPC codes distributions. Section III describes the finite-length optimization of hybrid LDPC codes using their binary images. Simulation results are presented in section IV. Conclusions are finally given in the last section.

[1]This work was supported by the French Armament Procurement Agency (DGA).

II. The Class of Hybrid LDPC Codes

A. Presentation and notations

A non-binary hybrid LDPC code is defined as an LDPC code whose variable nodes belong to finite sets of different orders [6].



$h_{i1}(c_1) + h_{i2}(c_2) + h_{i3}(c_3) = 0, \qquad h_{ij}(c_j) \in G(q_3)$

defines a component code in the group $G = G(q_1) \times G(q_2) \times G(q_3)$

Fig. 1. Factor graph of parity-check of an hybrid LDPC code.

As shown on figure 2, every symbols of a hybrid LDPC code can belong to different order finite groups, and hence the variable node outcoming messages are of different sizes. Therefore, a non-binary hybrid LPDC code is defined on a group which is the cartesian product of different finite groups $\mathcal{G} = G(q_{min}) \times \ldots \times G(q_{max})$ with q_{min} and q_{max} the minimum and maximum orders of codeword symbols, respectively. The function nodes on each edge are general applications from the group of the variable node to the group of the check node. Therefore, the check node messages are all in the group of the check node, and hence they are all of same size. A hybrid check node is hence a usual parity check on groups. An edge of the factor graph carries two kinds of messages, messages

978-1-4244-2863-2/08/$25.00 © 2008 IEEE

of size q_1 (or, e.g., q_2) and q_3 on figure 2. The function node corresponding to the linear application makes the components of the two types of messages correspond to each other. The applications between different groups can be of any types.

In order to explain the decoding algorithm for hybrid LDPC codes, it is useful to interpret a parity check of a hybrid code as a special case of a parity check built on the highest order group of the symbols of the row, denoted $G(q_3)$, and have a look at the binary image of the equivalent code as it has been done in [8] for finite fields. For codes defined over Galois fields, the nonzero values of $\mathbf{H}$ correspond to the powers of the companion matrix [9] of the finite field primitive element and are typically rotation matrices (because of the cyclic property of the Galois fields). In the case of hybrid LDPC codes, a nonzero value is a function that connects a row in $G(q_3)$ and a column in $G(q_1)$, i.e., that maps the q_1 symbols of $G(q_1)$ into a subset of q_1 symbols that belongs to $G(q_3)$. In our case, such an application is linear and hence its equivalent binary representation is a matrix of dimension $p_3 \times p_1$, with $p_k = \log_2(q_k)$.

We restrict ourselves in this work to hybrid LDPC codes with applications which are linear, and whose check nodes are in groups with orders higher than the variable nodes they are connected to. The linear applications are hence full column rank. A full rank component sub-matrix of size $p_3 \times p_1$ will be denoted in the following a *non-zero cluster*, which is the direct generalization of non-zero values for non-binary LDPC codes defined in fields. The transformation of the vector message is denoted extension from $G(q_1)$ to $G(q_3)$ when passing through the function node from symbol node to check node, and truncation from check node to symbol node.

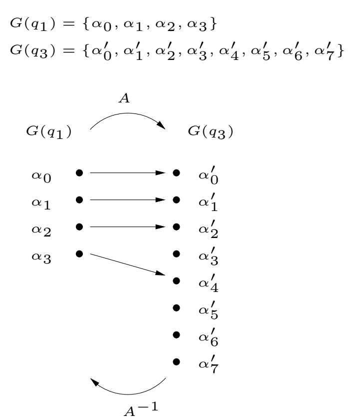


Fig. 2. A full column rank linear application.

Let A be an element of the set of linear applications from $G(q_1)$ to $G(q_3)$ which are full-rank. Im(A) denotes the image of A (that is injective since dim(Im(A))=rank(A)=p_1). The notations are the ones of figure 2.

$$
\begin{aligned}
A: \quad G(q_1) &\rightarrow G(q_3) \\
\alpha_i &\rightarrow \alpha'_j = A(\alpha_i)
\end{aligned}
$$

