Turbo Codes Construction for Robust Hybrid Multitransmission Schemes

Fulvio Babich and Francesca Vatta

Original scientific paper

***Abstract*—In certain applications the user has to cope with some random packet erasures due, e.g., to deep fading conditions on wireless links, or to congestion on wired networks. In other applications, the user has to cope with a pure wireless link, in which all packets are available to him, even if seriously corrupted. The ARQ/FEC schemes already studied and presented in the literature are well optimized only for one of these two applications. In a previous work, the authors aimed at bridging this gap, giving a design method for obtaining hybrid ARQ schemes that perform well in both conditions, i.e., at the presence of packet erasures and packet fading. This scheme uses a channel coding system based on partially-systematic periodically punctured turbo codes. Since the computation of the transfer function and, consequently, the union bound on the Bit or Frame Error Rate of a partially-systematic punctured turbo code becomes highly intensive as the interleaver size and the puncturing period increase, in this work a simplified and more efficient method to calculate the most significant terms of the average distance spectrum of the turbo encoder is proposed and validated.**

***Index Terms*—Hybrid ARQ, Turbo codes, Puncturing schemes, Multi-transmission schemes.**

I. INTRODUCTION

Certain applications require transmission schemes performing well both at the presence of packet erasures and packet fading. A typical example may be a non-stationary time-varying fading channel, where errors tend to be bursty. During a long deep fade, a packet is received severely corrupted (packet fading) or completely lost (packet erasure) due to bursty errors or packet header errors in the network. A kind of error control in these environments can be performed by adopting an ARQ scheme alone. However, although ARQ systems are simple, easy to implement and provide high system reliability, they suffer a rapid decreasing of the throughput with increased channel error rates. On the other hand, forward error correction (FEC) systems maintain constant the throughput, irrespective of the channel error rates. However, FEC systems have two major drawbacks. First, when a received sequence is detected in error, the sequence has to be decoded and the decoder output has to be delivered to the user regardless of whether it is correct or incorrect. Since the probability of a decoding error is usually greater than the probability of an undetected error, FEC systems are not highly reliable. Second, in order to achieve high system reliability, a long powerful code must be used, which can correct a large number of error patterns. This makes the decoder hard to implement and expensive.

The advantage, typical of ARQ systems, of obtaining high reliability can be coupled with the advantage of FEC systems to provide constant throughput even in poor channel conditions. Such a system, which is a combination of the two basic error control schemes FEC and ARQ, is called a hybrid ARQ system. Hybrid ARQ schemes are of great interest for transmission systems that use time-varying channels, such as the mobile channel, and erasure channels, such as the heterogeneous networks, due to their intrinsic capability to adapt to different channel conditions.

Hybrid ARQ schemes can be classified into two great categories, namely into memoryless and memory ARQ (MARQ) [1]. In conventional memoryless hybrid ARQ schemes the received erroneous frames are discarded and their retransmission is requested. Type-I hybrid ARQ schemes fall into this category [2] and are best suited for channels with a fairly stationary level of noise and interference. In memory ARQ schemes the received erroneous frames are retained and then combined with different strategies by the receiver to reconstruct the original error free frame. It is evident that memory ARQ schemes are more suitable over non-stationary time-varying channels since they allow a dynamic degree of FEC encoding. However, the memory is not the only property that guarantees a good adaptation to the channel conditions. Another important property of ARQ schemes is the self-decodability of additional transmissions, which makes the difference in channels where a frame can be severely damaged [1].

In a previous work [3] the authors aimed at bridging this gap, giving a design method for obtaining a hybrid ARQ scheme, using a rate-compatible turbo codes family, performing well both at the presence of packet erasures and packet fading. In particular, the goal was to obtain a FEC/ARQ scheme performing

1) better, w.r.t. hybrid ARQ schemes already proposed in the literature, when the user has to cope with some random packet erasures (due, e.g., to deep fading conditions on wireless links, or to congestions on wired networks), i.e., when the number of lost data packets (erasures on an erasure channel) is hard to predict;
2) as well as hybrid ARQ schemes proposed in the litera-

Fulvio Babich and Francesca Vatta are with DI3, University of Trieste, Trieste, Italy.

This work was supported by the University of Trieste through the "Finanziamento per i Ricercatori di Ateneo (FRA 2009)" grant.

Part of this work was presented at the IEEE International Conference on Communications, Istanbul, Turkey, 2006 and at the International Conference on Software, Telecommunications and Computer Networks, Split - Hvar - Dubrovnik, Croatia, 2011.

ture when all the packets are available to the final user, even if seriously corrupted, as on the pure wireless link.

To this end, to guarantee a good behaviour on the erasure channel the hybrid ARQ scheme should be designed so that its performance depends simply from the number of received packets, and not from the particular packets that are received; moreover, to guarantee a good performance also on the pure wireless link, the hybrid ARQ scheme should be designed so that it guarantees a rapid convergence of the repetition algorithm.

With these two features, the main disadvantages of classical hybrid ARQ schemes are bypassed. These schemes, in fact, are in general constructed so that the first transmission attempt includes all (or the majority of) information bits. In this sense, the first transmission attempt is the most important one, and if an erasure occurs on this packet, the performance of the overall hybrid ARQ scheme is seriously compromised. A way of bypassing this problem is to transmit the information bits at each transmission attempt, as done in classical systematic complementary hybrid ARQ schemes. This can guarantee that the performance depends simply from the number of received packets, but unfortunally it cannot guarantee a rapid convergence of the repetition algorithm, since, at each repetition, the overall code rate is lessened too slowly.

A novel robust and efficient hybrid multitransmission scheme has been proposed in [3]. This scheme, which uses a channel coding system based on partially-systematic periodically punctured turbo codes, performs particularly well on block erasure channels, i.e., when the number of erasures due, e.g., to deep fading conditions on wireless links, or to congestions on wired networks, is hard to predict. Moreover, it behaves quite well also when the packets may be received seriously corrupted, as on the pure wireless link. However, the evaluation of the transfer function and, consequently, of the union bound on the Bit or Frame Error Rate (BER or FER) of a partially-systematic punctured turbo code becomes computationally intensive as the interleaver size and the puncturing period increase.

