

## Achieving a near-optimum erasure correction performance with low-complexity LDPC codes

Gianluigi Liva<sup>1,\*,\dagger</sup>, Balazs Matuz<sup>1</sup>, Enrico Paolini<sup>2</sup> and Marco Chiani<sup>2</sup>

<sup>1</sup>*Institute of Communication and Navigation, Deutsches Zentrum für Luft- und Raumfahrt (DLR),  
82234 Wessling, Germany*

<sup>2</sup>*DEIS/WiLAB, University of Bologna, via Venezia 52, 47521 Cesena (FC), Italy*

### SUMMARY

Low-density parity-check (LDPC) codes are shown to tightly approach the performance of idealized maximum distance separable (MDS) codes over memoryless erasure channels, under maximum likelihood (ML) decoding. This is possible down to low error rates and even for small and moderate block sizes. The decoding complexity of ML decoding is kept low thanks to a class of decoding algorithms, which exploit the sparseness of the parity-check matrix to reduce the complexity of Gaussian elimination. ML decoding of LDPC codes is reviewed at first. A performance comparison among various classes of LDPC codes is then carried out, including a comparison with fixed-rate Raptor codes for the same parameters. The results confirm that a judicious LDPC code design allows achieving a near-optimum performance over the erasure channel, with very low error floors. Furthermore, it is shown that LDPC and Raptor codes, under ML decoding, provide almost identical performance in terms of decoding failure probability vs. overhead. Copyright © 2010 John Wiley & Sons, Ltd.

KEY WORDS: LDPC codes; Raptor codes; LT codes; binary erasure channel; maximum likelihood decoding; Gaussian elimination; MBMS; packet-level coding

### 1. INTRODUCTION

Low-density parity-check (LDPC) codes [1] exhibit an extraordinary performance under iterative (IT) decoding over a wide range of communication channels. Some classes of LDPC code ensembles under IT decoding can even asymptotically approach with an arbitrarily small gap the binary erasure channel (BEC) capacity [2–4]. Some problems, however, arise when constructing finite length  $(n, k)$  LDPC codes according to the so far proposed asymptotically optimal ensembles. Owing to the sub-optimality of IT decoding, at high error rates the performance curve, although usually good, denotes a coding gain loss with respect to that of

---

\*Correspondence to: Gianluigi Liva, Institute of Communication and Navigation, Deutsches Zentrum für Luft- und Raumfahrt (DLR), 82234 Wessling, Germany.

<sup>\dagger</sup>E-mail: gianluigi.liva@dlr.de

Contract/grant sponsor: European Community; contract/grant number: IST-27393

the same code under maximum likelihood (ML) decoding. Moreover, at lower error rates the performance curve typically exhibits a high error floor caused by the presence of small size stopping sets [5]. In general, lowering the IT error floor implies a sacrifice in terms of coding gain at high error rates.

Let us consider a  $(n, k)$  binary linear block code  $C$ , where  $n$  is the codeword length and  $k$  is the code dimension. Let  $\mathbf{H}$  be a parity-check matrix for the code and  $\mathbf{x} \in C$  be a generic codeword, such that  $\mathbf{x}\mathbf{H}^T = 0$ . Decoding of  $C$  over the BEC is equivalent to solving the system of linear equations [6]

$$\mathbf{x}_{\bar{K}}\mathbf{H}_{\bar{K}}^T = \mathbf{x}_K\mathbf{H}_K^T. \quad (1)$$

In (1),  $\mathbf{x}_{\bar{K}}$  denotes the set of erased codeword bits (system unknowns), while  $\mathbf{x}_K$  denotes the set of non-erased codeword bits. Analogously,  $\mathbf{H}_{\bar{K}}$  is the matrix composed of the columns of  $\mathbf{H}$  corresponding to  $\mathbf{x}_{\bar{K}}$ , while  $\mathbf{H}_K$  is the matrix composed of the columns of  $\mathbf{H}$  corresponding to  $\mathbf{x}_K$ . Different decoding algorithms attempt to solve (1) with different approaches and with a different complexity. Among them, ML decoding consists of solving (1) by Gaussian elimination (GE) performed on the matrix  $\mathbf{H}_{\bar{K}}$ . Its complexity is in general cubic with the dimension of the system, so that the overall complexity is  $O(n^3)$ . In general, ML decoding becomes impractical for large block lengths. For LDPC codes, IT decoding consists of attempting to solve (1) by recursively processing one equation at time. This approach is in general sub-optimum but has the advantage to exhibit a linear  $O(n)$  complexity [7]. It is important to point out that, for LDPC codes, ML decoding is feasible for longer codes than random codes (e.g.  $n$  up to thousands of symbols), taking advantage of the sparseness of the parity-check matrix. Efficient ways of implementing ML decoders for LDPC codes exploiting the parity-check matrix sparseness can be found, for example, in [8].

Failures of the IT decoder over the BEC are due to stopping sets. Because there exist sets of codeword bit positions that represent stopping sets for the IT decoder, but not for the ML decoder, a decoding strategy more powerful than simple IT decoding consists of performing IT decoding at first and, on an IT decoder failure, employing the ML decoder to try to resolve the residual stopping set (for instance, see [9]). This *hybrid IT/ML decoder* achieves the same performance as an ML decoder. The performance curve obtained after the ML step achieves a higher coding gain than IT decoding and a lower error floor. In fact, the subset of variable nodes (VNs) corresponding to the support of any codeword<sup>‡</sup> is a stopping set for the IT decoder [10]. These stopping sets cannot be resolved by an ML decoder as well because they lead to a matrix  $\mathbf{H}_{\bar{K}}$  whose rank is smaller than the number of unknowns of (1). On the other hand, the stopping sets of the IT decoder that do not include the support of any codeword can be resolved by an ML decoder: the error floor under ML decoding depends solely on the distance spectrum of the code.

A thorough performance analysis of LDPC codes under reduced-complexity ML decoding is provided in this article. A class of fixed-rate Raptor codes is used as a benchmark for these performance evaluations. Fixed-rate Raptor codes are obtained from the rate-less codes recommended for the Multimedia Broadcast Multicast Service (MBMS) by selecting *a priori* the codeword length  $n$ . Raptor codes are universally recognized as the state-of-the-art codes for erasure channels, and they have been under investigation for fixed-rate applications within the

<sup>‡</sup>The set of positions of a binary vector corresponding to the '1' bits is known as the support of the vector.

Digital Video Broadcasting (DVB) standards family [11] and have been included in the informative part of the multiprotocol encapsulation (MPE)-iFEC specification [12]. As for LDPC codes, also for Raptor codes efficient ML decoders are available [13].

The article is organized as follows. Applications of the proposed approach are briefly discussed in Section 2. Reduced-complexity ML decoding of LDPC codes over erasure channels is reviewed in Section 3, including code design issues. A class of fixed-rate Raptor codes is introduced in Section 4, together with a summary on their ML encoder/decoder implementations. Section 5 provides simulation results for both LDPC and fixed-rate Raptor codes. Conclusions follow in Section 6.

## 2. APPLICATIONS

The main application of erasure correcting codes in wireless communication systems is strictly related to the concept of packet loss recovery. Let us focus on the simple case of linear block codes. At the physical layer, a linear block code is usually employed to protect a frame by adding redundancy. This redundancy is exploited on the receiver side to recover from errors introduced by the communication channel. A  $(n, k)$  linear block code can be used also at higher layers of the protocol stack, to counteract packet losses. Here, the *symbols* the code operates on are not bits, but packets of bits all having the same length<sup>§</sup>. Therefore, the code is usually referred to as a *packet-oriented erasure correcting code* (or as a *packet erasure codes*). As an example,  $k$  data packets can be encoded through a systematic  $(n, k)$  packet-oriented code, producing  $(n-k)$  redundant packets. Each of the  $n$  packets is encoded into a physical layer frame by a bit-oriented error-correcting code (e.g. convolutional code, Reed-Solomon code, turbo code, etc.) and by an error detection code such as a cyclic redundancy check (CRC) code. On the receiver side, after physical layer decoding, the undecodable frames are usually discarded, leading to packet losses at the upper layers. However, if the number of lost packets is sufficiently small with respect to the number of redundant packets, the missing packets can be restored by the erasure decoder.

Packet erasure codes are usually adopted in wireless communications when link disruptions or long fading events take place, and whenever retransmissions are either impossible or highly expensive in terms of resources. Examples of applications for which the use of packet erasure codes is foreseen are listed next.

- *Wireless video/audio streaming.* Link-layer coding is currently applied to the video streams within the framework of the DVB-H/SH standards. In such a context, packet erasure codes take care of fading mitigation, which is crucial especially in the case of mobile users, in challenging propagation environments (urban/suburban and land-mobile satellite channels). Recall that, in such cases, the physical layer forward error-correction scheme may not provide time diversity sufficient to cope with the coherence time of the channel (unless long-time interleavers are employed, with obvious drawbacks in terms of signal processing complexity). A capacity-approaching performance is highly desirable to increase the service availability. Mobile applications require low-complexity decoders as well.

