A Gallager-Tanner construction based on convolutional codes

Sandrine VIALLE, Joseph BOUTROS

Motorola Research Center, Paris
ENST, Communications & Electronics Department
Email : *NAME*@com.enst.fr

December 1, 1998

Keywords : Low density codes, Turbo codes, Asymptotically good codes.

Abstract

Generalized low density codes are built by applying a Tanner-like construction to binary recursive systematic convolutional codes. The Gallager-Tanner construction is restricted to 2 levels only. We describe the structure of a GLD code and show how to compute its ensemble performance. We also prove that RSC based GLD codes are asymptotically good. A parity-check interpretation of turbo codes is given for both parallel and serial concatenations.

1 Introduction

An efficient channel coding scheme has to imitate random codes. To make it feasible, such a scheme is generally based on simple structured elementary codes linked via a pseudo-random interleaver. Low density parity-check (LDPC) codes developed by Gallager [1] are a good example of error-correcting codes imitating random coding.
A binary LDPC code (N, K) of length N and dimension K is defined by a set of $N-K$ interleaved parity-check equations (PCEs) making the $(N-K) \times N$ matrix H. The low density of H is due to the limited number of 1's in each PCE and the limited weight of its columns. It is also proved [1] that the low density of H reduces significantly the complexity of the LDPC iterative decoder. Figure 1 shows an LDPC matrix of size 9×12. This matrix is obtained from the concatenation of $J = 3$ submatrices H_i, $i = 1, 2, 3$. The first submatrix H_1 contains 3 disjoint PCEs and the other ones are given by applying a random column permutation π to H_1, i.e. $H_2 = \pi_1(H_1)$ and $H_3 = \pi_2(H_1)$. The PCEs weight is $n = 4$ and the columns weight is $J = 3$. Gallager's codes are asymptotically good in the sense of the minimum distance criterion if $J \geq 3$.

The matrix representation of an LDPC code is equivalent to a bipartite graph showing the structure of the code. The left part of the graph has N nodes (bit nodes) and the right one has $N-K$ nodes (subcode nodes). The graph is regular and the degrees of the bit nodes and the subcode nodes are J and n respectively. For example, the code of Figure 1 can be graphically represented with a bipartite graph having 12 bit nodes and 9 subcode nodes.
The generalization of the graphical representation described above generates a Tanner code [2]. In fact, each subcode node of an LDPC code is associated to an $(n = 4, k = 3, 2)$ single parity-check (SPC) code. A Tanner code is built from a graph where the subcode nodes are associated to a more general linear (n, k, d_{Hmin}) code, e.g. BCH codes.

In this paper, we apply a Tanner-like construction to binary recursive systematic convolutional (RSC) codes [3] to build a generalized low density (GLD) code. The matrix representation of a GLD code is similar to an LDPC representation where the PCEs derived from the SPC code are replaced by non-disjoint PCEs derived from an RSC code. In the sequel, we restrict the Gallager-Tanner construction to $J = 2$ levels only. We describe the structure of a GLD code and show how to compute its average weight distribution, i.e. its ensemble performance. We also prove that RSC based GLD codes (with 2 levels) are asymptotically good. A parity-check interpretation of turbo codes [4] is given for both parallel and serial concatenations, where turbo codes are described as a special case of Low Density Parity Check codes.

The structure of the generalized low density code is given in section 2 and its performance is analysed in section 3. A Gallager-Tanner interpretation of turbo codes is made in section 4 before sketching the simulation results on a gaussian channel.

$$
\begin{array}{c}
\xleftrightarrow{\quad N=12 \quad} \\
\xleftrightarrow{n=4} \\
\left|\begin{array}{cccccccccccc}
1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1
\end{array}\right. \\
\hline
\pi_1 \quad \left|\begin{array}{cccccccccccc}
0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\
1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{array}\right. \quad N-K \\
\pi_2 \\
\hline
\left|\begin{array}{cccccccccccc}
0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\
0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0
\end{array}\right. \\
J = 3 \text{ submatrices}
\end{array}
$$

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

2 The GLD code structure

For simplicity reasons, we consider only RSC codes of rate $r = \frac{k}{k+1}$. The RSC encoder reads k information bits and generates an additional parity bit. The total number of coded bits at its output is denoted by $n = k + 1$. The constraint length is $L = \nu + 1$ and the RSC code trellis has 2^ν states. This convolutional code is defined by n generator polynomials, $g_0(x), g_1(x), \ldots, g_k(x)$. The n output sequences $s_i(x)$ and the k input sequences $e_i(x)$ are related by the following equations

$$s_i(x) = e_i(x) \quad for\ i = 1, \ldots, k \qquad and \quad s_0(x) = \sum_{i=1}^{k} \frac{g_i(x)}{g_0(x)} e_i(x) \tag{1}$$

