Generalized Low Density Codes* : Approaching the channel capacity with simple and easily decodable block codes

J. Boutros†, O. Pothier‡, G. Zémor†

† ENST, 46 Rue Barrault, 75013 Paris, France
‡ LEP, 22 Avenue Descartes, 94453 Limeil-Brévannes, France
email : *NAME* @enst.fr

January 31, 1998

1 Introduction

We build a new class of pseudo-random error correcting codes (called GLD codes) by generalizing the Gallager's construction of low density parity check codes (LDPC) [1]. Each parity check equation of an LDPC code (N, K) is replaced by the parity check matrix of a small linear code (n, k) called the *constituent code*.

LDPC codes are usually defined by their parity check matrix, but they can also be described with a bipartite graph. The left part of the graph contains the code symbols and the right one contains the parity check nodes. A parity check node is associated to the trivial parity check code $(n, n-1)$. This representation of LDPC codes has been used by Sipser and Spielman [2] to study the influence of the graph expansion on the code parameters.

The graphical representation of block codes has been first exploited and generalized by Tanner [3]. Tanner codes based on a bipartite deterministic graph are obtained by replacing the $(n, n-1)$ code associated to one parity check node with a less trivial constituent code (n, k). Thus, building a Tanner code on a random graph (instead of a deterministic one) is a second method to construct GLD codes.

*Submitted to IEEE-ITW 98, Killarney, Ireland.

In the sequel, we restrict our description of GLD codes to their matrix representation. As explained in the next section, the GLD class is obtained from the intersection of two or more interleaved subcodes. The subcodes of length N are a direct sum of N/n constituent codes.

GLD codes seems to perform as well as Turbo codes [4] for both small and large block lengths. It is also proved that GLD codes are asymptotically optimal in the sens of the minimum distance criterion.

2 Structure of the GLD code

Figure 1 shows the parity check matrix H of an LDPC code (N, K) with length $N = 12$ and rate $R \geq 1/4$. The matrix H is the concatenation of $J = 3$ submatrices. The first submatrix H_1 of size 3×12 defines a subcode by the direct sum of three parity check codes $(n, n-1)$ with $n = 4$. The whole matrix H is obtained by concatenating H_1, $H_2 = \pi_1(H_1)$ and $H_3 = \pi_2(H_1)$, where π_1 and π_2 are two pseudo-random column permutations. This example can be generalized to build any LDPC code (N, K) using $J-1$ permutations. Gallager showed that LDPC codes are aysmptotically good when $J \geq 3$. He also described an iterative decoding algorithm similar to turbo decoding [4] and he exploited the low density of the parity check matrix to reduce the decoder complexity when computing the a posteriori probabilities.

$$
\begin{array}{c|cccccccccccc|c}
 & \multicolumn{12}{c}{N = 12} & \\
 & \multicolumn{4}{c}{n = 4} & & & & & & & & & \\
 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \\
 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & \\
 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & \\
\hline
\pi_1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & \\
 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & N-K \\
 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & \\
\hline
\pi_2 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & \\
 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & \\
 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & \\
\end{array}
$$

$J = 3$ submatrices

Figure 1: Example of an LDPC matrix with $J = 3$ levels.

Each line of the LDPC matrix H is a parity check equation defined by the $(n, n-1)$ parity code. We replace this line by $n-k$ lines including one copy of the parity check matrix H_0 of a constituent code $C_0(n,k)$. This operation is depicted on Figure 2. The first submatrix produces the direct sum of N/n identical codes $C_0(n,k)$. The matrix H has J submatrices derived by interleaving the columns of the first submatrix. This type of parity check matrices H defines the class of GLD codes. Thus, a GLD code C is the intersection of J subcodes C_j, i.e. $C = \bigcap_{j=1}^{J} C_j$ where $C_{j+1} = \pi_j(C_1)$ for $j = 1 \dots J-1$, and $C_1 = C_0 \oplus \dots \oplus C_0$. If C_0 has a rate $r = k/n$, the total rate of the GLD code is $R = 1 - J(1-r)$. Note that it is not possible to define the GLD code as a serial (neither parallel, nor hybrid) concatenation of two or multiple constituent codes.

In our study, we considered binary GLD codes with only $J = 2$ levels (the total rate is $R = 2r - 1$) based on binary Hamming codes. As shown in the next section, we need only $J = 2$ (i.e. one interleaver) to make the GLD code asymptotically optimal. For practical applications, efficient GLD codes can be built from primitive, shortened or extended binary BCH codes.