---

<sup>§</sup>Throughout the article, by *symbol*, we denote either a bit or a packet of bits. In this article, we will usually assume a bit-oriented perspective, the extension to packets of bits being straightforward.

- *File delivery in broadcasting/multicasting networks.* Reliable file delivery in broadcasting/multicasting networks finds a very favorable solution in erasure correcting codes. In such a scenario, reliability cannot be guaranteed by any automatic repeat request (ARQ) mechanism, due to the broadcast nature of the channel. Packet erasure codes would limit (or avoid) the usage of packet retransmissions, leading to a more efficient use of the available bandwidth.
- *File delivery in point-to-point communications.* Also in point-to-point links, file delivery may require further protection at upper layers. This is true especially if retransmissions are impossible (due to the absence of a return channel or due to long round-trip delays).
- *Deep space communications.* Deep space communication has been always an ideal application field for error-correcting codes. The Consultative Committee for Space Data Systems is currently investigating the adoption of packet erasure codes to further protect the telemetry down-link, especially for deep space missions, which are not suitable for ARQ, and in optical down-links, which are affected by bursty signal outages due to the turbulence of the propagation medium and/or to pointing errors. Within these contexts, the possibility of processing the data offline, together with the relatively low-data rates (up to some Mbps), makes ML decoding of linear block codes a concrete solution, even in the absence of low-complexity decoders. A mandatory feature is instead represented by low-complexity encoder implementations.

### 3. LDPC CODES AND ML DECODING

This section is organized in a twofold way. First, the main concepts of the ML decoder of [8] will be explained. Second, new code designs for ML decoding will be presented and evaluated with regard to complexity and performance.

#### 3.1. Efficient ML decoding of LDPC codes over erasure channels

Efficient implementation of GE on sparse matrices of large size and constructed on finite fields is a widely investigated topic (e.g. [14, 15]). An effective approach, sometimes referred to as *structured Gaussian elimination*, consists of converting the system of sparse linear equations into a non-sparse system whose unknowns form a small subset of the original set of unknowns. These unknowns are referred to as the *pivots* or *reference variables* and are chosen in such a way that their knowledge is sufficient to resolve all the other unknowns by simple back-substitution operations. Therefore, to solve the linear system, it is sufficient to run a brute-force GE only to decode the pivots<sup>†</sup>.

Within the framework of ML decoding of LDPC codes over erasure channels, the idea of structured GE has been applied in [8]. A brief overview of this approach is reviewed next. For the sake of clarity, and without losing generality, let us apply column permutations to arrange the parity-check matrix  $\mathbf{H}$  as in (1): the left part shall contain all the columns related to known VNs ( $\mathbf{H}_K$ ), whereas the right part shall be made up of all the columns related to erased VNs ( $\mathbf{H}_{\bar{K}}$ ). Thus, to solve the unknowns, we proceed as follows [8]:

<sup>†</sup>It is worth observing that ML decoding over a binary erasure channel can also be performed using a bit guessing approach [16]. This approach, however, is not practical over packet erasure channels, as in this case all the bits composing an erased packet should be guessed.

- Perform diagonal extension steps on  $\mathbf{H}_{\bar{k}}$ . This results in the submatrices  $\mathbf{B}$ , as well as  $\mathbf{P}$  that is in a lower triangular form, and columns that cannot be put in lower triangular form (columns of matrices  $\mathbf{A}$  and  $\mathbf{S}$ ). The VNs corresponding to the latter set of columns form the above-mentioned pivots (or reference variables) (see Figure 1(a)).
- Zero out the matrix  $\mathbf{B}$  through row summations only. All the remaining unknown variables can be now obtained by linear combination of pivots and known variables only (Figure 1(b)).
- Resolve the pivots by performing brute-force GE only on the rows of the system involving  $\mathbf{A}'$  (Figure 1(c)). If the pivot recovery step has been successful, the remaining unknown variables can be easily obtained thanks to the lower triangular structure of  $\mathbf{P}$ .

The main strength of this algorithm lies in the fact that GE is performed only on  $\mathbf{A}'$  and not on the entire set of unknown variables. Therefore, it is of great interest to keep the dimensions of  $\mathbf{A}'$  as small as possible. This can be obtained by both sophisticated ways of choosing the pivots [8] and by a judicious code design [9]. Besides, to reduce the complexity further, the brute-force GE step on  $\mathbf{A}'$  could be replaced by matrix decomposition algorithms [17].

Note that the ML decoder for an  $(n, k)$  LDPC code operates on a sparse matrix with at most  $n-k$  columns and  $n-k$  rows. The relevance of this consideration will become more clear after the description of the ML Raptor decoder [13] provided in Section 4.

### 3.2. On the code design

Let us consider an LDPC code and its Tanner graph. Let us denote by  $\lambda_i$  and  $\rho_i$  the fractions of edges incident on the VNs of degree  $i$  and on the check nodes (CNs) of degree  $i$ , respectively. Moreover, let us define the polynomials  $\lambda(x) = \sum_{i \geq 2} \lambda_i x^{i-1}$  as the VN degree distribution from an edge perspective and  $\rho(x) = \sum_{i \geq 2} \rho_i x^{i-1}$  as the CN degree distribution from an edge perspective, respectively. Then, the LDPC code ensemble  $C(n, \lambda, \rho)$  is defined as the set of all LDPC codes of length  $n$  sharing the same polynomials  $\lambda(x)$  and  $\rho(x)$  [18]. The pair  $(\lambda, \rho)$  is usually referred to as the degree distribution pair. For any LDPC code in the ensemble, the code rate fulfills  $R \geq 1 - \int_0^1 \rho(x) dx / \int_0^1 \lambda(x) dx$ , where the right hand side of this inequality is known as the design rate of the ensemble. Note that an LDPC code ensemble may be described also from a node perspective, through the polynomials  $A(x) = \sum_{i \geq 2} A_i x^i$  and  $P(x) = \sum_{i \geq 2} P_i x^i$ . Here,  $A_i$  and  $P_i$  represent the fractions of VNs and CNs of degree  $i$ , respectively.

Consider an LDPC code ensemble  $C(n, \lambda, \rho)$  under IT decoding, and assume transmission over the BEC. Assume the codeword length tends to infinity. Then, the asymptotic threshold  $\varepsilon_{IT}$

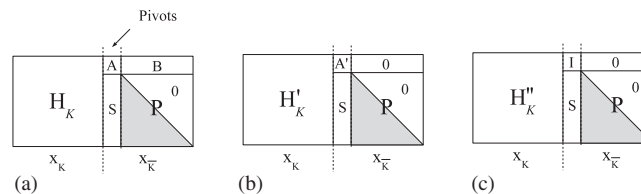


Figure 1. ML decoder as in [8]. (a) Pivots selection within  $\mathbf{H}_{\bar{k}}$ . (b) Zeroing of matrix  $\mathbf{B}$ . (c) GE on  $\mathbf{A}'$ .

is defined as the supremum value of the channel erasure probability  $\varepsilon$ , such that the bit error probability tends to zero as the number of decoding iterations tends to infinity [18]. Similarly, we can define an asymptotic threshold  $\varepsilon_{\text{ML}}$  under ML decoding [19].

A common way to design an LDPC code, with a certain code rate  $R$ , for the BEC consists of the selection of a proper degree distribution pair (or protograph [20]) with design rate  $R$  and offering a satisfying IT decoding threshold  $\varepsilon_{\text{IT}}$  subject to eventual further constraints (in terms, for instance, of growth rate of the stopping set size distribution [21]). An  $(n, nR)$  LDPC code is then picked from the ensemble defined by the above-mentioned degree distribution pair (or protograph). The selection may be performed after some girth optimization techniques. Such a design technique does not necessarily address the need to a good LDPC code for ML decoding. For example, we can look at Table I, where the asymptotic thresholds over the BEC of some regular LDPC ensembles (with rate 1/2 and 2/3), under both IT decoding ( $\varepsilon_{\text{IT}}$ ) and ML ( $\varepsilon_{\text{ML}}$ ) decoding are reported. For both rates, we see that the larger  $\varepsilon_{\text{IT}}$  the smaller  $\varepsilon_{\text{ML}}$ . Note that the improvement given by the ML decoder is usually large.

Therefore, within the context of LDPC code design for ML decoding, a different figure shall be put in the focus of the degree distribution optimization, namely, the ML decoding threshold  $\varepsilon_{\text{ML}}$ . A method for deriving a tight upper bound on the ML threshold for an LDPC ensemble has been developed in [19]. For a broad set of ensembles (including the regular ones and the irregular ones whose extrinsic information transfer (EXIT) curve presents a single jump [19]), the upper bound on  $\varepsilon_{\text{ML}}$  can be derived as follows.

