A Survey and Tutorial on Low-Complexity Turbo Coding Techniques and a Holistic Hybrid ARQ Design Example

Hong Chen, Robert G. Maunder, *Member, IEEE* and Lajos Hanzo, *Fellow, IEEE*

***Abstract*—Hybrid Automatic Repeat reQuest (HARQ) has become an essential error control technique in communication networks, which relies on a combination of arbitrary error correction codes and retransmissions. When combining turbo codes with HARQ, the associated complexity becomes a critical issue, since conventionally iterative decoding is immediately activated after each transmission, even though the iterative decoder might fail in delivering an error-free codeword even after a high number of iterations. In this scenario, precious battery-power would be wasted. In order to reduce the associated complexity, we will present design examples based on Multiple Components Turbo Codes (MCTCs) and demonstrate that they are capable of achieving an excellent performance based on the lowest possible memory octally represented generator polynomial $(2,3)_o$. In addition to using low-complexity generator polynomials, we detail two further techniques conceived for reducing the complexity. Firstly, an Early Stopping (ES) strategy is invoked for curtailing iterative decoding, when its Mutual Information (MI) improvements become less than a given threshold. Secondly, a novel Deferred Iteration (DI) strategy is advocated for the sake of delaying iterative decoding, until the receiver confidently estimates that it has received sufficient information for successful decoding. Our simulation results demonstrate that the MCTC aided HARQ schemes are capable of significantly reducing the complexity of the appropriately selected benchmarkers, which is achieved without degrading the Packet Loss Ratio (PLR) and throughput.**

***Index Terms*—Multiple-component turbo codes, ARQ, automatic repeat request, EXIT chart, iterative detection.**

I. Turbo Coding Complexity and Motivation

IN TELECOMMUNICATION networks, reliable transmission constitutes one of the ultimate design objectives. Forward Error Correction (FEC) [1] and Automatic Repeat reQuest (ARQ) [2] are the most salient solutions, which are capable of enhancing the achievable transmission reliability.

The family of FEC codes is capable of recovering the information bits by incorporating carefully controlled redundancy based on different codes. The first FEC code was the single error correcting Hamming code [3], which was invented in 1950. Since then, more potent codes have been developed, such as Bose-Chaudhuri-Hocquenghem (BCH) block codes, Convolutional Codes (CC) and turbo codes as well as fountain codes. Figure 1.1 of [1] outlines the brief history of FEC codes. The ratio of the number of information bits to the total number of information and parity bits defines the normalized throughput or the coding rate. Shannon's channel capacity determines the upper bound of the coding rate that any FEC code may be able to achieve at a certain Signal Noise Ratio (SNR). Since the transmission of these parity bits requires an increased bandwidth, the maximum coding rate that an FEC code can have, while still recovering the information bits becomes a useful criterion for quantifying the capability of FEC codes. Their decoding complexity is also a critical factor in the evaluation of FEC codes. Researchers have studied the associated tradeoff between these two aspects, when choosing a specific FEC code for a communication system.

A. Turbo Code Complexity

As one of the most powerful codes in the FEC family, turbo codes [4] have shown capacity-achieving capability by combining two parallel concatenated Recursive Systematic Convolutional (RSC) codes at the transmitter. At the receiver, iterative exchange of soft information is carried out between the two so-called Bahl, Cocke, Jelinek and Raviv (BCJR) decoders [5]. The BCJR decoder is also often referred to as the Maximum A *posteriori* (MAP) algorithm, which estimates a decoded bit by selecting the specific transition path having the maximum *a posteriori* probability among all transition paths from one state to the next in the decoder's trellis [6]. Since the calculation of each transition probability involves the exploration of all possible paths through the trellis, the complexity of the BCJR decoder is potentially high, especially when several iterations are used for exchanging soft information between two BCJR decoders.

Researchers have been striving for reducing the complexity of turbo codes, approaching the problem from all aspects. First of all, they aimed for simplifying the MAP algorithm itself. As a result, the Max-Log-MAP algorithm proposed in 1995 [7] significantly decreased The MAP algorithm's complexity, whilst imposing only a modest performance degradation. This was achieved by transforming the multiplications involved in the MAP algorithm to low-complexity additions carried out

Manuscript received April 23, 2012; revised November 5, 2012. The financial support of the EPSRC, UK, of the RCUK under the auspices of the India-UK Advanced Technology Center (IU-ATC), that of the EU's CONCERTO project and the European Research Council's Advanced Fellow Grant is gratefully acknowledged. H. Chen would also like to acknowledge the support of the Fundamental Research Funds for the Central Universities, China under Grants No. ZYGX2011J058.

H. Chen is with the School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 610054, China (e-mail: hong.chen@surrey.ac.uk). She is also with the CCSR, University of Surrey, GU2 7XH, U.K.

R. G. Maunder and L. Hanzo are with School of Electronics and Computer Science, University of Southampton, UK (e-mail: lh@ecs.soton.ac.uk).

Digital Object Identifier 10.1109/SURV.2013.013013.00079

in the logarithmic domain, and then further approximating the calculation of its basic function of $\ln\left(\sum_i e^{x_i}\right)$ by a single term, namely by the maximum one. In order to avoid the modest, but not negligible performance degradation of the Max-Log-MAP decoder, Robertson *et al.* [7] suggested employment of the Jacobean logarithm for correcting the approximation, which may be expressed as [7]:

$$\ln\left(e^{x_1} + e^{x_2}\right) = max(x_1, x_2) + f(\cdot), \tag{1}$$

where the function $f(\cdot)$ represents a correction term, which may be pre-stored in a look-up table storing the values of $ln\left(1 + e^{-|x_1 - x_2|}\right)$. This technique is referred to as the Log-MAP algorithm, which is invoked in most of today's turbo decoding implementations. Naturally, the Max-Log-MAP algorithm has a reduced complexity, which is achieved at the cost of suffering a slight performance degradation. By contrast, the Log-MAP algorithm imposes a somewhat higher complexity. In 2004, Park [8] combined these two algorithms for the sake of achieving a near-MAP performance at the cost of a reduced complexity.

Even though the per-iteration complexity has been reduced by the above-mentioned innovative solutions, the complexity of turbo codes remains substantial due to their iterative nature. It has been observed that the first few iterations tend to result in the most substantial performance improvements. However, after these high-gain initial iterations, the improvements typically become marginal. Hence, it has been suggested in [9] to curtail iterations, when the Mutual Information (MI) improvements between the soft-information and its hard-decision version become low. The Early Stopping (ES) decoding strategies proposed by numerous researchers [10]–[28] may be classified into the following categories:

1) Setting the number of iterations

Conventional turbo codes tend to employ a fixed number of iterations, albeit this is somewhat wasteful. In [10], Kim *et al.* set the number of iterations according to the estimated Channel State Information (CSI), which reflects the current channel SNR. The appropriate number of iterations was then pre-determined for different CSIs in [10] based on the extrinsic information improvements experienced in consecutive iterations of the BCJR decoders. Furthermore, the authors of [11] exploited the cross correlation between the Log-Likelihood Ratios (LLRs) for finding the specific SNR threshold, at which the Extrinsic Information Transfer (EXIT) chart has a 'just' open tunnel. For the SNRs below this threshold, the number of iterations is set to zero, otherwise it is set to a fixed number.

2) Early stopping based on MI thresholding

In turbo codes, the mutual information increase achieved by each iteration may be represented by diverse variables. Since the amount of MI improvement eventually converges to zero, the MI improvement may be used as a stopping criterion, because if $M(i)$ is below the threshols of $T(i)$, indicating marginal iteration gains, then the iterations may be curtailed. More specifically, this implies that the iterative decoding operations will be curtailed, when an appropriately chosen function of the MI $M(i)$ becomes less than the threshold $T(i)$ at the i*th* iteration. The threshold $T(i)$ can be determined by observing the MI results of offline simulations. The ES strategies of [12], [13] quantified $M(i)$ in terms of the Cross Entropy (CE) between the distributions of the *a posteriori* LLRs generated by the two BCJR decoders, namely

$$M(i) = \sum_k \frac{|\triangle L_{e_2}^{(i)}(\tilde{u}_k)|^2}{exp(|L_1^{(i)}(\tilde{u}_k)|)}, \tag{2}$$

where L_{e_2} denotes the extrinsic LLRs generated by the second BCJR decoder, L_1 represents the *a posteriori* LLRs of the first BCJR decoder, while $\tilde{u}_k$ is the k*th* estimated information bit. When the CE becomes less than the threshold of $T(i) = 10^{-3} \cdot M(1)$, the iterations are stalled. By contrast, the authors of [14] used the average entropy for representing $M(i)$. The average entropy per bit for a packet having a length of N may be calculated as

$$M(i) = -\frac{1}{N}\sum_{k=1}^{N}[p(\tilde{u}_k = 0|r)\log p(\tilde{u}_k = 0|r) + \\ p(\tilde{u}_k = 1|r)\log p(\tilde{u}_k = 1|r)], \tag{3}$$

where $p(\tilde{u}_k = 0|r)$ represents the *a posteriori* probability of the estimated bit $\tilde{u}_k = 0$ based on the received symbol r. Bearing in mind that the *a posteriori* LLR output of the BCJR decoder represents the logarithmic form of the *a posteriori* probability, Equation 3 may also be expressed in terms of the *a posteriori* LLRs. Furthermore, Chen et al. [14] also suggested specific threshold values for different SNR ranges.

The following contributions further developed the ES philosophy based on the polarity changes of the LLRs [15], [16] during iterative decoding, since the hard decision output only depends on the polarity of the LLRs. The techniques proposed in [15] observed the relative frequency of polarity changes at the i*th* iteration and introduced the Sign Change Ratio (SCR) as another stopping criterion. The MI improvement $M(i)$ was estimated by the number of polarity changes in L_{e_2} between two consecutive iterations over the packet length N and the corresponding thresholds $T(i)$ set for the SCR in [15] were 0.005 to 0.03. Furthermore, the authors of [16] extended the SCR stopping criterion to the Sign Difference Ratio (SDR), which calculates the ratio of the sign differences between the *a priori* and the extrinsic LLRs. In [17], [18], the absolute value of the LLR was invoked as a metric used for ES. Explicitly, the minimum absolute value of the LLRs was suggested for characterizing the $M(i)$ in [17], while the mean of the absolute LLR values was advocated in [18].

3) Hard Decision Aided ES

Hard Decision Aided (HDA) ES was proposed in [15], where hard decisions were invoked at each iteration for the *a posteriori* LLRs $L_2^{(i-1)}(\tilde{\mathbf{u}})$ and $L_2^{(i)}(\tilde{\mathbf{u}})$ was acquired from the second BCJR decoder. If the estimated bits of both HDA decoding operations agreed with each other, iterative decoding was concluded. The HAD may save more iterations at low to medium SNRs, but may increase the complexity at high SNRs. Therefore, Taffin [19] derived a generalized HDA algorithm, which calculates the ratio of the number of different HDA bits between two consecutive iterations over the entire packet length N. Furthermore, different thresholds were derived for high SNRs and low/medium SNRs. An Improved HDA (IHDA) was then proposed in [20], which achieved a similar performance to the original HDA of [15] without the extra storage requirement of HDA, since it simply compared the agreement between the hard decision versions of the *a priori* LLRs and of the *a posteriori* LLRs gleaned from the second BCJR decoder at the i*th* iteration.

4) Cyclic Redundancy Check Check

Cyclic Redundancy Check (CRC) aided Turbo-CRC decoding has become a widely used method of controlling the activation/deactivation of iterative decoding [21]–[23], which combines turbo codes with CRC codes for the sake of detecting, whether there are any residual errors after turbo decoding. According to this method, each turbo encoded packet will be passed to a CRC encoder for appending a number of check bits. At the receiver, the classic CRC check will be applied to the estimated bits, which are hard-decided on the basis of the *a posteriori* LLRs gleaned from the second BCJR decoder [21] or in fact from both BCJR decoders [22]. If the CRC check indicates a decoding success, the iterative decoding will be ceased. This CRC aided ES technique is capable of indicating perfect decoding. However, it fails to reduce the number of decoding iterations for transmissions over hostile channels.

5) The input-output consistency check.

The decoding process conceived for the *a posteriori* LLRs of the parity bits is similar to that of the original information bits, although it is not necessary for the BCJR decoder to output them at the end of the decoding process. Based on this fact, the authors of [24], [25] proposed an ES strategy referred to as the input-output consistency check, where the BCJR decoder outputs the *a posteriori* LLRs of both the information bits as well as of the parity bits. After each iteration, the estimated information bits and parity bits may be obtained by subjecting these *a posteriori* LLRs to hard decision. The estimated information bits will be encoded by the same convolutional encoder as that employed at the transmitter. Then, the resultant output parity bits will be compared to the parity bits estimated from the *a posteriori* LLRs. If they are identical, the iterative decoding stops; otherwise, it will continue until the maximum number of iterations has been exhausted.

6) Bit-based ES

Recently, it was found in [26] that the convergence speed of each bit during the iterative decoding is different. More explicitly, it was demonstrated that a substantial fraction of *a posteriori* LLRs of the information bits become stable after a few iterations. Hence, we refer to them as the 'converged' bits. The remaining modest number of 'un-converged' bits require more decoding iterations for achieving a reliable estimation. Using this property, the bit-based ES strategies proposed in [26]–[28] curtail the iterative decoding for the 'converged bits', when a certain MI threshold has been reached by them. More explicitly, the iterative decoding will only continue for the specific bits having a low reliability, which may be quantified in terms of their LLR magnitudes [26]. The decoding complexity imposed is therefore reduced by processing a reduced number of bits in the second stage. In order to improve the achievable decoding performance, the authors of [27], [28] grouped the non-converged bits along with their adjacent converged bits into a set and invoked partial iterative decoding for these small sets. The benefit of this technique is that the high-reliability converged bits assist in improving the reliability of the non-converged bits.

