Flexible On-line Construction of IRA Codes for Packet Erasure Correction with Application to Aeronautical Communications

Gianluigi Liva and Paola Pulini
Institute of Communication and Navigation
Deutsches Zentrum fur Luft- und Raumfahrt (DLR)
82234 Wessling, Germany
{Gianluigi.Liva,Paola.Pulini}@dlr.de

Marco Chiani
DEIS, WiLAB
University of Bologna
via Venezia 52, 47023 Cesena (FC), Italy
marco.chiani@unibo.it

Abstract— In many applications erasure correcting codes are used to recover packet losses at high protocol stack layers. The objects (e.g. files) to be transmitted often have variable sizes, resulting in a variable number of packet to be encoded by the packet-level encoder. In this paper, algorithms for the (on-line) flexible design of parity-check matrices for irregular-repeat-accumulate codes are investigated. The proposed algorithms allow designing in fast manner parity-check matrices that are suitable for low-complexity maximum-likelihood decoding. The code ensembles generated by the proposed algorithms are analyzed via extrinsic information transfer charts. Numerical results show how the designed codes can attain codeword error rates as low as 10^{-5} without appreciable losses w.r.t. the performance of idealized maximum-distance separable codes. The application of the proposed techniques to the upcoming aeronautical communication standard is investigated, proving the efficiency and the flexibility of the approach.

I. INTRODUCTION

The application of error control codes to protocol stack layers different from the physical one gained a large interest during the past decade [1], [2]. Such coding techniques deal with the use of a linear (block) code applied to *encoding units* (symbols) that are usually packets with constant size. In this context, a packet level encoder receives as input a set of k packets, and produces as output $n > k$ packets. Assuming systematic encoding, the final set of packets comprises the k information packets together with $m = n - k$ parity packets. At the receiver, after the physical layer decoding, the packets that have been validated (e.g. by an error detection code) are forwarded to the packet level decoder. The corrupted packets are marked as lost. Therefore, the upper layers deal with packet erasures. The packet level decoder may recover the erased packets by means of the parity packets. Packet level codes are employed in multimedia wireless broadcasting systems (see for example the DVB-H/SH standards [3], [4]), in multicast scenarios [5], and are currently invested in space communication systems [6].

In [7], [8], a novel approach to the design of low-density parity-check (LDPC) codes [9] and decoders for the binary erasure channel (BEC) [10] was proposed, showing how a judicious code design together with an efficient maximum likelihood (ML) decoding algorithm permits to achieve near-optimum performance (i.e., close to those of an idealized maximum distance separable (MDS) code) down to low error rates on the memory-less BEC with LDPC codes. The results were extended to correlated erasure channels in [11]. In [12] enhancements of the original LDPC ML decoding algorithm proposed in [13] were introduced, demonstrating decoding speeds up to several hundreds Mbps on an actual satellite communication link.

In many applications, the object (e.g., the file) to be transmitted has a variable size, resulting in a variable number of packets to be encoded. One may design a code with large input block size k and may perform code shortening for all the cases where the number of packets to be encoded is lower. In a similar manner, if the code rate has to be adapted to specific channel conditions, a low-rate code may be designed, and higher rates may be obtained by puncturing. However, shortening does not allow complete flexibility, since a maximum value of k has to be a-priori selected. Moreover, the selection of the codeword symbols to be zero-padded has to be performed in a careful manned to avoid degradations of the iterative (IT) decoding threshold.[1] Puncturing a large fraction of symbols may pose some complexity issues for the ML decoder since the size of the system of equations to be solved is associated with the number of parity equations of the mother (unpunctured) code. We thus investigate a flexible *on-line* code construction technique which allows designing in a fast (algorithmic) manner reasonably-good parity-check matrices for an arbitrary set of the (n, k) parameters.[2] The code construction shall be based on a simple algorithm available at both the receiver and the transmitter. The adaptive packet level coding scheme may work as follows. Once the encoder receives the set of k packets to be encoded and the target code

---

[1]Recall that a large gap between the IT decoding threshold and the BEC Shannon limit is responsible on one hand of poor performance under IT decoding, and on the other hand of a large decoding complexity when ML decoding is used [8].

[2]A similar approach has been adopted for the Raptor codes standardized in the Multimedia Broadcast Multicast Service (MBMS) [5], [14] standard.

rate R is selected, the set of parameters $\{n, k, \kappa\}$ is signaled to the receiver, being κ a compact parameter that drives the code construction (the significate of κ will be clarified in the following). On the transmitter side, a code $\mathfrak{C}_{(n,k,\kappa)}$ is generated and used to encode the set of k packets. At the receiver, the $\{n, k, \kappa\}$ parameters are used to reconstruct the parity-check matrix of $\mathfrak{C}_{(n,k,\kappa)}$, which is then used for the erasure decoding. We propose two code constructions, both leading to irregular repeat-accumulate (IRA) codes [15] whose parity-check matrix is built according to specific (code-rate dependent) degree distributions. The construction of the parity-check matrix is based in both cases on a pseudo-random approach. With this respect, the parameter κ is the seed for the pseudo-random number generator used for the construction of the code parity-check matrix.