The PCEs of the RSC code are defined by the syndrome equation easily derived from (1)

$$g_0(x)s_0(x) + g_1(x)s_1(x) + \ldots + g_k(x)s_k(x) = 0 \tag{2}$$

The above equation produces the parity-check matrix H_{RSC} of the convolutional code. As an example, the matrix below is associated to a four-state RSC of rate $r = 1/2$ with generators $g_0 = 7$ and $g_1 = 5$ in octal notation,

$$H_{RSC} = \begin{pmatrix}
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1
\end{pmatrix} \tag{3}$$

Note that this convolutional code ($g_0 = 7, g_1 = 5$) has been converted to a ($N = 16, K_1 = 6$) linear block code. The trellis termination needs 2 branches ($\lceil \frac{\nu}{k} \rceil$ branches in general) and it occupies the last 4

columns and the last 2 rows of H_{RSC}. The even columns are associated to information bits and the odd columns to parity bits. Note also that the PCEs of a convolutional code are not disjoint.
The weight distribution of an RSC code (viewed as an (N, K_1) block code) is computed using the transfer function method described in [5]. The effect of the trellis termination phase can be neglected, i.e. $\frac{K_1}{N} \approx r$. The number of codewords of weight ℓ is denoted $N_1(\ell)$, $\ell = 0 \ldots N$.

Let C_1 be an (N, K_1, d_1) linear binary block code built from an RSC code. A second (N, K_1, d_1) block code $C_2 = \pi(C_1)$ is constructed by a random interleaving of C_1.

Definition (GLD code with two identical constituents)
A GLD code C is an (N, K, d_{Hmin}) linear block code equal to the intersection of C_1 and C_2.
The above definition is similar to that of GLD construction based on block codes [6] such as primitive, extended or shortened BCH codes. It can be easily shown that $R = \frac{K}{N} = 2r - 1$ when the permutation π is random. The average minimum Hamming distance d_{Hmin} is obtained from the average weight distribution given in theorem 1. The structure of the GLD parity-check matrix is illustrated in Figure 2.

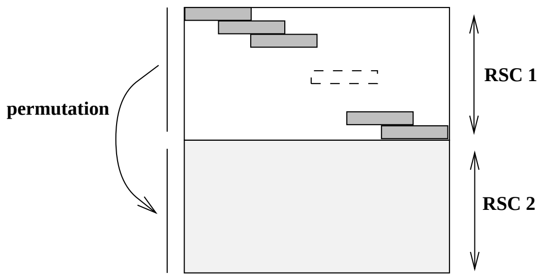

Figure 2: Structure of a GLD parity-check matrix based on two convolutional codes.

The GLD code $C = C_1 \bigcap C_2$ has a simple graphical representation. Each convolutional code C_i is drawn as a chain [7] where a supernode includes the encoder state, the n coded bits and the corresponding channel output (observation). Thus, the Bayesian network [7] of C is obtained by linking the supernodes of the two chains via the interleaver. For example, if $r = 3/4$, each supernode is linked to the opposite code via 4 branches (see Figure 3). The iterative decoding of C is done by propagating the belief in the network (practically by a forward-backward algorithm [8] applied successively on each chain).

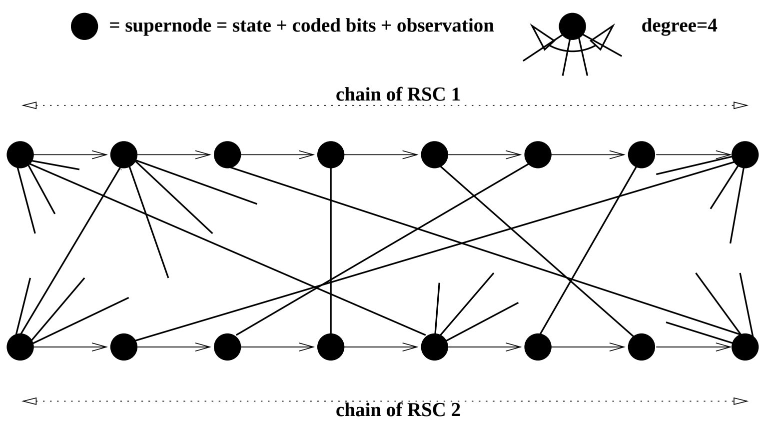

Figure 3: The Bayesian network of a GLD code based on two RSC rate 3/4 codes.