B. Motivation and Organization

Based on the Log-MAP algorithm and on the above-mentioned ES strategies, turbo codes have found applications in diverse scenarios. Their application may also be combined with other techniques, such as for example, Hybrid ARQ (HARQ) techniques, or with Decode and Forward (DF) relaying. In these applications, the complexity may become high, since the iterative decoding process may be activated multiple times. Specifically, turbo coded HARQ, which will be detailed in Section III, performs turbo decoding after each reception of the (re)transmitted packets. Likewise, a turbo coded relay-aided network may activate turbo decoding both at the relay and at the destination. The packets received both from the source and from the relay also have to be combined with the aid of soft extrinsic information exchange, which is reminiscent of turbo decoding at the destination. However, the iterative detection complexity imposed may be further increased, when invoking multiple relays.

From a theoretical perspective, both HARQ and relay-aided networks rely on the incremental redundancy extracted from the retransmissions or from the relays. However, the ES strategies reviewed in this paper never considered this additional information increment. Yet, it is plausible that initially it may be unnecessary to activate any iterations during the HARQ process, since the decoding convergence may be substantially accelerated by the forthcoming incremental information, hence resulting in a reduced number of iterations. Furthermore, the previously proposed complexity reduction methods never considered the influence of the specific component convolutional codes, especially the memory length m of their forward and feedback Generator Polynomials (GPs), which determines the number of states in the trellis. As mentioned above, the MAP algorithm will estimate a specific bit by evaluating all the probabilities of all possible state transitions for that particular bit in the trellis. Since the number of trellis states increases exponentially with the memory length of the GPs, the complexity is dramatically reduced, if m is a small value, whist maintaining an unimpaired performance.

In this paper, we set out to reduce the complexity of turbo coded applications by explicitly considering both the incremental information gleaned, as well as the influence of the component codes' memory length. HARQ will be considered as our design example for discussing the complexity reduction achieved with the aid of information combination. Nevertheless, the proposed methods are general, hence they are also applicable to turbo coded relay networks. More explicitly, our novel contributions may be formulated as follows,

- Multiple Component Turbo Codes (MCTCs) are designed, which have the shortest possible memory length GPs of $(2, 3)_o$, where the underscore o indicates the octal representation.
- Based on these MCTCs, we conceived an ES strategy, which curtails iterations, when ever the MI increment becomes less than a specific threshold, which may be determined by an offline training process.
- A so-called Deferred Iteration (DI) method was proposed for delaying the commencement of iterations, until sufficient information is deemed to be received.

The paper is organized as follows. In Section II, both HARQ arrangements and the techniques of information combination are reviewed. The existing turbo coded HARQ schemes are discussed in Section III, where an equation is conceived for calculating the complexity of turbo coded HARQ. Furthermore, in Section IV, we demonstrate that MCTCs exhibit an attractive performance even when the GPs having the lowest possible memory of $m = 1$ represented by $(2, 3)_o$ are adopted. Therefore, MCTCs may be beneficially invoked for constructing low-complexity turbo-coded HARQ schemes. Furthermore, in Section V two methods are proposed for finding an optimal number of iterations for a specific normalized target throughput and Packet Loss Ratio (PLR) performance. More specifically, the ES strategy to be detailed in Section V-A curtails the iterative decoding process after each transmission attempt, when the MI improvement of additional iterations becomes low. By contrast, in Section V-B, we introduce the DI method, which delays the commencement of iterative decoding, until the total MI is deemed to be sufficiently high for guaranteeing successful decoding with a high probability.

II. HARQ Introduction

When the receiver detects that a packet has been correctly received, it will feed back a positive ACKnowledgement (ACK) message to the transmitter. Otherwise, it may send back a negative ACK or wait for the transmitter to time out. The transmitter will continue retransmitting the specific information packet, until it receives the ACK message or a maximum retry limit is reached. In the early ARQ designs, each retransmitted packet was independently processed. Therefore, a packet may become discarded, regardless of how many retransmissions are allowed, if all transmissions encounter hostile channel conditions. However, in more sophisticated schemes, all corrupted replicas may be combined with the aid of soft decisions in order to successfully decode the original information bits. This is philosophically similar to the action of repetition codes [29]. In other words, this forms a naive example of the combination of FEC and ARQ, which is referred to as HARQ.

HARQ schemes overcome the disadvantages of FEC and ARQ. More specifically, FEC may waste bandwidth by unnecessarily transmitting parity bits for protecting the transmissions over a channel having a good condition, while ARQ degrades the throughput drastically for transmissions over a hostile channel. In general, HARQ schemes combine ARQ with FEC techniques, as illustrated in Figure 1. It may be observed that each information packet will be accompanied by a number of check bits of an error detection code, for example a CRC scheme. These check bits are denoted by ‘D’ in Figure 1. Then, the resultant packet is passed to the FEC encoder. The encoded packet represented by ‘Q’ in Figure 1 may have different forms depending on the specific type of the HARQ scheme. For example, it may be the input packet accompanied by a number of error correction parity bits. In this case, the packet length is increased. Alternatively, the packet may only contain parity bits, while having either the same or a potentially reduced length in comparison to the original packet, if it was prepared for retransmissions, which may hence be referred to as incremental redundancy.

At the receiver, the FEC decoder performs decoding based on all the received bits, where different HARQ schemes have different packet combining strategies during this decoding operation. The decoding may only process the currently received packet $\tilde{Q}$, as seen in Figure 1. However, combining the previous corrupted replicas with the current one for the sake of decoding them together becomes a better method. If decoding errors still persist in the estimated message, the current replica is discarded, triggering a retransmission by sending a Negative ACKnowledgement (NACK) or by simply waiting for the transmitter to time out. This process is repeated, until a retry limit is reached or the packet is correctly received.

Based on the development of FEC techniques, HARQ schemes have been classified into Type-I, Type-II and Type-III categories, which overcame the shortcomings of the previous versions. Furthermore, the achievable throughput also benefits from combining the retransmitted packets, which evolved from using the naive technique of no combining at all in the early days, to combining parts of the retransmitted packets and finally to combining all received replicas.

A. Three Types of HARQ

As discussed before, HARQ schemes combine the advantages of ARQ and FEC arrangements for the sake of achieving a high throughput. However, for the earliest concept of HARQ proposed and analyzed in 1970 [30], the authors of [31] claimed that separate ARQ or FEC may attain a better performance than HARQ when considering different SNR regions. The straightforward combination of ARQ and FEC is referred to as Type-I HARQ. More explicitly, the FEC-encoded packet ‘Q’ seen in Figure 1 is the original packet, which is complemented by a number of parity bits. This packet will be (re)transmitted for all (re)transmissions. At the receiver, all corrupted packets will be thrown away. Each FEC decoding action is performed solely for a single encoded packet.

Figure 1 of [32] was reproduced here for completeness as Figure 2, which shows the achievable throughput of a single retransmission attempt, which we refer to as conventional ARQ. This ARQ scheme is also compared to both Type-I HARQ as well as to Type-II HARQ in Figure 2. Observe that Type-I HARQ is capable of improving the attainable throughput of conventional ARQ, apart from Bit Error Ratios (BER) below about 10^{-4}. This is because the FEC scheme’s coding gain results in a reduced number of retransmissions required for the successful delivery of a packet. However, if the channel conditions are sufficiently good, the resultant reduced number of errors may not require any FEC parity bits. In this situation, the resultant normalized throughput is reduced by the unnecessary parity bits. By contrast, if the channel conditions are hostile, the FEC code may not be capable of correcting all errors for each transmission. Hence, the normalized throughput tends to zero, similar to that of conventional ARQ. Most of the early HARQ schemes designed during the 1970s and early 1980s [30], [33]–[36] belong to the family of Type-I HARQs.

Type-II HARQ was proposed by Lin and Yu [32] in 1982. In order to eliminate the throughput degradation of Type-I HARQ for transmission over a benign channel, the first

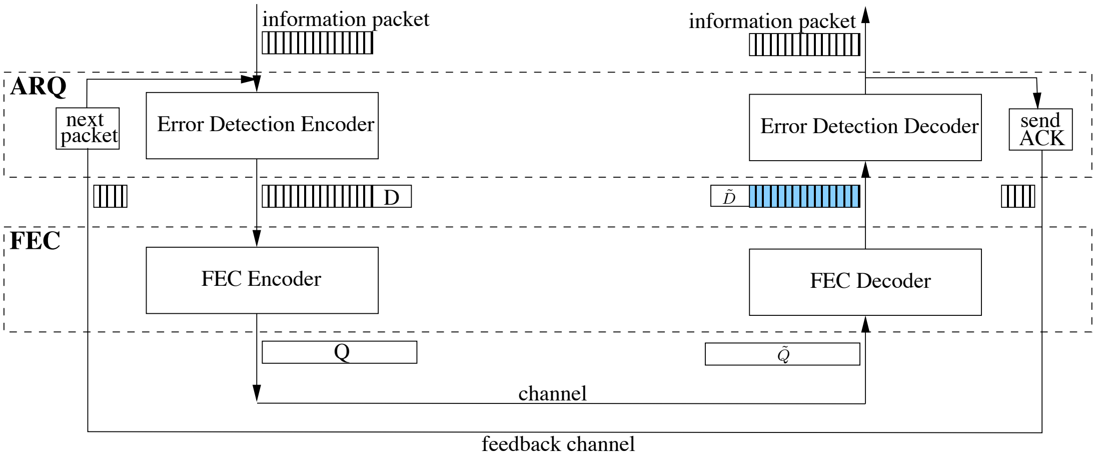


Fig. 1. The general illustration of HARQ, where the Error Detection Encoder (EDE) and FEC encoder generate the codewords Q to be transmitted. The received codewords $\tilde{Q}$ are FEC decoded and in the presence of detection errors further redundancy is requested from the transmitter.

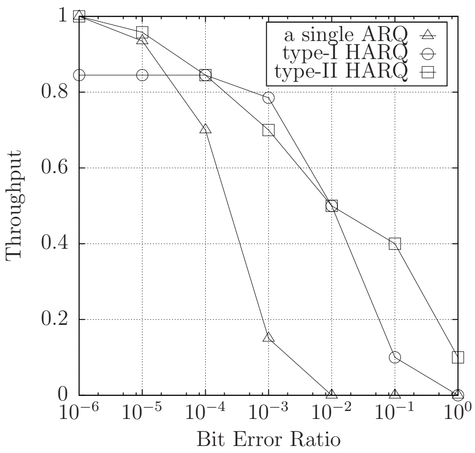


Fig. 2. The normalized throughput of a single ARQ, type-I HARQ and type-II HARQ ©Lin [32].

transmission of Type-II HARQ relies on the original packet without adding any redundancy. More specifically, the first transmitted encoded packet 'Q' of Figure 1 is the original packet, dispensing with FEC. If the channel conditions are good, this uncoded packet may indeed be received correctly. Otherwise, the second transmission will be requested. Only parity bits are transmitted during the second transmission. Then at the receiver, FEC decoding is performed based on the combination of both these new parity bits and of the corrupted original packet of the prior transmission attempt. If the FEC decoder fails to correct all errors, the uncoded original packet will be transmitted during the third transmission attempt. At this time, it will be combined with the second transmitted parity bits for FEC decoding. If errors still exist, the same sequence of parity bits is transmitted again during the fourth transmission. This process is repeated until the packet is correctly received or the retry limit is reached.

Type-II HARQ proposed by Lin and Yu of [32] has the highest throughput for transmission over a benign channel among the conventional ARQ, Type-I and Type-II HARQ schemes. However, the normalized throughput becomes lower than that of Type-I HARQ in the middle of the BER range seen in Figure 1 of [32]. This is because it has to combine two transmissions for the sake of commencing FEC decoding, while Type-I HARQ carries parity bits by each transmission. However, this throughput decrease of Type-II HARQ may be overcome by carefully designing the coding rate of FEC codes for each transmission. For example, the authors of [37] and [38] employed Rate-Compatible Punctured Convolutional (RCPC) codes [39] for the sake of adapting the coding rate for Type-II HARQ.

Like Type-II HARQ, Type-III HARQ uses different redundant information during each transmission attempt, but each of them is self-decodable. In more detail, each transmitted packet 'Q' of Figure 1 may contain an increasing amount of redundant information, which is provided by the FEC encoder. The original information packet may be recovered by decoding each packet of 'Q', when the channel is benign, before it is combined with all previously received replicas. This measure is capable of enhancing the probability of successful delivery for the pure systematic bits in Type-II HARQ schemes, as well as of decreasing the decoding complexity for the combination of all retransmitted packets. The authors of [40]–[43] characterized the attainable performance of Type-III HARQ schemes, which exhibited an improved performance compared to the Type-I and Type-II HARQ schemes.

B. *Combining Transmissions*

During the evolution of HARQ schemes, minimizing the required number of retransmissions has received a significant research attention, because unnecessary retransmissions result in reducing the effective throughput. Researchers have conceived a large number of strategies for improving the throughput efficiency of HARQ schemes, such as adapting the coding rate of FEC codes, or using soft information to detect errors for the sake of eliminating the CRC overhead, and so on. We summarize the literature addressing the normalized throughput of HARQ schemes in Table I. Furthermore, the most efficient packet combining techniques will be detailed in this section.

A typical approach has been that of combining the various corrupted retransmitted components in order to provide a more

TABLE I
MAJOR CONTRIBUTIONS ADDRESSING THE THROUGHPUT IMPROVEMENT OF HARQ SCHEMES.