In the next decade, aeronautical communications will be likely based on the integration of several systems for aeronautical communications into a global heterogeneous communications network [16]. The overall system composed by a ground segment, a satellite segment, and a dedicated airport surface system, will be eventually complemented by ad-hoc networks between aircrafts [17], answering to the demand of high availability, robustness, and capacity. Existing infrastructures, e.g. the satellite communications for The Air Traffic Control (ATC)/Air Traffic Management (ATM) which are already available for aeronautical passenger communications (APC), may be re-used for this purpose. Being the proposed packet-level coding scheme independent by the underlying system (e.g., by integration in a dedicated sub-layer), it can in fact be considered an upgrade on top of the various standards employed for the different links. As an example, we illustrate the performance improvement provided by the application of the proposed scheme to the new (under development) C-Band Aeronautical Mobile Airport Communications System (AeroMACS) based on the IEEE 802.16e standard [18], [19].

The paper is organized as follows. In Section II the two families of flexible IRA codes are described. The performance on the BEC is analyzed in Section III. The application to the AeroMACS system is presented in Section IV, together with numerical results. Conclusions follow in Section V.

II. Families of Flexible IRA Codes

We consider next the construction of the parity-check matrix of an (n, k) IRA code [15].[3] The $m \times n$ IRA code parity-check matrix is partitioned in two parts, i.e. $\mathbf{H} = [\mathbf{H}_u | \mathbf{H}_p]$, with $m = n - k$ being the number of parity-check equations (and hence of check nodes (CNs) in the bipartite graph associated with $\mathbf{H}$). In our construction, $\mathbf{H}_u$ is a $m \times k$ unstructured low-density (random) matrix, while $\mathbf{H}_p$ is the usual $m \times m$ dual-diagonal matrix. The construction algorithms focus on the systematic part only ($\mathbf{H}_u$). In the next subsection, we make use of the following notation. We denote by Λ_i the fraction of columns of weight i in $\mathbf{H}$ (and hence the fraction of degree-i

variable nodes (VNs) in the bipartite graph associated with $\mathbf{H}$), whereas Φ_i denotes the fraction of columns of weight i in ($\mathbf{H}_u$). We denote by Ψ_i (Ω_i) the fraction of rows of $\mathbf{H}$ ($\mathbf{H}_u$) with Hamming weight i. Clearly, Ψ_i is also the fraction of degree-i CNs in the bipartite graph associated with $\mathbf{H}$. We associate to Λ_i, Φ_i, Ψ_i, Ω_i the polynomial representations $\Lambda(x) = \sum_i \Lambda_i x^i$, $\Phi(x) = \sum_i \Phi_i x^i$, $\Psi(x) = \sum_i \Psi_i x^i$, $\Omega(x) = \sum_i \Omega_i x^i$. We refer to $\{\Lambda_i\}$ ($\{\Phi_i\}$) as the node-oriented VN degree distribution for $\mathbf{H}$ ($\mathbf{H}_u$), and to $\{\Psi_i\}$ ($\{\Omega_i\}$) as the node-oriented CN degree distribution for $\mathbf{H}$ ($\mathbf{H}_u$). The edge-oriented degree distributions for the complete matrix are defined as $\{\lambda_i\}$ and $\{\rho_i\}$ for VNs and CNs respectively. λ_i represents the fraction of edges in the bipartite graph representation of $\mathbf{H}$ which are connected to VNs of degree i, while ρ_i represents the fraction of edges in the bipartite graph representation of $\mathbf{H}$ which are connected to CNs of degree i. The polynomial representations of λ_i and ρ_i are given by $\lambda(x) = \sum_i \lambda_i x^{i-1}$, $\rho(x) = \sum_i \rho_i x^{i-1}$. Due to the structure of $\mathbf{H}_p$, the relations between node-perspective degree distributions for the full $\mathbf{H}$ matrix and those for the systematic part $\mathbf{H}_u$ are

$$\Lambda(x) = \Phi(x)R + x^2(1 - R) \tag{1}$$

$$\Psi(x) = \Omega(x)x^2 \tag{2}$$

with $R = k/n$ being the rate of the IRA code[4].

A. Fully-Random Construction

The first on-line construction technique works as follows. For a given VN degree distribution $\Phi(x)$ of $\mathbf{H}_u$, a vector $\mathbf{u} = (u_0, u_1, \ldots, u_{k-1})$ of k VN degree values is derived. For the generic l-th column of $\mathbf{H}_u$, u_l indexes $q_0, q_1, \ldots, q_{u_l-1}$ comprised between 0 and $m - 1$ are generated by means of a pseudo-random number generator (e.g., a simple linear congruential generator may be used). The elements of the l-th column of $\mathbf{H}_u$ corresponding to the indexes $q_0, q_1, \ldots, q_{u_l-1}$ are hence set to 1, i.e. $\mathbf{h}_{u(q_j,l)} = 1 \quad \forall j = 0 \ldots u_l$. The construction algorithm is summarized next.