Definition 1: The extension $\mathbf{y}$ of the probability vector $\mathbf{x}$ by A is denoted by $\mathbf{y} = \mathbf{x}^{\times A}$ and defined by, for all $j = 0, \ldots, q_3 - 1$,

$$
\begin{aligned}
&\text{if } \alpha'_j \notin \mathrm{Im}(A), \quad y_j = 0 \\
&\text{if } \alpha'_j \in \mathrm{Im}(A), \quad y_j = x_i \text{ with } i \text{ such that } \alpha'_j = A(\alpha_i)
\end{aligned}
$$

Although A is not bijective, we define A^{-1} the pseudo-inverse of A, by

$$
\begin{aligned}
A^{-1}: \quad \mathrm{Im}(A) &\rightarrow G(q_3) \\
\alpha'_j &\rightarrow \alpha_i \text{ with } i \text{ such that } \alpha'_j = A(\alpha_i)
\end{aligned}
$$

Definition 2: The truncation $\mathbf{x}$ of the probability vector $\mathbf{y}$ by A^{-1} is denoted by $\mathbf{x} = \mathbf{y}^{\times A^{-1}}$ and defined by, for all $i = 0, \ldots, q_1 - 1$,

$$x_i = y_j \text{ with } j \text{ such that } \alpha'_j = A(\alpha_i)$$

In addition, we consider a particular subclass of hybrid LDPC codes which has two constraints, as depicted on figure 3. We

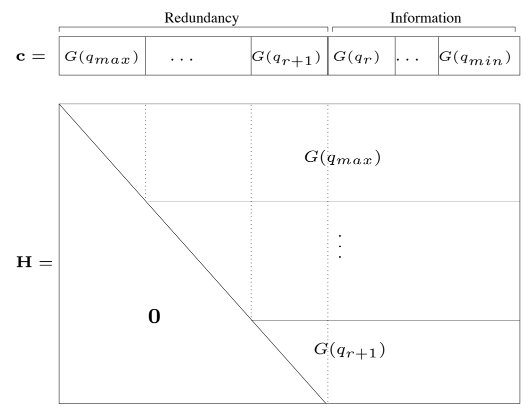


Fig. 3. Hybrid codeword with the specific sorting of groups, and upper-triangular parity-check matrix of a hybrid LDPC code.

hierarchically sort the different group orders in the rows of the parity-check matrix and also in the codeword. It is not difficult to extend the results and algorithms to more general structures, as long as the hybrid LDPC code stays linear in the product group. The second constraint is that we choose to work only with upper-triangular parity-check matrices in order to avoid use of generator matrix to encode. With this specific structure, it is clear that encoding is feasible in linear time by backward computation of the redundancy symbols.

B. Parameterization of Hybrid LDPC family

An edge of the Tanner graph of an hybrid LDPC code has four parameters (i, q_k, j, q_l). A hybrid LDPC code is then represented by $\pi(i, j, k, l)$ which is the proportion of edges connecting variable nodes of degree i in $G(q_k)$, to check nodes of degree j in $G(q_l)$. With four parameters, the parameterization of hybrid LDPC codes is hence very rich, and highlights the generality of this class of codes, which includes

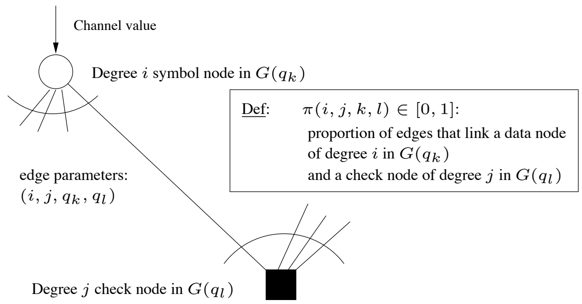


Fig. 4. Parameterization of hybrid LDPC codes.

classical irregular binary and non-binary LPDC codes, and which allows more degrees of freedom. Hence, looking for good LDPC codes for a given channel in the huge class of hybrid LDPC codes increases the chances to find such a good code, since the search space is bigger than with classical LDPC codes. The results presented in this paper show that this framework is well fitted to low-rate coding schemes.