Author(s)	Contribution
Krishna *et al.* 1987 [44]	generalized the Type-II HARQ schemes based on a new class of linear codes, whose encoder/decoder configuration does not change as the length of the code is varied.
Morgera *et al.* 1989 [45]	employed soft-decision decoding in generalized HARQ schemes. Simulation results had shown the improvement in throughput efficiency.
Kousa *et al.* 1991 [46]	employed Hamming codes in a cascaded manner for adaptive HARQ schemes, where the coding rate could be adaptively matched to the prevailing channel condition.
Rice 1994 [47]	compared and analyzed the throughput performance of two HARQ schemes using Viterbi soft decoding, one of which still depended on a high rate CRC to detect errors. The other rejected the packets having a low reliability during the Viterbi decoding process.
Coulton *et al.* 2000 [48]	proposed a HARQ scheme for soft information based turbo codes, which employed the mean value of soft output to reject or accept the decoded packets, rather than the CRC. Furthermore, four retransmission options were implemented and compared, which retransmitted different parts of the encoded packet.
Buckley *et al.* 2000 [49]	proposed a method of predicting the presence of errors after turbo decoding by observing the cross entropy of the component decoders. This may improve the reliability and throughput despite reducing average decoding complexity.
Choi *et al.* 2001 [50]	compared three adaptive HARQ schemes based on Reed-Solomon (RS) codes. The RS code rate and frame length were adapted to the estimated channel conditions in different ways for three schemes.
Harvey 2004 [51]	exploited the characteristics of the Viterbi algorithm to reduce the computational and memory requirements in adaptive HARQ. At the receiver, the memory and processing power could be allocated dynamically for improving the transmission reliability.
Cao *et al.* 2004 [52]	proposed a retransmission strategy based on segment-selective repetition for turbo coded HARQ schemes. Only the most severely corrupted segments identified by the lowest LLR values would be retransmitted. As a result, the throughput was significantly improved.
Holland *et al.* 2005 [53]	proposed a soft combining method for HARQ schemes based on turbo codes. The post-decoding BER was significantly decreased by accumulating the *a priori* values from the channel output for each additional retransmission.
Zhang *et al.* 2006 [54]	proposed an adaptive HARQ scheme for multicast scenarios, where the error-control scheme was dynamically adapted to both the packet-loss level and Quality of Service (QoS) requirements.
Oteng-Amoako *et al.* 2006 [55]	presented HARQ schemes using asymmetric turbo codes, where the component codes have different structures or polynomial orders.
Chiti *et al.* 2007 [56]	introduced a modified HARQ scheme based on turbo codes. The performance had been noticeably improved by applying soft-combining techniques for amalgamating the *a priori* input and the channel input of the turbo decoder.
Heo *et al.* 2008 [57]	combined Raptor codes with HARQ schemes, and proposed a low complexity decoding algorithm, which efficiently combined the symbols during the multiple transmissions.
Mielczarek *et al.* 2008 [58]	introduced two types of NACK messages for improving the throughput of HARQ schemes using a BCJR decoder. One of the NACK type was conceived for conventional retransmissions, while the other specified the actual positions of the subblocks for which the additional parity bits will be retransmitted.
Fricke *et al.* 2009 [59]	proposed reliability-based HARQ schemes, whose retransmission criteria used the maximum tolerable Bit Error Probability (BEP) or Word Error Probability (WEP), instead of using the conventional CRC.

reliable decision for the original bits. Two main combining strategies have been proposed, namely Chase combining [60] and the transmission of Incremental Redundancy (IR) [61]. Chase combining achieves a diversity gain by beneficially combining the identical data replicas conveyed during different retransmissions. By contrast, IR conveys different redundant information during each transmission attempt, which may be combined and reconstructed by a single FEC decoder at the receiver. More recently, the employment of incremental redundancy has found applications in cooperative networks [62], [63].

Soft combining [64], [53] techniques conceived for Chase combining and IR may be beneficially invoked for iterative soft-decision-based FEC decoders, such as turbo codes, where soft information represented by the LLRs [1] is exchanged between the constituent BCJR decoders [5]. Soft decision aided Chase combining simply adds the LLRs of several packet replicas, which contain the same bit sequences received from different transmissions. Actually, the iterative decoding process of turbo codes relies on IR-aided soft combining, since each transmitted IR packet will provide additional channel information or *a priori* information for the iterative decoding ensuing after each transmission. For example, the High Speed Downlink Packet Access (HSDPA) protocol [65] uses a punctured $R = \frac{1}{3}$-rate turbo code as the basis of its HARQ scheme. Here, whenever a retransmission is received, the corresponding LLRs will be added to those that have either been received from previous transmissions, or have been used for providing soft information for bits that were punctured during previous transmissions.

III. TURBO CODED HARQ SCHEMES

As mentioned above, turbo codes [4], [6], [9] are characterized by an iterative exchange of increasingly reliable soft information between the constituent BCJR [5] decoders, which are concatenated in parallel and separated by an interleaver. Owing to their near-capacity performance, turbo codes can be successfully combined with HARQ schemes [66]–[68], in order to achieve a high normalized throughput. In these turbo HARQ schemes, the transmitter continually transmits turbo-encoded IR or repeated packets to the receiver, where BCJR

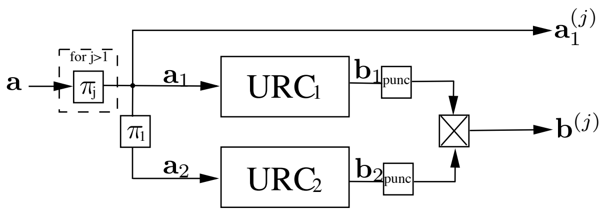


Fig. 3. The encoder structure of Narayanan's turbo coded HARQ scheme, where URC represents a Unity Rate Code [69].

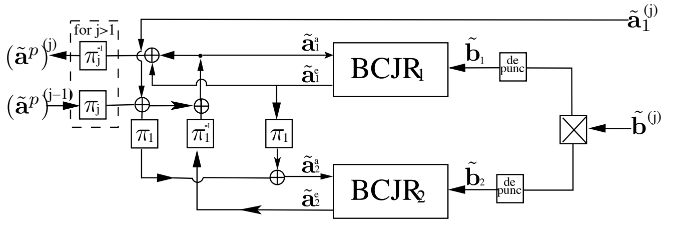


Fig. 4. The decoder structure of Narayanan's turbo coded HARQ scheme.

decoding operations may be performed iteratively following the reception of each transmission.

In the early turbo coded HARQ scheme proposed by Narayanan and Stuber [66], a systematic rate-$\frac{1}{2}$ turbo encoded packet generated from two differently interleaved versions of a given information packet is transmitted during every repeated transmission. Figure 3 illustrates its encoder structure at the jth transmission. For each retransmission, the information bits are interleaved separately and differently by π_j $(j > 1)$, as seen in Figure 3, before they are turbo encoded. When relying on $\frac{1}{2}$-rate encoding, each transmitted packet includes $\mathbf{a}_1^j$ and $\mathbf{b}^j$ generated by puncturing two URCs. At the receiver, the classic twin-component turbo decoder shown in Figure 4 performs iterative decoding for the currently received packet. If an error is found in the decoded packet, the output *a posteriori* LLRs $(\tilde{\mathbf{a}}^p)^{(j)}$ will act as the *a priori* information for the next transmission. This combines the information gleaned from previous transmissions and acccelarates the convergence of turbo decoding in later transmissions. However, the PLR and throughput performance of this turbo HARQ scheme is not so attractive, since the *a priori* LLRs 'inherited' from the previous $\frac{1}{2}$-rate turbo decoding may not provide reliable assistance for the current turbo decoding.

In a recent paper [67], Souza *et al.* proposed a HARQ scheme that amalgamated Chase combining with IR in a systematic TCTC. Figures 5 and 6 have shown the encoder and decoder structures of Souza's HARQ scheme. In more detail, the transmitter of Souza's HARQ sequentially transmits the systematic bit sequence $\mathbf{a}$ and the parity bit sequences $\mathbf{b}_1$ and $\mathbf{b}_2$, which are encoded by two URC encoders having octally represented memory-3 GPs of $(17, 15)_o$. More specifically, the transmitted sequences are equivalent to $(\mathbf{a}, \mathbf{b}_1), \mathbf{b}_2, \mathbf{a}, \mathbf{b}_1, \mathbf{b}_2, \cdots$ respectively for the first, second, third transmissions and so on, where $(\mathbf{a}, \mathbf{b}_1)$ are transmitted together during the first transmission, as seen in Figure 5. The receiver activates iterative decoding by exchanging extrinsic information between the two parallel concatenated BCJR decoders seen in Figure 6 after the second IR transmission. For the first transmission, BCJR decoding is performed only once, since only a single parity bit sequence, namely $\mathbf{b}_1$ is available. From the third IR transmission onwards, the repeated packet replica's LLRs are directly added to those gleaned from the previous transmissions, as seen in Figure 6. Souza's scheme is the best known arrangement at the time of writing, since it was demonstrated that Souza's scheme is capable of outperforming the Narayanan - Stueber scheme of [66]. We will use Souza's scheme as our benchmarker later.

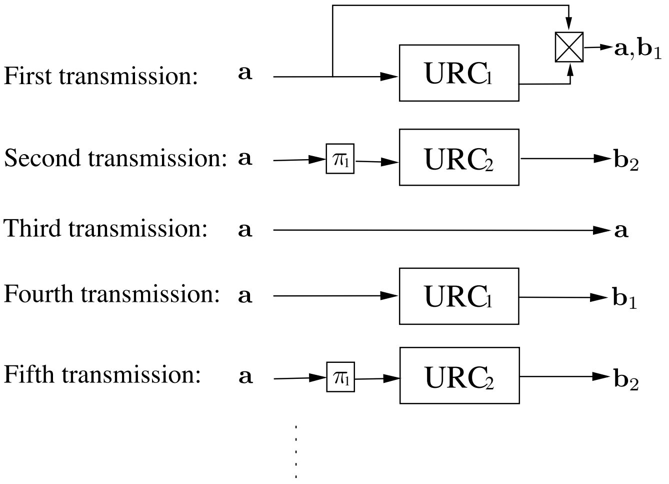


Fig. 5. The encoder structure of Souza's turbo coded HARQ scheme.

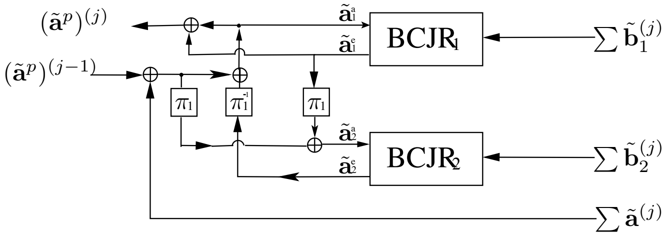

$(\tilde{\mathbf{a}}^p)^{(j)}$
$(\tilde{\mathbf{a}}^p)^{(j-1)}$
$\tilde{\mathbf{a}}_1^a$
$\tilde{\mathbf{a}}_1^e$
BCJR1
$\sum \tilde{\mathbf{b}}_1^{(j)}$
π_1
π_1^{-1}
π_1
$\tilde{\mathbf{a}}_2^a$
$\tilde{\mathbf{a}}_2^e$
BCJR2
$\sum \tilde{\mathbf{b}}_2^{(j)}$
$\sum \tilde{\mathbf{a}}^{(j)}$

Fig. 6. The decoder structure of Souza's turbo coded HARQ scheme.

Figures 7 and 8 illustrate the encoder and decoder structures of a LTE HARQ scheme, which also adopts a TCTC having memory-3 polynomials of $(15, 13)_o$. The LTE standard specifies a particular interleaver, and a so-called 'rate matching' operation is invoked for selecting specific bits for transmission, rather than transmitting all bits [70]. More explicitly, the standard defines its own interleaver i.e. π_1 in figure 7 for employment between two parallel concatenated turbo encoders/decoders for a range of specific packet lengths. Furthermore, the systematic bit sequence $\mathbf{a}$ and the two parity bit sequences $\mathbf{b}_1$, $\mathbf{b}_2$ are interleaved again, according to the standard's so-called sub-block interleavers, namely π_2, π_3 and π_4 in Figure 7. The interleaved systematic bits are entered into a circular buffer, where the terminology is justified, since the starting point of each transmission will revert to the beginning of the buffer, when it reaches the end of the buffer. The interleaved parity bits are entered at the back of the circular buffer, where the odd positions are from $\mathbf{b}_1$ and the even positions are from $\mathbf{b}_2$. Next, a sequence of transmitted bits is selected from a specific starting point of this circular buffer

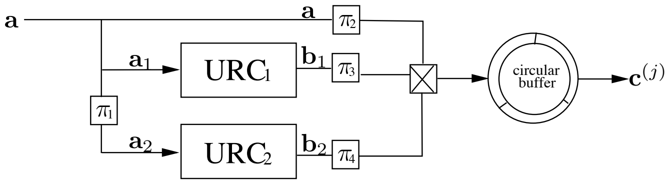


Fig. 7. The encoder structure of LTE turbo coded HARQ scheme.

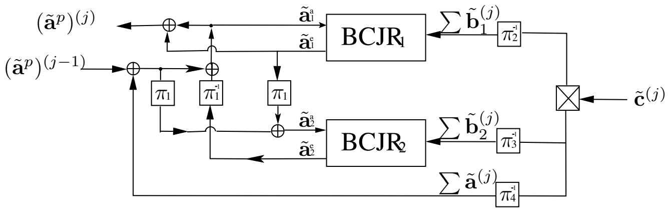


Fig. 8. The decoder structure of LTE turbo coded HARQ scheme.

seen in Figure 7, where the number of bits is determined by the LTE standard. This starting point advances along the circular buffer, based on an LTE-standard-specific equation, which is a function of the transmission frame index j. As a result, turbo decoding can be activated right after the first frame's transmission, since it contains some of the systematic bits as well as some of the two parity bit sequences. It may be observed in Figure 8 that the repeated LLRs are also Chase combined with the corresponding previously received replicas at the receiver.