3 Performance analysis of GLD codes

Given a fixed RSC constituent code C_1, the average weight distribution over the whole ensemble of random interleavers is stated by the following theorem.

Theorem 1 (weight distribution)
Let C be an (N, K) GLD code. Then, the average number $N(\ell)$ of codewords of C with weight ℓ is

$$N(\ell) = \frac{N_1(\ell)^2}{\binom{N}{\ell}} \tag{4}$$

where $N_1(\ell)$ is the weight distribution of the constituent RSC code.

Proof. We have $N(\ell) = \binom{N}{\ell} \times P(\ell)$, where $P(\ell)$ is the probability that a weight-ℓ word c chosen at random belongs to C. But if $P_i(\ell)$ is the probability that $c \in C_i$, then $P(\ell) = P_1(\ell)P_2(\ell)$ since $C = C_1 \bigcap C_2$.
By replacing $P_1(\ell) = P_2(\ell) = \frac{N_1(\ell)}{\binom{N}{\ell}}$ we obtain the announced result. *QED.*

Formula (4) has been applied to a rate 1/2 GLD code C based on an 8-state RSC code. The starting tail of the distribution is shown in Table 1. It can be easily shown that $Prob(d_{Hmin} \leq D) \leq \sum_{\ell=d_1}^{D} N(\ell)$. By taking the right hand side of the previous relation equal to 1, we can compute an upper bound for the minimum Hamming distance of C

$$\sum_{\ell=d_1}^{\Delta} N(\ell) = 1 \qquad and \qquad d_{Hmin} \leq \Delta \tag{5}$$

As an example, from Table 1 we obtain $d_{Hmin} \leq 18$.

Weight ℓ	Coefficient $N(\ell)$	Weight ℓ	Coefficient $N(\ell)$
4	3.4E-3	17	0.37
5	1.4E-3	18	0.91
6	8.9E-4	19	2.3
7	6.3E-4	20	6.1
8	1.3E-3	21	16.6
9	2.0E-3	22	46.5
10	3.1E-3	23	134.0
11	4.8E-3	24	396.3
12	8.6E-3	25	1.19E+3
13	1.6E-2	26	3.71E+3
14	3.4E-2	27	1.16E+4
15	7.2E-2	28	3.75E+4
16	0.16	29	1.22E+5

Table 1: The starting tail of the average weight distribution, C_1 is an 8-state (13,7,15,17) RSC.

The output weight distribution is sufficient for computing the bit error probability when Maximum Likelihood (ML) decoding of C. Actually, the interleaver acts on all coded bits, so that they are equally protected. Thus, the input-output enumerating function [9][10] is not needed to evaluate the ML bound. Finally, we can write the following union bound

$$P_{eb} \leq \sum_{\ell=d_1}^{N} \frac{\ell}{N} \times N(\ell) \times Q(\sqrt{R\ell\frac{2E_b}{N_0}}) \tag{6}$$

where E_b/N_0 is the signal-to-noise ratio per bit and $Q(x)$ is the error function.
Figure 4 shows the average ML bound for two values of the code length, $N = 200$ and $N = 800$, when C_1 has an 8-state trellis.

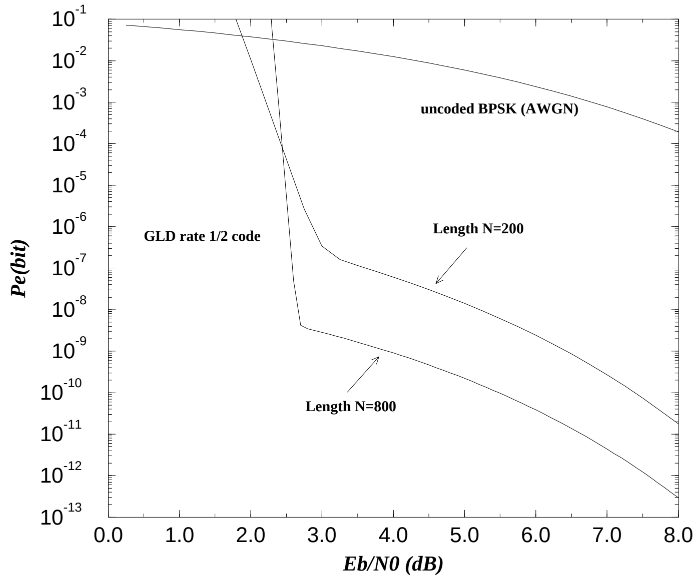

Figure 4: ML Performance of the GLD code built from the 8-state rate 3/4 RSC (13,7,15,17). The GLD code length is $N = 200$ and $N = 800$, total rate $R = 0.5$, on AWGN channel.