Algorithm 1 fully-random construction (FRC)

1: Calculate the column weights $\mathbf{u} = (u_0, u_1, \ldots, u_{k-1})$ for the systematic part of $\mathbf{H}$ according to $\Phi(x)$.
2: **for** $l = 0 : k - 1$ **do**
3:     **for** $j = 0 : u_l - 1$ **do**
4:         Generate a random integer number $q \sim \mathcal{U}(0, m-1)$
5:         **while** $h_{u(q,l)} == 1$ **do**
6:             Generate a random integer number $q \sim \mathcal{U}(0, m - 1)$
7:         **end while**
8:         Set $h_{u(q_j,l)} = 1$
9:     **end for**
10: **end for**

---

[3]The proposed construction techniques can be easily extended to the case of generalized IRA codes [20].

[4]In the derivation of the relations (1) and (2) we neglected the presence of a weight-1 column in the matrix $\mathbf{H}_p$.

This method builds a random matrix $\mathbf{H}_u$ whose column weights fulfill the degree distribution specified by $\Phi(x)$. Indeed, no control at all is performed on the graph girth. Even more important, there is no control on the final CN degree distribution. Hence, the ensemble is completely specified by the distribution $\Phi(x)$ and we will denote this code ensemble by $\mathcal{C}_1(\Phi(x), k, R)$. The ensemble can be analyzed in the asymptotic setting for $k \to \infty$ by extrinsic information transfer (EXIT) analysis [21]. While the VN degree distribution for the complete matrix $\mathbf{H}$ is already available by (1), we need to derive the expected degree distribution for CNs. For this purpose, we define $c = \Phi'(1)$ as the average column weight of $\mathbf{H}_u$, and by $r = k\delta$ the average row weight of $\mathbf{H}_u$. It turns that a generic element of $\mathbf{H}_u$ is a 1 with probability $\delta = c/m$. The probability that a row of $\mathbf{H}_u$ possesses i 1s is given by

$$\Omega_i = \binom{k}{i}\delta^i(1-\delta)^{k-i}.$$

Hence, the average CN distribution for $\mathbf{H}_u$ is

$$\Omega(x) = \sum_i \Omega_i x^i = \sum_i \binom{k}{i}(x\delta)^i(1-\delta)^{k-i} = (1 - \delta x)^k.$$

By letting $n \to \infty$,

$$\Omega(x) = \lim_{n \to \infty}\left(1 - \frac{r}{k}x\right)^k = \exp(-rx).$$

The knowledge of $\Omega(x)$ allows to derive the CN degree distribution for $\mathbf{H}$ thanks to (2). Out of $\Lambda(x)$ and $\Psi(x)$, the iterative decoding threshold for the $\mathcal{C}_1(\Phi(x), k, R)$ ensemble can be computed and used as cost function for optimization, e.g. via differential evolution [22]. Being this construction incapable of controlling the girth of the graph, high error floors are expected. Nevertheless, we will consider the FRC as a reference for a more sophisticated construction provided in the next subsection.

B. Permutation-Based Construction

The construction method presented next does not increase the matrix generation complexity w.r.t. to the one described in Section II-A. However, it permits to keep a control on the CN degree distribution, and to limit somehow the amount of short cycles. As before, the construction algorithm starts with the derivation of the vector $\mathbf{u} = (u_0, u_1, \ldots, u_{k-1})$ containing the k column weights for the matrix $\mathbf{u}$, according to a specific degree distribution $\Phi(x)$. The algorithm proceeds as follows. The vector $\mathbf{v} = (0, 1, 2, \ldots, m-1)$ is permuted according to a random permutation pattern, leading to the permutation vector $\pi = (\pi_0, \pi_1, \ldots, \pi_{m-1})$. The values of the first u_0 elements of π are then assigned to $q_0, q_1, \ldots, q_{u_0-1}$, i.e. $q_j = \pi_j$ for $j = 0 \ldots u_0 - 1$, and the elements of the 0-th column of $\mathbf{H}_u$ corresponding to the indexes $q_0, q_1, \ldots, q_{u_0-1}$ are hence set to 1, i.e. $\mathbf{h}_{u(q_j,0)} = 1 \qquad \forall j = 0 \ldots u_0$. For the column with index 1, the next u_1 elements of π are extracted, i.e. $q_j = \pi_{u_0+j}$, for $j = 0 \ldots u_1 - 1$, and are used to define the elements of the column to be set at 1. For the column with index 2, the next u_2 elements of π are extracted, i.e.