The number of iterations following each reception is closely related to the complexity of turbo HARQ schemes. Generally, there is a trade-off between the achievable throughput and the complexity imposed by the turbo HARQ schemes. Naturally, the complexity is increased when more than necessary BCJR iterations are performed during the iterative decoding process following the reception of each transmission. For example, the above turbo HARQ schemes [67], [71] performed a sufficiently high number of BCJR decoder executions following each and every IR transmission in order to ensure that iterative decoding convergence had been achieved. In this way, they minimized the number of IR transmissions required, hence maximizing the throughput, albeit at the cost of imposing an increased complexity. On the other hand, the attainable throughput degrades, when sufficient IR contributions have been received for facilitating error-free decoding, but insufficient BCJR iterations have been performed.

The complexity of a tubo decoder may be quantified in terms of the total number of trellis states per bit [1], which is expressed as $Complexity = 2^m \cdot K$, where 'm' is the number of memory elements employed in the convolutional encoders' generator polynomial. Hence, 2^m denotes the number of states in the corresponding trellis diagram, and 'K' is the total number of BCJR operations performed during the iterative decoding process. Since the complexity exponentially increases with 'm', using the generator polynomial of $(2, 3)_o$ having the lowest possible memory length of '$m = 1$' is desirable.

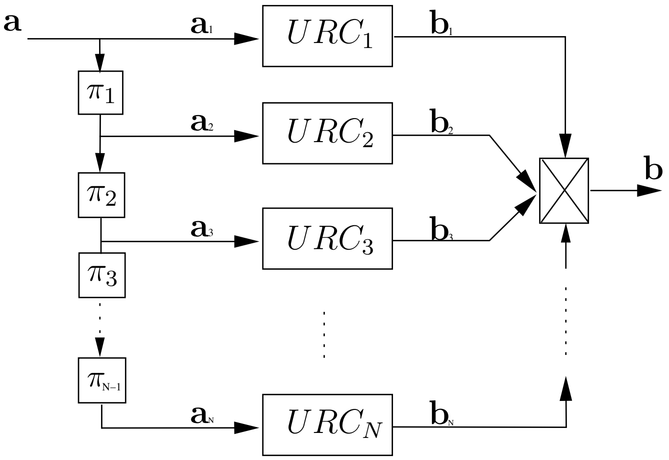


Fig. 9. The encoder structure of a MCTC using N URC components.

IV. The Foundations of Low-Complexity HARQ: Multiple Component Turbo Codes

MCTCs are constituted by the parallel concatenation of more than two component codes. They may outperform TCTCs, despite relying on low-complexity low-memory GPs, and therefore become the best choice for low-complexity HARQ schemes. In this section, we will analyze MCTCs and characterize their performance. Our system model employs URCs as the component codes of the proposed MCTCs. Figure 9 shows the encoder's structure, where the source information is $\mathbf{a}$ and its interleaved copies $\{\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3, \cdots, \mathbf{a}_N\}$ are entered into the N encoders. The multiplexer seen at the output of the encoder in Figure 9 assembles the encoded bits $\mathbf{b}$, where the resultant bit stream has N times the original sequence length. Hence, the overall coding rate becomes $\frac{1}{N}$.

Figure 10 shows the corresponding N BCJR decoders' structure, where each BCJR decoder has two inputs, namely the *a priori* LLRs $\tilde{\mathbf{a}}_i^a$ combined from all other decoders' extrinsic LLRs and the channel's output information $\tilde{\mathbf{b}}_i$. Then each BCJR decoder outputs its own extrinsic LLRs $\tilde{\mathbf{a}}_i^e$. The decoding process proceeds as follows: when a specific BCJR decoder takes control, the decoding iterations by exchange extrinsic information among all N BCJR decoders, until the point of convergence is reached or the affordable number of iterations was exhausted. Then, the recovered bits are decided upon, based on their *a posteriori* LLRs $\tilde{\mathbf{a}}_1^p$.

A. A Semi-Analytical Tool: EXIT charts

The performance of turbo codes has been classically evaluated in terms of their BER versus SNR characteristics. The BER curves of turbo codes may be divided into three regions: the high-BER region at low SNRs; the rapid BER reduction region at medium SNRs, which may also be referred to as the 'turbo-cliff' region; and the lowest-BER region at high SNRs, which is also referred to as the error floor region. These three regions correspond to different decoding convergence scenarios of turbo codes. Ten Brink [72] introduced the concept of EXIT charts for analyzing the convergence behavior of iterative decoding, which is exemplified at the right of Figure 12. Since then, EXIT charts became widely used as

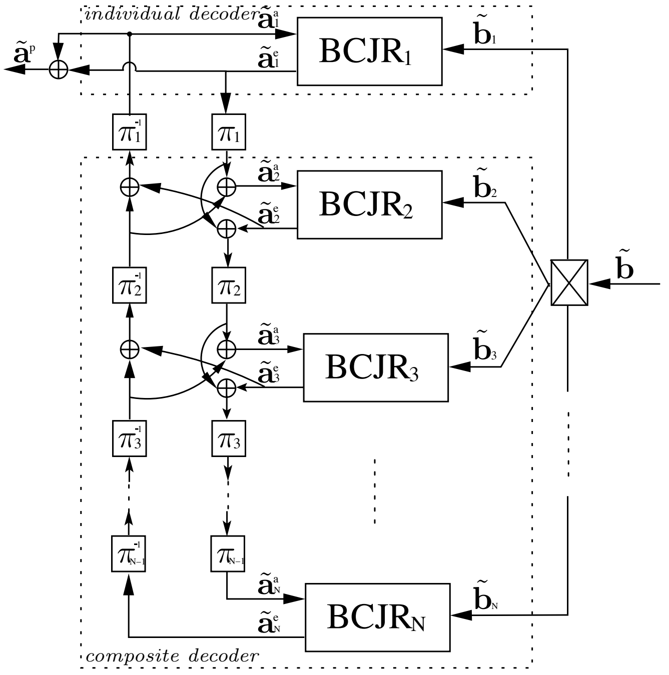


Fig. 10. The decoder structure of a MCTC using N BCJR decoders. Its logical partitioning is showed in the dashed boxes.

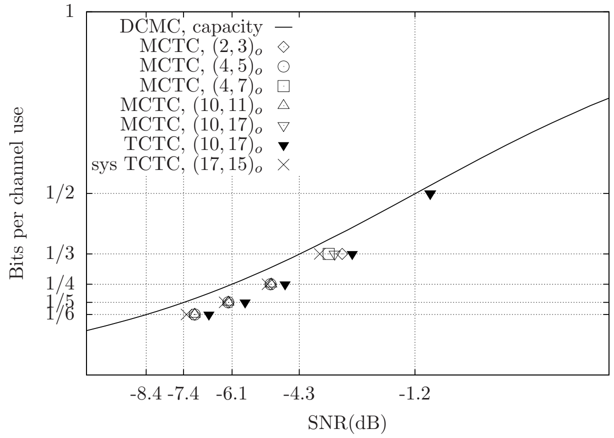


Fig. 11. Throughput versus open-tunnel-SNR threshold for both MCTCs, as well as systematic and non-systematic TCTCs using various generator polynomials, when communicating over a Binary Phase-Shift Keying (BPSK) modulated uncorrelated Rayleigh fading channel.

an effective semi-analytical tool of designing turbo codes and other iterative detection techniques.

A classic EXIT chart describes the extrinsic MI exchange between two parallel concatenated BCJR decoders. The extrinsic MI provided by a BCJR decoder is denoted by $I(\tilde{\mathbf{a}}_i^e)$, which may be approximately calculated from the extrinsic LLRs $\tilde{\mathbf{a}}_i^e$ using the following equation introduced in [73]:

$$I(\tilde{\mathbf{x}}) \approx 1 - \frac{1}{L}\sum_{j=1}^{L} H_{\mathrm{b}}\left(\frac{e^{+|\tilde{x}_j|/2}}{e^{+|\tilde{x}_j|/2} + e^{-|\tilde{x}_j|/2}}\right) , \quad (4)$$

where H_{b} represents the binary entropy function, $\tilde{\mathbf{x}}$ is a general notation for a vector of LLRs, and L denotes the length of the vector $\tilde{\mathbf{x}}$. There are two curves in a classic EXIT chart, each reflecting a BCJR decoder's EXIT function F_{exit}, which may be expressed as:

$$I(\tilde{\mathbf{a}}_i^e) = F_{exit}\left(I(\tilde{\mathbf{a}}_i^a)\right) \ . \quad (5)$$

where the independent variable $I(\tilde{\mathbf{a}}_i^a)$ denotes the *a priori* MI input of the $BCJR_i$ decoder, which may also be calculated by replacing $\tilde{\mathbf{x}}$ with $\tilde{\mathbf{a}}_i^a$ in Equation 4. Since the output extrinsic MI $I(\tilde{\mathbf{a}}_1^e)$ of the $BCJR_1$ decoder will be passed to the $BCJR_2$ decoder as the *a priori* MI input during the iterative decoding process, classic EXIT charts use the horizontal axis for representing the independent variable of the $BCJR_1$ decoder's EXIT function, while the vertical axis represents that of the $BCJR_2$ decoder's. More explicitly, the independent variable of the $BCJR_1$ decoder's EXIT function - namely $I(\tilde{\mathbf{a}}_1^a)$ - is represented along the X-axis of EXIT charts, while the independent variable of the $BCJR_2$ decoder's - namely $I(\tilde{\mathbf{a}}_2^a)$ - is scaled along the Y-axis of EXIT charts. Hence, the Monte-Carlo simulation based decoding trajectory along the tunnel between two EXIT curves shows the MI exchange process and demonstrates whether the stair-case-shaped decoding trajectory reaches the point $(1,1)$ of perfect convergence to an infinitesimally low BER, indicating whether potentially successful decoding can or cannot be achieved.

In order to exploit the relationship between EXIT charts and the BER, we consider the example of non-systematic turbo codes having the octal generator polynomials of $(8,F)_o$ and transmit a sufficiently long packet over an uncorrelated Rayleigh fading channel. As seen in Figure 12, the EXIT chart at the SNR of 0dB has an open tunnel and the arrows show the relationship between the BER and EXIT charts at this SNR. It may be observed that the BER curve is gradually shifted to the left, while the stair-case-shaped decoding trajectory is approaching the point of perfect convergence to a vanishing low BER at $(1,1)$. Since each step along the decoding trajectory indicates a single BCJR operation, the BER becomes about 2×10^{-1} after the 3*rd* BCJR operation at 0dB. When the decoding continues after 6 BCJR operations, the BER descends to 10^{-1}, as seen in Figure 12; while after 12 operations, it falls down to slightly higher than 2×10^{-2}. Finally, when the decoding trajectory reaches the point of $(1,1)$, the number of errors becomes close to 0.

B. Achievable Performance

The MCTC decoder of Figure 10 may be partitioned into two logical parts, which are are surrounded by the dashed rectangles. One of them is constituted by an individual BCJR decoder, while the other so-called composite decoder consists of the remaining $(N-1)$ components. EXIT charts [1] [72] may be applied for visualizing the extrinsic information exchange between these two logical partitions. Then, it becomes possible to determine the 'open tunnel SNR threshold', which is defined as the minimum SNR for which the EXIT chart has an open tunnel.

The 'open tunnel SNR threshold' may be used for characterizing the capability of channel codes to approach the capacity. In order to compare this capability of MCTCs and TCTCs relying on different GPs, Figure 11 illustrates the open-tunnel-SNR thresholds for both MCTCs, as well as for

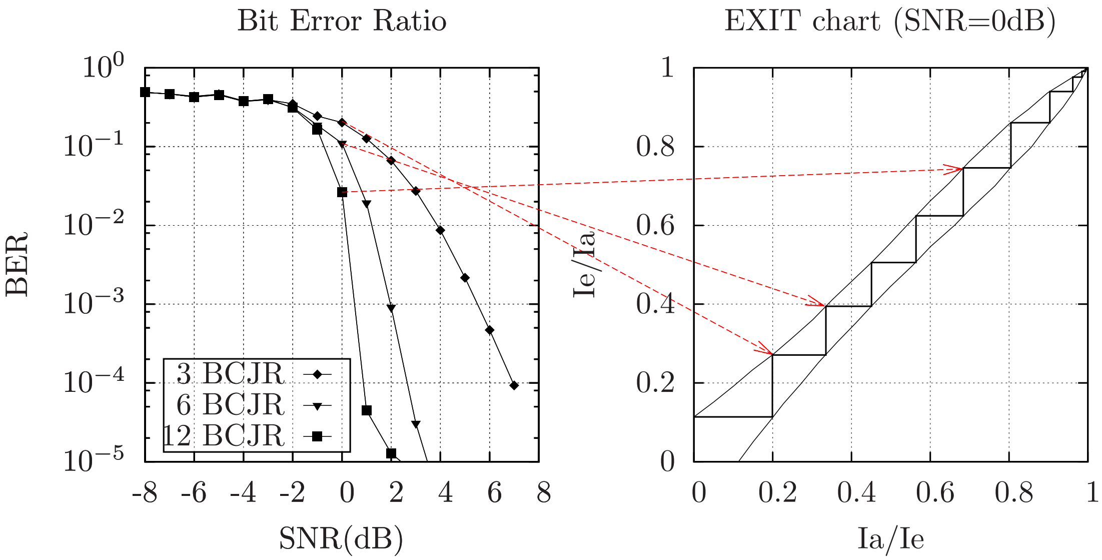


Fig. 12. The relationship between EXIT charts and the BER.

systematic and non-systematic TCTCs. Here, only the lowest open-tunnel SNRs are recorded among all the polynomials having a memory length of $m \leq 3$. Furthermore, the encoders of TCTCs employ two URC encoders for generating the encoded bit sequences $\mathbf{b_1}$ and $\mathbf{b_2}$, which are transmitted the required number of times in order to achieve the desired normalized throughput or rate of R bits per channel use. More explicitly, the output sequence is arranged to be $(\mathbf{b_1}, \mathbf{b_2}, \mathbf{b_1})$ for non-systematic TCTCs and $(\mathbf{a}, \mathbf{b_1}, \mathbf{b_2})$ for systematic TCTCs, when a throughput of $R = \frac{1}{3}$ bits per channel use is desired. For $R = \frac{1}{4}$, it is $(\mathbf{b_1}, \mathbf{b_2}, \mathbf{b_1}, \mathbf{b_2})$ for non-systematic TCTCs and $(\mathbf{a}, \mathbf{b_1}, \mathbf{b_2}, \mathbf{a})$ for systematic TCTCs. Similarly, the above-mentioned Chase combining technique is applied at the receiver of TCTCs. It may be observed in Figure 11 that MCTCs are capable of operating closer to the Discrete-input Continuous-output Memoryless Channel's (DCMC) capacity than non-systematic TCTCs, albeit they are slightly outperformed by their systematic TCTC counterparts. However, systematic TCTCs have to depend on the $m = 3$ polynomial $(17, 15)_o$, while MCTCs benefit from having lower complexities per BCJR algorithm activation than both systematic and non-systematic TCTCs.