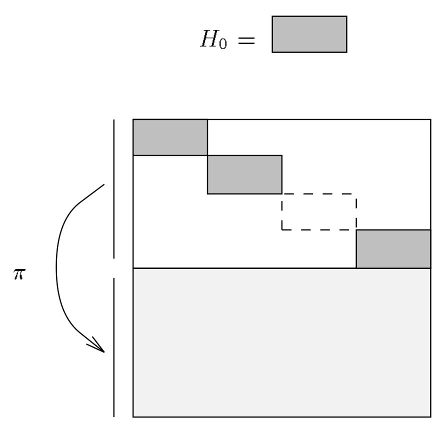

Figure 2: Structure of a GLD parity check matrix ($J = 2$).

3 Ensemble performance

Without loss of generality, we restrict the performance study in this section to the case of a two levels GLD code based on the binary Hamming code $C_0(7,4)$. We also consider a BSC channel with a transition probability $0 < p < 1/2$ (the study is intractable on the AWGN channel).

Let us start by computing the average weight distribution of the ensemble of GLD codes built with $C_0(7,4)$ and a random column permutation π. In other words, the weight coefficients are obtained by averaging over the interleaver π. The moment-generating function $g(s)$ of C_0 is given by :

$$g(s) = \frac{1 + 7e^{3s} + 7e^{4s} + e^{7s}}{16}$$

The first subcode C_1 of length N is the direct sum of N/n independent codes C_0. Hence, its moment-generating function $G(s)$ is simply a power of $g(s)$:

$$G(s) = g(s)^{N/n} = \sum_{\ell} Q(\ell) e^{\ell s}$$

where $Q(\ell)$ is the probability that a vector of weight ℓ belongs to C_1. Since the total number of codewords in C_1 is $(2^k)^{N/n}$, then the average number in C_1 of coderwords of weight ℓ is $N_1(\ell) = 2^{(kN/n)} Q(\ell)$. Exploiting the fact that C_1 and $C_2 = \pi(C_1)$ are totally independent, the probablity that a vector of weight ℓ belongs to $C = C_1 \bigcap C_2$ can be written as :

$$P(\ell) = \left(\frac{N_1(l)}{\left(\begin{array}{c} N \\ \ell \end{array} \right)} \right)^2$$

Finally, the average number of codewords in C having weight ℓ is :

$$\overline{N(\ell)} = \left(\begin{array}{c} N \\ \ell \end{array} \right) \times P(\ell) = \frac{2^{(2kN/n)} Q(\ell)^2}{\left(\begin{array}{c} N \\ \ell \end{array} \right)}$$

By upperbounding the coefficient $Q(\ell)$ with $G(s)e^{-\ell s}$, and by applying the Stirling approximation (valid for large N), we get an upperbound on the average number of codewords of weight ℓ in the GLD code (we omit the details) :

$$\overline{N(\ell)} \leq C(\lambda, N) \times e^{-NB(\lambda)}$$

where $\lambda = \ell/N$ is the normalized weight.
The two functions $C(\lambda, N)$ and $B(\lambda)$ are expressed as follows :

$$C(\lambda, N) = \sqrt{2\pi N \lambda (1-\lambda)} \times e^{1/(12N\lambda(1-\lambda))}$$

$$B(\lambda) = H(\lambda) - \frac{2}{n} \left[\mu(s) + k \ log2 \right] + 2s\lambda$$

where $H(\lambda)$ is the natural entropy function and $\mu(s) = log(g(s))$. The upperbound has been optimized and the optimal value of s is related to the weight by $\lambda = \mu'(s)/n$.

The exponent function $B(\lambda)$ is sketched in Figure 3. Asymptotically, when $N \to \infty$, the average number $\overline{N(\ell)}$ of codewords of weight ℓ goes to zero if $B(\lambda) > 0$. The first value of $\lambda \in]0 \ldots 1/2[$ corresponding to a sign transition gives us a lower bound on the minimum distance $\delta(C) = d_{min}(C)/N$ of the GLD code. As seen in Figure 3, $\delta \geq 0.186$. Thus, C is asymptotically optimal with $d_{min} \geq 0.186N$ and $R = 1/7$ (the Gilbert-Varshamov bound gives $\delta_0 = H_2^{-1}(1-R) = 0.281$).