$q_j = \pi_{u_0+u_1+j}$, for $j = 0 \ldots u_2 - 1$, and are used to define the elements of the column to be set at 1. The process continues for l steps, until the number of remaining elements in π is less than the column weight under consideration, i.e. when $\sum_{w=0}^{l} u_w > m$. When this happens, a new permutation vector is generated, and the above described procedure restarts from the l-column. More specifically, the first u_l elements of π are extracted, i.e. $q_j = \pi_j$, for $j = 0 \ldots u_l - 1$, and are used to define the elements of the column to be set at 1. The procedure is iterated until the last column of $\mathbf{H}_u$ has been filled.

Algorithm 2 permutation-based construction (PBC)

1: Calculate the parity-check matrix column weights $\mathbf{u} = (u_0, u_1, \ldots, u_{k-1})$ according to $\Phi(x)$.
2: Generate a random permutation vector $\pi = (\pi_0, \pi_1, \ldots, \pi_{m-1})$ with m elements
3: Set $\mu = 0$
4: **for** $l = 0 : k - 1$ **do**
5:     **if** $\mu + u_l - 1 > m$ **then**
6:         Generate a new random permutation vector $\pi = (\pi_0, \pi_1, \ldots, \pi_{m-1})$ with m elements
7:         Set $\mu = 0$
8:     **end if**
9:     **for** $j = 0 : u_l - 1$ **do**
10:         Set $q_j = \pi_{j+\mu}$
11:         Set $h_{u(q_j,l)} = 1$
12:     **end for**
13:     Set $\mu = \mu + u_l$
14: **end for**

The above-described algorithm permits to achieve a nearly-constant CN degree since, in the block of columns taking their non-null elements row indexes from the same permutation vector, the row weight is (almost) uniformly 1. Moreover, no loops are present among the columns composing such a block. We refer to this code ensemble as $\mathcal{C}_2(\Phi(x), k, R)$.

The absence of a complete girth optimization may still result in error floors at moderate error probabilities, especially when short codes (e.g. $k < 500$ symbols) have to be designed. We propose hence an enhancement based on the concatenation with an outer (k, k') binary linear random block code. More specifically, the outer code parity-check matrix $\mathbf{H}_o$ is a $(k - k') \times k$ in the form $\mathbf{H}_o = [\mathbf{H}_{ou}|\mathbf{I}]$ with $\mathbf{I}$ being the $(k - k') \times (k - k')$ identity matrix and $\mathbf{H}_{ou}$ being a $(k - k') \times k'$ matrix whose elements are set to 0 or 1 with uniform probability.[5] The parity-check matrix of the code obtained by the concatenation of the outer random code with the inner IRA one has thus the

---

[5]The proposed on-line construction requires that the algorithms discussed above are implemented at both the transmitter and the receiver. Furthermore, the transmitter shall signal to the receiver the (n, k) parameters together with the seed κ used to initialize the random number generator used by the algorithms. If pre-coding is used, the dimension of the pre-code k' shall be signaled too. The degree distribution $\Phi(x)$ must be known to the receiver as well. A possibility is to pre-compute capacity-approaching distributions for a range of code rates, and to store them in a tables on both the encoder and the decoder side.

form

$$\mathbf{H}' = \left[\begin{array}{c|c} \mathbf{H}_o & \mathbf{0} \\ \hline \mathbf{H}_u & \mathbf{H}_p \end{array}\right].$$

III. PERFORMANCE

In this section simulation results on the memory-less BEC are presented for codes constructed with the methods of Section II. We will restrict to the case of coding rate $1/2$. For the codes generation we will use the distribution $\Phi_1(x) = 0.543x^3 + 0.102x^4 + 0.008x^5 + 0.020x^6 + 0.008x^7 + 0.008x^8 + 0.047x^9 + 0.266x^{10}$. When used for the FRC, this distribution leads to the code ensemble characterized by an IT decoding threshold $\epsilon_{IT}^* = 0.4788$ and to an ML one $\epsilon_{ML}^* = 0.4869$ (see the ensemble EXIT function [23] in Figure 1(a)). When used to generate codes according to the PBC, the related ensemble possesses an IT decoding threshold $\epsilon_{IT}^* = 0.4673$ and an ML one $\epsilon_{ML}^* = 0.4954$ (see the ensemble EXIT function in Figure 1(b)). In both cases, the gap between the IT and the ML thresholds is limited and hence the ML decoding complexity shall be low [8], [12]. Two block lengths are investigated, i.e a short one ($n = 512$) and a moderate-large one ($n = 2048$).
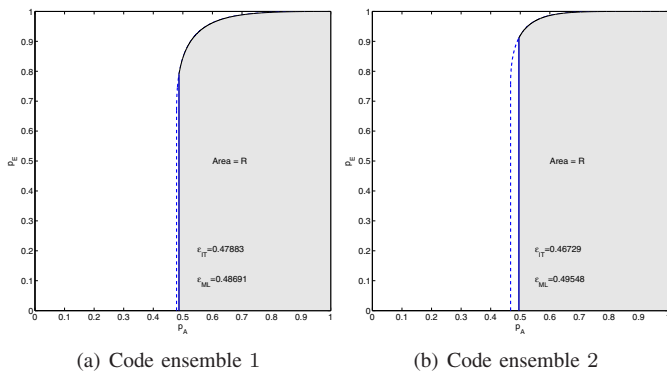

Fig. 1. EXIT function for the FRC (left) and the PBC (right) ensembles based on $\Phi_1(x)$ with $R = 1/2$.