In this paper a fast method to calculate the most significant terms of the transfer function of a punctured turbo code is applied and validated. The method is able to quickly determine the same well performing puncturing patterns, for the scenario addressed in [3], but it may applied to more general situations (see, e.g., [4] and [5]).

The paper is organized as follows. Section II introduces an overview of the ARQ schemes under investigation including the novel hybrid multitransmission scheme introduced in [3]. Section III depicts the hybrid ARQ communication system assumptions. In Section IV we recall the results of the encoder analysis and design conducted in [3] and we introduce a rapid method to calculate the most significant terms of the transfer function of a partially-systematic punctured turbo code. This rapid method is shown to give the same results of the classical more complex method using the union bound paradigm we have used in [3]. Finally, in Section V the main conclusions are summarized.

II. Hybrid ARQ schemes overview

Hybrid ARQ schemes may be classified depending on the content of subsequent retransmissions and on their features. Define as *self-decodable* an ARQ scheme for which user data may be recovered from every single retransmission and *balanced* an ARQ scheme for which all retransmissions include the same amount of bits. Observe that, with every scheme type, the first transmission attempt must be self-decodable.

We consider, for comparison, two well known hybrid ARQ schemes, namely the type II ARQ algorithm, called *incremental*, and the type III ARQ algorithm, called *complementary*.

The type II ARQ algorithm adopts an incremental parity retransmission scheme [6], [7] in which additional parity bits alone are sent in subsequent retransmissions. Thus, after every retransmission, a richer set of parity bits is available at the receiver, improving the probability of reliable decoding. Usually, information cannot be recovered from parity bits alone for the retransmissions following the first one. Thus, this scheme is not self-decodable. Moreover, in general it may not be balanced, since the retransmissions may include different amounts of bits. For a fair comparison, however, we restrict our attention to balanced incremental schemes. Being the code rate variable for each retransmission, a very low number of retransmissions may be necessary to receive the packet correctly.

The *type III* ARQ scheme [8] includes both user data and complementary parity bits in every retransmission. Due to repeated information of user data bits, this scheme is less efficient than the type II one. Also in this case, after every retransmission, a richer set of parity bits is available at the receiver, improving the probability of reliable decoding. Moreover, by adopting combining techniques, this reliability is further improved. In this scheme, information can be recovered from every transmission. Thus, this scheme is self-decodable. Moreover, it is also balanced, since all retransmissions include the same amount of bits. However, since the code rate decreases slowly at each retransmission, quite a high number of retransmissions may be necessary to receive the packet correctly.

We have proposed in [3] a hybrid ARQ algorithm using a puncturing scheme designed so that the advantages of incremental and complementary schemes are maintained without the corresponding drawbacks. The scheme is called *robust, efficient and balanced* (*REB*), since it is designed so that all the transmission attempts are self-decodable and include the same amount of bits. These properties of the REB scheme guarantee,

1) that the performance of the repetition algorithm depends simply from the number of received packets, and not from the particular packets that are received;
2) a rapid convergence of the repetition algorithm itself.

The REB ARQ scheme is meant to be incremental but symmetrical, so that to guarantee the same usefulness of each transmission attempt, thanks to its symmetry, and also the rapid convergence of the overall repetition process, thanks to its incremental redundancy, constructed so that the overall code rate is lessened as much as possible at each repetition attempt.

The REB scheme is designed so that information can be recovered from every transmission, as in type III schemes, and a very low number of retransmissions may be necessary to receive the packet correctly, as in type II schemes. These features make REB schemes attractive in erasure channels, where entire packets may be lost. In these channels, their performance was shown in [3] to be better than incremental schemes' performance, since they are self-decodable, and better than complementary schemes' performance, since a lower number of retransmissions may be necessary to receive the packet correctly (lower than the number of retransmissions needed by complementary schemes). For the same reasons, on fading channels, the REB schemes' performance was shown in [3] to be better than complementary schemes' performance and comparable to incremental schemes' performance, since, although REB schemes may need a higher number of retransmissions to receive the packet correctly (higher than the number of retransmissions needed by incremental schemes) they are self-decodable, and thus the information can be recovered even in case of a deep fade condition (this is not possible when using incremental schemes). An example of REB transmission scheme is depicted in Section III-B.

III. Hybrid ARQ communication system assumptions

A. Punctured Turbo Codes

We assume to use a mother code of rate $R = 1/3$, from which all other rates are obtained by puncturing. The mother code is assumed to be a 1/3 eight state turbo code, given by the concatenation of two rate 1/2 convolutional codes, with generators (g_1, g_2)=(13,17) in octal form.

The data source packet of length I (equal to the turbo code interleaver length) is input to the rate-1/3 turbo mother code, thus obtaining a block of size $N = 3I$. The resulting block of code symbols is partitioned into sublocks of size (N/P), where P is the so-called puncturing period. The code construction allows for a family of codes of rates

$$R_l = \frac{P}{l},\ l = P, P+1, ..., P/R \tag{1}$$

For each value of l, we define a binary $3 \times P$ puncturing matrix $\mathbf{a}(l)$. If $a_{ij}(l) = 1$, the j-th bit of each sublock of size (I/P), into which the i-th systematic-parity stream is partitioned, belongs to the subcode of rate R_l (≤ 1). The index $i = 1$ indicates the systematic stream, the index $i = 2$ the parity 1 stream, and the index $i = 3$ the parity 2 stream at the turbo encoder output.