C. BER Performance

As a further insight, Figure 13 compares the BERs that can be achieved both by MCTCs and by TCTCs, when fixed decoding complexities are used for recovering the information transmitted over a BPSK-modulated uncorrelated Rayleigh fading channel using an interleaver length of 2048 bits. Again, the complexity recorded in Figure 13 is defined as $Complexity = 2^m \cdot K$. As expected, Figure 13 shows that higher affordable complexities result in improved BER performances. The MCTCs have significantly steeper turbo cliffs and significantly lower error floors than the corresponding non-systematic TCTCs at all the complexities considered. As a result, at a complexity of 48, for example, the MCTCs offer 3.5dB to 4.5dB gain over the non-systematic TCTCs at a BER of 10^{-6}. By contrast, the MCTCs have slightly flatter turbo cliffs accompanied by perceivably lower error floors, compared to the systematic TCTCs having the same coding rates and complexities. For example, at a complexity of 96, the turbo cliff SNRs of the MCTCs and the systematic TCTCs are similar. Still considering the complexity of 96, the BER of MCTCs may become vanishingly low without exhibiting an error floor, whereas the residual BERs of the systematic TCTCs are on the order of 10^{-7}.

V. Early Stopping and Deferred Iteration Aided HARQ Based on MCTC

Since MCTCs outperform TCTCs at the same complexity and have shown attractive performance even using the lowest possible memory-1 generator polynomial of $(2, 3)_o$, this motivates the design of our MCTC aided HARQ scheme. As seen in Figure 14, the transmitter of our proposed MCTC aided HARQ transmits the encoded bits $\mathbf{b}_i$ at regular intervals, until a positive ACK message is received or the transmission count i reaches the affordable retry limit. At the receiver, the multiple-component turbo decoder of Figure 15 is used. More specifically, its operation is as follows. Whenever a sequence of encoded LLRs $\tilde{\mathbf{b}}_i$ is received during the i^{th} IR-transmission, the $N = i$-component MCTC shown in Figure 15 is activated. Whether the iterative decoding should or should not be activated depends on the decision of our Deferred Iteration (DI) strategy to be detailed in Section V-B. If the EXIT tunnel is predicted to be open by the DI strategy for the i-component MCTC decoder, the iterative decoding process commences and continues until the CRC succeeds or conditions required by the ES approach to be proposed in Section V-A are satisfied. The flow chart of the DI and ES aided MCTC HARQ scheme is illustrated in Figure 16, which will be frequently referred to during our further discourse.

HARQ is capable of automatically accommodating the channel quality fluctuations, since the ACK messages immediately curtail retransmissions upon correctly decoding the current packet. Therefore, the MCTC-aided HARQ behaves identically to the TCTC aided HARQ schemes at high SNRs.

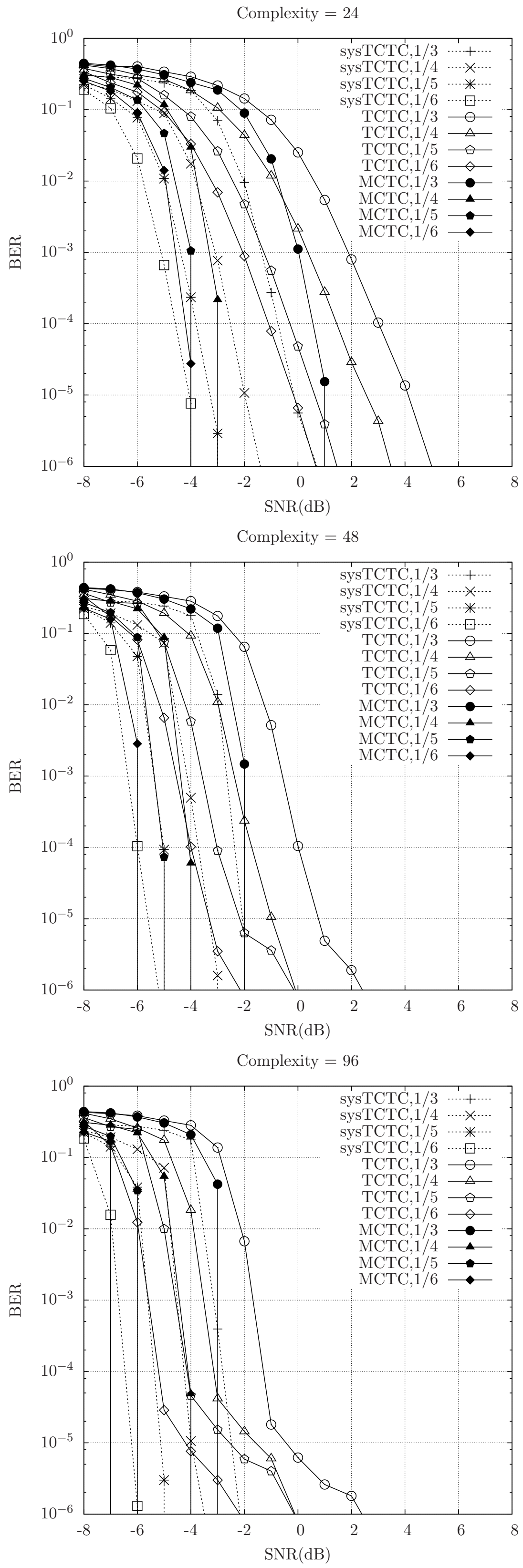


Fig. 13. The BER versus SNR for MCTCs and TCTCs having different complexities for transmission over an uncorrelated Rayleigh channel. All the non-systematic TCTCs having different rates R employ the polynomial of $(10, 17)_o$, and all the systematic TCTCs having different rates R employ the polynomial of $(17, 15)_o$, while the $R = \frac{1}{3}$ MCTC employs $(4, 7)_o$ and the MCTCs having $R < \frac{1}{3}$ employ $(2, 3)_o$.

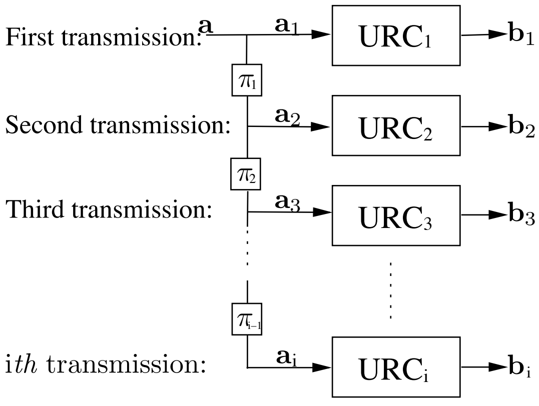


Fig. 14. The encoder structure used in the MCTC HARQ regime.

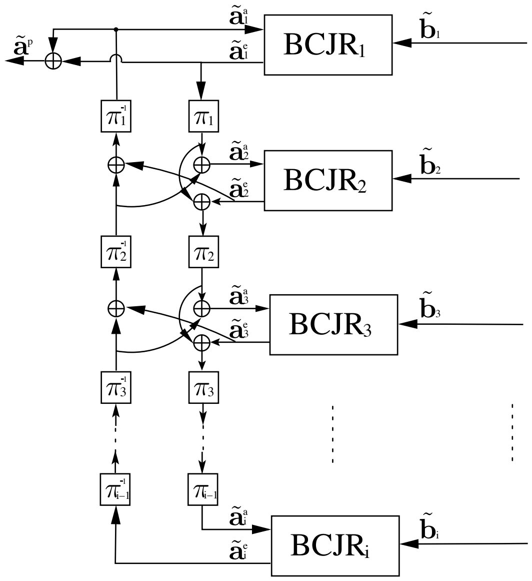


Fig. 15. The decoder structure used in the MCTC HARQ scheme after i IR-transmissions.

More explicitly, if $\tilde{\mathbf{b}}_1$ received during the first transmission is successfully decoded, an ACK message is fed back to deactivate any further transmissions. Otherwise, a second transmission is required. At this stage, a twin-component turbo decoder will perform iterative decoding by exchanging extrinsic information between $\tilde{\mathbf{b}}_1$ and $\tilde{\mathbf{b}}_2$. Following the third transmission, a three-component decoder is activated at the receiver, which may be followed by the third, fourth and so on transmissions, as and when needed for hostile channels. Given the appealing BER performance of MCTCs characterized in Section IV, we expect MCTC aided HARQ to achieve a high performance at a low complexity, as a benefit of our ES and DI strategies.

A. Early Stopping Strategy

The ES strategy is portrayed in the box at the bottom right corner of Figure 16 surrounded by the dotted line, which is discussed below. In classic turbo codes operating without ARQ, these ES approaches determine the specific instant of curtailing iterative decoding by estimating the expected BER performance. However, in turbo HARQ schemes, the ES

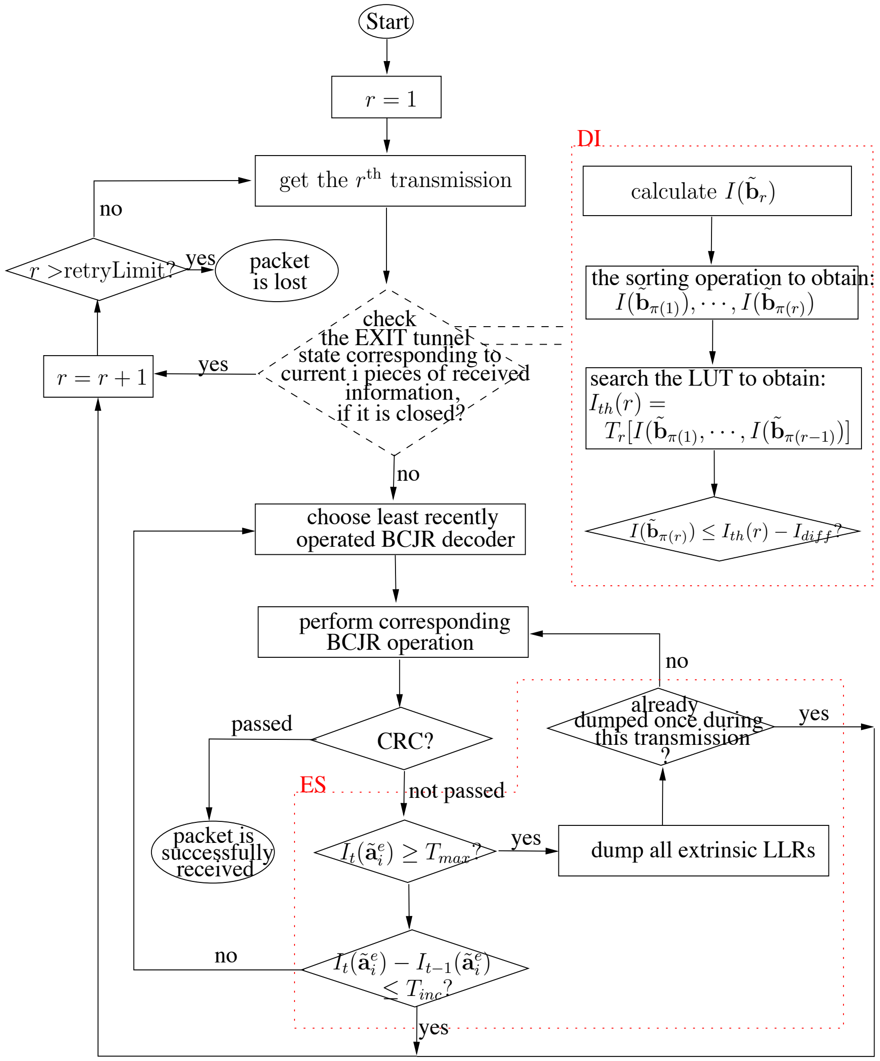

Start
$r = 1$
get the r^{th} transmission
no
r >retryLimit?
yes
packet is lost
check the EXIT tunnel state corresponding to current i pieces of received information, if it is closed?
yes
$r = r + 1$
no
choose least recently operated BCJR decoder
perform corresponding BCJR operation
no
passed
CRC?
not passed
packet is successfully received
already dumped once during this transmission ?
yes
$I_t(\tilde{\mathbf{a}}_i^e) \geq T_{max}$?
yes
dump all extrinsic LLRs
no
$I_t(\tilde{\mathbf{a}}_i^e) - I_{t-1}(\tilde{\mathbf{a}}_i^e) \leq T_{inc}$?
yes
DI
calculate $I(\tilde{\mathbf{b}}_r)$
the sorting operation to obtain: $I(\tilde{\mathbf{b}}_{\pi(1)}), \cdots, I(\tilde{\mathbf{b}}_{\pi(r)})$
search the LUT to obtain: $I_{th}(r) = T_r[I(\tilde{\mathbf{b}}_{\pi(1)}, \cdots, I(\tilde{\mathbf{b}}_{\pi(r-1)})]$
$I(\tilde{\mathbf{b}}_{\pi(r)}) \leq I_{th}(r) - I_{diff}$?
ES

Fig. 16. The flow chart of the DI and ES aided MCTC HARQ scheme based on the look-up table.

approach decides when to request a new incremental transmission, rather than increasing the number of BCJR operations for the current codeword, hence striking a tradeoff between the attainable throughput and the complexity imposed.