- Consider an LDPC code randomly drawn from the ensemble  $C(n, \lambda, \rho)$ . The EXIT function [22] under IT decoding can be derived in terms of *extrinsic erasure probability* at the output of the decoder (denoted by  $p_E$ ) as a function of the *a priori erasure probability* input to the decoder (denoted by  $p_A$ ). In the limit where  $n \rightarrow \infty$ , the EXIT function of the ensemble defined by the degree distribution pair  $(\lambda, \rho)$  is a function of  $(\lambda, \rho)$ . It can be expressed in parametric form by the pair of simultaneous equations

$$p_A = \frac{x}{\lambda(1 - \rho(1 - x))}, \quad (2)$$

$$p_E = A(1 - \rho(1 - x)), \quad (3)$$

where  $x \in [x_{\text{BP}}, 1]$ ,  $x_{\text{BP}}$  being the value of  $x$  for which  $p_A = \varepsilon_{\text{BP}}$ , and where  $A(x) = \sum A_i x^i$ ,  $A_i$  being the fraction of degree- $i$  VNs. The EXIT functions of two regular LDPC code ensembles are displayed in Figure 2 (dashed lines).

Table I. ML and IT decoding threshold for regular LDPC ensembles vs. the Shannon limit,  $\varepsilon_{\text{Sh}}$ .

Ensemble	$\varepsilon_{\text{ML}}$	$\varepsilon_{\text{IT}}$	$\varepsilon_{\text{Sh}}$
(3, 6)	0.4881	0.4294	0.5000
(4, 8)	0.4977	0.3834	0.5000
(5, 10)	0.4994	0.3416	0.5000
(6, 12)	0.4999	0.3075	0.5000
(3, 9)	0.3196	0.2828	0.3333
(4, 12)	0.3302	0.2571	0.3333
(5, 15)	0.3324	0.2303	0.3333

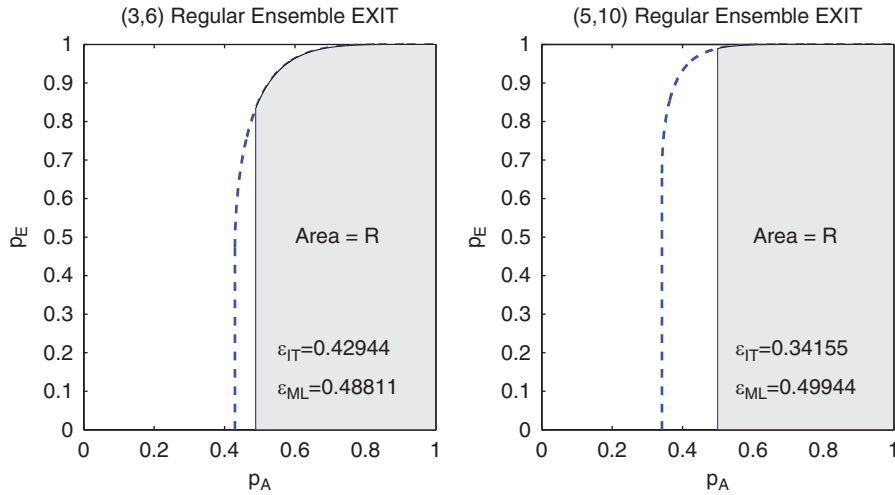


Figure 2. EXIT functions for the (3,6) and the (5,10) regular LDPC ensembles. Dashed lines represent the IT decoder EXIT functions. Solid lines are placed in correspondence of the ML thresholds upper bounds.

- Owing to the area theorem [23], the area below the EXIT function under ML decoding, must equal the code rate  $R$ . As the EXIT function defined by (2) and (3) assumes IT decoding, the area below the corresponding EXIT curve is equal to or larger than the code rate.
- Consider the extrinsic erasure probability at the output of an ML and of an IT decoder. Obviously,  $p_E^{ML} \leq p_E^{IT}$ .
- Therefore, by drawing a vertical line on the EXIT function plot of the ensemble, in correspondence with  $p_A = p_A^*$  and such that

$$\int_{p_A^*}^1 p_E(p_A) dp_A = R,$$

we obtain an upper bound on the ML threshold, i.e.  $\varepsilon_{ML} \leq p_A^*$  (cf. Figure 2).

It was illustrated in [19] that this bound on  $\varepsilon_{ML}$  is very tight for regular LDPC ensembles, and for ensembles whose EXIT curve under IT decoding presents one *jump* (for further details, see [19]). It was also shown how slightly different (but still rather simple) techniques to obtain tight bounds are applicable also in the other cases. Extensions of the above-mentioned techniques can be applied to other code ensembles, provided the EXIT curve under IT decoding is available. For example, for protograph LDPC ensembles, a rather simple approach consists of applying EXIT analysis for protograph to obtain the EXIT curve under IT decoding for a given protograph ensemble [24]. The upper bound on the ML threshold can then be obtained as for the standard LDPC ensemble defined by a degree distribution pair. An example of the EXIT curve under IT decoding for an accumulate-repeat-accumulate (ARA) ensemble [25] is depicted in Figure 3 (specifically, for the ARA ensemble with repetition rate 3, referred as AR3A ensemble), together with the derivation of the corresponding upper bound on  $\varepsilon_{ML}$ . Proofs of the tightness of the bound for protograph ensembles are beyond the scope of this article. For the regular ensembles, the improvement given by the ML decoder is usually large (see Table I).

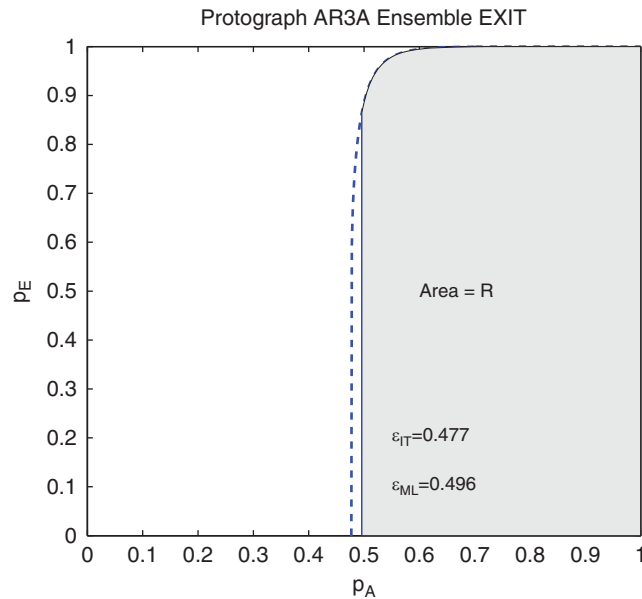


Figure 3. EXIT function for an accumulate-repeat-accumulate (ARA) ensemble. Dashed line represents the IT decoder EXIT function. Solid line is placed in correspondence of the ML thresholds upper bounds.

A rule of thumb for the design of capacity-approaching LDPC codes under ML decoding consists in the selection of sufficiently dense parity-check matrices, by keeping for instance a relatively large average check node degree. To give an idea, we found that for rate 1/2 LDPC ensembles, an average check node degree  $d_c \geq 9$  is sufficient to provide ML thresholds close to the Shannon limit [9]. This heuristic rule seems to work for both regular and irregular ensembles.

### 3.3. Generalized irregular repeat accumulate codes with low-complexity ML decoder

In [9] it is discussed that for LDPC codes a good IT decoding threshold is desirable under hybrid IT/ML decoding as it helps to reduce the decoder complexity. More specifically, in [9] some simple design rules are provided, leading to codes with good IT thresholds, near-Shannon-limit ML thresholds, low error floors, with simple (turbo-code-like) encoders. The proposed code design leads to a class of generalized irregular repeat accumulate (GeIRA) codes [26] tailor-made for efficient ML decoding. Numerical results on GeIRA codes with different coding rates/block lengths are provided in Section 5.

## 4. FIXED-RATE RAPTOR CODES AND ML DECODING

Raptor codes were introduced by Shokrollahi in [27]. They are an instance of the concept of *fountain code*<sup>||</sup> [28] and, thanks to the large degrees of freedom in parameter choice, they can be

<sup>||</sup>Commonly, the expression ‘fountain code’ is used to refer to a code which can produce on-the-fly any desired number of encoded symbols from  $k$  information symbols.

applied to several systems, increasing their reliability. Recently, a fully specified version of Raptor codes has been approved to efficiently disseminate data over a broadcast network (MBMS service [13]). A  $(n, k)$  fixed-rate Raptor code can be obtained by limiting to  $n$ , the amount of symbols produced by the Raptor encoder. Fixed-rate Raptor codes derived from the MBMS standard have been investigated for the MPE level protection within the DVB standards family and have been included in the informative part of the MPE-iFEC standard [11]. In the following, we will provide first a description of the Raptor codes specified in [13], including some insights on their encoding and recommended decoding algorithms.