C. Decoding algorithm for hybrid LDPC codes

It is possible to generalize the Belief propagation decoder to hybrid LDPC codes since parity equations are defined on groups, and the function nodes corresponding to the non-zero clusters of the matrix $\mathbf{H}$ consist in extension and truncation of messages. Even for those very specific structures, it has been shown that it is possible to derive a fast version of the decoder using FFTs [10]. One decoding iteration in the probability domain with a flooding schedule is composed of [6]:

- Step 1 **Symbol node update** in $G(q_k)$: product of incoming messages
- Step 2 **Message extension** $G(q_k) \rightarrow G(q_l)$
- Step 3 **Parity-Check update** in $G(q_l)$ in the Fourier domain
  - FFT of size q_l
  - Term-by-term product of FFT vectors
  - IFFT of size q_l
- Step 4 **Message truncation** from $G(q_l) \rightarrow G(q_k)$

Although we do not focus on simplified decoders, hybrid LDPC codes are compliant with reduced complexity non-binary decoders which have been presented recently in the literature [11], [12].

D. Asymptotic optimization using a Gaussian approximation

In [7][13], it has been proved that, on a binary-input symmetric channel (BISC), we can assume that the all-zero codeword is transmitted because the hybrid decoder preserves the symmetry of messages, which entails that the probability of error is independent of the transmitted codeword. By exhibiting a stability condition thanks to density evolution analysis, we have also proved that the probability of error is able to drop infinitely close to zero, i.e., that hybrid LDPC are codes with threshold behavior. Due to the high complexity of tracking all the parameters of multi-dimensionnal density evolution for non-binary LDPC codes, we have approximated the densities by using a Gaussian approximation, and hence reduced the problem to one scalar parameter analysis. By doing so, the analysis of hybrid LDPC codes has been carried out for the binary-input additive white Gaussian noise (BIAWGN) channel. An optimization procedure has been derived, which gives the optimal code profile, either the connection profile, or the group order profile. It depends on which parameters are fixed. We refer the reader to [6][7][13] for the details on this EXIT charts optimization of hybrid LDPC codes.

The example of low-rate code optimization is specific. We look for the best group order profile of variable nodes, while the connexion profile is fixed. We fix the check node parameters (group order $G(q_{red})$ and connection profile), independently from the other parameters. All symbol nodes belong to groups of order lower or equal than the group of the check nodes they are connected to. For finite length code optimization, we choose the Tanner graph to be ultra-sparse, i.e. with a regular connection profile $(d_v = 2, d_c)$. This choice of strictly regular LDPC code with minimum symbol degree $d_v = 2$ will be argued in section III. With this a priori fixed values for the mentioned parameters, the parametrization of hybrid LDPC codes restricts to (as in [13]):

$$\pi(i,j,k,l) = \delta(i,d_v)\delta(j,d_c)\pi(k)\delta(l,red)$$

where $\delta(.,.)$ is the Kronecker symbol. Since the variable nodes belong to different order groups, the rate of the graph is different of the code rate. We denote by $\mathcal{I}$ the indices of the group order of information symbols. In other words, any information symbols is in $G(q_k)$ with $k \in \mathcal{I}$. For a strictly regular graph with rate $R_{graph} = 1 - \frac{d_v}{d_c}$, the fact that the codeword symbols belong to groups of different orders induces an actual code rate R of:

$$R = \frac{R_{graph} \sum_{k \in \mathcal{I}} \pi(k) \log_2(q_k)}{R_{graph} \sum_{k \in \mathcal{I}} \pi(k) \log_2(q_k) + (1 - R_{graph}) \log_2(q_{red})} \tag{1}$$

It is worthy to note that, due to the fact that every check node is in a group of order higher or equal to the groups of variable nodes, the graph rate R_{graph} is always greater than or equal to the code rate. This means that iterative decoding is performed on a higher rate graph, which is very interesting for low rate applications. Indeed, the very slow convergence of BP decoding on classical low rate LDPC codes is due to the sparsity of the associated Tanner graph.

By limiting the choice of the hybrid LDPC codes to graphs which are regular of type $(d_v = 2, d_c)$, the minimum reachable code rate is limited by the maximum group order. The curves plotted on figure 5 depict the maximum reachable graph rate for a given code rate when the maximum group order varies between $\mathrm{G}(16)$ and $\mathrm{G}(1024)$. We see on this figure that with $(2, d_c)$ regular hybrid LDPC codes, we cannot reach code rates lower than 0.047 for $\mathrm{G}(1024)$.