The ES approach of Figure 16 is specifically designed for turbo HARQ, based on the MI improvement after each BCJR operation. It introduces two MI thresholds, namely T_{dump} and T_{conv}, where T_{conv} is proposed for dynamically adjusting the number of BCJR operations that are performed following each IR-transmission. We refer to this threshold as the 'convergence threshold'. The second MI threshold, namely T_{dump} is proposed to guarantee a near-zero PLR for the MCTC aided HARQ scheme relying on an infinite number of IR-transmissions. We refer to this threshold as the 'dumping threshold'. Both T_{dump} and T_{conv} are detailed below with reference to Figure 16.

1) Convergence Threshold: T_{conv}. The MI between an LLR sequence and the corresponding hard-decision based bit sequence provides a quantitative indication of our confidence in the corresponding hard decisions. Equation 4 maps the sequence of LLRs to a confidence metric confined to the range of $[0, 1]$, where 0 implies no confidence, while 1 means absolute confidence. By observing the EXIT charts of turbo codes such as Figure 12, we note that the extrinsic information increases along the decoding trajectory of an open or closed tunnel. Our ES strategy estimates the MI increment between two BCJR operations performed by each individual decoder. When the MI increment of any of the BCJR decoders drops below a threshold, this implies that the iterative decoding has converged and using further iterations no longer enhances the attainable decoding performance. The MI increment is expressed by $I_t(\tilde{\mathbf{a}}_i^e) - I_{t-1}(\tilde{\mathbf{a}}_i^e)$ in Figure 16, where $I_t(\tilde{\mathbf{a}}_i^e)$ is the extrinsic MI obtained after the t^{th} operation of BCJR_i and $I_0(\tilde{\mathbf{a}}_i^e)$ is initially assumed to be 0. Therefore, when the MI increment falls below a certain threshold, this may be considered as the stopping criterion, which we refer to as the convergence threshold T_{conv}.

2) Dumping Threshold: T_{dump}. Turbo coded HARQ schemes have an inherent problem, which prevents them from guaranteeing a vanishingly low PLR, even when an infinite number of IR-transmissions is permitted. That is, turbo decoders are capable of converging to a legitimate bit-

TABLE II
THE PARAMETERS OF INVESTIGATING THE VARIATION OF THE TURBO CODED HARQ THROUGHPUT AND COMPLEXITY WITH THE VALUE OF STOPPING THRESHOLDS T_{conv} AND T_{dump}.

Retry Limit	infinite
Packet Length	100-bit, 1000-bit, 10000-bit
T_{conv}	1.0, 0.1, 0.01, 0.001, 0.0001
T_{dump}	0.9, 0.99, 0.999
Modulation Scheme	BPSK
Channel Type	quasi-static Rayleigh fading

TABLE III
THE PREFERRED THRESHOLDS FOR DIFFERENT PACKET LENGTHS FOR THE MCTC AIDED HARQ SCHEME.

Packet Length	100 bits	1000 bits	10000 bits
T_{dump}	0.99	0.999	0.999
T_{conv}	0.017	0.009	0.0032

TABLE IV
THE SIMULATION PARAMETERS FOR COMPARING THE PLR, THROUGHPUT AND COMPLEXITY PERFORMANCE FOR SOUZA'S SYSTEMATIC TCTC AIDED HARQ SCHEME AND OUR MCTC AIDED HARQ SCHEME.

Retry Limit	6
Packet Length	1000-bit
Stopping Strategy	ES (or) 10 BCJR operations
T_{conv}	0.009 for the MCTC HARQ scheme 0.1 for Souza's scheme
T_{dump}	0.999 for both schemes
Modulation Scheme	BPSK
Channel Type	quasi-static Rayleigh fading

sequence for an MI of approximately 1, but they may still output the wrong bit sequence. This occurs when the received soft-sequences are more similar to that of the incorrect bit sequence than to the correct sequence. This effect accounts for the error floor in the BER performance of turbo decoders observed at high SNRs, where the EXIT chart tunnel is wide open. In the turbo coded HARQ scheme, the decoding of the earlier few IR-transmissions may converge towards the wrong but legitimate bit sequence associated with a high confidence, like in a standard turbo code. In this case, the CRC fails and hence further IR-transmissions are requested. Unfortunately, the influence of these later IR-transmissions may become insufficiently decisive to guide the decoder away from the wrong bit sequence associated with MI ≈ 1 and towards the correct one, regardless of how many IR-transmissions are received.

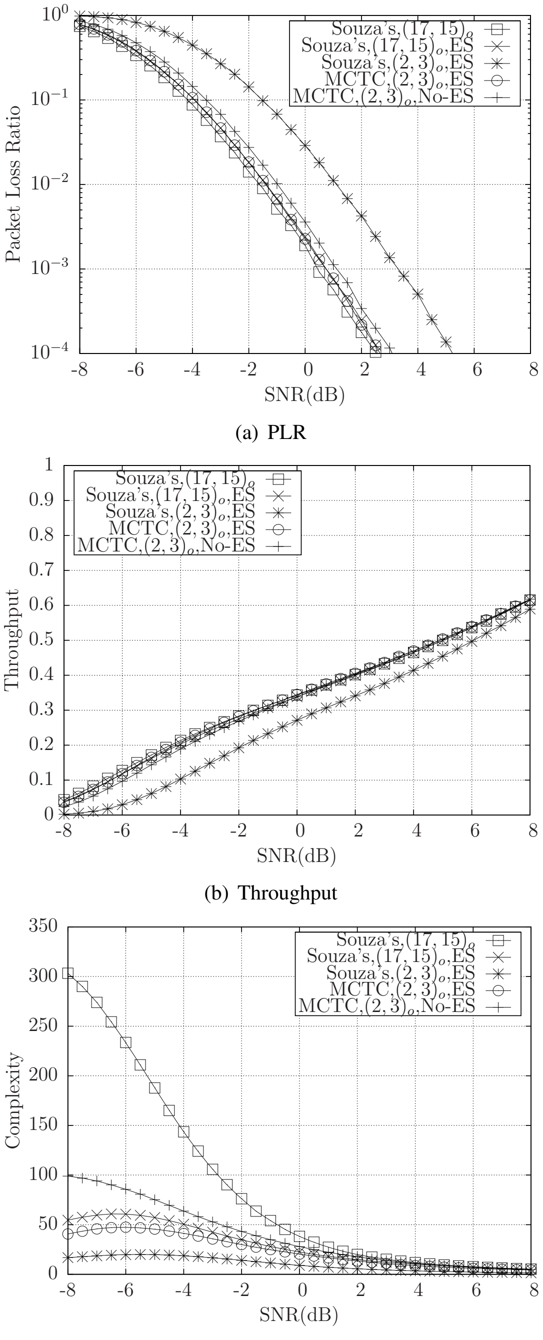
A dumping threshold T_{dump} is employed in Figure 16 to circumvent this problem as detailed below. When the MI of the extrinsic LLRs $\tilde{\mathbf{a}}_i^e$ obtained after some BCJR operation become higher than T_{dump}, while the CRC condition of Figure 12 has not been satisfied, all extrinsic LLRs are reset to zero and an improved iterative decoding procedure is activated, since the decoder is now in possession of potentially numerous received replicas of the original information. For most practical cases, this re-initialized iterative process may lead to a better chance of successful decoding. In the exceptionally rare case when the MI obtained following a BCJR operation becomes larger again than T_{dump} without satisfying the CRC, the receiver dumps all extrinsic LLRs once more and requests a new IR-transmission. This process is repeated until the packet is correctly decoded.

3) Determining the Stopping Thresholds: An off-line training may be employed to find the preferred values of stopping thresholds. During the off-line simulations, we determined both the throughput and the complexity associated with each of the 15 combinations of $T_{conv} \in \{1.0, 0.1, 0.01, 0.001, 0.0001\}$ and $T_{dump} \in \{0.9, 0.99, 0.999\}$, for a range of channel SNRs and packet lengths. Table II summarized the parameters of this offline training.

The simulation results showed that as the convergence threshold T_{conv} was increased, both the throughput and the complexity was reduced for all the packet lengths considered. Fortunately, the throughput reduction was relatively modest, while the complexity was significantly reduced, as T_{conv} was increased from 0.0001 to 0.01. Since achieving a high throughput is one of the prime design targets of HARQ schemes, we specify an average throughput reduction of 0.005 as the maximum loss that can be tolerated, when optimizing the corresponding T_{conv} values for all three packet lengths. We appropriately adjusted the T_{conv} values in order to obey this throughput reduction limit. Quantitatively, we found that the preferred values were $T_{conv} = 0.017$ for the 100-bit, $T_{conv} = 0.009$ for the 1000-bit and $T_{conv} = 0.0032$ for the 10000-bit packet lengths, as seen in Table III, which yielded the lowest complexity for MCTC aided HARQ, when the maximum throughput loss was allowed to be 0.005. As the packet length increases, it becomes more beneficial to complete more decoding iterations, since this allows the decoder to widely disseminate the soft information throughout the packet. As a benefit, the reliable bits assist the decoder in correcting the less reliable bits. This explains why T_{conv} reduces and T_{dump} increases, as the packet length increases in Table III, allowing more decoding iterations to take place.

4) ES Performance: In order to demonstrate the advantages of our ES approach portrayed at the bottom right corner of Figure 12, the PLR, the throughput and the complexity metrics are benchmarked against Souza's systematic TCTC aided HARQ scheme [67] detailed in Section III, when invoking our MCTC aided HARQ scheme. The simulation parameters are summarized in Table IV, where a pre-defined number of 10 BCJR operations was adopted in Souza's original paper following each transmission.

Figure 17-(a) illustrates the attainable PLR versus SNR performance, while Figure 17-(b) shows the associated throughput versus SNR trends. It can be observed from these two figures that our ES-based MCTC aided HARQ scheme using the $m = 1$ GPs of $(2, 3)_o$ succeeds in maintaining a similar PLR and throughput performance as that offered by Souza's more complex scheme using the $m = 3$ polynomials of $(17, 15)_o$, regardless of whether the proposed ES approach



(a) PLR

(b) Throughput

(c) Complexity

Fig. 17. PLR, throughput and complexity versus SNR for MCTC aided HARQ schemes and Souza's systematic TCTC aided HARQ schemes.

is adopted or not. However, both the PLR and throughput are significantly reduced if the polynomials are changed to $(2,3)_o$ for Souza's scheme detailed in Section III, showing that using this low-complexity polynomial pair is only appropriate for the proposed MCTC HARQ scheme. This is because in a two-dimensional EXIT chart, the $(2,3)_o$ GPs create a tunnel that requires a high SNR to become open. However, the MCTC EXIT charts discussed in Section IV-A have a higher dimensionality, which causes the tunnel to become open at lower SNRs, when employing the $(2,3)_o$ polynomial.

Figure 17-(c) shows the complexity benefits achieved by the proposed ES-based MCTC aided HARQ scheme, which has a significantly lower complexity than Souza's original systematic TCTC HARQ scheme relying on the $m = 3$ polynomials of $(17,15)_o$. Specifically, for SNRs below 0dB, the complexity is reduced by 50% to 85%. At higher SNRs, the complexity is similar, because the CRC of Figure 16 is typically satisfied for all schemes after a few BCJR operations. On the other hand, if Souza's systematic TCTC HARQ scheme detailed in Section III and using the $m = 3$ polynomials of $(17,15)_o$ adopts our proposed ES approach, its complexity can also be significantly reduced. However, its complexity still remains higher than that of our MCTC aided HARQ scheme, because it has to adopt those higher memory polynomials. Although Souza's scheme [67] using the ES approach of Figure 16 is capable of achieving the lowest complexity for the $m = 1$ polynomials of $(2,3)_o$, this is achieved at the cost of degrading the PLR and throughput performance, as demonstrated in Figures 17-(a) and 17-(b).

The PLR, throughput and complexity curves were recorded for the MCTC HARQ scheme without ES in Figure 17, which further demonstrate how critical the careful employment of ES is for a turbo HARQ scheme. The PLR and throughput performances recorded for the No-ES MCTC HARQ scheme exhibit an insignificant degradation. Furthermore, the complexity more than doubles compared to that of the ES aided MCTC HARQ scheme of Figure 16 for low SNRs, namely below -2dB. This illustrates that a fixed number of 10 BCJR operations is insufficient for reaching the minimum PLR and maximum throughput that the MCTC HARQ scheme achieved for most situations, when the EXIT tunnel is open. By contrast, 10 BCJR operations appear to be excessive for most situations associated with closed EXIT tunnels, hence imposing an excessive complexity.

B. Deferred Iterations

Having characterized the ES philosophy of Figure 16, let us now focus our attention on our second complexity reduction measure, namely on the DI technique of Figure 16. Explicitly, the complexity of MCTC aided HARQ schemes may be further reduced by our Look-Up Table (LUT) based DI method detailed at the top right corner of Figure 16. The DI method proposed in this section delays the iterative decoding until the receiver estimates that it has received sufficient information for successful decoding, which may be represented by the emergence of an open tunnel in the EXIT chart corresponding to all received packet replicas. Therefore, the specific MI for which a marginally open tunnel appears when combining all previous $(i-1)$ MI contributions will become the threshold that has to be satisfied by the i^{th} reception. More specifically, if the MI received during the i^{th} reception is higher than this threshold denoted by $I_{th}(i)$, the EXIT tunnel is deemed to be open and hence the iterative decoding should be triggered without any further delay. Otherwise, iterative decoding will be deferred, when the tunnel is deemed to be closed. This eliminates futile iterations and hence reduces the complexity, as well as the power consumption.