The performance for the $(512, 256)$ and the $(2048, 1024)$ codes generated according to the FRC of Section II-A is depicted in Figure 2 and in Figure 3 respectively. The performance is presented in terms of codeword error rate (CER) vs. erasure probability ϵ under ML decoding. We note that in the $n = 512$ case the CER performance remains close to the Singleton bound (representing the performance of an idealized MDS code) down to CER $\simeq 10^{-1}$, while for the longer block (Figure 3) the performance curve follows quite closely the Singleton bound down to CER $\simeq 10^{-2}$. Below those values, in both cases the CER curves show high error floors.

The PBC permits to limit the presence of loops in the graph associated with the generated matrix, resulting in lower error floors w.r.t. the FRC. This can be observed in both Figure 2 and in Figure 3. When the short block size is considered, the floor reduction is modest (less than one order of magnitude in the CER), while it is more significant in the case of $n = 2048$. Here, the CER floor is lowered by almost 2 orders of magnitude. For sake of completeness, the performance under IT decoding is provided as well, showing the good behavior of

the code also when decoded with the IT approach. A further reduction of the error floors is obtained by using an outer random code according to the approach discussed in Section II-B. In Figure 2 and in Figure 3 the performance of the PBC IRA code with different (k, k') random outer codes is depicted for ML decoding. For the short block size case (Figure 2), we investigated the effect of $(256, 251)$ and of $(256, 246)$ random outer codes. In the first case, the addition of 5 parity-check equations leads to a code rate reduction to $R' = 0.490$, while in the second case the additional 10 parity-check equations lower to the code rate down to 0.480. In both cases, we provide the associated Singleton bounds as reference. The adoption of the $(256, 251)$ outer code permits to lower the error floor of the IRA code by more than 1 order of magnitude. A large improvement is achieved by increasing the redundancy of the outer code to 10. In this case, no floors are observed down to CER $\simeq 10^{-6}$. When considering the $n = 2048$ case, we applied a $(1024, 1014)$ random out code. Here, the code rate is reduced to $R' = 0.495$. No error floor is observed down to CER $\simeq 10^{-5}$, with a performance that is tightly approaching the Singleton bound for $(2048, 1014)$ codes.

For our simulations, we used the efficient ML decoder described in [13] together with the Maximum Column Weight pivoting algorithm of [12]. The decoding complexity of the algorithm of [13] is dominated by the last decoding step, which consists in solving (via Gaussian elimination) a dense system of binary linear equations in α unknowns, where α unknowns are referred to as *pivots* or *reference variables*. An indirect measure of the decoding complexity is given by the average number of pivots $\overline{\alpha}$ to be solved at a certain erasure probability. Focusing on the case with $\epsilon = 0.5$ and on $n = 512$, the FRC leads to a lower decoding complexity w.r.t the PBC, $\overline{\alpha} = 10$ vs. $\overline{\alpha} = 12$, which is in accordance with the $\epsilon_{ML}^* - \epsilon_{IT}^*$ argument of [8]. Considering the corresponding PBC case, the concatenation with an outer code tends to increase the decoding complexity (i.e., the average number of pivots) by a factor that is related to the number of rows $k - k'$ of $\mathbf{H}_o$. However, even considering the case with $k - k' = 10$, the average number of pivots is kept small, $\overline{\alpha} = 16$, and hence high decoding speeds can be achieved.[6]

IV. APPLICATION TO AERONAUTICAL COMMUNICATIONS

We analyzed the code construction described in the previous sections in the context of the forward link (FL)[7] of the airport surface communications, i.e. the upcoming AeroMACS system. The airport environment may be classified in areas referred to as apron, taxi, parking and runway [19], characterized by different propagation conditions. The taxi and runway scenarios are related to moderate mobility phases and are characterized by line of sight (LOS) conditions. The parking

---

[6]Recall that for a $(2048, 1024)$ IRA code decoding speeds higher than 1.5 Gbps via software decoder implementation of the ML decoding algorithm were demonstrated in [12].