The Raptor code can be viewed as the concatenation of several codes. Let us consider the systematic Raptor encoder specified in [13] that is also depicted in Figure 4. The innermost code is a non-systematic Luby-transform (LT) code [29] with  $L$  input symbols  $\mathbf{F}$ , producing the encoded symbols  $\mathbf{E}$ . The symbols  $\mathbf{F} = [\mathbf{D}^T | \mathbf{D}_s^T | \mathbf{D}_h^T]^T$  are known as the *intermediate symbols*, and are generated through a pre-coding, made up of some outer high-rate block coding, performed on the  $k$  symbols  $\mathbf{D}$ . The  $s$  intermediate symbols  $\mathbf{D}_s$  are known as the *LDPC symbols*, while the  $h$  intermediate symbols  $\mathbf{D}_h$  are known as the *half symbols*. The combination of pre-code and LT code produces a non-systematic Raptor code. The parameters  $s$  and  $h$  are functions of  $k$ , according to [13]. Some pre-processing has to be put before the non-systematic Raptor encoding to obtain a systematic one. Such a pre-processing consists of a rate-1 linear code generating the  $k$  symbols  $\mathbf{D}$  from the  $k$  information symbols  $\mathbf{C}$ .

LT codes are the first practical implementation of fountain codes. A unique encoded symbol ID (ESI) is assigned to each encoded symbol. Starting from an ESI  $i$ , the encoded symbol  $E_i$  is computed by xor-ing a subset  $\Theta_i$  of  $d_i$  intermediate symbols. The number  $d_i$ , known as the *degree* associated with the encoded symbol  $E_i$  is a random integer between 1 and  $L$ : the  $d_i$  intermediate symbols are chosen at random according to a specific probability distribution. As a consequence, to recover the information symbols the decoder needs both the set of encoded symbols  $E_i$  and of the corresponding  $\Theta_i$ . This last information can either be explicitly transmitted or obtained by the decoder through the same pseudo-random generator used for the encoding, starting from ESIs, which have therefore to be sent together with the corresponding encoded symbols (as depicted in Figure 4).

Some of the main properties of LT codes are that the encoder can generate as many encoded symbols as desired and that the decoder is able to recover the block of source symbols from any set of received encoded symbols whose number is only slightly greater than that of the source symbols (in fact the code claims a low amount of overhead). A Raptor code, whose core consists of an LT code, inherits such properties.

#### 4.1. Fixed-rate Raptor generator matrix

Considering a systematic Raptor code as a finite length  $(n, k)$  linear block code (fixed-rate Raptor code), we can ask what is the structure of its generator matrix. This problem is addressed next for the Raptor code specified in [13].\*\* The generator matrix of the first pre-coding stage is given by  $[\mathbf{I}_k | \mathbf{G}_{\text{LDPC}}^T]^T$ . According to the specifications in [13],  $\mathbf{G}_{\text{LDPC}}$  consists of columns all of weight equal to 3, regardless the value of  $k$ . On the other hand, the generator matrix of the second pre-coding stage is given by  $[\mathbf{I}_{s+k} | \mathbf{G}_H^T]^T$ , where  $\mathbf{G}_H$  is a  $(h \times (s+k))$  matrix consisting of

\*\*Throughout this section, the vectors are intended as column vectors (if not stated differently) and the generator matrix of a  $(n, k)$  linear block code is expressed as a  $(n \times k)$  matrix.

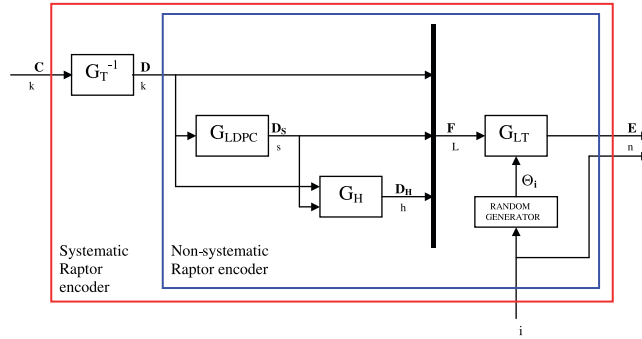


Figure 4. Block diagram of the systematic Raptor encoder specified in [13].

columns all of constant weight: each column is an element of the Grey sequence of weight  $h'$ , where  $h' = \lceil h/2 \rceil$ . Finally, let us denote by  $\mathbf{G}_{LT}$  the  $(n \times L)$  LT code generator matrix (regarded as a finite length  $n$  linear block code). It is built in such a way that the row of index  $i$  has  $d_i$  ones in  $\Theta_i$  positions, where  $d_i$  and  $\Theta_i$  are derived from the ESI  $i$ , through pseudo-random algorithms described in [13]. Next, we use the notation  $\mathbf{G}_{LT}(i_1, i_2, \dots, i_r)$  to denote the  $(r \times L)$  submatrix of  $\mathbf{G}_{LT}$  composed of the rows with indexes  $(i_1, i_2, \dots, i_r)$ . The notation  $\mathbf{G}_{LT}$  is equivalent to  $\mathbf{G}_{LT}(1, \dots, n)$ .

The  $L = k + s + h$  intermediate symbols  $\mathbf{F}$  are obtained from  $\mathbf{D}$  as

$$\mathbf{F} = \begin{bmatrix} \mathbf{D} \\ \mathbf{D}_s \\ \mathbf{D}_h \end{bmatrix}$$

through the relations

$$\mathbf{D}_s = \mathbf{G}_{LDPC} \cdot \mathbf{D}, \quad (4)$$

$$\mathbf{D}_h = \mathbf{G}_H \cdot \begin{bmatrix} \mathbf{D} \\ \mathbf{D}_s \end{bmatrix}. \quad (5)$$

The intermediate symbols  $\mathbf{F}$  are the inputs to the LT encoder for deriving the  $n$  encoded symbols  $\mathbf{E}$  as

$$\mathbf{E} = \mathbf{G}_{LT} \cdot \mathbf{F}. \quad (6)$$

Let us subdivide  $\mathbf{G}_{LT}$  as

$$\mathbf{G}_{LT} = [\mathbf{G}_{LT}^I \quad \mathbf{G}_{LT}^{II} \quad \mathbf{G}_{LT}^{III}],$$

where the sizes of the three submatrices are  $(n \times k)$ ,  $(n \times s)$  and  $(n \times h)$ , respectively. If also  $\mathbf{G}_H$  is subdivided as

$$\mathbf{G}_H = [\mathbf{G}_H^I \quad \mathbf{G}_H^{II}],$$

that is into two submatrices whose sizes are  $(h \times k)$  and  $(h \times s)$ , respectively, then the non-systematic Raptor code generator matrix can be expressed as

$$\begin{aligned} \mathbf{G}_{R,n-sys} &= \mathbf{G}_{LT}^I + \mathbf{G}_{LT}^{II} \cdot \mathbf{G}_{LDPC} \\ &\quad + \mathbf{G}_{LT}^{III} (\mathbf{G}_H^I + \mathbf{G}_H^{II} \cdot \mathbf{G}_{LDPC}), \end{aligned}$$

which satisfies the relation:

$$\mathbf{E} = \mathbf{G}_{R,n-\text{sys}} \cdot \mathbf{D}.$$

Let us now subdivide  $\mathbf{G}_{R,n-\text{sys}}$  into two submatrices  $\mathbf{G}_{R,n-\text{sys}}^I$  and  $\mathbf{G}_{R,n-\text{sys}}^{II}$ , whose sizes are  $(k \times k)$  and  $((n - k) \times k)$ , respectively:

$$\mathbf{G}_{R,n-\text{sys}} = \begin{bmatrix} \mathbf{G}_{R,n-\text{sys}}^I \\ \mathbf{G}_{R,n-\text{sys}}^{II} \end{bmatrix}.$$

For a systematic code, the following holds

$$E_i \equiv C_i \quad \forall i = 1, \dots, k,$$

and therefore

$$\begin{bmatrix} \mathbf{G}_{R,n-\text{sys}}^I \\ \mathbf{G}_{R,n-\text{sys}}^{II} \end{bmatrix} \cdot \mathbf{D} = \begin{bmatrix} \mathbf{E}_{[1,\dots,k]} \\ \mathbf{E}_{[k+1,\dots,n]} \end{bmatrix}, \quad (7)$$

$$= \begin{bmatrix} \mathbf{C} \\ \mathbf{E}_{[k+1,\dots,n]} \end{bmatrix}. \quad (8)$$

We have introduced in (7) the notations  $\mathbf{E}_{[1,\dots,k]}$  and  $\mathbf{E}_{[k+1,\dots,n]}$  to denote the first  $k$  and the last  $n-k$  encoded symbols, respectively.

We can state that the pre-processing matrix generating  $\mathbf{D}$  from  $\mathbf{C}$  can be obtained by

$$\mathbf{G}_T^{-1} = (\mathbf{G}_{R,n-\text{sys}}^I)^{-1}$$

and, as a consequence, the systematic Raptor code generator matrix is

$$\mathbf{G}_{R,\text{sys}} = \begin{bmatrix} \mathbf{I}_k \\ \mathbf{G}_{R,n-\text{sys}}^{II} \end{bmatrix}. \quad (9)$$

In (9)  $\mathbf{I}_k$  denotes the  $(k \times k)$  identity matrix. Obviously,  $\mathbf{G}_{R,n-\text{sys}}^I$  can be inverted if and only if it has full rank  $k$ . By initializing the random generator of inner LT code through the so-called *systematic index* (defined in [13]), this property is fulfilled for all  $k = 4, \dots, 8192$ .