Due to the constraint of the grah rate, we do not have many degrees of freedom for choosing the group order profile when

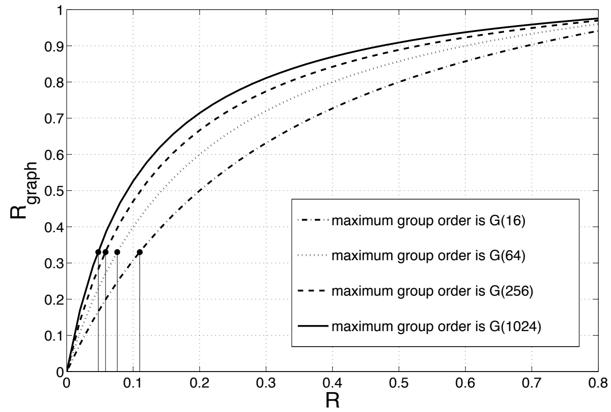


Fig. 5. Maximum graph rate R_{graph} reachable in terms of the code rate R.

the target code rate is small. That is why, except for code rate $R = 1/6$, we do not use the optimization of group order profile by EXIT chart techniques. Moreover, in this work, we focus on designing codes with good performance at small to moderate codeword lengths, and we have therefore conducted a deep optimization procedure on the topological properties of the non-binary hybrid Tanner graph, which is explained in next section.

III. Finite-length optimization

This section presents an extension of optimization methods that has been described in [8] for finite length non-binary LDPC codes. We address the problem of the selection and the matching of the parity check matrix $\mathbf{H}$ nonzero clusters. In this section, we assume that the connectivity profile and group order profile of the graph have been optimized. With the knowledge of the graph connectivity, we run a progressive-edge-growth (PEG) algorithm [14] in order to build a graph with a high girth.

The method is based on the binary image representation of $\mathbf{H}$ and of its components, i.e. the non-zero clusters of the hybrid code in our case (cf. section II-A). First, the optimization of the rows of $\mathbf{H}$ is addressed to ensure good waterfall properties. Then, by taking into account the algebraic properties of closed topologies in the Tanner graph, such as cycles or their combinations, an iterative method is used to increase the minimum distance of the binary image of the code by avoiding low weight codewords.

A. Row optimization

Based on the matrix representation of each nonzero entry, we give thereafter the equivalent vector representation of the parity check equations associated with the rows of H.

Let $\mathbf{x} = [x_0 \dots x_{N-1}]$ be a codeword in $\mathcal{G} = G(q_{min}) \times \dots \times G(q_{max})$, and let p_j be the number of bits representing the binary map of symbol $x_j \in G(2^{p_j})$, $j = 0, \dots N-1$. For the $i-$th parity equation of H in the group $G(2^{p_i})$, we have the following vector equation:

$$\sum_{j:H_{ij}\neq 0} H_{ij}\mathbf{x_j} = \mathbf{0} \tag{2}$$

where H_{ij} is the $p_i \times p_j$ binary matrix representation of the non-zero cluster, $\mathbf{x_j}$ is the vector representation (binary map) of the symbol x_j. The all zero component vector is noted $\mathbf{0}$.

Considering the i-th parity check equation as a single component code, we define $\mathbf{H_i} = [H_{ij_0} \dots H_{ij_m} \dots H_{ij_{d_c-1}}]$ as its equivalent binary parity check matrix, with $\{j_m : m = 0 \dots d_c - 1\}$ the indices of the nonzero elements of the i-th parity-check equation. The size of $\mathbf{H_i}$ is $p_i \times (p_{ij_0} + \dots + p_{ij_{d_c-1}})$, with p_i and p_{ij_k} the extension orders of the groups of the check node and the k-th connected variable node, respectively. Let $\mathbf{X_i} = [\mathbf{x_{j_0}} \dots \mathbf{x_{j_{d_c-1}}}]^t$ be the binary representation of the symbols of the codeword $\mathbf{x}$ involved in the $i-$th parity check equation. When using the binary representation, the i-th parity check equation of $\mathbf{H}$ (2), can be written equivalently as $\mathbf{H_i}\mathbf{X_i}^t = \mathbf{0}^t$.