[7]In the aeronautical context, the FL is usually referred to as the link from the control tower to the aircraft, while the reverse link (RL) represents the link from the aircraft to the control tower.
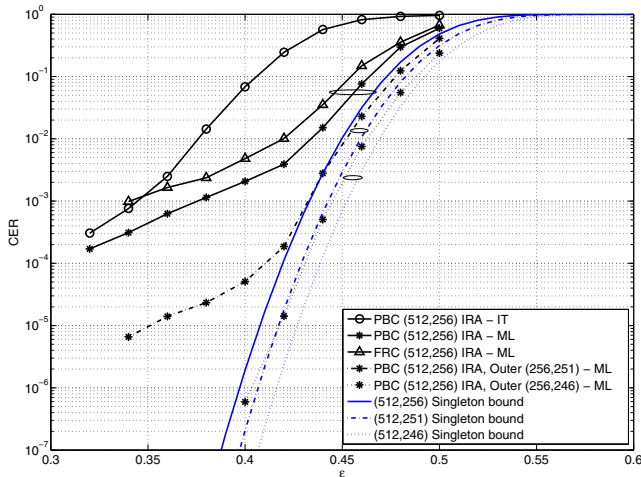
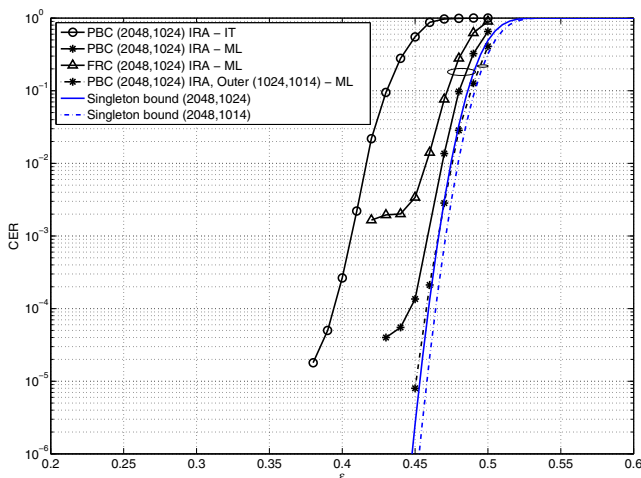Fig. 2.   CERs for $n = 512$, $R \simeq 0.5$ FRC and PBC IRA codes.

Fig. 3.   CERs for $n = 1024$, $R \simeq 0.5$ FRC and PBC IRA codes.

and apron scenarios correspond to areas close to the buildings and are characterized by limited mobility. The presence of the buildings may obstruct the LOS w.r.t. the control tower, stressing the performance of the system. While in LOS conditions the AeroMACS system achieves low bit error probabilities at moderate signal-to-noise ratios, [19], in non line of sight (NLOS) scenarios improvements to the system performance are required. In this context, packet-level coding may enhance the AeroMACS system by allowing coding on large blocks, thus exploiting time diversity. Moreover, packet-level coding may be particulary useful whenever large amounts of information have to be broadcasted to several aircrafts. In this case, rate-compatible (fountain) packet-level codes would efficiently complement Automatic Repeat reQuest (ARQ) protocols [2]. For the simulations, we considered a system based on the orthogonal frequency division multiple access (OFDMA) mode of the WiMAX standard with 512 sub-carriers and a bandwidth of 5 MHz. According to the AeroMACS specification, time division duplex (TDD) is considered. The frame (5 ms long) includes both FL and RL sub-frames. We considered FL

sub-frames of 24 OFDM symbols, a cyclic prefix (CP) of $1/8$ of the symbol length and QPSK sub-carrier modulation. Convolutional coding with rate $1/2$ is applied over 10 slots (being a slot the minimum amount of information allocable to a user), corresponding to 960 bits.

We focused on the most critical scenario, i.e. the parking one. The stochastic airport channel model presented in [19] has been used, which is based on the wide sense stationary uncorrelated scattering (WSSUS) model adapted to the peculiarities of the airport environment. The main channel parameters include Rice factor K equal to 0 dB, 12 taps, a delay spread of 1.25 μs and maximum Doppler spread of 40 Hz. Linear interpolation in the frequency domain for the channel coefficients, based on the pilot tones and tailored for the WiMAX frame structure, has been used.

The performance of the scheme (without packet-level coding) is depicted in Figure 4, in terms of packet error rate (PER) vs. E_b/N_0 (being E_b the energy per information bit and N_0 the one-sided noise power spectral density). The limited time diversity leads to a lack of steepness in the curve. A PER $\simeq 10^{-2}$ is achieved at $E_b/N_0 = 12$ dB. Note that according to simulation parameters, the duration of a FL sub-frame is 2.5 ms, while the channel coherence time is roughly 50 ms. We hence applied different packet-level codes directly at the link layer (i.e., considering as encoding symbols the units encoded by the convolutional code). We selected code rates between $4/5$ and $19/20$, and block sizes of 250, 500 and 1000 symbols. Considering a frame duration of 5 ms (which includes both the FL and the RL sub-frames), the latency introduced by the packet-level codes spans from 1.25 s (for $n = 250$ symbols) to 5 s (for $n = 1000$ symbols), bringing sufficient time diversity to counteract moderate-short outages. The introduced latency is indeed acceptable for file delivery applications, while it is too large for delay-sensitive applications (such as voice communications).