Gallager [1] showed that LDPC codes based on simple SPC equations are asymptotically good, i.e. $d_{Hmin} \geq \delta N$ where δ is a positive constant, when $J = 3$ levels. It has been recently proved [6] that Tanner codes based on bipartite graphs and BCH codes are asymptotically good with $J = 2$ levels only. The theorem stated below proves that GLD construction with convolutional codes satisfies the same minimum distance property. Notice that theorems 1 & 2 are not limited to RSC codes and are also valid for non-systematic non-recursive convolutional (NRNSC) codes. In the SISO decoding of a systematic code, the a posteriori probability (APP) depends on the a priori probabilities of information bits and their channel observation. Thus, RSC codes exhibit a slightly better performance than NRNSC codes, when iterative decoding is applied to the whole concatenation.

Theorem 2 (asymptotically good)
The GLD code C built from the rate k/n convolutional code $C_1(N, K_1, d_1)$ is asymptotically good. When N is large enough, the normalized minimum distance $\delta_{min} = d_{Hmin}/N$ is lower bounded by a positive constant δ. The constant δ is the smallest positive (non-zero) root of the equation $B(\lambda) = 0$, where $B(\lambda) > 0$ for $\lambda \in]0 \ldots \delta[$ and

$$B(\lambda) = H(\lambda) - 2\Big(\frac{1}{n} - \frac{\lambda}{n} + \frac{\lambda}{d_1}\Big) H\Big(\frac{\lambda}{\frac{d_1}{n} - \beta\lambda d_1}\Big) \tag{7}$$

$H(x) = -xlog(x) - (1-x)log(1-x)$ is the entropy function and β is a positive constant depending on the convolutional code transfer function.

Proof. We first compute an asymptotic upper bound for $N(\ell)$ from formula (4). When the code length N is large enough, the Stirling approximation gives

$$\frac{1}{\sqrt{2\pi N\lambda(1-\lambda)}} \exp\left(NH(\lambda) - \frac{1}{12N\lambda(1-\lambda)}\right) \le \binom{N}{\ell} \le \frac{1}{\sqrt{2\pi N\lambda(1-\lambda)}} \exp(NH(\lambda)) \tag{8}$$

where $\lambda = \frac{\ell}{N}$ is the normalized Hamming weight, $0 \le \lambda < 0.5$.
By introducing the number $N_1(\ell, e)$ of codewords of C_1 of weight ℓ formed from the concatenation of e consecutive simple error events in the trellis of C_1, we can write the weight distribution of C_1 as a sum over all possible combinations of error events, i.e.

$$N_1(\ell) = \sum_{e=1}^{e_{max}} \binom{\frac{N}{n} - \rho(e,\ell) + e}{e} N_1(\ell, e) \tag{9}$$

where $\rho(e,\ell)$ is the total number of branches in the e error events of total weight ℓ. Moreover $\binom{\frac{N}{n} - \rho(e,\ell) + e}{e}$ is maximal for $e = e_0$, consequently we obtain:

$$N_1(\ell) \le \binom{\frac{N}{n} - \rho(e_0,\ell) + e_0}{e_0} \sum_{e=1}^{e_{\max}} N_1(\ell, e)$$

Using (8), the above equation becomes

$$N_1(\ell) \le A(N,l) \exp\left(\Big(\frac{N}{n} - \rho(e_0,\ell) + e_0\Big) H\Big(\frac{e_0}{\frac{N}{n} - \rho(e_0,\ell) + e_0}\Big)\right)$$

where $A(N,l) = \frac{1}{\sqrt{2\pi e_0\left(1 - \frac{e_0}{\frac{N}{n} - \rho(e_0,\ell) + e_0}\right)}} \sum_{e=1}^{e_{\max}} N_1(\ell, e)$.

Similarly, by substituting equation (9) in (4) we have:

$$N(\ell) \le C(N,\lambda) \exp(-Nf(\lambda)) \tag{10}$$

where the expression of the exponent function is

$$f(\lambda) = H(\lambda) - 2\Big(\frac{1}{n} - \frac{\rho}{N} + \frac{e_0}{N}\Big) H\Big(\frac{e_0}{\frac{N}{n} - \rho(e_0,\ell) + e_0}\Big)$$

According to inequality (10), $f(\lambda)$ can be lowerbounded while keeping the inequality satisfied. The positive part $\frac{1}{n} - \frac{\rho}{N} + \frac{e_0}{N}$ can be maximized as well $\frac{e_0}{\frac{N}{n} - \rho + e_0}$. We need then to bound ρ and to upperbound e_0.