Now, let us compute the maximal value of p for which the word error probability P_{ew} of an ML decoder goes to zero when N is arbitrarily large. An upperbound on P_{ew} is obtained by assuming that a decoding error occurs when at least half of the codeword non zero symbols are covered. If j denotes the channel error weight, ℓ the weight of a coderword and i the number of covered non zero bits, we have :

$$P_{ew} \leq \sum_{j=1}^{N} p^j (1-p)^{N-j} \sum_{\ell=d_{min}}^{N} \overline{N(\ell)} \sum_{i=\ell/2}^{\ell} \binom{\ell}{i} \binom{N-\ell}{j-i}$$

When N is large enough, an expression similar to the upperbound on $\overline{N(\ell)}$ can be found for P_{ew}. After some algebraic manipulations, we get (details omitted) :

$$P_{ew} \leq D(N,p) \times e^{-NE(p)}$$

where the exponent function $E(p)$ is given by :

$$E(p) = Minimum_\lambda \quad \left[B(\lambda) + H(p) - \lambda log2 - (1-\lambda) H(\frac{p - \lambda/2}{1-\lambda}) \right]$$

Figure 4 shows $E(p)$ versus p. From this curve, we conclude that an ML decoder for the GLD code achieves $P_{ew} \to 0$ if $p < 0.277$ (not far from 0.281 given by the BSC channel capacity).

4 Simulation results

Two different GLD codes have been tested over an additive white Gaussian noise channel (AWGN). The modulation is a BPSK with symbols equal to

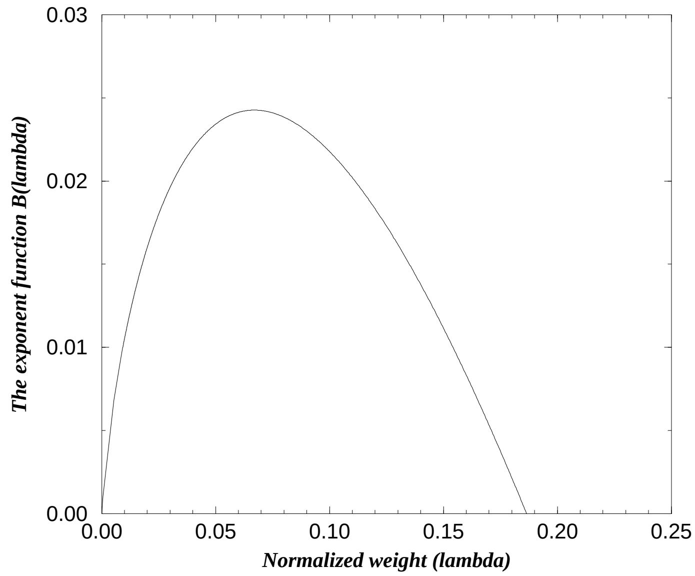

Figure 3: The exponent function $B(\lambda)$ versus the normalized weight λ. The low density code is built from the $(7,4)$ Hamming code.

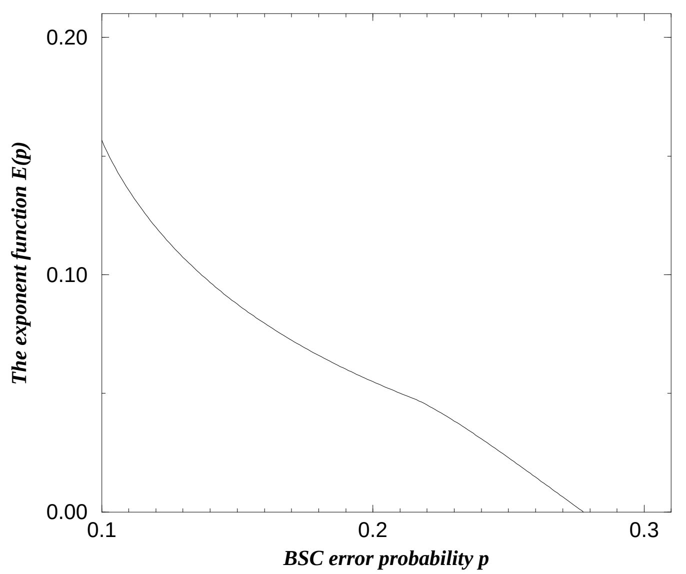

Figure 4: The exponent function $E(p)$ versus the BSC channel transition probability p. The low density code is built from the $(7,4)$ Hamming code.

$\pm\sqrt{2RE_b}$, where E_b is the average energy per information bit. We used a forward-backward algorithm to decode the constituent code C_0 and an iterative procedure (similar to turbo decoding) to compute the a posteriori probablities of the code symbols.

The first code, suitable for mobile radio transmissions or small frame systems, has length $N = 810$. Its performance (BER versus the number of decoding iterations) is shown in Figure 5 for $E_b/N_0 = 1.5,\ 2.0\ and\ 2.7\ dB$.

The second code, suitable for deep space communications or image transmissions, has length $N = 65534$. Figure 6 shows its BER versus the decoding iteration number. This code achieves zero error probability at $1.8dB$ with a rate $R = 0.67$. Its performance is 0.6dB away from the capacity limit (1.2dB for $R = 0.67$ and a BPSK input).