#### 4.2. Raptor encoding

The relations (4), (5) and (6) can conveniently be represented as

$$\mathbf{A} \cdot \mathbf{F} = \begin{bmatrix} \mathbf{0} \\ \mathbf{E}_{[1,\dots,n]} \end{bmatrix},$$

whereby  $\mathbf{A}$  is a  $((s+h+n) \times (s+h+k))$  binary matrix called *encoding matrix*, whose structure is shown in Figure 5. Here,  $\mathbf{I}_s$  is the  $(s \times s)$  identity matrix,  $\mathbf{I}_h$  is the  $(h \times h)$  identity matrix and  $\mathbf{Z}$  is the  $(s \times h)$  all-zero matrix. The matrix  $\mathbf{A}$  doesn't properly represent the Raptor code generator matrix (which is defined in (9) instead), but includes the set of constraints imposed by the pre-coding and LT coding together. We next use the notation  $\mathbf{A}(i_1, i_2, \dots, i_r)$  to indicate the  $((s+h+r) \times L)$  submatrix of  $\mathbf{A}$  obtained by selecting only the rows of  $\mathbf{G}_{LT}$  with indexes  $(i_1, i_2, \dots, i_r)$ . Again,  $\mathbf{A}$  is equivalent to  $\mathbf{A}(1, \dots, n)$ .

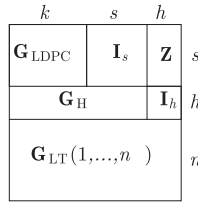


Figure 5. Structure of the encoding matrix  $\mathbf{A}$  for a  $(n; k)$  Raptor code specified in [13] ( $L = k + s + h$ ).

A possible Raptor encoding algorithm exploits a submatrix of  $\mathbf{A}$ . Such a matrix, consisting of the first  $L$  rows of  $\mathbf{A}$ , is used to obtain  $\mathbf{F}$  solving the system of linear equations:

$$\mathbf{A}(1, \dots, k) \cdot \mathbf{F} = \begin{bmatrix} 0 \\ \mathbf{C} \end{bmatrix}.$$

At this point, it is sufficient to multiply  $\mathbf{F}$  by the LT generator matrix to produce the encoded symbols  $\mathbf{E}$ , according to (6).

#### 4.3. Raptor ML decoding

The most direct way to decode the received sequence lies in inverting each encoding step of Figure 4. In this case we work on individual sub-codes. When using ML decoding at each sub-code, such a method requires the inversion of a matrix for each code, so it doesn't appear to be the best solution from the computational viewpoint [30]. Moreover, if the number of received encoded symbols does not exceed the number of source symbols  $k$  by a sufficiently large overhead, it shows a high failure probability.

For example, let us assume that only a subset of encoded symbols of ESIs  $(i_1, i_2, \dots, i_r)$  are available at the decoder. The first step the decoder should perform is to solve the system of linear equations:

$$\mathbf{G}_{\text{LT}}(i_1, i_2, \dots, i_r) \cdot \mathbf{F} = \mathbf{E}_{[i_1, i_2, \dots, i_r]}.$$

The matrix  $\mathbf{G}_{\text{LT}}(i_1, i_2, \dots, i_r)$  has  $(r \times L)$  size and, obviously, the necessary condition to solve the system is that  $r \geq L$ . If such a condition is not fulfilled, the decoding fails. It means that to recover the source symbols the decoder requires at least  $L$  encoded symbols (let us recall that  $L = k + s + h$ ).

Such a method doesn't exploit the fact that the  $L$  intermediate symbols are not independent from each other, are subject to the pre-coding constraints, instead. Therefore, to obtain the intermediate symbols  $\mathbf{F}$  by using a submatrix of  $\mathbf{A}$  (which considers such constraints) turns out to be a far more efficient solution.

According to the above-mentioned assumption, the first decoding step will turn into:

$$\mathbf{A}(i_1, i_2, \dots, i_r) \cdot \mathbf{F} = \begin{bmatrix} 0 \\ \mathbf{E}_{[i_1, i_2, \dots, i_r]} \end{bmatrix},$$

where  $\mathbf{A}(i_1, i_2, \dots, i_r)$  is a  $((s+h+r) \times L)$  matrix, as defined above. The system can be solved by GE (ML decoding) only if  $s+h+r \geq L$ , that is  $r \geq k$  (note that this is a necessary condition for successful ML decoding, not a sufficient one). In this way, the number of encoded symbols required at the decoder is typically lower compared with that in the previous case and, notably, is close to the number of source symbols  $k$  [27]. Once  $\mathbf{F}$  is known, the source symbols  $\mathbf{C}$  are easily

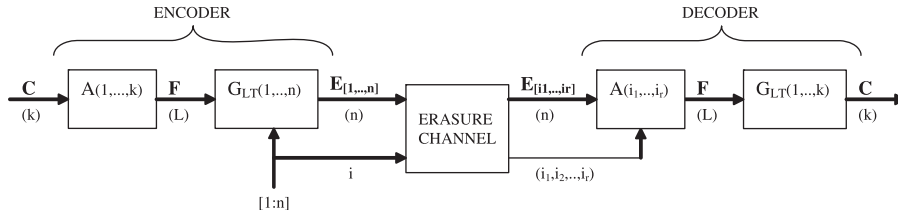


Figure 6. Overview of the encoding and decoding process for the systematic Raptor code specified in [13].

recovered by

$$\mathbf{C} = \mathbf{G}_{\text{LT}}(1, \dots, k) \cdot \mathbf{F}.$$

To conclude, when the described encoding and decoding algorithms are employed, both the encoding and the decoding are performed by making use of operations which are analogous in the two cases (Figure 6).

#### 4.4. Some remarks on the decoding complexity of LDPC and fixed-rate Raptor codes

If we take into consideration the first decoding step, an algorithm to perform GE in a more efficient way on  $\mathbf{A}(i_1, i_2, \dots, i_r)$  has been proposed in [13, Annex E]. This algorithm shares some similarities with that proposed in [8] for LDPC codes. In both cases, the erased symbols are solved by means of structured GE, exploiting the sparse nature of the equations to reduce the size of the matrix on which brute-force GE is performed. The targets of structured GE are  $\mathbf{H}_{\bar{\mathcal{K}}}$  for LDPC codes and  $\mathbf{A}$  for Raptor codes. Consider now an  $(n, k)$  LDPC code and its fixed-rate Raptor counterpart. Suppose also an erasure pattern (introduced by the communication channel) leading to a small overhead  $\delta$ , i.e. that the amount of correctly received symbols is  $k + \delta$ . On the LDPC code side, structured GE will be performed on  $\mathbf{H}_{\bar{\mathcal{K}}}$  with size  $(n-k) \times (n-k-\delta)$ . For the Raptor code, structured GE will work on  $\mathbf{A}$  with size  $(k + \delta + s + h) \times (k + s + h)$ . Hence, while for the LDPC code the complexity of the ML decoder is driven by  $(n-k)$  (i.e. the amount of redundancy, thus by the code rate  $R$ ), for the Raptor code the complexity depends just on  $k$  (i.e. it's code rate independent). The result is that for high rates ( $R > 1/2$ ) LDPC codes have an inherent advantage in complexity. On the other hand, for lower rates Raptor codes shall be preferable from a complexity viewpoint.

## 5. NUMERICAL RESULTS

In this section, some numerical results will be provided for LDPC and fixed-rate Raptor codes under ML decoding over the BEC. The performance is provided in terms of codeword error rate (CER) vs. the channel erasure probability  $\varepsilon$ . The section is organized as follows. First, some performance bounds for a  $(n, k)$  linear block code over the BEC are reviewed. Then, the performance of some moderate block-length LDPC codes is provided. The comparison with fixed-rate Raptor codes is presented in a dedicated subsection. Finally, some results for a protograph-based ARA code are given.

5.1. Bounds on the code performance

A lower bound on the decoding failure probability  $P_e$  for an  $(n,k)$  linear block code over a BEC with erasure probability  $\varepsilon$  is given by the well-known Singleton bound [31]:

$$P_e \geq \sum_{i=n-k+1}^n \binom{n}{i} \varepsilon^i (1-\varepsilon)^{n-i}. \tag{10}$$

There exist only a few binary codes achieving (10) with equality. However, for generic  $n$  and  $k$  we call idealized MDS code an idealized binary  $(n,k)$  linear block code achieving (10) with equality. An upper bound on the CER for random code ensembles was introduced by Berlekamp [6]. The bound can be expressed as:

$$\bar{P}_e \leq \sum_{i=0}^{n-k} \binom{n}{i} \varepsilon^i (1-\varepsilon)^{n-i} 2^{-(n-k-i)} + \sum_{i=n-k+1}^n \binom{n}{i} \varepsilon^i (1-\varepsilon)^{n-i}, \tag{11}$$

where  $\bar{P}_e$  represents the average error probability for the  $(n,k)$  random code ensemble. Even if (11) constitutes an upper bound to the error probability, for sufficiently large block lengths such bound can be considered as a good benchmark for the code performance [32].