For a given ℓ, the maximal number of simple error events can be upper bounded by $\frac{\ell}{d_1}$, as d_1 is the minimal weight of an error event. The number of branches ρ is simply lowerbounded by $\rho \ge \frac{\ell}{n}$. On the other hand, we can upperbound ρ by $\beta\ell$, where β is a constant given by the trellis (or state diagram) cycle maximizing the ratio ρ/ℓ for C_1. Figures 5, 6 and 7 illustrate this special cycle for three different codes and the resulting value of β.

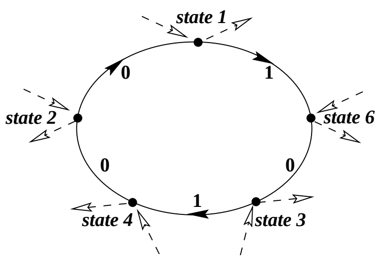

Figure 5: The RSC (13,7,15,17) cycle maximizing ρ/ℓ: cycle length=5, weight=2, $\beta = 5/2$.

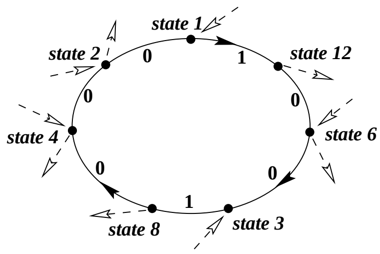

Figure 6: The RSC (23,31,35,37) cycle maximizing ρ/ℓ: cycle length=7, weight=2, $\beta = 7/2$.

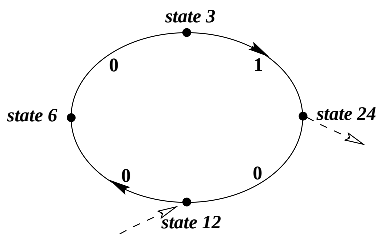

Figure 7: The RSC (45,63,67,75) cycle maximizing ρ/ℓ: cycle length=4, weight=1, $\beta = 4$.

Consequently, we obtain a lower bound for $f(\lambda)$:

$$f(\lambda) \geq H(\lambda) - 2\left(\frac{1}{n} - \frac{\lambda}{n} + \frac{\lambda}{d_1}\right) H\left(\frac{\lambda}{\frac{d_1}{n} - \beta\lambda d_1 + e_0\frac{d_1}{N}}\right) \tag{11}$$

When N is large, $e_0\frac{d_1}{N}$ can be neglected (this has no influence on the starting tail of the weight distribution) and finally we have

$$N(\ell) \leq C(N, \lambda)\exp(-NB(\lambda)) \tag{12}$$

where $B(\lambda)$ is defined by expression (7). Q.E.D.

Expressions of $B(\lambda)$ are given in Table 3 for different constituent RSC codes. $B(\lambda)$ is also sketched on figure 8. The values of δ are listed in the last column of Table 3. Notice that the Gilbert-Varshamov bound produces a minimum distance of $\delta_0 = H_2^{-1}(1-R) = H_2^{-1}(1/2) = 0.11$. We also computed the upperbound Δ from equation (5) for the GLD code based on (13,7,15,17) and for different values of N. Both bounds are shown in Figure (9) where Δ exhibits a linear behavior similar to δN.

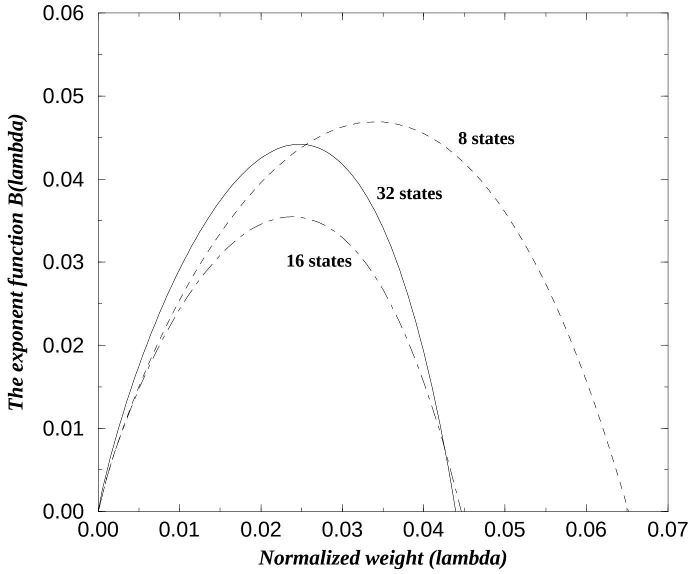

Figure 8: The exponent function $B(\lambda)$ versus the normalized weight λ.