PBCs have been considered, with $\Phi_2(x) = x^5$, which (for a reference rate $R = 9/10$) provides an iterative decoding threshold $\epsilon_{IT} = 0.0733$ and a ML one $\epsilon_{ML} = 0.0992$. No outer code has been used.[8] In Table I, the performance in terms of PER after packet-level decoding is provided for several PBC IRA codes. The Singleton bound[9] is provided as reference. The PBC codes attain in almost all the cases the Singleton bound.

Figure 4 illustrates the PER vs. E_b/N_0 for different rate-$9/10$ PBC IRA codes of different lengths. Again, the performance achievable with idealized MDS codes (Singleton bound) is provided as reference. Already with a block length of 250 symbols, the PBC allows tightly approaching the bound.
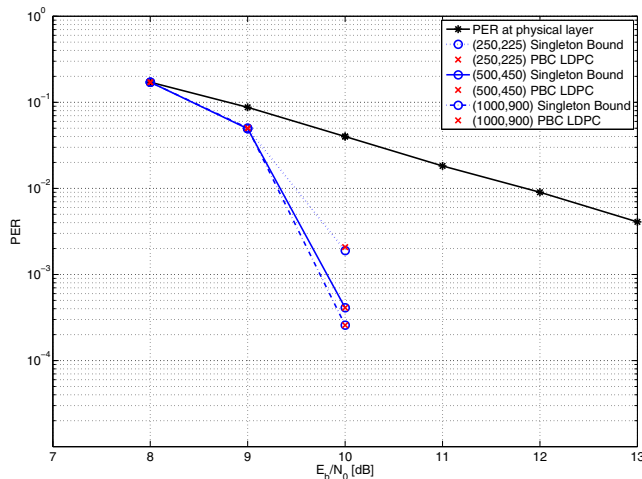
[8]For high code rates (e.g., $R \geq 4/5$) the optimization of the degree distributions may leave place to the use of *sufficiently-dense* near-regular distributions. In fact, for high code rates near-regular distributions have IT thresholds close to the Shannon limit, allowing (in the finite length setting) approaching the Singleton bound while keeping the decoding complexity limited [8].

[9]The Singleton bound represents the performance of an idealized MDS code with a given block length, n, and code dimension k. More specifically, for each set of (n, k) parameters, we analyzed the sequence of packet erasures after physical layer decoding, and we assumed that in a block of n packets, is $n - k$ or less have been erased, the entire block is recovered.

TABLE I

PACKET ERROR RATES FOR VARIOUS (n, k) PBC IRA CODES COMPARED WITH THE SINGLETON BOUND (IN BRACKETS).

(n, k)	$E_b/N_0 = 8$ dB	$E_b/N_0 = 9$ dB	$E_b/N_0 = 10$ dB
-	1.71×10^{-1}	8.27×10^{-2}	4.01×10^{-2}
$(250, 225)$	1.71×10^{-1} (1.71×10^{-1})	4.93×10^{-2} (4.93×10^{-2})	2.20×10^{-3} (1.88×10^{-3})
$(500, 400)$	1.17×10^{-1} (1.17×10^{-1})	1.00×10^{-2} (1.00×10^{-2})	-
$(500, 450)$	1.71×10^{-1} (1.71×10^{-1})	4.93×10^{-2} (4.93×10^{-2})	4.10×10^{-4} (4.10×10^{-4})
$(1000, 800)$	1.22×10^{-1} (1.22×10^{-1})	4.25×10^{-3} (4.25×10^{-3})	-
$(1000, 900)$	1.71×10^{-1} (1.71×10^{-1})	5.02×10^{-2} (5.02×10^{-2})	2.57×10^{-4} (2.57×10^{-4})
$(1000, 950)$	1.71×10^{-1} (1.71×10^{-1})	6.25×10^{-2} (6.25×10^{-2})	1.45×10^{-2} (1.45×10^{-2})

Fig. 4. Packet error rate after physical layer decoding, and after the application of $R = 9/10$ packet level IRA codes (various block lengths).

The gain at PER $\simeq 10^{-2}$ w.r.t. the curve without packet-level coding is ~ 2.5 dB.

V. CONCLUSIONS

Two algorithms for the on-line design of parity-check matrices for IRA codes have been introduced. The proposed algorithms allow a fast design of parity-check matrices that are suitable for low-complexity ML decoding. The code ensembles generated by the proposed algorithms have been analyzed via extrinsic information transfer charts. On of the proposed constructions can attain codeword error rates as low as 10^{-5} without appreciable losses w.r.t. the performance of idealized MDS codes. The proposed techniques have been applied to a packet-level coding scheme for the upcoming aeronautical communication standard, proving the efficiency and the flexibility of the approach.