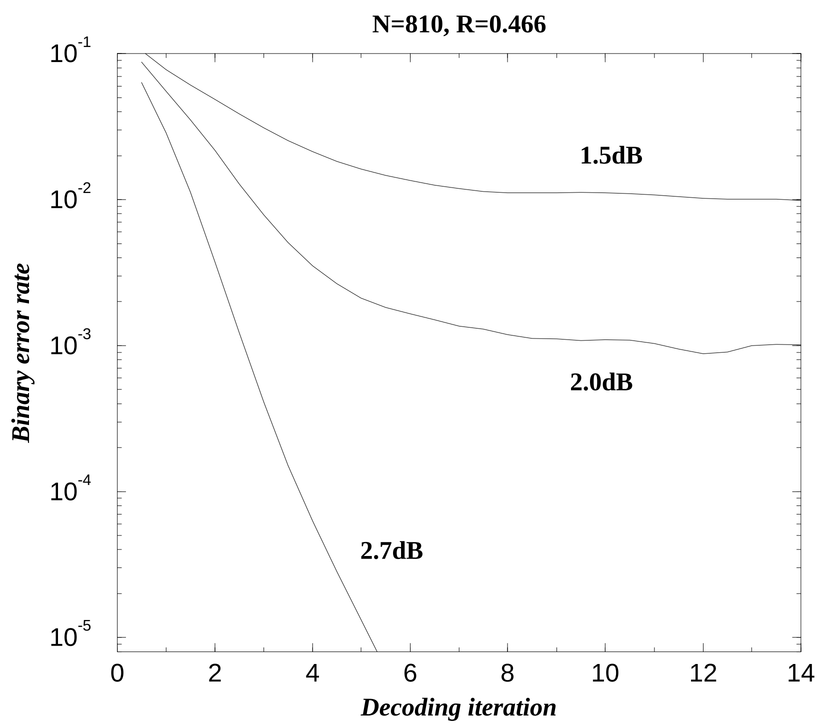

Figure 5: Performance of the GLD code built from the $(15, 11)$ Hamming code, length $N = 810$, total rate $R = 0.466$, on AWGN channel.

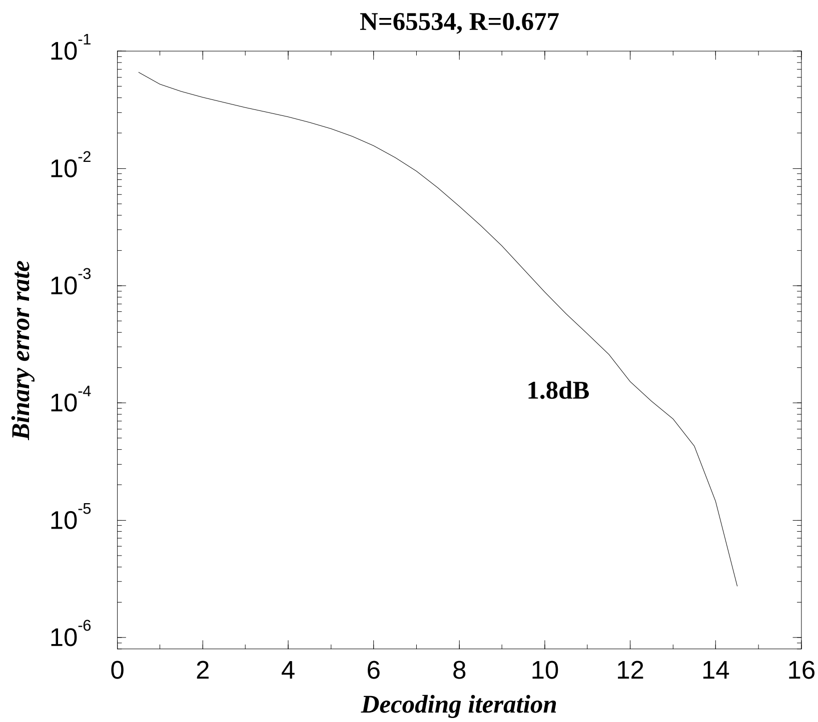

Figure 6: Performance of the GLD code built from the $(31, 26)$ Hamming code, length $N = 65534$, total rate $R = 0.677$, on AWGN channel.

References

[1] R.G. Gallager: Low-density parity-check codes, MIT Press, 1963.

[2] M. Sipser and D.A. Spielman: "Expander codes", 1994.

[3] R.M. Tanner: "A recursive approach to low complexity codes", *IEEE Trans. on Information Theory*, Vol. IT-27, Sept 1981.

[4] C. Berrou, A. Glavieux, P. Thitimajshima : "Near Shannon limit error-correcting coding and decoding : turbo-codes," *Proceedings of ICC'93*, Genève, pp. 1064-1070, Mai 1993.