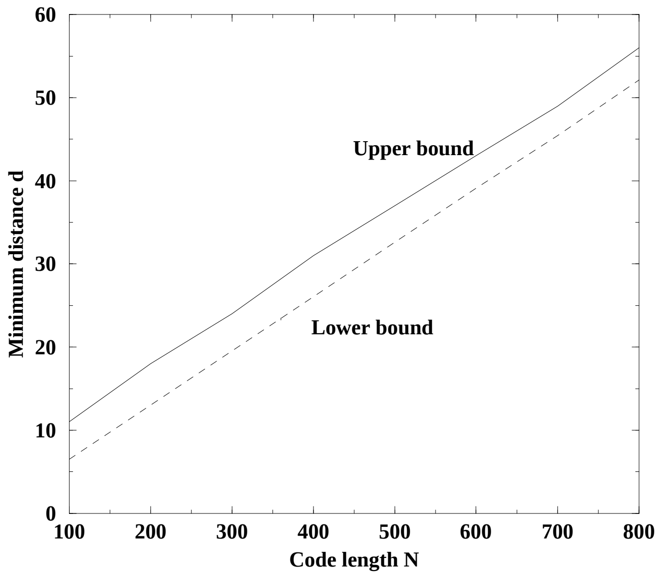

Figure 9: Upper bound and lower bound on the minimum distance.

Number of States	Generator polynomials	$B(\lambda)$	$\delta = \frac{d_{Hmin}}{N}$
8	$13, 7, 15, 17$	$H(\lambda) - \frac{1}{2}H(\frac{\lambda}{1-10\lambda})$	0.0652
16	$23, 31, 35, 37$	$H(\lambda) - \frac{1}{2}H(\frac{\lambda}{1-14\lambda})$	0.0449
32	$45, 63, 67, 75$	$H(\lambda) - \frac{1}{2}(1 - \frac{\lambda}{5})H(\frac{\lambda}{\frac{5}{4}-20\lambda})$	0.0439

Table 2: Lower bound on the minimum distance of three GLD codes.

4 A Gallager-Tanner interpretation of PCCCs and SCCCs

Parallel concatenated convolutional codes (PCCCs) [4] and serial concatenated convolutional codes (SCCCs) [11] can be described as the intersection of two (or more) interleaved convolutional codes.

Let us consider a classical (PCCC) turbo code C with an interleaver of size K and two constituents. A parity-check matrix H_1 can be defined for the first constituent $C_1(N_1, K)$. The $(N_1 - K) \times N_1$ matrix H_1 is similar to that given by equation (3), but the columns associated to information bits are now grouped together on the left side. Next, extend C_1 by adding $N_1 - K$ zero columns to H_1 (at the right side). This extended code is denoted C_{1ext}. The parity-check matrix H_{1ext} is written horizontally as 3 blocks : the first block of size $(N_1 - K) \times K$ defines the part of the PCEs associated to information bits, the second block of size $(N_1 - K) \times (N_1 - K)$ corresponds to parity check bits and the last block is null.
The turbo code C is equal to the GLD code obtained from the intersection of C_{1ext} and $C_2 = \pi(C_{1ext})$, where the special interleaver π acts randomly on the K columns of the first block of H_{1ext} and permutes the second and the third blocks.

Let us now consider a serial (SCCC) turbo code C with an interleaver of size N_1 and two constituents. The parity-check matrix H_1 of C_1 (the outer code) is of size $(N_1 - K) \times N_1$. Extend H_1 by adding $N - N_1$ zero columns and denote the extended constituent by C_{1ext}. H_{1ext} has the same structure as described above for the parallel turbo code. The inner code $C_2(N, N_1)$ has a parity-check matrix H_2 with no zero columns, where the N_1 columns associated to information bits are grouped in the left side.
The serial turbo code C is equal to the GLD code obtained from the intersection of C_{1ext} and $\pi(C_2)$, where the special interleaver π acts on the N_1 first columns of H_2.

In both cases, parallel and serial concatenations, the interleaver of a GLD code equivalent to a turbo code does not act on all coded bits. Thus, the formula $R = 2r - 1$ is no more valid. We have $R = r/(2 - r)$ and $R = r_1 r_2$ for PCCCs and SCCCs respectively.
PCCCs exhibit an interleaver gain of $\frac{1}{N}$ when both constituents are RSC codes [9]. SCCCs exhibit an interleaver gain of $\frac{1}{N^{(1+d_1)/2}}$ when the inner code is RSC. Unfortunately, using the proof of theorem 2, it can be easily shown that the bit error probability of a GLD code with a random interleaving of all coded bits is proportionnal to $\sqrt{N}$ at least, i.e. GLD codes do not exhibit an interleaving gain.