We define $d_{min}(i)$ as the minimum distance of the binary code associated with $\mathbf{H_i}$. As described in [8], a d_c-tuple of d_c linear applications is chosen in order to maximize the minimum distance $d_{min}(i)$ of the code corresponding to the ith row of $\mathbf{H}$, $i = 0, \dots, M-1$. As shown in [8] for strictly regular non-binary LDPC codes, picking up the nonzero values in the selected d_c-tuple lowers the threshold of the optimized code ensemble, suggesting that the waterfall region of the error performance curve can be improved by selecting carefully the rows of the parity check matrix. For non-binary hybrid LDPC codes, we adopt the same strategy, and choose for $\mathbf{H_i}$ a binary linear component code with the highest minimum distance achievable with the dimensions of $\mathbf{H_i}$. For example, let $\mathbf{H_i}$ be obtained from a $d_c = 3$ check node with the three symbols belonging to $G(2^8) \times G(2^8) \times G(2^2)$, $\mathbf{H_i}$ has size (8×18) and the highest possible minimum distance is $d_{min}(i) = 5$ [15]. For hybrid LDPC codes, even if the connection degree is constant for all check nodes, the dimensions of the component code $\mathbf{H_i}$ could differ and depend on the symbols orders which appear in $\mathbf{X_i}$.

B. Avoiding low weight codewords

We now address the problem of designing codes with good minimum distance. It has been shown in [8] that the error floor of non-binary LDPC codes based on ultra-sparse ($d_v = 2$) graph is not uniquely due to pseudo-codewords, but also to low weight codewords. We adopt for hybrid LDPC codes the same strategy that has been introduced in [8], which aims at avoiding the low weight codewords which are contained in the smallest cycles. In order to do so, we first extract and store the cycles of the Tanner graph with length belonging to $\{g, \dots, g + gap\}$, where g is the girth and gap is a small integer such that the number of cycles with size $g + gap$ is manageable.

As in the previous section, we consider the binary images of cycles as component codes. Let $\mathbf{Hc_k}$ be the binary image of the k-th stored cycle. Since we consider $(2, d_c)$ codes, if some columns of $\mathbf{Hc_k}$ are linearly dependent, so will be the columns of $\mathbf{H}$ (see [8] for more details). This means that a codeword of a cycle is also a codeword of the whole code. The proposed approach is hence to avoid low weight codewords by properly choosing the nonzero clusters implied in the cycles, so that no codeword of low-weight is contained in the cycles. This is achieved by ensuring that the binary matrices $\mathbf{Hc_k}$ corresponding to the cycles have full column rank. The iterative procedure that we use in this optimization step is essentially the same as the one depicted in [8]. Briefly speaking, in each step of the iterative procedure, we change the values of a limited number of non-zero clusters in order to maximize the number of cycle component codes $\mathbf{Hc_k}$ which have full rank. Thus, the matrix of a cycle should be full rank to cancel the cycle. Contrarily to classical non-binary LDPC codes for which the matrix of a cycle is squared, the matrix of a cycle of a hybrid LDPC code is rectangular, with more rows than columns. This means that we will have more degrees of freedom to cancel the cycles in hybrid LDPC codes. Hybrid LDPC codes are therefore well-suited to this kind of finite-length optimization procedure.

IV. Simulation results

The considered channel is the BPSK-AWGN channel. We compare the performance of our proposed hybrid LDPC codes with existing good codes [3][4]. The performance curves of hybrid LDPC codes have been obtained for a maximum number of BP decoding iterations fixed to 500. K_{bit} is the number of information bits.

For a code rate $R = \frac{1}{6}$, a regular graph $(d_v = 2, d_c = 3)$ is considered, and the proportion of group orders has been optimized with EXIT charts techniques. With the order of the check nodes being fixed to $G(q_{max}) = G(256)$, the code resulting from the optimization has three different group orders $G(256) - G(16) - G(8)$.

For a code rate $R = \frac{1}{12}$, a regular graph $(d_v = 2, d_c = 3)$ is also considered, and the code group order profile has three different group orders $G(256) - G(4) - G(2)$. For this case, none EXIT chart technique was used for the optimization of the profile, but the finite length optimization technique is applied.