Figure 18 illustrates the EXIT chart of a 3-component MCTC, which is drawn using the partitioning method introduced in Section IV-B. The MIs $I(\tilde{\mathbf{b}}_1) = 0.15$ and

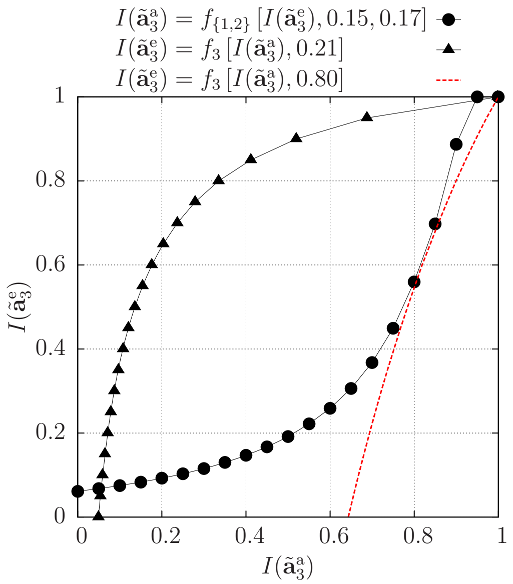
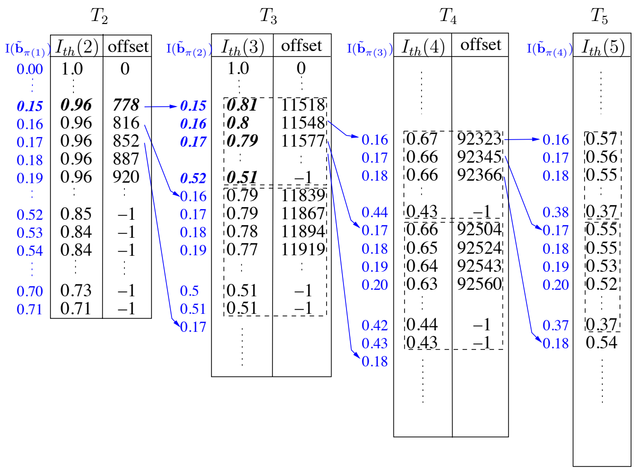


Fig. 18. The EXIT chart of a 3-component MCTC.



$I(\tilde{\mathbf{b}}_{\pi(1)})$	$I_{th}(2)$	offset
0.00	1.0	0
⋮	⋮	
0.15	***0.96***	***778***
0.16	0.96	816
0.17	0.96	852
0.18	0.96	887
0.19	0.96	920
⋮	⋮	⋮
0.52	0.85	−1
0.53	0.84	−1
0.54	0.84	−1
⋮	⋮	⋮
0.70	0.73	−1
0.71	0.71	−1

$I(\tilde{\mathbf{b}}_{\pi(2)})$	$I_{th}(3)$	offset
	1.0	0
	⋮	⋮
0.15	***0.81***	11518
0.16	***0.8***	11548
0.17	***0.79***	11577
	⋮	⋮
0.52	***0.51***	−1
0.16	0.79	11839
0.17	0.79	11867
0.18	0.78	11894
0.19	0.77	11919
	⋮	⋮
0.5	0.51	−1
0.51	0.51	−1
0.17	⋮	

$I(\tilde{\mathbf{b}}_{\pi(3)})$	$I_{th}(4)$	offset
	⋮	
0.16	0.67	92323
0.17	0.66	92345
0.18	0.66	92366
	⋮	
0.44	0.43	−1
0.17	0.66	92504
0.18	0.65	92524
0.19	0.64	92543
0.20	0.63	92560
	⋮	
0.42	0.44	−1
0.43	0.43	−1
0.18	⋮	

$I(\tilde{\mathbf{b}}_{\pi(4)})$	$I_{th}(5)$
	⋮
0.16	0.57
0.17	0.56
0.18	0.55
	⋮
0.38	0.37
0.17	0.55
0.18	0.55
0.19	0.53
0.20	0.52
	⋮
0.37	0.37
0.18	0.54
	⋮

Fig. 19. A subset of the MCTC HARQ LUT for $i = 2, 3, 4$ and 5, where the input MI step-size is $G = 0.01$.

$I(\tilde{\mathbf{b}}_2) = 0.17$ which are calculated on the channel's output LLRs $\tilde{\mathbf{b}}_1$ and $\tilde{\mathbf{b}}_2$ from the first and second transmission, correspond to the composite EXIT function $I(\tilde{\mathbf{a}}_3^{\mathrm{a}}) = f_{\{1,2\}}[I(\tilde{\mathbf{a}}_3^{\mathrm{e}}), 0.15, 0.17]$ of Figure 18. This EXIT chart analysis reveals that if the LLRs $\tilde{\mathbf{b}}_3$ supplied by the the third transmission have an MI of at least $I_{th}(3) = 0.80$, then an open EXIT chart tunnel will be created with the EXIT function $I(\tilde{\mathbf{a}}_3^{\mathrm{e}}) = f_3[I(\tilde{\mathbf{a}}_3^{\mathrm{a}}), 0.80]$. In this case, there is a high probability that the information bit sequence $\mathbf{a}_1$ can indeed be successfully recovered by activating the iterative decoding process. If this is not the case, e.g. we have $I(\tilde{\mathbf{b}}_3) = 0.21$ as seen in Figure 18, then the start of the iterative decoding process may be deferred, until an open EXIT chart tunnel is deemed to have been created. Here, the threshold value of $I_{th}(3) = 0.80$ could be revealed to the receiver by rounding the values of $I(\tilde{\mathbf{b}}_1) = 0.15$ and $I(\tilde{\mathbf{b}}_2) = 0.17$ to two decimal places and using them as the address of an LUT comprising $(101)^2$ entries.

In order to accurately predict the EXIT tunnel's open/closed states, the threshold $I_{th}(i)$ should be determined by an offline-training and stored in advance. An LUT is designed to store all thresholds for all possible MI combinations for the r-component MCTCs, where r ranges from 2 to the maximum affordable retry limit R. For example, it may be readily shown that for a 3-component MCTC, the number of LUT entries for all combinations of $I(\tilde{\mathbf{b}}_1)$ and $I(\tilde{\mathbf{b}}_2)$ becomes $(101)^2 \approx 10000$, when the step size of the MI region $[0, 1]$ is assumed to be $G = 0.01$. Generally, a total of $\sum_{r=2}^{R}(101)^{r-1}$ entries are required for the LUT of MCTC aided HARQ, which potentially makes the storage requirements of the LUT rather excessive and hence the implementation of the offline-training may become unattractive. However, our proposed sophisticated LUT design significantly reduces the number of LUT entries that must be trained and stored. Hence, below a more practical DI scheme is devised for MCTC HARQ, which is capable of facilitating a significantly reduced decoding complexity.

1) Minimizing the Storage Requirements: The LUT designed for MCTC HARQ schemes may be defined as a set of sub-tables T_i generated for describing the relationship between any particular set of $(i-1)$ MIs and the minimum supplemental MI $I_{th}(i)$, i.e. the threshold required for creating an open EXIT chart tunnel, according to:

$$I_{th}(i) = T_i\left[I(\tilde{\mathbf{b}}_1), I(\tilde{\mathbf{b}}_2), \cdots, I(\tilde{\mathbf{b}}_{i-1})\right], \tag{6}$$

where $2 \leq i \leq (R-1)$. Our DI strategy aided MCTC HARQ schemes does not require the R^{th} sub-table T_R, since the receiver should always exploit its final - namely the R^{th} opportunity of activating the turbo decoder after the reception of the R^{th} IR transmission.

- **Sorting LUT entries**

The LUT of MCTC HARQ only records the required thresholds $I_{th}(i)$ for a limited set of quantized and sorted $(i-1)$ MI values appearing in an ascending order, i.e. satisfying $I(\tilde{\mathbf{b}}_{\pi(1)}) \leq I(\tilde{\mathbf{b}}_{\pi(2)}) \leq ... \leq I(\tilde{\mathbf{b}}_{\pi(i-1)}) \leq I_{th}(i)$, where π contains the unique integers of $1, ..., (i-1)$ used for appropriately permuting the original IR transmission order. This method avoids storing a large amount of redundant entries, since for example $I_{th}(4) = T_4(0.21, 0.15, 0.17)$ is identical to $I_{th}(4) = T_4(0.15, 0.17, 0.21)$.

Figure 19 displays a subset of the LUTs recorded for BPSK transmission over a quasi-static Rayleigh fading channel, as considered in Section V-B2. As seen in Figure 19, the LUT is composed of several sub-tables, each of which corresponds to different IR indices i. Each sub-table T_i - except for the last one having the index $(R-1)$ - contains two columns. The first one stores the MI threshold $I_{th}(i)$, while the other provides an offset, which can assist the indexing of the next sub-table T_{i+1}. More explicitly, the first element of each row in the sub-table T_i stores the $I_{th}(i)$ value that is valid for a particular set of MIs $\{I(\tilde{\mathbf{b}}_{\pi(1)}), I(\tilde{\mathbf{b}}_{\pi(2)}), ..., I(\tilde{\mathbf{b}}_{\pi(i-1)})\}$. The second element of each row stores a specific offset of the sub-table T_{i+1}, which indicates the starting index of the rows related to the sub-table T_i within T_{i+1}. Specifically, these rows are displayed in dashed boxes in Figure 19, all of which store the threshold MIs $I_{th}(i+1)$ corresponding to the sets of

$\{I(\tilde{\mathbf{b}}_{\pi(1)}), I(\tilde{\mathbf{b}}_{\pi(2)}), ..., I(\tilde{\mathbf{b}}_{\pi(i-1)}), I(\tilde{\mathbf{b}}_{\pi(i)})\}$, where $I(\tilde{\mathbf{b}}_{\pi(i)})$ is varied, but the previous $(i-1)$ IR MI values remain the same as in conjunction with the equivalent row in T_2.

Considering $I(\tilde{\mathbf{b}}_{\pi(1)}) = 0.15$ in Figure 19 as an example for providing further explanations, its corresponding threshold information is $I_{th}(2) = T_2(0.15) = 0.96$, implying that the minimum IR MI required for creating an open tunnel is 0.96, when the received MI of the first transmission is 0.15. This threshold is stored at the row index of $I(\tilde{\mathbf{b}}_{\pi(1)}) \cdot \frac{1}{G} = 15$ in sub-table T_2 having $G = 0.01$, as seen in Figure 19. Due to the sorting of $I(\tilde{\mathbf{b}}_{\pi(1)}) \leq I(\tilde{\mathbf{b}}_{\pi(2)}) \leq I_{th}(3)$, the initial value of $I(\tilde{\mathbf{b}}_{\pi(2)})$ is 0.15, thence we fix $I(\tilde{\mathbf{b}}_{\pi(1)}) = 0.15$ and increase $I(\tilde{\mathbf{b}}_{\pi(2)})$ from 0.15 by $G = 0.01$ each step (printed in bold fonts) to explore all possible $I_{th}(3)$ values corresponding to them. The operations continue until the $I_{th}(3)$ value required for perfect convergence becomes less than the current $I(\tilde{\mathbf{b}}_{\pi(2)})$ value, again, owing to the above-mentioned ordering. The resultant $I_{th}(3)$ values corresponding to $I(\tilde{\mathbf{b}}_{\pi(1)}) = 0.15$ and the incremental $I(\tilde{\mathbf{b}}_{\pi(2)})$ values are recorded in a block of continuous rows, as illustrated in the first dashed rectangle of sub-table T_3 in Figure 19. These 38 rows of sub-table T_3 have the offsets ranging from 778 to 815, which correspond to the 38 incremental $I(\tilde{\mathbf{b}}_{\pi(2)})$ values ranging from 0.15 to 0.52, as seen in the index area of sub-table T_3 in Figure 19. Here, the index of this block in sub-table T_3 starts at 778, since the rows 0 to 14 in sub-table T_2 have a total of 778 entries in sub-table T_3.

There are some special cases to be considered in Figure 19, for example, when the fixed $I(\tilde{\mathbf{b}}_{\pi(1)})$ has a larger value, such as $I(\tilde{\mathbf{b}}_{\pi(1)}) = 0.52$. Then, owing to the above-mentioned ordering, the incremental $I(\tilde{\mathbf{b}}_{\pi(2)})$ values start from 0.52. In this situation, the threshold IR MI $I_{th}(3) = 0.15$ corresponding to $\{0.52, 0.52\}$ becomes less than the current $I(\tilde{\mathbf{b}}_{\pi(2)}) = 0.52$. This suggests that no entries will be stored in sub-table T_3 for the fixed $I(\tilde{\mathbf{b}}_{\pi(1)})$ value of 0.52. We tag the corresponding offset as -1 for these cases.

- **Searching the LUT**

Based on the above structure of the LUT, the search operation may be carried out at a low complexity using the offsets that are stored in the sub-tables. More specifically, the expression of 'offset+$[I(\tilde{\mathbf{b}}_{\pi(i)}) - I(\tilde{\mathbf{b}}_{\pi(i-1)})] \cdot 100$' may be applied recursively, where an initial 'offset' of 0 and $I(\tilde{\mathbf{b}}_0) = 0$ are assumed for $i = 1$, namely searching a certain $I_{th}(2)$ from the sub-table T_2. Once an offset of -1 has been found, this indicates that the EXIT tunnel is definitely open, since the received MIs are always sorted in an ascending order before the search is activated.