As an example, suppose we construct a rate 4/5 code with $P = 8$, thus, from (1), we should take $l = 10$. A possible puncturing matrix could, for example, take the form

$$\mathbf{a}(10) = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{2}$$

namely, $a_{1j}(10) = 1$, $\forall j$, and one element $a_{ij}(10) = 1$ in the second and third row (for $i = 2, 3$). This means that all the systematic bits are preserved after puncturing, together with only one parity-1 and one parity-2 bits. This code is systematic and, hence, certainly decodable, in the sense that the receiver can invoke the iterative turbo decoding algorithm, and this algorithm does converge above a certain, generally low, SNR threshold.

By allowing the puncturing of systematic bits, we obtain a so-called *partially-systematic* code. To allow convergence of the iterative turbo decoding algorithm, the puncturing pattern should be carefully chosen, in the sense that we should avoid catastrophic and non-invertible codes, since these codes do not allow convergence of the iterative turbo decoding algorithm.

A code is said to be *invertible* if, knowing only the parity-check digits of a code vector, the corresponding information digits can be uniquely determined [9]. Being the turbo code linear, the invertibility is guaranteed iff only one input pattern gives the all zero codeword, i.e., the zero-weight input pattern. If there are non-zero-weight input patterns giving the all zero codeword, the corresponding code is certainly non-invertible. In the following, we report a rate 4/5 code with $P = 8$ and $l = 10$ which is not invertible. Its puncturing matrix could, for example, take the form

$$\mathbf{a}(10) = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{3}$$

It is easy to verify from the trellis of this code that there are input patterns of weight multiple of 3 giving an all-zero codeword, due to the particular puncturing pattern.

A rate 4/5 code with $P = 8$ and $l = 10$ which is invertible and also non catastrophic could have a puncturing matrix of the form

$$\mathbf{a}(10) = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{4}$$

or of the form

$$\mathbf{a}(10) = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{5}$$

These two codes are both non-catastrophic and invertible: they have however different output spectra, which give different performances, as shown in Section IV.

Finally, a code is said to be *catastrophic* if there are self-loops in its trellis for which the input weight is increased without increasing the output distance of the code. In the following, we report a rate 4/5 code with $P = 8$ and $l = 10$ which is catastrophic but invertible. Its puncturing matrix could, for instance, take the form

$$\mathbf{a}(10) = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{6}$$

1st attempt	u_1	u_2	u_3	u_4	u_5	u_6	u_7	u_8	p_{15}	p_{21}
2nd attempt	u_1	u_2	u_3	u_4	u_5	u_6	u_7	u_8	p_{13}	p_{27}
3rd attempt	u_1	u_2	u_3	u_4	u_5	u_6	u_7	u_8	p_{11}	p_{25}
4th attempt	u_1	u_2	u_3	u_4	u_5	u_6	u_7	u_8	p_{18}	p_{24}
5th attempt	u_1	u_2	u_3	u_4	u_5	u_6	u_7	u_8	p_{14}	p_{28}
6th attempt	u_1	u_2	u_3	u_4	u_5	u_6	u_7	u_8	p_{17}	p_{22}
7th attempt	u_1	u_2	u_3	u_4	u_5	u_6	u_7	u_8	p_{12}	p_{26}
8th attempt	u_1	u_2	u_3	u_4	u_5	u_6	u_7	u_8	p_{16}	p_{23}

(a)

1st attempt	u_1	u_2	u_3	u_4	u_5	u_6	u_7	u_8	p_{15}	p_{21}
2nd attempt	p_{11}	p_{13}	p_{14}	p_{17}	p_{18}	p_{22}	p_{23}	p_{25}	p_{26}	p_{27}
3rd attempt	u_1	u_3	u_4	u_5	u_7	u_8	p_{12}	p_{16}	p_{24}	p_{28}

(b)

1st attempt	u_1	u_2	u_3	u_4	u_5	p_{13}	p_{14}	p_{15}	p_{21}	p_{22}
2nd attempt	u_4	u_5	u_6	u_7	u_8	p_{16}	p_{17}	p_{18}	p_{24}	p_{25}
3rd attempt	u_1	u_2	u_3	u_7	u_8	p_{11}	p_{12}	p_{13}	p_{27}	p_{28}

(c)

Fig. 1. Hybrid ARQ schemes under evaluation: (a) complementary, (b) incremental and (c) REB.

B. The protocols under comparison

Balanced ARQ schemes have been taken into consideration in [3] for performance evaluation, and are recalled here for self-consistency sake. For these schemes, all the attempts include the same amount of channel bits.

Fig. 1 describes an example of the schemes considered, in which the information bits are encoded by a rate 1/3 binary turbo code. In this example, the rate of the first transmission of an ARQ cycle is assumed to be $r_1 = 4/5$. The interleaver length is $I = 1024$ bits. The transmission patterns depicted in the figure are applied periodically with period $P = 8$.

The information bits in a transmission attempt are denoted by u_i, the parity check bits at the first constituent code output by p_{1i}, and the parity check bits at the second constituent code output by p_{2i}.

For the complementary scheme [(a) in the figure], all information bits are transmitted in each attempt $(u_1, \cdots, u_8)$, while complementary parity bits are transmitted in subsequent attempts (observe that there are 8 different groups of parity bits). The code used in the first transmission attempt is the best systematic code with rate $r_1^C = \frac{I}{I+P_1^C} = r_1$, where P_1^C are the parity bits transmitted at the first attempt (i.e., p_{15} and p_{21} transmitted with period 8). With *best* we mean that the code let us obtain *the best performance at lower SNRs*, i.e., a superior performance for ARQ applications. The codes used in the i-th transmission attempt following the first one are the best systematic ones with rate $r_i^C = \frac{I}{I+P_1^C+P_2^C+\cdots+P_i^C}$, where P_i^C are the parity bits transmitted at the i-th attempt. To make an example, the code rate of the 2$^{\text{nd}}$ transmission attempt is $r_2^C = 2/3$, the code rate of the 3$^{\text{rd}}$ transmission attempt is $r_3^C = 4/7$, and so on. Please note that each transmission attempt is self-decodable, namely that the codes used at each attempt are of the type (2), i.e., systematic and, hence, certainly decodable.