On figure 6, for $K_{bit} \simeq 200$, the hybrid LDPC code of code rate $1/6$ outperforms with 0.3 dB gain the ZH code of code rate $1/6$. Additionnaly, our hybrid code has no observed error floor up to a BER=10^{-7}. When comparing the computer simulation of the hybrid code with the union bound of ZH code, we observe that the BER of the hybrid LDPC code has gain of about one decade at $E_b/N_0 = 2$dB. Since union bounds are tight upper bounds on BER performances [3] for Turbo-Hadamard codes, we can predict from the figure that the error floors of our two simulated codes will be lower than the error floors of Turbo-Hadamard codes with random interleaver. Indeed, the minimum distance of our hybrid LDPC code has been estimated thanks to the impulse method [16] and is upper bounded by $d_{min} = 80$, which is by far superior

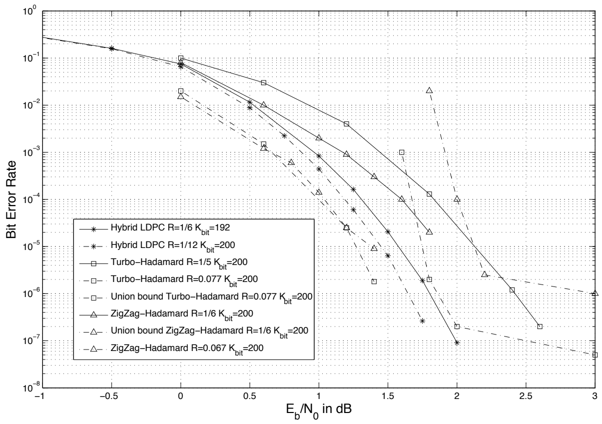


Fig. 6. Comparison of hybrid LDPC code with Tubo Hadamard codes (TH) taken from [3] and Zigzag Hadamard (ZH) codes taken from [4], for an information blocklength of $K_{bit} \simeq 200$.

of the minimum distance that can be achieved with TH or ZH codes.

The hybrid LDPC code of code rate $R = 1/12 = 0.083$ has poorer performance in the waterfall region than TH and ZH codes with comparable rates, but has much lower error floor when comparing the computer simulations to the union bound of the code rate 0.077 TH code. Indeed, its minimum distance is upper bounded by $d_{min} = 125$. Hence, although this $R = 0.083$ code suffers from 0.1 to 0.2 dB loss compared with the rate 0.077 TH code, the good error floor properties highlight the interest of hybrid LDPC codes for lower rates.

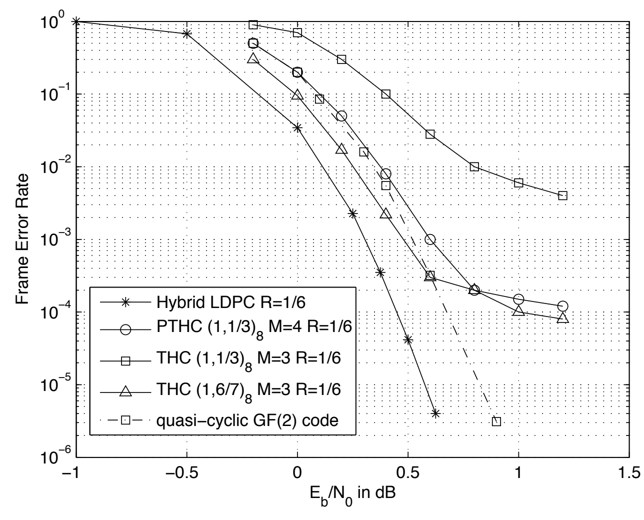


Fig. 7. Comparison of hybrid LDPC code with Tubo Hadamard codes and punctured Turbo Hadamard (PTH) are taken from [17], for an information blocklength of $K_{bit} \simeq 1000$.

On figure 7, the hybrid LDPC code for code rate $1/6$ is the same as previously described, but optimized with the method presented in section III for $K_{bit} \simeq 1000$. The hybrid LDPC code outperforms all the other plotted codes, both in the error floor and in the waterfall regions. All Hadamard like codes have high error floors, but it is especially worthy to note that, in the two SNR regions our code is also better than a quasi-cyclic code in GF(2) designed to achieve low error-floor [18]. This is promising for working on very low code rate frameworks using hybrid LDPC codes. Following the ideas of Turbo-Hadamard codes, it would be interesting to combine a very small repetition code with a hybrid LDPC code in order to reach lower code rates. We plan to address this issue in a future work.

It is worthy to note that the better performance of hybrid LDPC codes over codes based on Hadamard codes are obtained with no complexity increase. Indeed, the complexity of these codes is dominated by the complexity of the fast Hadamard transform, which is $O(2^r)$, where r is the order of the Hadamard code. The complexity of hybrid LDPC codes is dominated by the fast Fourier transform at check nodes $O(2^q)$, where q is the maximum group order. The complexity of Hadamard type codes and hybrid LDPC codes is hence equivalent. However, contrary to TH codes, one should note that hybrid LDPC codes are suitable for decoding with reduced complexity and no loss, as described in [12].