5.2. Moderate block size LDPC codes

The performance of some moderate length LDPC codes is provided in Figures 7–9. In Figure 7, the CER for a (2048, 1024) GeIRA code from [9] is presented. The code is picked from an LDPC ensemble with  $\varepsilon_{IT} = 0.480$  and  $\varepsilon_{ML} = 0.496$ . The code performance, under ML decoding, tightly approaches the Singleton bound. The IT decoding curve, although not so far from the state-of-the-art for iteratively decoded codes, lies quite far from the bound. The sub-optimality of the IT curve is therefore not due to the code by itself but to the sub-optimality of the decoder.

The result is confirmed for a family of rate-compatible GeIRA codes with code rates ranging from 1/2 to 4/5 and input block size  $k = 502$  (Figure 8). The higher rates are obtained by

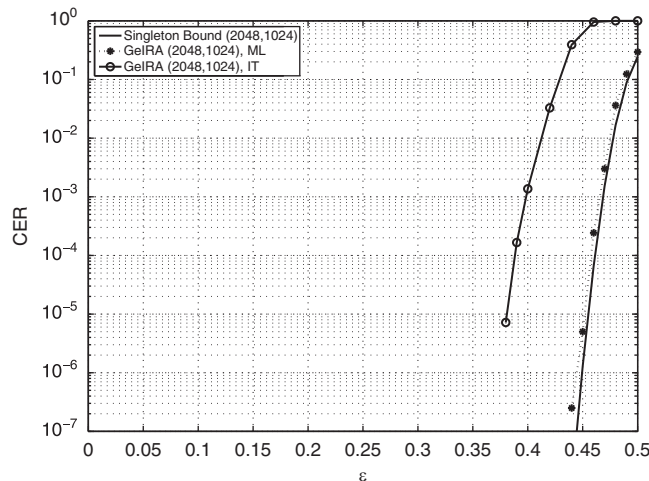


Figure 7. Codeword error rate for a (2048,1024) GeIRA code. The solid line represents the Singleton bound on the CER.

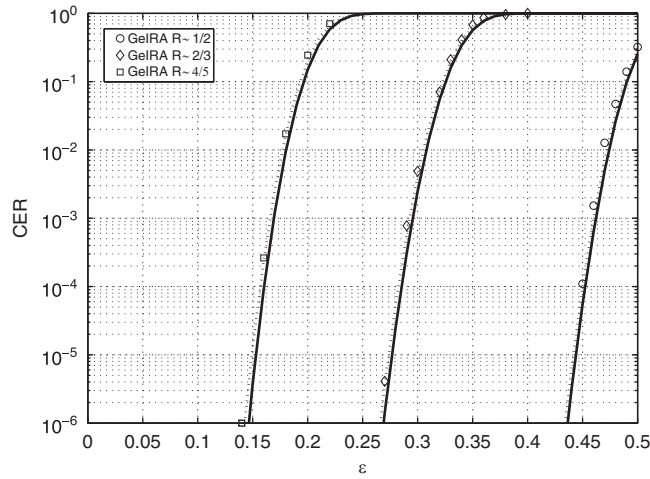


Figure 8. Codeword error rates for a family of GeIRA codes with input block size  $k = 502$  and code rates spanning from  $1/2$  to  $4/5$ . The solid lines represent the respective Singleton bounds on the CER, while dotted lines represent the respective Berlekamp random coding bounds.

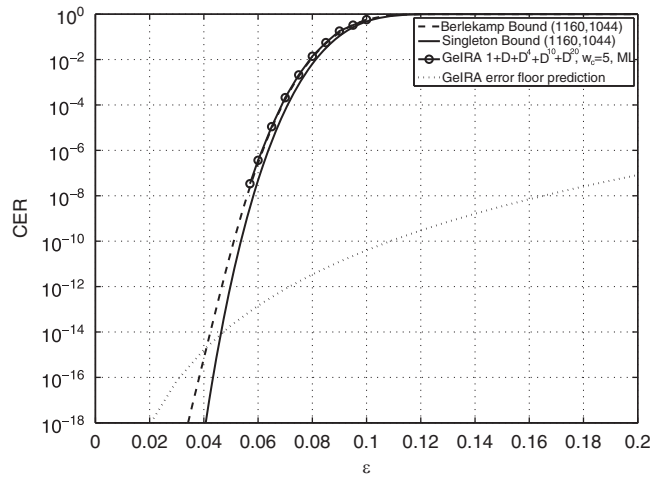


Figure 9. Codeword error rate for a (1160,1044) GeIRA code. The performance is compared to the Berlekamp bound and to the Singleton bound.

puncturing the mother  $R = 1/2$  code, which has been derived from the construction proposed in [9]. For the code rates under investigation, the performance is uniformly close to the corresponding Singleton bound, down to low codeword error rates ( $CER \approx 10^{-6}$ ). We remark that rate-compatibility allows using the LDPC codes as fountain codes, with the only limitation due to the lowest possible code rate, which is given by the mother code. In Figure 9, the codeword error rate for a (1160, 1044)  $R = 9/10$  code is shown. The code is a near-regular GeIRA code with almost constant column weight  $w_c = 5$  and feedback polynomial given by

$1+D+D^4+D^{10}+D^{20}$ . The ML threshold is  $\varepsilon_{\text{ML}} = 0.0994$ , whereas  $\varepsilon_{\text{IT}} = 0.0699$ . In addition, in this case, the error performance curve matches the Berlekamp bound down to low error rates. The minimum distance of this code (and its corresponding multiplicity) has been evaluated by [33]. An error floor estimation has been carried out by means of the truncated union bound on the codeword error probability, which is given by

$$P_e \simeq A_{\min} \varepsilon^{d_{\min}}, \quad (12)$$

where  $A_{\min}$  represents the minimum distance multiplicity. Four codewords at  $d_{\min} = 11$  have been found, leading to the error floor estimation provided in Figure 9. Even though such results represent only an estimation of the actual error floor, they are quite remarkable. The code performance would in fact deviate remarkably from the Singleton bound just at error rates below  $10^{-14}$ .

### 5.3. Comparisons with fixed-rate Raptor codes

A comparison with fixed-rate Raptor codes specified in the MBMS standard is provided next. In Figure 10, the decoding failure probability (i.e. the CER) as a function of the overhead is depicted for the codes specified in [13] and for some GeIRA codes. The overhead  $\delta$  is here defined as the number of codeword symbols that are correctly received in excess respect to  $k$  (recall that  $k$  represents the minimum amount of correctly received bits allowing successful decoding with an ideal MDS code). The comparison is carried out for various block sizes. There is basically no difference in performance between the MBMS Raptor codes and properly designed LDPC codes under ML decoding. As already pointed out in [34], the decoding failure probability vs. overhead does not seem to depend on the input block size.

A comparison between a (512,256) fixed-rate Raptor code and a near-regular GeIRA code from [9] with constant column weight  $w_c = 4$  is provided in Figure 11. In the waterfall region, the two codes exhibit almost the same performance. A minimum distance estimation according to [33] was conducted on the two codes. For the (512, 256) fixed-rate Raptor code, the minimum distance is given by  $d_{\min} = 25$ , with  $A_{\min} = 2$ . The lowest Hamming-weight codewords can be obtained by feeding the encoder with the  $k$ -bits input sequences  $\mathbf{u}^{(1)}, \mathbf{u}^{(2)}$ , where the non-null bits are

$$u_{13}^{(1)}, u_{21}^{(1)}, u_{32}^{(1)}, u_{39}^{(1)}, u_{63}^{(1)}, u_{90}^{(1)}, u_{91}^{(1)}, u_{95}^{(1)}, u_{98}^{(1)}, u_{102}^{(1)}, u_{115}^{(1)}, u_{118}^{(1)}, u_{133}^{(1)}, u_{142}^{(1)}, u_{181}^{(1)}, u_{230}^{(1)}, u_{243}^{(1)}, u_{247}^{(1)}$$

and

$$u_6^{(2)}, u_{13}^{(2)}, u_{18}^{(2)}, u_{75}^{(2)}, u_{88}^{(2)}, u_{101}^{(2)}, u_{123}^{(2)}, u_{131}^{(2)}, u_{140}^{(2)}, u_{143}^{(2)}, u_{176}^{(2)}, u_{220}^{(2)}, u_{231}^{(2)}, u_{243}^{(2)},$$

being  $u_0^{(1)}$  and  $u_0^{(2)}$  the first bit of  $\mathbf{u}^{(1)}$  and  $\mathbf{u}^{(2)}$ , respectively. For the GeIRA code, the estimated minimum distance is  $d_{\min} = 40$ , with multiplicity  $A_{\min} = 2$ . In both cases, the estimated minimum distance is quite large, and would permit to achieve very low error floors. For the Raptor code, the error floor estimation predicts a deviation from the Berlekamp bound at  $\text{CER} \simeq 10^{-11}$ , whereas for the GeIRA code the error floor would appear at  $\text{CER} \simeq 10^{-20}$ . The latter result is quite impressive, and would suggest the use of the near-regular GeIRA construction for applications<sup>††</sup> requiring very low error

<sup>††</sup>Almost all the current wireless systems adopting erasure-correcting codes have requirements which are usually much above the error floor of the Raptor code.