On the other hand, the specific structure of the interleaver makes the results on GLD code performance inappropriate for turbo codes. Thus, GLD codes are good for the minimum distance criterion, but it is well known that Turbo codes are not asymptotically good in this sense [9][10][11].

5 Simulation results

The iterative decoding of GLD codes is similar to the SISO decoding of turbo codes. Simulation results presented here are obtained by applying the modified forward-backward algorithm to the constituent code C_1 and its interleaved version. Figures 10, 11, 12, 13 show the bit error rate function of the signal-to-noise ratio E_b/N_0 on a gaussian channel. Two code lengths have been tested, $N = 800$ and $N = 2000$. For each length, two different constituent codes have been compared. The first code C_1 has a rate $r = 3/4$

and is obtained by puncturing the rate 1/2 16 state (23,35) RSC code. The second code C_1 is a true rate 3/4 8 state (13,7,15,17) RSC code.

6 Conclusions

We built generalized low density parity-check codes from the intersection of two randomly interleaved convolutional codes. These codes belong to the Tanner family based on bipartite graphs. It has been proved that such GLD codes are asymptotically good but do not have an interleaving gain. Their average minimum Hamming distance is relatively high, e.g. $52 \leq d_{Hmin} \leq 56$ for a code length $N = 800$ based on an 8 state RSC code. We showed also that parallel and serial turbo codes can be viewed as a special case of GLD codes.

References

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

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

[3] L.H. Charles Lee : Convolutional coding, fundamentals and applications, Artech House, 1997.

[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.

[5] D. Divsalar, S. Dolinar, F. Pollara, R.J. McEliece: "Transfer function bounds on the performance of Turbo codes," *TDA Progress Report* 42-122, August 1995.

[6] J. Boutros, O. Pothier, G. Zémor : "Generalized Low Density (Tanner) Codes : Approaching the channel capacity with simple and easily decodable block codes," *ENST - Philips Research Report*, January 1998, also to appear in ICC'99.

[7] B.J. Frey and F.R. Kschischang : "Probability propagation and iterative decoding," *Allerton Conference*, October 1996.

[8] L.R. Bahl, J. Cocke, F. Jelinek, J. Raviv : "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. on Inf. Theory*, vol. 20, pp. 284–287, March 1974.

[9] S. Benedetto, G. Montorsi : "Design of parallel concatenated convolutional codes," *IEEE Trans. Com.*, vol. 44, no. 5, pp. 591-600, May 1996.

[10] L.C. Perez, J. Seghers, D.J. Costello : "A distance spectrum interpretation of turbo codes," *IEEE Trans. on Inf. Theory*, vol. 42, no. 6, pp. 1698-1709, November 1996.

[11] S. Benedetto, G. Montorsi, D. Divsalar, F. Pollara : "Serial concatenation of interleaved codes : Performance analysis, design and iterative decoding," *TDA Progress Report 42-126*, JPL, August 1995.

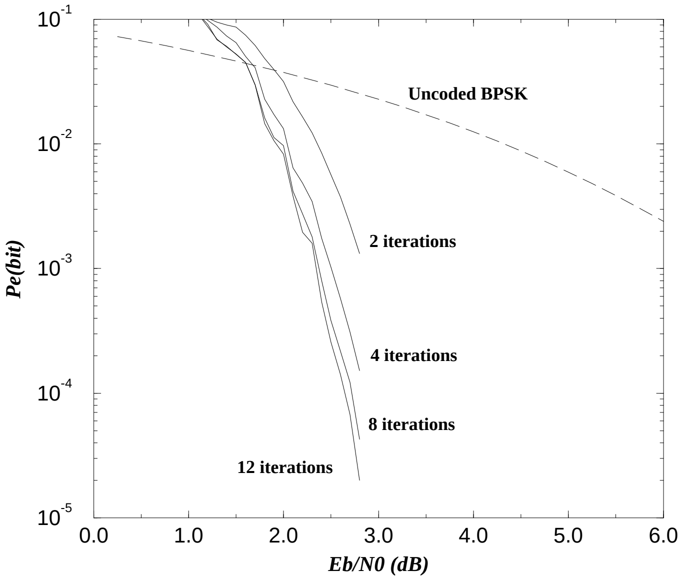

Figure 10: Iterative decoding of a GLD code, $R = 0.5$, C_1 is a 16 state punctured (23,35) RSC, N=800.