For the incremental scheme [(b) in the figure], all information bits are transmitted at the first attempt $(u_1, \cdots, u_8)$, together with the first group of parity bits (p_{15} and p_{21}), while the remaining parity bits are transmitted in the second and third attempt. Then the process starts again (if necessary). Storing the previously received data allows the concept of incremental redundancy to be exploited. With incremental redundancy, the system begins by transmitting at the highest FEC code rate. The code used in the first transmission attempt is the best systematic code with rate $r_1^I = \frac{I}{I+P_1^I} = r_1$, where P_1^I are the parity bits transmitted at the first attempt (i.e., p_{15} and p_{21} transmitted with period 8). If a retransmission is requested, then only some of the previously unused parity bits are transmitted. The code used in the second transmission attempt is the best systematic code with rate $r_2^I = \frac{I}{I+P_1^I+P_2^I}$, where P_2^I are the parity bits transmitted at the second attempt. In the example, $r_2^I = 2/5$ (please note that $r_2^C > r_2^I$). With each retransmission, more and more of the parity bits are transmitted and the codeword is strengthened until eventually the receiver has all the parity bits. In the example, the receiver has all the parity bits at the 3$^{\text{rd}}$ transmission attempt, i.e., the code used in the third transmission attempt has rate $r_3^I = \frac{I}{I+P_1^I+P_2^I+P_3^I} = 1/3$, where P_3^I are the parity bits transmitted at the third attempt. Incremental redundancy allows a fast reduction of the effective code rate at each transmission attempt, until the packet can be successfully decoded. Thus, as it can be expected, the throughput of the type II ARQ scheme is significantly higher than it is for the type III ARQ one. Please note that, in this example, the first and third transmission attempts are self-decodable (only the second one is not self-decodable). In general, i.e., with higher initial rates r_1, only the first attempt would be self-decodable.

For the REB scheme [(c) in the figure], some information bits are transmitted at each attempt together with some parity bits. Their number must be such that the first constituent code rate $R_{\text{upper}} \leq 1$ at every transmission attempt. Moreover, they must be chosen so that the invertibility of the used code is guaranteed at each attempt, i.e., to be of the type (4) or (5). In this example, the type (5) code has been selected since its spectral characteristics allow a better performance in the region of interest, as shown in the following Section IV.

We need codes of the type (4) or (5) since it is desirable to split the encoded sequence into subsequences to be sent in successive transmissions. Thus, a basic requirement is that each subsequence must permit the recovery of the original information in case of no errors [10]. The code used in the first transmission attempt is the best partially systematic code with rate $r_1^R = \frac{I}{I_1^R+P_1^R} = r_1$, where I_1^R are the systematic bits transmitted at the first attempt (i.e., $u_1, \cdots, u_5$ transmitted with period 8) and P_1^R are the parity bits transmitted at the first attempt (i.e., p_{13}, p_{14}, p_{15}, p_{21} and p_{22} transmitted with period 8). The code used in the i-th transmission attempt is the best partially systematic code with rate $r_i^R = \frac{I}{I_1^R+I_2^R+\cdots+I_i^R+P_1^R+P_2^R+\cdots+P_i^R}$, where I_i^R are the systematic bits newly transmitted at the i-th attempt (i.e., not transmitted at either previous attempt) and P_i^R are the parity bits newly transmitted at the i-th attempt. To make an example, the code rate of the 2^{nd} transmission attempt is $r_2^R = 4/9$ and the code rate of the 3^{rd} transmission attempt is $r_3^R = 4/11$. Please note that each transmission attempt is self-decodable and that the partially systematic codes used in each attempt, if taken alone, have the same performance. Thus, each combination of these partially systematic codes let us obtain the same performance, which does not rely on which of the three groups of bits is available at the receiver, but simply on the number of groups which are not erased or seriously corrupted.

IV. Optimal puncturing analysis and design

A. Partially-systematic puncturing analysis

A binary $3 \times P$ puncturing matrix $\mathbf{a}(l)$, as those shown in Section III-A, can be described also by its permeability rates ρ [11]. Let's define the permeability rate ρ_u as the proportion of information bits that are not punctured, i.e., survive after puncturing, at the output of the turbo encoder. ρ_u is simply given, in our example, by the number of ones in the first row of $\mathbf{a}(l)$. Let's define, consequently, the permeability rates ρ_{p_1} and ρ_{p_2} as the proportion of parity 1 and parity 2 bits, respectively, that are not punctured at the output of the turbo encoder. Again, ρ_{p_1} and ρ_{p_2} are simply given, in our example, by the number of ones in the second and third row, respectively, of $\mathbf{a}(l)$. Thus, the relation between the permeability rates and the turbo code rate is given by:

$$\frac{1}{R} = \rho_u + \rho_{p_1} + \rho_{p_2} \tag{7}$$

The puncturing design should let us obtain codes performing better at lower SNRs, i.e., with a superior performance for ARQ applications. It is critical for the REB scheme, since the codes needed in this protocol must be chosen so that their invertibility is guaranteed at each attempt, and so that their spectral characteristics allow a better performance in the region of interest.

To make an example on how the codes should be chosen for this scheme, consider Fig. 1 (c), referring to the REB scheme. The code chosen for the first transmission attempt has the puncturing matrix (5). Please, notice that the codes used in all other transmission attempts are obtained through a quasi-cyclic permutation of the puncturing pattern over the puncturing period.