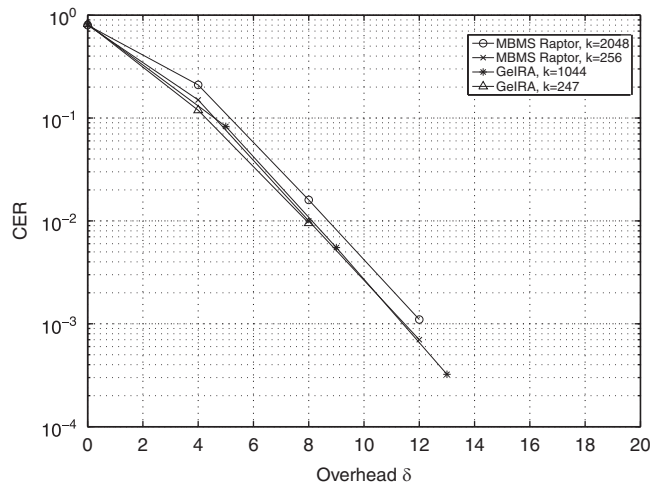


Figure 10. Codeword error rate vs. overhead  $\delta$  for the MBMS Raptor code [34] and for two GeIRA codes with parameters (1160, 1044) and (512, 247).

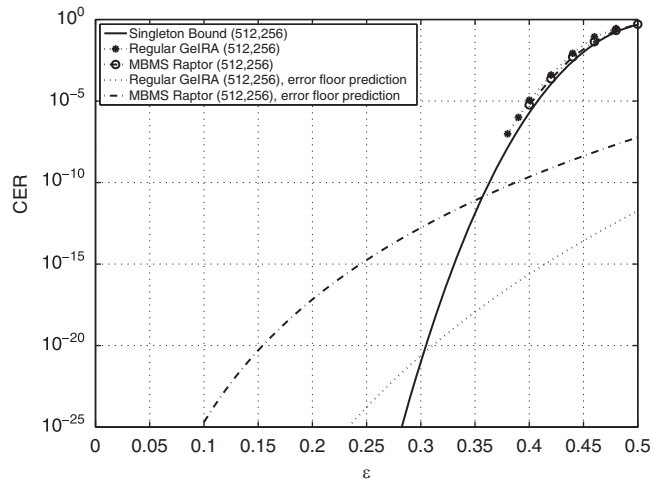


Figure 11. Codeword error rates and error floor predictions for (512, 256) Raptor and GeIRA codes.

floors. A final remark on the minimum distance evaluation for fixed-rate Raptor codes. The minimum distance evaluation has been applied to fixed-rate MBMS Raptor codes with various block lengths. For a (128,64) Raptor code, the lowest weight codeword found by [33] was 14 ( $A_{\min} = 2$ ). In the case of a (2048,1024) Raptor code,  $d_{\min} = 26$  ( $A_{\min} = 2$ ). Recalling the result for the (512,256) Raptor code ( $d_{\min} = 25$ ), it appears from this preliminary analysis that for fixed-rate Raptor codes the minimum distance might scale sub-linearly with the block length.

5.4. ML decoding of a (1024, 512) ARA code

In this subsection, we provide some numerical results dealing with ML decoding of a (1024,512) ARA code. The ARA protograph ensemble is defined by the base matrix [35]

$$\mathbf{B} = \begin{pmatrix} 2 & 1 & 1 & 1 & 0 \\ 1 & 2 & 1 & 1 & 0 \\ 2 & 0 & 0 & 0 & 1 \end{pmatrix},$$

where the first column corresponds to punctured VNs. Its IT decoding threshold is  $\epsilon_{IT} = 0.477$ . The upper bound on the ML threshold is  $\epsilon_{ML} \leq 0.496$  (see Figure 3). The code performance is shown in Figure 12, for both IT and ML decoding. The gain obtained by the ML decoder in the waterfall region (the error rate performance is actually quite close to the Singleton bound) indicates that the bound on the ML threshold is quite tight. Both the IT and the ML curves at low error rates present an evident error floor, due to the presence of 16 codewords with Hamming weight 10 in the codeword set.

6. Concluding remarks

In this article, we provided some insights on the code design for ML-decoded LDPC codes on the erasure channel, together with an overview on efficient ML decoding algorithms. The complexity on the decoder side can be kept low with a proper code design. Such an approach allows to design codes with a large flexibility in terms of block lengths and code rates. A comparison with ML-decoded fixed-rate Raptor codes (derived from the MBMS specification) has been carried out as well. Our results show that, even at short block sizes, LDPC codes under ML decoding can tightly approach the Singleton bound down to very low error rates as their Raptor counterpart. In some cases, the estimated error floor for the LDPC code is much lower than the estimated error floor of the corresponding fixed-rate Raptor code. ML-decoded LDPC codes represent therefore a practical tool to approach the ideal MDS codes performance, down to very low error rates, and with limited decoding complexity.

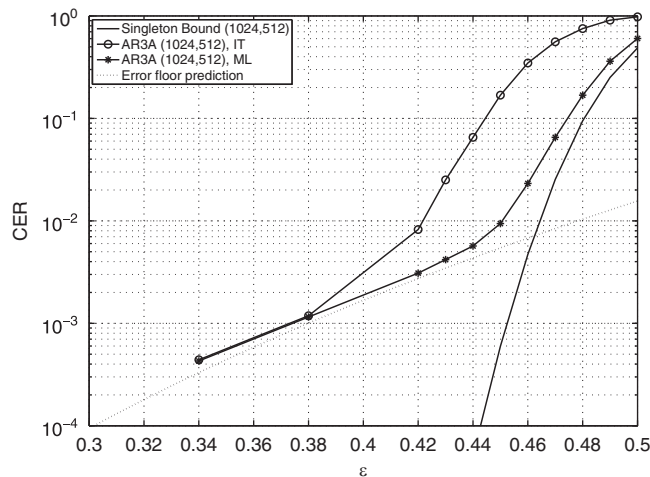


Figure 12. Codeword error rates for a (1024, 512) accumulate-repeat-accumulate code under IT and ML decoding.

## ACKNOWLEDGEMENTS

The study was supported in part by the European Community under Seventh Framework Programme grant agreement ICT OPTIMIX nINFSO-ICT-214625 and in part by the EC-IST SatNEx-II project (IST-27393). The authors thank Michela Varrella for her work on an earlier version of this manuscript.