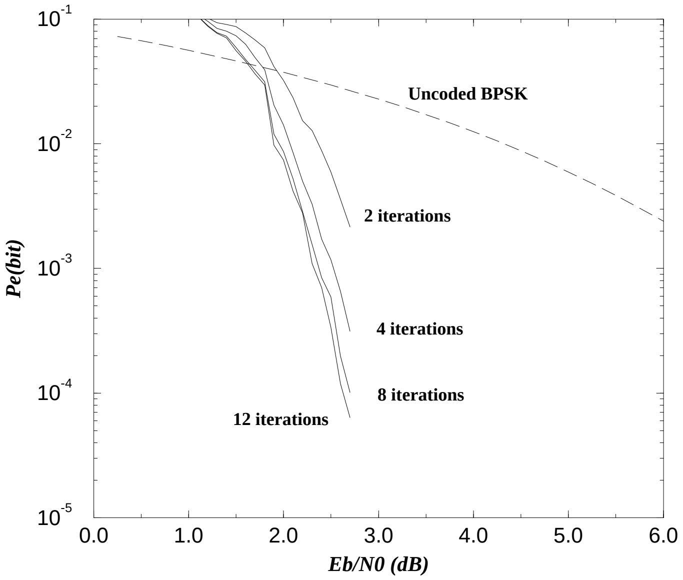

Figure 11: Iterative decoding of a GLD code, $R = 0.5$, C_1 is an 8 state (13,7,15,17) RSC, N=800.

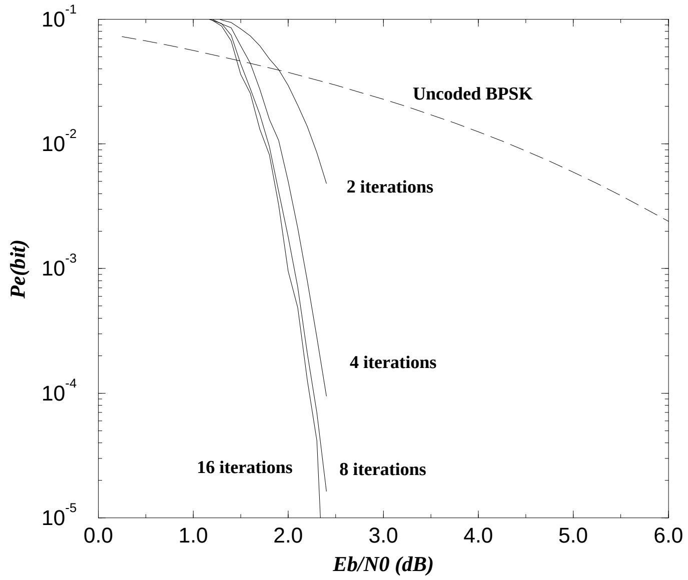

Figure 12: Iterative decoding of a GLD code, $R = 0.5$, C_1 is a 16 state punctured (23,35) RSC, N=2000.

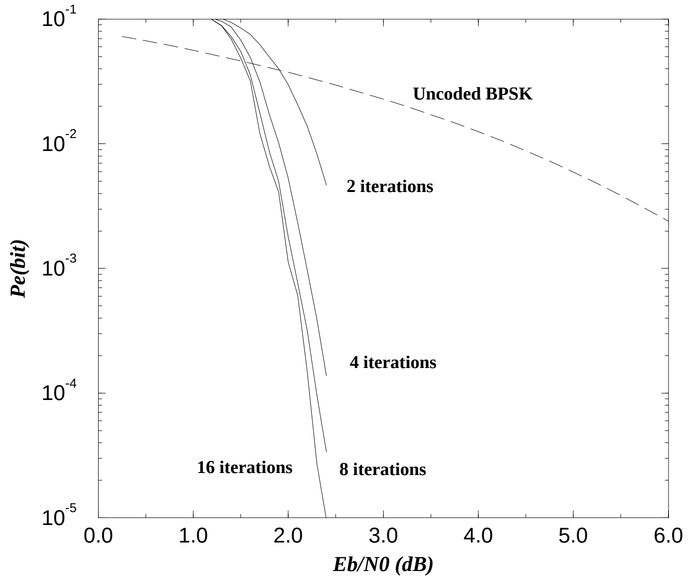

Figure 13: Iterative decoding of a GLD code, $R = 0.5$, C_1 is an 8 state (13,7,15,17) RSC, N=2000.