First of all we shall focus on the maximum number of information bits that can be maintained to maximize d_{eff}, which is the minimum output weight given by a weight-two input. It is well known, in fact, that the codes for which the d_{eff} is maximum perform better in the error floor region and that, lowering the number of information bits (and incrementing, correspondingly, the number of parity check bits) to obtain a code of a given rate, d_{eff} will be maximized, as shown in Fig. 2, thus lowering the error floor [12]. However, we should remind that information bits make iterative decoding

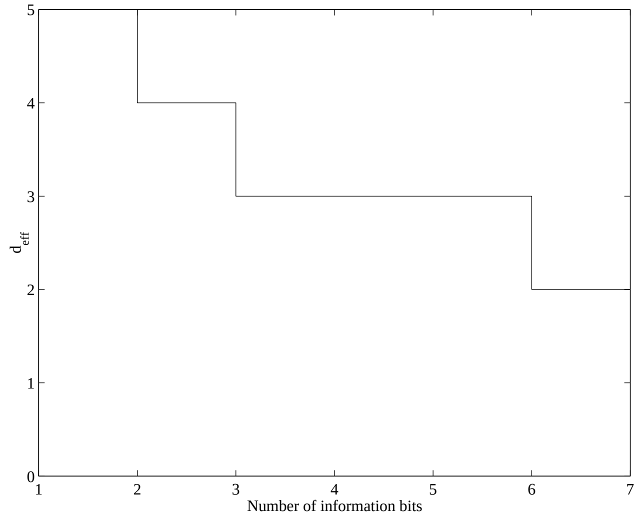

Fig. 2. d_{eff} output distance vs. the number of information bits used to construct a code of rate 4/5.

easier, yielding better convergence abscissæof the iterative decoding algorithm, which are fundamental in the region of low SNRs [12]. Thus, the best code design should adopt a compromise between the maximization of d_{eff}, obtained when $(\rho_{p_1} + \rho_{p_2})$ is maximized, and the minimization of the convergence abscissa, obtained when ρ_u is maximized. In this sense, the choice we have done of selecting the number of surviving information bits equal to the number of surviving parity bits, i.e., $\rho_u = \rho_{p_1} + \rho_{p_2}$, represents a good compromise between the two above mentioned goals.

Now, once we have fixed ρ_u, ρ_{p_1}, and ρ_{p_2} to obtain a rate 4/5 code, we have many choices to do on the positions that the information and the parity check bits should have inside the puncturing period (i.e., on the positions of the ones inside the puncturing matrix). For instance, other codes of this type (i.e., with 5 ones in the first row) could have a puncturing matrix like this

$$\mathbf{a}(10) = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{8}$$

or like this

$$\mathbf{a}(10) = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{9}$$

In order to perform a choice among the puncturing positions, the weight enumerators and union bound analysis has to be performed.

B. Weight enumerators and union bound analysis

We follow the approach of Benedetto and Montorsi [13] for parallel concatenated convolutional codes (PCCCs). The component codes of the parallel concatenated codes are connected through uniform random interleavers. The important property of the uniform interleaver is that its output depends only on the input weight w, not on the distribution of the weight within the input word. A uniform interleaver of length K maps an input word of weight w into all of its $\binom{K}{w}$ possible permutations with equal probability. As a consequence, the use of the uniform interleaver drastically simplifies the performance evaluation of turbo codes.

Denote by w_m the minimum weight of an input sequence generating an error event of the parallel concatenated code C, and by h_m and h_M the minimum and maximum weight, respectively, of the codewords of C. Also, let $A^{C}_{w,h}$ denote the Input-Output Weight Enumerating Function (IOWEF), i.e., the average number of codewords in code C with input weight w and output weight h. Similarly, we define $A^{C_U}_{w,h_U}$ and $A^{C_L}_{w,h_L}$ for the upper constituent code C_U and the lower constituent code C_L, respectively. The bit error probability of a PCCC over an additive white Gaussian noise channel can be upper bounded by [13]

$$P_b(e) \leq \frac{1}{2} \sum_{h=h_m}^{h=h_M} \sum_{w=w_m}^{K} \frac{w}{K} A^{\mathcal{C}}_{w,h} \mathrm{erfc}\left(\sqrt{\frac{hRE_b}{N_0}}\right) \tag{10}$$

where $N_0/2$ is the two-sided noise power spectral density and E_b is the energy per information bit.

Likewise, the frame error probability is upper bounded by

$$P_w(e) \leq \frac{1}{2} \sum_{h=h_m}^{h=h_M} A^{\mathcal{C}}_{h} \mathrm{erfc}\left(\sqrt{\frac{hRE_b}{N_0}}\right) \tag{11}$$

where $A^{\mathcal{C}}_h = \sum_{w=w_m}^{K} A^{\mathcal{C}}_{w,h}$.

$A^{\mathcal{C}}_{w,h}$ can be calculated by replacing the actual interleaver with the uniform interleaver [13] and exploiting its properties. The uniform interleaver of length K transforms an input sequence of weight w at the input of the upper constituent encoder into all its distinct $\binom{K}{w}$ permutations. As a consequence, each input sequence of the upper code of weight w, through the action of the uniform interleaver, enters the lower constituent encoder generating $\binom{K}{w}$ codewords of the lower code. The IOWEF of the overall PCCC can then be evaluated from the knowledge of the IOWEFs of C_U and C_L [13]:

$$A^{C}_{w,h} = \frac{A^{C_U}_{w,h_U} A^{C_L}_{w,h_L}}{\begin{pmatrix} K \\ w \end{pmatrix}} \tag{12}$$

where h, h_U, and h_L are related by the equation $h = h_U + h_L$.

The puncturing positions are selected so that to obtain the minimum residual Bit Error Rate (BER) and Frame Error Rate (FER) in the waterfall region, i.e., beneath the cutoff rate of the code, since a good performance in this region is important for the applications considered in this paper.

The computation of the union bounds (10) and (11) on the BER and FER, respectively, of a punctured turbo code becomes more and more heavy as the interleaver size K and the puncturing period increase [14]. However, a rapid method to calculate the most significant terms of the distance spectrum of a punctured turbo code, which can be used as a close approximation of the union bound, can be applied.