REFERENCES

[1] M. Luby, M. Mitzenmacher, M. A. Shokrollahi, D. A. Spielman, and V. Stemann, "Practical loss-resilient codes," in *Proc. 29th Symp. Theory Computing*, 1997, pp. 150–159.

[2] J. Byers, M. Luby, and M. Mitzenmacher, "A digital fountain approach to reliable distribution of bulk data," *IEEE Journal on Selected Areas in Communications*, vol. 20, no. 8, pp. 1528–1540, Oct. 2002.

[3] G. Faria, J. Henriksson, E. Stare, and P. Talmola, "DVB-H: Digital Broadcast Services to Handheld Devices," *Proceedings of the IEEE*, vol. 94, no. 1, pp. 194–209, Jan 2006.

[4] "Framing structure, channel coding and modulation for Satellite Services to Handheld devices (SH) below 3GHz," Digital Video Broadcasting (DVB)," Blue Book, 2007.

[5] 3GPP TS 26.346 V6.1.0, "Technical specification group services and system aspects; multimedia broadcast/multicast service; protocols and codecs," June 2005.

[6] G. P. Calzolari, M. Chiani, F. Chiaraluce, R. Garello, and E. Paolini, "Channel coding for future space missions: New requirements and trends," *Proc. IEEE*, vol. 95, no. 11, pp. 2157–2170, Nov. 2007.

[7] E. Paolini, G. Liva, B. Matuz, and M. Chiani, "Generalized IRA erasure correcting codes for hybrid Iterative/Maximum Likelihood decoding," *IEEE Communications Letters*, vol. 12, no. 6, pp. 450–452, June 2008.

[8] E. Paolini, G. Liva, M. Varrella, B. Matuz, and M. Chiani, "Low-Complexity LDPC Codes with Near-Optimum Performance over the BEC," in *Proc. 4th Advanced Satellite Mobile Systems Conference*, Bologna, August 2008.

[9] R. Gallager, *Low-Density Parity-Check Codes*. Cambridge, MA: M.I.T. Press, 1963.

[10] P. Elias, "Coding for two noisy channels," in *In Proc Information Theory: Third London Symposium*. London: Butterworth Scientific, Ed. C. Cherry, 1955, pp. 61–74.

[11] G. Liva, B. Matuz, Z. Katona, E. Paolini, and M. Chiani, "On Construction of Moderate-Length LDPC Codes over Correlated Erasure Channels," in *IEEE Int. Conf. on Communications*, Dresden, Germany, June 2009.

[12] B. Matuz, G. Liva, E. Paolini, and M. Chiani, "Pivoting Algorithms for Maximum Likelihood Decoding of LDPC Codes over Erasure Channels," in *Proc. IEEE Globecomm*, Honolulu, USA, Nov. 2009.

[13] D. Burshtein and G. Miller, "An efficient maximum likelihood decoding of LDPC codes over the binary erasure channel," *IEEE Transactions on Information Theory*, vol. 50, no. 11, nov 2004.

[14] M. Shokrollahi, "Raptor codes," *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2551–2567, June 2006.

[15] H. Jin, A. Khandekar, and R. McEliece, "Irregular repeat-accumulate codes," in *Proc. Int. Symp. on Turbo codes and Related Topics*, Sept. 2000, pp. 1–8.

[16] F. Schreckenbach, M. Schnell, S. Scalise, C. Kissling, "NEWSKY Networking The Sky For Aeronautical Communications," in *Ka and Broadband Communications Conference*, Turin, Italy, Sept. 2007.

[17] D. Medina, F. Hoffmann, F. Rossetto, C. H. Rokitansky, "A Crosslayer Geographic Routing Algorithm for the Airborne Internet," in *Proc. IEEE Int. Conf. on Communications*, Cape Town, South Africa, May 2010.

[18] "IEEE 802.16e System Profile Analysis for FCIs Airport Surface Operation," 2009, European Oragnization for the Safety of Air Navigation.

[19] P. Pulini, "Forward Link Performance Analysis for the Future IEEE 802.16-based Airport Data-Link," in *Proc. IEEE Int. Conf. on Communications*, Cape Town, South Africa, May 2010.

[20] G. Liva, E. Paolini, and M. Chiani, "Simple reconfigurable low-density parity-check codes," *IEEE Communications Letters*, vol. 9, no. 3, pp. 258–260, Mar. 2005.

[21] S. ten Brink, "Convergence behavior of iteratively decoded parallel concatenated codes," *IEEE Transactions on Communications*, vol. 49, pp. 1727–1737, Oct. 2001.

[22] R. Storn and K. Price, "Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, December 1997.

[23] C. Measson, A. Montanari, T. Richardson, and R. Urbanke, "Life above threshold: From list decoding to area theorem and mse," in *Proc. IEEE Information Theory Workshop*, San Antonio, USA, October 2004.