V. Conclusions

We have shown that hybrid LDPC codes can be very good candidates for efficient low rate coding schemes. We have used two methods to optimize them: the code profile can be optimized thanks to EXIT charts techniques whereas the algebraic properties are optimized for finite length performances. The finite length performance of low rate hybrid LDPC codes compare very well to existing Turbo Hadamard or Zigzag Hadamard codes. Especially, hybrid LDPC codes exhibit very good minimum distances and error floor properties. To reach lower code rates, the interest of hybrid LDPC codes in concatenated systems will be investigated.

References

[1] A. Viterbi, "Very low rate convolutional codes for maximum theoretical performance of spread-spectrum multiple-access channels," *IEEE Journal on Selected Areas on Communications*, vol. 8, pp. 641–649, May 1990.

[2] T. Richardson, A. Shokrollahi, and R. Urbanke, "Design of capacity-approaching irregular LDPC codes," *IEEE Transactions on Information Theory*, vol. 47, pp. 619–637, February 2001.

[3] L. Ping, W. Leung, and K. Wu, "Low-rate Turbo-Hadamard codes," *IEEE Transactions on Information Theory*, vol. 49, pp. 3213–3224, December 2003.

[4] G. Yue, W. Leung, L. Ping, and X. Wang, "Low rate concatenated Zigzag-Hadamard codes," in *Proceedings of International Conference on Communications*, (Istanbul, Turkey), June 2006.

[5] M. González-López, F. J. Vázquez-Araújo, L. Castedo, and J. Garcia-Frias, "Layered LDGM codes: a capacity-approaching structure for arbitrary rates," in *Proc. ISWCS*, (Trondheim, Norway), September 2007.

[6] L. Sassatelli and D. Declercq, "Non-binary hybrid LDPC codes: Structure, decoding and optimization," in *Proceedings of IEEE Information Theory Workshop*, (Chengdu, China), October 2006.

[7] L. Sassatelli and D. Declercq, "Analysis of non-binary hybrid LDPC codes," in *Proceedings of IEEE International Symposium on Information Theory*, (Nice, France), June 2007.

[8] C. Poulliat, M. Fossorier, and D. Declercq, "Design of regular (2,dc)-LDPC codes over GF(q) using their binary images," *accepted in IEEE Transactions on Communications*, 2007.

[9] F. MacWilliams and N. Sloane, *The theory of error-correcting codes*. North Holland, 1978.

[10] A. Goupil, M. Colas, G. Gelle, and D. Declercq, "FFT-based BP decoding of general LDPC codes over Abelian groups," *IEEE Transactions on Communications*, vol. 55, pp. 644–649, April 2007.

[11] D. Declercq and M. Fossorier, "Decoding algorithms for nonbinary LDPC codes over GF(q)," *IEEE Transactions on Communications*, vol. 55, pp. 633–643, April 2007.

[12] A. Voicila, D. Declercq, F. Verdier, M. Fossorier, and P. Urard, "Low complexity decoding for non-binary LDPC codes in high order fields," *accepted for publication in IEEE Transactions on Communications*, 2008.

[13] L. Sassatelli and D. Declercq, "Non-binary hybrid LDPC codes," *submitted to IEEE Transactions on Information Theory*, second stage revision.

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

[15] A. Brouwer and T. Verhoeff, "An updated table of minimum distance for binary linear codes," *IEEE Transactions on Information Theory*, vol. 39, pp. 662–677, March 1993.

[16] X.-Y. Hu and M. Fossorier, "On the computation of the minimum distance of low-density patity-check codes," in *Proceedings of IEEE International Conference on Communications*, (Paris), June 2004.

[17] N. Shimanuki, B. Kurkoski, K. Yamagichi, and K. Kobayashi, "Improvements and extensions of low-rate Turbo-Hadamard codes," in *Proceedings of ISITA*, (Seoul, Korea), October 2006.

[18] G. Liva, W. Ryan, and M. Chiani, "Design of quasi-cyclic Tanner codes with low error floors," in *Proceedings of IEEE International Symposium on Turbo Codes*, (Munich, Germany), April 2006.