In particular, the following procedure has been applied in [3] and is described here more in detail. Given the IOWEF $A^{C}_{w,h}$ of a punctured parallel concatenated code, calculated employing a uniform interleaver approach [13], its residual BER and FER can be calculated by (10) and (11), respectively. However, it was shown in [13] that minimum information weight codewords are the principal contributors to residual BER and FER, as the size K of the uniform interleaver increases, and that, when Recursive Systematic Convolutional (RSC) constituent encoders are used, this minimum information weight is equal to two.

Thus, for large interleaver sizes, it follows that the input weight $w = 2$ is the dominant contributor to the union bounds (10) and (11) on the BER and FER, respectively. Therefore, Eqs. (10) and (11) can be rewritten as:

$$P_b(e) \leq \frac{1}{2} \sum_{h=h_m}^{h=h_M} \frac{2}{K} A^{\mathcal{C}}_{2,h} \mathrm{erfc}\left(\sqrt{\frac{hRE_b}{N_0}}\right) \tag{13}$$

$$P_w(e) \leq \frac{1}{2} \sum_{h=h_m}^{h=h_M} A^{\mathcal{C}}_{2,h} \mathrm{erfc}\left(\sqrt{\frac{hRE_b}{N_0}}\right) \tag{14}$$

Thus, instead of the minimization of (10) and (11), the minimization of (13) and (14) may be performed, which is computationally less cumbersome. This leads to the minimization of $\sum_{h=h_m}^{h=h_M} A^{\mathcal{C}}_{2,h}$. In Fig. 2 in [3] we have reported $\sum_{h=h_m}^{h=h_M} A^{\mathcal{C}}_{2,h}$ for the code (5) (solid line), for the code (8) (dotted line) and for the catastrophic code (9) (dashed line), which has the best cumulative function (since it is minimal), as it usually happens when catastrophic codes are considered. Of course, the code having the minimal $\sum_{h=h_m}^{h=h_M} A^{\mathcal{C}}_{2,h}$ must be chosen among the invertible and non-catastrophic codes, and thus, in this sense, the code (5) we have selected in [3] is the best.

To further simplify the choice of the puncturing pattern positions, an alternative simpler analysis method was presented in [15] and is recalled in the following subsection.

C. Puncturing design through a simplified distance spectrum of the resulting turbo code

One very widely used approximation of the $\mathrm{erfc}(x)$ function is the Chernoff bound:

$$\mathrm{erfc}(x) \leq 2\exp(-x^2) \tag{15}$$

This bound can be tightened as:

$$\mathrm{erfc}(x) \leq \exp(-x^2) \tag{16}$$

Using the approximation (16) and switching to the natural logarithmic expression, the puncturing pattern can be selected as the one minimizing the following:

$$\max_h \left[\ln A^{\mathcal{C}}_{2,h} - \frac{hRE_b}{N_0}\right] \tag{17}$$

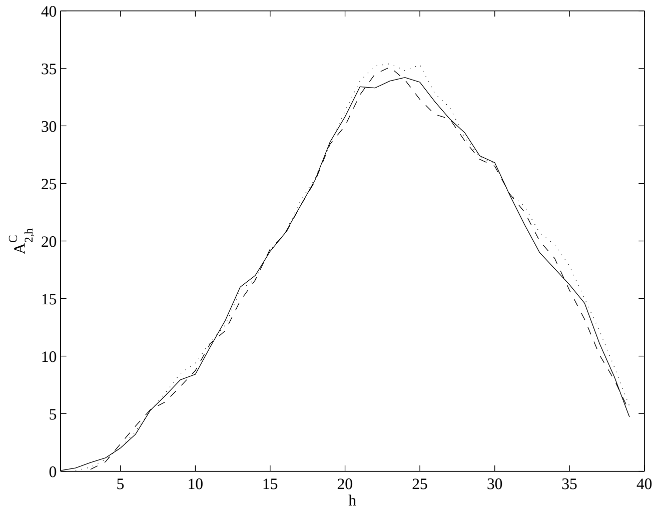

Fig. 3. The output spectrum $A^{\mathcal{C}}_{2,h}$ vs. the output distance h of three rate 4/5 codes. Solid line: code (5); dotted line: code (8); dashed line: code (9).

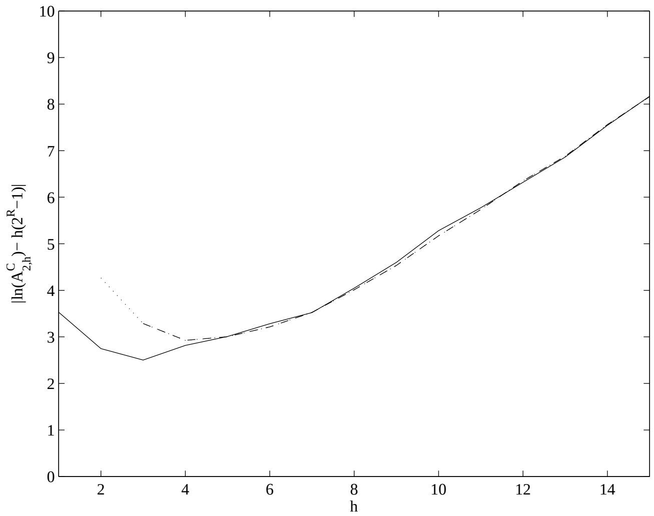

Fig. 4. The output spectrum $|\ln A^{\mathcal{C}}_{2,h} - h(2^R - 1)|$ vs. the output distance h of three rate 4/5 codes. Solid line: code (5); dotted line: code (8); dashed line: code (9).

with $\frac{E_b}{N_0}$ given by $\frac{2^R-1}{R}$, which is the cutoff rate of the code, having assumed a BPSK modulation.