## REFERENCES

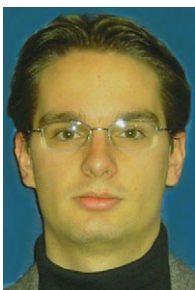
1. Gallager RG. *Low-Density Parity-Check Codes*. M.I.T. Press: Cambridge, MA, 1963.
2. Luby M, Mitzenmacher M, Shokrollahi M, Spielman D. Efficient erasure correcting codes. *IEEE Transactions on Information Theory* 2001; **47**(2):569–584.
3. Oswald P, Shokrollahi MA. Capacity-achieving sequences for the erasure channel. *IEEE Transactions on Information Theory* 2002; **48**(12):364–373.
4. Pfister HD, Sason I, Urbanke R. Capacity-achieving ensembles for the binary erasure channel with bounded complexity. *IEEE Transactions on Information Theory* 2005; **51**(7):2352–2379.
5. Di C, Proietti D, Telatar I, Richardson T, Urbanke R. Finite-length analysis of low-density parity-check codes on the binary erasure channel. *IEEE Transactions on Information Theory* 2002; **48**(6):1570–1579.
6. Berlekamp E. The technology of error-correcting codes. *Proceedings of IEEE* 1980; **68**(5):564–593.
7. Richardson T, Urbanke R. *Modern Coding Theory*. Cambridge University Press: Cambridge, 2008.
8. Burshtein D, Miller G. An efficient maximum likelihood decoding of LDPC codes over the binary erasure channel. *IEEE Transactions on Information Theory* 2004; **50**(11):2837–2844.
9. Paolini E, Liva G, Matuz B, Chiani M. Generalized IRA erasure correcting codes for hybrid iterative/maximum likelihood decoding. *IEEE Communication Letters* 2008; **12**(6):450–452.
10. Hiroto M, Konishi Y, Morii M. On the probabilistic computation algorithm for the minimum-size stopping sets of LDPC codes. In *Proceedings of 2008 IEEE International Symposium on Information Theory*, Toronto, Canada, July, 2008; 259–299.
11. Framing structure, channel coding and modulation for Satellite Services to Handheld devices (SH) below 3GHz. Digital Video Broadcasting (DVB), Blue Book, 2007.
12. MPE-iFEC specification. Digital Video Broadcasting (DVB). *Technical Report A131*, version 3.0.8, November 2008.
13. 3GPP TS 26.346 V7.4.0. Technical specification group services and system aspects; multimedia broadcast/multicast service; protocols and codecs. 2007.
14. Odlyzko A. Discrete logarithms in finite fields and their cryptographic significance. In *Proceedings of EUROCRYPT 84 Workshop on Advances in Cryptology: Theory and Application of Cryptographic Techniques*, Lecture Notes in Computer Science, vol. 209/1985. Springer: New York, 1985; 224–314.
15. LaMacchia B, Odlyzko A. Solving large sparse linear systems over finite fields. In *Proceedings of the 10th Annual International Cryptology Conference on Advances in Cryptology*, Lecture Notes in Computer Science, vol. 237. Springer: London, U.K., 1990; 109–133.
16. Measson C, Montanari A, Urbanke R. Maxwell construction: the hidden bridge between iterative and maximum a posteriori decoding. *IEEE Transactions on Information Theory* 2008; **54**(12):5277–5307.
17. Horn RA, Johnson CR. *Topics in Matrix Analysis*. Cambridge University Press: Cambridge, 1994.
18. Richardson T, Shokrollahi M, Urbanke R. Design of capacity-approaching irregular low-density parity-check codes. *IEEE Transactions on Information Theory* 2001; **47**(2):619–637.
19. Measson C, Montanari A, Richardson T, Urbanke R. Life above threshold: from list decoding to area theorem and MSE. In *Proceedings of 2004 IEEE Information Theory Workshop*. San Antonio, TX, U.S.A., October, 2004.
20. Thorpe J. Low-density parity-check (LDPC) codes constructed from protographs. *JPL INP, Technical Report*, 2003; 42–154.
21. Orlitsky A, Viswanathan K, Zhang J. Stopping set distribution of LDPC code ensembles. *IEEE Transactions on Information Theory* 2005; **51**(3):929–953.
22. ten Brink S. Convergence behavior of iteratively decoded parallel concatenated codes. *IEEE Transactions on Communication* 2001; **49**(10):1727–1737.
23. Ashikhmin A, Kramer G, ten Brink S. Extrinsic information transfer functions: model and erasure channel properties. *IEEE Transactions on Information Theory* 2004; **50**(11):2657–2673.
24. Liva G, Chiani M. Protograph LDPC codes design based on EXIT analysis. In *Proceedings of 2007 IEEE Global Telecommunications Conference*, Washington, DC, U.S.A., April, 2007; 3250–3254.
25. Abbasfar A, Disvalar D, Yao K. Accumulate-repeat-accumulate codes. *IEEE Transactions on Communication* 2007; **55**(4):692–702.
26. Liva G, Paolini E, Chiani M. Simple reconfigurable low-density parity-check codes. *IEEE Communication Letters* 2005; **9**(3):258–260.
27. Shokrollahi M. Raptor codes. *IEEE Transactions on Information Theory* 2006; **52**(6):2551–2567.

28. Byers J, Luby M, Mitzenmacher M. A digital fountain approach to reliable distribution of bulk data. *IEEE Journal of Selected Areas and Communication* 2002; **20**(8):1528–1540.
29. Luby M. LT codes. In *Proceedings of the 43rd Annual IEEE Symposium on Foundations of Computer Science*, Vancouver, Canada, November, 2002; 271–282.
30. Luby M, Watson M, Gasiba T, Stockhammer T, Xu W. Raptor codes for reliable download delivery in wireless broadcast systems. In *Proceedings of 2006 IEEE Consumer Communications and Networking Conference*, Las Vegas, NV, U.S.A., January, 2006; 1:192–197.
31. MacWilliams FJ, Sloane N. *The Theory of Error Correcting Codes*. North-Holland Mathematical Library: Amsterdam, 1977.
32. MacMullan S, Collins OM. A comparison of known codes, random codes, and the best codes. *IEEE Transactions on Information Theory* 1998; **44**(7):3009–3022.
33. Hu X.-Y, Fossorier M, Eleftheriou E. On the computation of the minimum distance of low-density parity-check codes. In *Proceedings of 2004 IEEE International Conference on Communications*, Paris, France, June, 2004; 767–771.
34. Luby M, Gasiba T, Stockhammer T, Watson M. Reliable multimedia download delivery in cellular broadcast networks. *IEEE Transactions on Broadcasting* 2007; **53**(1):235–246.
35. Liva G, Song S, Lan L, Zhang Y, Ryan W, Lin S. Design of LDPC codes: a survey and new results. *Journal of Communication Software and Systems* 2006; **2**(3):191–211.

## AUTHORS' BIOGRAPHIES



**Gianluigi Liva** was born in Spilimbergo, Italy, on July 23, 1977. He received the MS degree in Electrical Engineering in 2002, and the PhD degree in 2006 at DEIS, University of Bologna (Italy). His main research interests include satellite communication systems and error control coding (with emphasis on LDPC codes and Turbo-like codes) for wireless fading channels. Since 2003 he has been involved in the research of near Shannon limit channel codes for high data rate CCSDS (Consultative Committee for Space Data Systems) missions, in collaboration with the European Space Operations Centre of the European Space Agency (ESAESOC). From October 2004 to April 2005 he was researching at the University of Arizona, where he was involved in the design of low-complexity coding systems for high data rate Mars links. He is currently with the Institute of Communications and Navigation, at the German Aerospace Center (DLR), in Wessling, where he is involved in the research of FEC and modulation techniques for mobile satellite systems. In 2010 he has been appointed as guest lecturer for Channel Coding at the Institute for Communications Engineering (LNT) of the Technische Universität München (TUM). He is active in the DVB-SH and in the DVB-RCS Mobile standardization groups. He is an IEEE member and he serves IEEE as a reviewer for Transactions, Journals and Conferences. He received the 2007 IST Mobile and Wireless Communication Summit Best Paper Award.



**Balázs Matuz** was born 1982 in Budapest, Hungary. He received his Diploma degree in Electrical Engineering and Information Technology from the Technical University of Munich (TUM) in 2007. Since then he has been pursuing a PhD at German Aerospace Center (DLR) in Oberpfaffenhofen, Germany. His main research interests are in physical and upper layer channel coding, as well as in channel modeling.



**Enrico Paolini** received the Dr. Ing. Degree (with honors) in Telecommunications Engineering and the PhD degree in Telecommunications Engineering from the University of Bologna, Italy, in 2003 and 2007, respectively. While working towards the PhD degree, he was Visiting Research Scholar at the University of Hawaii at Manoa. Currently, he holds a postdoctoral position from the Department of Electronics, Computer Science and Systems (DEIS) of the University of Bologna, Italy. His research interests include error-control coding (with emphasis on LDPC codes and their generalizations, iterative decoding algorithms, reduced-complexity maximum likelihood decoding for erasure channels), and distributed radar systems based on ultra-wideband. In the field of error correcting codes, he has been involved since 2004 in activities with the European Space Agency (ESA). Dr. Paolini is a member of the IEEE Communications Society and of the IEEE Information Theory Society.



**Marco Chiani** was born in Rimini, Italy, in April 1964. He received the Dr. Ing. degree (magna cum laude) in Electronic Engineering and the PhD degree in Electronic and Computer Science from the University of Bologna in 1989 and 1993, respectively. Dr. Chiani is a Full Professor at the II Engineering Faculty, University of Bologna, Italy, where he is the Chair in Telecommunication. During the summer of 2001 he was a Visiting Scientist at AT&T Research Laboratories in Middletown, NJ. He is a frequent visitor at the Massachusetts Institute of Technology (MIT), where he presently holds a Research Affiliate appointment. Dr. Chiani's research interests include wireless communication systems, MIMO systems, wireless multimedia, low density parity check codes (LDPC) and UWB. He is leading the research unit of University of Bologna on cognitive radio and UWB (European project EUWB), on Joint Source and Channel Coding for wireless video (European projects Phoenix-FP6 and Optimix-FP7), and is a consultant to the European Space Agency (ESA-ESOC) for the design and evaluation of error correcting codes based on LDPC for space CCSDS applications.

Dr. Chiani has chaired, organized sessions and served on the Technical Program Committees at several IEEE International Conferences. In January 2006 he received the ICNEWS award "For Fundamental Contributions to the Theory and Practice of Wireless Communications". He was the recipient of the 2008 IEEE ComSoc Radio Communications Committee Outstanding Service Award.

He is the past chair (2002–2004) of the Radio Communications Committee of the IEEE Communication Society and the past Editor of Wireless Communication (2000–2007) for the IEEE Transactions on Communications.