The two criteria (i.e., this one and the one proposed in the previous subsection) can be shown to be equivalent. In fact, given two puncturing patterns presenting the same output weight h distribution with input weight $w = 2$, the minimization of (13) and (14) leads to the minimization of $\sum_{h=h_m}^{h=h_M} A^{\mathcal{C}}_{2,h}$. However, considering the minimization of (17), which is computationally less cumbersome, in place of the minimization of (13) and (14), the same results can be obtained.

In Fig. 3 we report the output spectrum $A^{\mathcal{C}}_{2,h}$ for the code (5) (solid line), for the code (8) (dotted line) and for the catastrophic code (9) (dashed line). In Fig. 4 we report the values of

$$|\ln A^{\mathcal{C}}_{2,h} - \frac{hRE_b}{N_0}| = |\ln A^{\mathcal{C}}_{2,h} - h(2^R - 1)| \qquad (18)$$

being $\frac{E_b}{N_0}$ given by $\frac{2^R-1}{R}$, which is the cutoff rate of the code and $R = 4/5$. These values are shown for the code (5) (solid line), for the code (8) (dotted line) and for the catastrophic code (9) (dashed line).

Of course, following this simplified criterion, the best puncturing pattern can be selected as the one minimizing the following quantities:

$$\max_h \left[\ln A^{\mathcal{C}}_{2,h} - h(2^R - 1)\right] = \min_h |\ln A^{\mathcal{C}}_{2,h} - h(2^R - 1)| \qquad (19)$$

and thus, in this sense, as it was observed in the previous subsection, the code (5) is the best.

Although not rigorously proved [16], it is observed that for capacity approaching codes such as turbo codes, there is a tradeoff between low pinch-off thresholds, i.e., good waterfall performance, and high effective free distance values, determining good error-floor performance [13]. Actually, in all the cases examined, the distance spectrum of the codes with minimum residual FER in the waterfall region, presented low values of the effective free distance $d_{\mathrm{f,eff}}$, defined in [13] as the minimum weight of code sequences generated by input sequences of weight 2. Thus, a design guideline may be that of restricting the puncturing pattern search to the patterns giving codes with low values of $d_{\mathrm{f,eff}}$, since the obtained codes have a bad error-floor performance, but, as a tradeoff, a very good waterfall performance, as shown in Fig. 5, where the simulated residual FER performance of three rate 8/9 codes with different values of $d_{\mathrm{f,eff}}$ is shown on the AWGN channel. The thick solid curve in the figure shows the sphere-packing bound limiting performance [17], [18].

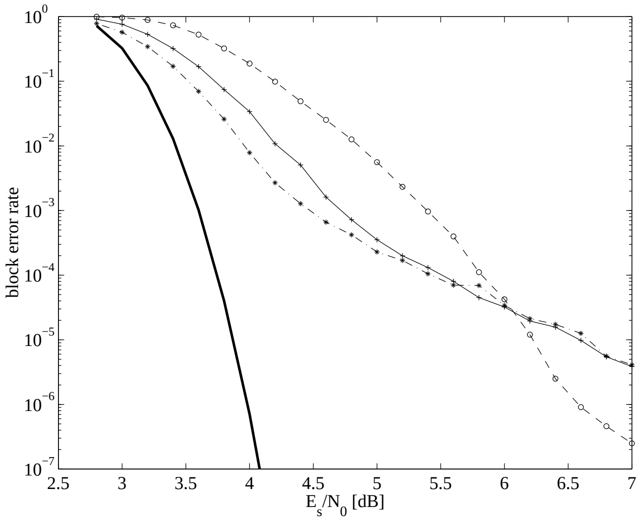

Fig. 5. Comparison of the residual FER performance obtained for 3 rate 8/9 codes with different values of $d_{\mathrm{f,eff}}$. Thick curve: sphere packing bound. Solid and dash-dotted curves: $d_{\mathrm{f,eff}} = 1$; dashed curve: $d_{\mathrm{f,eff}} = 2$.

V. Comments and Conclusions

In this paper we have investigated further on robust, efficient and balanced (REB) rate-compatible puncturing schemes for turbo codes. Their use is meant for hybrid ARQ schemes, in which the data transmission is supposed to go on progressively. The performance of the hybrid ARQ algorithm adopting these optimized puncturing schemes, called REB, has been compared in [3] with type II (incremental) and III (complementary) hybrid ARQ schemes' performance, adopting a channel coding system based on the use of non-optimized punctured turbo codes.

The REB scheme was shown in [3] to perform particularly well when the number of lost data packets (erasures on an erasure channel) is hard to predict. In particular, it was shown to have a good performance for each combination of SNR value and erasure rate, i.e., to be erasure resistant as much as complementary schemes and also efficient as much as incremental schemes.

In this paper we have investigated further on the encoder design for the REB scheme presented in [3]. This scheme uses a channel coding system based on punctured turbo codes. Since the computation of the transfer function and, consequently, the union bound on the Bit or Frame Error Rate (BER or FER) of a punctured turbo code becomes highly intensive as the interleaver size and the puncturing period increase, a rapid method to calculate the most significant terms of the transfer function of a punctured turbo code is proposed and validated.

The two criteria, namely the one proposed in [15] and the one proposed in [3], have been shown to be equivalent. In fact, considering the minimization of (17), which is computationally less cumbersome, in place of the minimization of (14), the same results, in terms of puncturing pattern choice, have been obtained with a significant gain in computation complexity.

Comparisons with other design tools, such as the EXIT charts [19] which may accurately model the code behavior in the waterfall region, are left for further investigation.

References

[1] S. Kallel, R. Link and S. Bakhtiyari, "Throughput performance of memory ARQ schemes", *IEEE Transactions on Vehicular Technology*, Vol. 48, No. 3, May 1999, pp. 891-899.

[2] S. Lin, D. J. Costello, Jr. and M. J. Miller, "Automatic-Repeat-Request Error Control Schemes", *IEEE Communications Magazine*, Vol. 22, No. 12, December 1984, pp. 5-17.

[3] F. Babich and F. Vatta, "Robust, efficient and balanced (REB) rate-compatible puncturing schemes for hybrid ARQ algorithms using turbo codes", *Proc. of the IEEE 2006 International Conference on Communications - ICC '06*, Istanbul, Turkey, June 11-15, 2006, pp. 1154-1159.

[4] I. Chatzigeorgiou, M. R. D. Rodrigues, I. J. Wassell, and R. Carrasco, "A novel technique for the evaluation of the transfer function of punctured turbo codes", *Proc. of the IEEE 2006 International Conference on Communications - ICC '06*, Istanbul, Turkey, June 11-15, 2006.

[5] I. Chatzigeorgiou, M. R. D. Rodrigues, I. J. Wassell, and R. Carrasco,"Analysis and design of punctured rate-1/2 turbo codes exhibiting low error floors", *IEEE Journal on Selected Areas in Communications*, Vol. 27, No. 6, August 2009, pp. 944-953.

[6] S. Kallel and D. Haccoun, "Generalized type II hybrid ARQ scheme using punctured convolutional coding", *IEEE Transactions on Communications*, Vol. 38, No. 11, November 1990, pp. 1938-1946.

[7] M. A. Kousa and M. Rahman, "An Adaptive Error Control System Using Hybrid ARQ Schemes", *IEEE Transactions on Communications*, Vol. 39, n. 7, July 1991, pp. 1049-1057.

[8] S. Kallel, "Complementary Punctured Convolutional (CPC) Codes and Their Applications", *IEEE Transactions on Communications*, Vol. 43, No. 6, June 1995, pp. 2005-2009.

[9] S. Lin and D. J. Costello, Jr., "Error control coding, fundamentals and applications", *Prentice-Hall, Inc.*, New Jersey, 1983.

[10] A. Dholakia, M. A. Vouk and D. L. Bitzer, "Variable-redundancy hybrid ARQ scheme using invertible convolutional codes", *Proc. of VTC '94*, Vol. 3, Stockholm, 8-10 June 1994, pp. 1417-1420.

[11] I. Land and P. Hoeher. Partially systematic rate 1/2 turbo codes. *Proc. of the II International Symposium on Turbo codes and related topics*. Brest, pp. 287–290, Sept. 2000.

[12] S. Benedetto and G. Montorsi, "Versatile bandwidth-efficient parallel and serial turbo-trellis-coded modulation", *Proc. of the II International Symposium on Turbo codes and related topics*, Brest, 4-7 September 2000, pp. 201-208.

[13] S. Benedetto and G. Montorsi, "Design of parallel concatenated convolutional codes", *IEEE Transactions on Communications*, Vol. 44, n. 5, May 1996, pp. 591-600.

[14] F. Babich, G. Montorsi and F. Vatta, "On rate-compatible punctured turbo codes (RCPTC) design", *EURASIP Journal on Applied Signal Processing*, Special Issue on Turbo Processing, Vol. 2005, No. 6, May 2005, pp. 784-794.

[15] F. Babich and F. Vatta, "Turbo Codes Construction for Robust Hybrid Multitransmission Schemes", *Proc. of the 2011 International Conference on Software, Telecommunications and Computer Networks - SOFTCOM '11*, Split - Hvar - Dubrovnik, Croatia, September 15 - 17, 2011.

[16] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara, "Serial concatenation of interleaved codes: performance analysis, design and iterative decoding", *IEEE Transactions on Information Theory*, Vol. 44, n. 3, May 1998, pp. 909-926.

[17] C. Schlegel and L. Perez, "On error bounds and turbo-codes", *IEEE Communications Letters*, Vol. 3, n. 7, July 1999, pp. 205-207.

[18] F. Babich, "On the performance of efficient coding techniques over fading channels", *IEEE Transactions on Wireless Communications*, Vol. 3, n. 1, January 2004, pp. 290-299.

[19] F. Babich, A. Crismani, and R. G. Maunder, "EXIT chart aided design of periodically punctured turbo codes", *IEE Electronic Letters*, Vol. 46, n. 14, July 2010, pp. 1001 1003.

Fulvio Babich Fulvio Babich received the doctoral degree, (Laurea), cum laude, in Electrical Engineering, at the University of Trieste, on July 1984. After graduation he was with Telettra, working on optical system design. Then he was with Zeltron, working on communication protocols. In 1992 he joined the Department of Electrical Engineering (DEEI) of the University of Trieste, where he is Associate Professor of Digital Communications. His current research interests are in the field of wireless networks and personal communications. He is involved in channel modeling, hybrid ARQ techniques, channel coding, cross-layer design, and multimedia transmission over heterogeneous networks. Fulvio Babich is senior member of IEEE.

Francesca Vatta received a Laurea in Ingegneria Elettronica in 1992 from University of Trieste, Italy. From 1993 to 1994 she has been with Iachello S.p.A., Olivetti group, Milano, Italy, as system engineer working on design and implementation of Computer Integrated Building (CIB) architectures. Since 1995 she has been with the Department of Electrical Engineering (DEEI) of the University of Trieste where she received her Ph.D. degree in telecommunications, in 1998, with a Ph.D. thesis concerning the study and design of source-matched channel coding schemes for mobile communications. In November 1999 she became assistant professor at University of Trieste. Starting in 2002, she spent several months as visiting scholar at the University of Notre Dame, Notre Dame, IN, U.S.A., cooperating with the Coding Theory Research Group under the guidance of Prof. D. J. Costello, Jr. Starting in 2005, she spent several months as visiting scholar at the University of Ulm, Germany, cooperating with the Telecommunications and Applied Information Theory Research Group under the guidance of Prof. M. Bossert. She is an author of more than 70 papers published on international journals and conference proceedings. Her current research interests are in the area of channel coding concerning, in particular, the analysis and design of concatenated coding schemes for wireless applications.