- **Larger step size and interpolation**

Although the LUT only records the sorted MIs, the size of the sub-table T_i increases approximately by an order of magnitude upon increasing i by one. However, increasing the step-size G has the potential of further decreasing the LUT size. In order to guarantee a high accuracy for the determination of the EXIT tunnel's open/closed state, the values of $I_{th}(i)$ in the LUT always maintain a step-size of 0.01, while $I(\tilde{\mathbf{b}}_1), I(\tilde{\mathbf{b}}_2), \cdots, I(\tilde{\mathbf{b}}_{i-1})$ tend to be increased in larger steps. As a result, the sub-table T_4 of Figure 19 changes to only store the threshold MIs at a lower resolution. For example, for a step-size of $G = 0.1$, it stores the threshold MIs such as $0.80 = T_4(0.1, 0.1, 0.1)$, $0.73 = T_4(0.1, 0.1, 0.2)$, $0.65 = T_4(0.1, 0.2, 0.2)$ and so on. For a set of $(i-1)$ received MIs represented at a higher resolution, we follow the classic multi-dimensional linear interpolation technique for the sake of estimating the supplemental MI required for achieving perfect decoding convergence and hence for triggering iterative decoding. This operation will require 2^{i-1} memory accesses to the LUT and $\sum_{k=0}^{i-2} 2^k$ linear interpolations for generating $I_{th}(i)$. Nonetheless, the complexity imposed is still far lower than the BCJR decoding complexity, provided that the value of i is moderate.

2) Minimizing the Offline-training Complexity: An offline training process was developed for generating the LUT of Figure 19 by finding the specific $I_{th}(i)$ values for all possible values of $I(\tilde{\mathbf{b}}_{\pi(1)}), I(\tilde{\mathbf{b}}_{\pi(2)}) \ldots I(\tilde{\mathbf{b}}_{\pi(i-1)})$. This process scans through the entire input MI range in steps of 0.01 - regardless of the specific value of G - in order to find the minimum MI for creating a marginally open EXIT tunnel, when it is combined with a certain set of MIs $I(\tilde{\mathbf{b}}_{\pi(1)}), I(\tilde{\mathbf{b}}_{\pi(2)}) \ldots I(\tilde{\mathbf{b}}_{\pi(i-1)})$. For the sake of efficient execution, we abandon the traditional LLR-histogram based experimental EXIT chart creation and instead we use a model based on the spline functions $I(\tilde{\mathbf{a}}_i^{\mathrm{e}}) = f_i'[I(\tilde{\mathbf{a}}_i^{\mathrm{a}}), I(\tilde{\mathbf{b}}_i)]$ by fitting the corresponding EXIT functions f_i. Each spline function f_i' consists of a group of linear polynomials having the following form

$$I(\tilde{\mathbf{a}}_i^{\mathrm{e}}) = \begin{cases} m_{i1} \cdot I(\tilde{\mathbf{a}}_i^{\mathrm{a}}) + n_{i1} & (0.0 \leq I(\tilde{\mathbf{a}}_i^{\mathrm{a}}) < 0.1) \\ m_{i2} \cdot I(\tilde{\mathbf{a}}_i^{\mathrm{a}}) + n_{i2} & (0.1 \leq I(\tilde{\mathbf{a}}_i^{\mathrm{a}}) < 0.2) \\ \qquad \vdots & \\ m_{i10} \cdot I(\tilde{\mathbf{a}}_i^{\mathrm{a}}) + n_{i10} & (0.9 \leq I(\tilde{\mathbf{a}}_i^{\mathrm{a}}) \leq 1.0), \end{cases} \tag{7}$$

where $m_{i1} \cdots m_{i10}$ and $n_{i1} \cdots n_{i10}$ were determined in advance for any $I(\tilde{\mathbf{b}}_i) \in [0, G, 2G, \cdots, 1]$.

The training is accelerated by iteratively invoking r spline functions f_i' corresponding to the MCTC decoder's r constituent components, since each call of this function only involves simple calculations. More specifically, all $I(\tilde{\mathbf{a}}_i^{\mathrm{e}}), i = 1, 2, \cdots, r$ are firstly initialized to 0. Then, we progress from f_1' to f_r', each time with a corresponding $I(\tilde{\mathbf{a}}_i^{\mathrm{a}})$ input calculated according to the following equation:

$$I(\tilde{\mathbf{a}}_i^{\mathrm{a}}) = J\left(\sqrt{\sum_{j=1, j\neq i}^{r} \left[J^{-1}(I(\tilde{\mathbf{a}}_j^{\mathrm{e}}))\right]^2}\right), \tag{8}$$

where the function $J(\cdot)$ and $J^{-1}(\cdot)$ can be found in the Appendix of [74]. This procedure continues, until the output extrinsic information $I(\tilde{\mathbf{a}}_i^{\mathrm{e}})$ approaches 1 or until the affordable number of iterations is exhausted.

The training process can be accelerated by exploiting the monotonically decreasing nature of the function T_i, which ensures that each consecutive row in the LUT of Figure 19 requires a slightly lower MI value for achieving convergence than the previous one. For example, where we have $I(\tilde{\mathbf{b}}_{\pi(1)}), I(\tilde{\mathbf{b}}_{\pi(2)}), I(\tilde{\mathbf{b}}_{\pi(3)}) = \{0.15, 0.15, 0.17\}$, the additional MI required is $I_{th}(4) = 0.66$ in Figure 19, which is slightly lower than the previous value of $I_{th}(4) = 0.67$, corresponding to $I(\tilde{\mathbf{b}}_{\pi(1)}), I(\tilde{\mathbf{b}}_{\pi(2)}), I(\tilde{\mathbf{b}}_{\pi(3)}) = \{0.15, 0.15, 0.16\}$.

TABLE V
THE NUMBER OF ENTRIES FOR 6 SUB-TABLES OF THE LUT FOR THE STEP-SIZES OF 0.01, 0.05 AND 0.1.

	T_2	T_3	T_4	T_5	T_6	T_7
G=0.01	71	1261	13337	95388	508767	2159981
G=0.05	16	80	264	602	1108	1752
G=0.1	9	32	70	110	137	164

TABLE VI
THE SIMULATION PARAMETERS FOR COMPARING THE PLR, THROUGHPUT AND COMPLEXITY PERFORMANCE WITH AND WITHOUT THE DI STRATEGY FOR MCTC AIDED HARQ, SOUZA'S SYSTEMATIC TCTC AIDED HARQ AND LET HARQ SCHEMES.

Retry Limit	6
Packet Length	48-bit, 480-bit, 4800-bit
Stopping Strategy	ES
T_{conv}	see Table VII
T_{dump}	0.99 for 48-bit packet length 0.999 for 480-bit, 4800-bit
Deferred Iteration	with (or) without
LUT step-size	0.1
I_{diff}	see Table VIII
Modulation Scheme	BPSK
Channel Type	quasi-static Rayleigh fading

TABLE VII
THE PREFERRED T_{conv} VALUES FOR 48, 480 AND 4800 BITS PACKET LENGTHS WHICH IS FOUND BY THE SIMILAR OFFLINE TRAINING IN SECTION IV, WHEN THE ES STRATEGY IS EQUIPPED AND THE ALLOWED THROUGHPUT LOSS IS LESS THAN 0.005.

	48 bits	480 bits	4800 bits
MCTC HARQ	0.018	0.011	0.005
Souza's HARQ	0.15	0.125	0.06
LTE HARQ	0.058	0.048	0.036

This implies that the investigation of each new threshold test may commence from the value of the previous one, in decremental steps of 0.01. Statistically speaking, only two or three steps are required for determining the additional MI that yields an open EXIT tunnel.

3) Storage Requirements of the LUT: Based on the above LUT structure and training process, Table V shows the number of entries for T_2 to T_7 sub-tables of the LUT for different step-sizes, namely $G \in \{0.01, 0.05, 0.1\}$.

Observe from Table V that the total number of entries is significantly reduced for the step-size of $G = 0.1$ in comparison to 0.01. Our simulation results not included here owing to space-limitation demonstrated that the PLR, throughput and complexity curves of $G = 0.01, 0.05$ and 0.1 recorded for the LUT based DI aided MCTC HARQ are almost identical. Therefore, $G = 0.1$ is the best choice for the LUT, since it requires the minimum storage. When designing T_2, T_3, T_4 and T_5 with the step-size of 0.1 for our later simulations to be detailed in Section V-B5 and supporting $R = 6$ IR transmissions, only 212 MI thresholds have to be stored in these four sub-tables of the LUT. The complexity of the associated multiple-dimensional interpolation includes at most 16 memory accesses and 15 simple linear interpolations. Furthermore, if the LUT has to support more than $R = 6$ IR transmissions, e.g. $R = 8$, we may combine T_2, T_3 having $G = 0.01$ with T_4 to T_7 also having $G = 0.1$ in order to strike a trade-off between the memory requirements and the multi-dimensional interpolation cost.

4) I_{diff} for short packets: Satisfying the condition of $I(\tilde{\mathbf{b}}_{\pi(r)}) \leq I_{th}(r)$ cannot always provide a sufficiently reliable judgement of whether the trajectory can or cannot navigate through the EXIT tunnel to the $(1,1)$ point. This is, because for short packets the trajectory may sometimes navigate through the tunnel that is marginally closed and vice versa [75]. Since our primary objective is to approach the maximum possible throughput, rather than waiting for the EXIT tunnel to open, it is desirable to allow iterative decoding to commence, even if the tunnel is marginally closed, especially when the packet length is short. This is achieved by modifying the threshold test according to $I(\tilde{\mathbf{b}}_{\pi(r)}) \leq I_{th}(r) - I_{diff}$, as seen in the flow chart of Figure 16, where I_{diff} is chosen to be the appropriate MI 'safety margin' for the specific packet length. More particularly, if I_{diff} is chosen to be too high, then iterative decoding might commence at too low MI values, when there is no chance for the trajectory to navigate through the tunnel, hence unnecessarily increasing the complexity. By contrast, if I_{diff} is chosen to be too low, then iterative decoding will be deferred, even when there is a fair chance for the trajectory to navigate through the 'just' closed tunnel. This may potentially reduce the throughput. For this reason, the appropriate values for I_{diff} for a range of packet lengths may be pre-determined by off-line simulations.

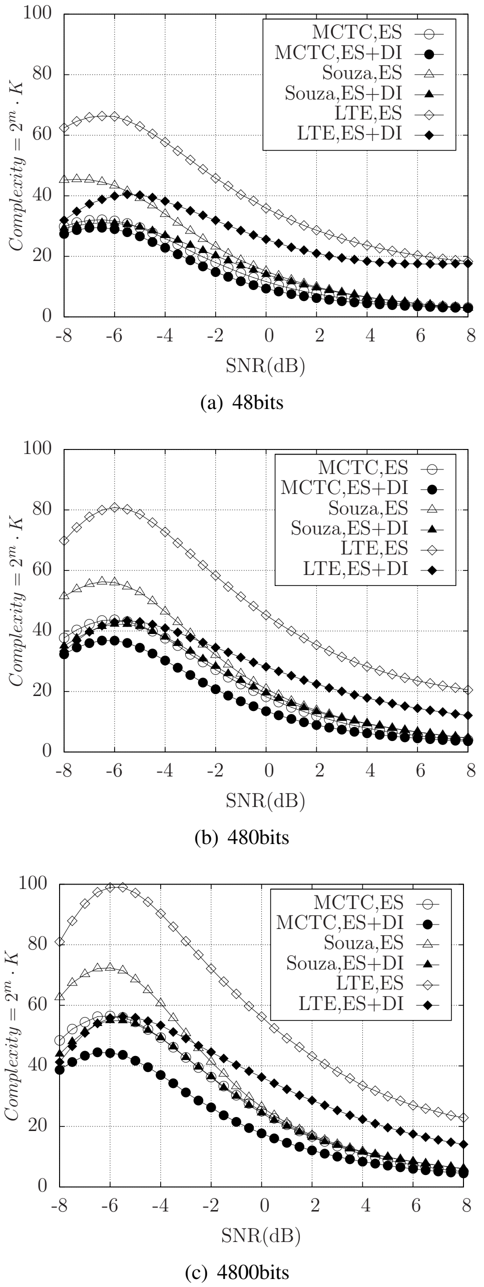
5) DI performance: In this section, we evaluate the PLR, throughput and complexity of our proposed MCTC HARQ scheme relying on the proposed ES and LUT based DI strategy. This was achieved by simulating the transmission of a statistically relevant number of packets over a BPSK-modulated quasi-static Rayleigh fading channel. We also apply the proposed ES and LUT based DI strategy to Souza's systematic TCTC HARQ [67] and to the LTE system's systematic TCTC HARQ [70], both of which were detailed in Section III. These three HARQ schemes are used as our benchmarkers. The simulation parameters are summarized in Table VI.

The appropriate I_{diff} values were selected for these packet lengths in order to limit the maximum normalized throughput loss imposed by the DI strategy to be as low as 0.003. Table VIII shows the preferred I_{diff} values for the three HARQ schemes considered. Observe from Table VIII that for Souza's HARQ scheme, the preferred I_{diff} values are all zeros, regardless of how short the packet length is. This is because the determination of the EXIT tunnel's open/closed state only starts after the third transmission, and because in practice we seldom have a 'just' open or 'just' closed EXIT tunnel.

Figure 20 shows the complexity versus SNR performance for the three HARQ schemes both with and without the DI strategy of Figure 16. As shown in Figure 20, the 'MCTC,ES+DI' HARQ scheme offers complexity reductions of approximately 10%, 20% and 20% for the packet lengths of 48, 480 and 4800 bits respectively, when compared to the

TABLE VIII
THE PREFERRED I_{diff} VALUES FOR 48, 480 AND 4800 BITS PACKET LENGTHS, WHEN THE ALLOWED MAXIMUM THROUGHPUT LOSS IS 0.003.

	48 bits	480 bits	4800 bits
MCTC HARQ	0.08	0.01	0.0
Souza's HARQ	0.0	0.0	0.0
LTE HARQ	0.11	0.0	0.0



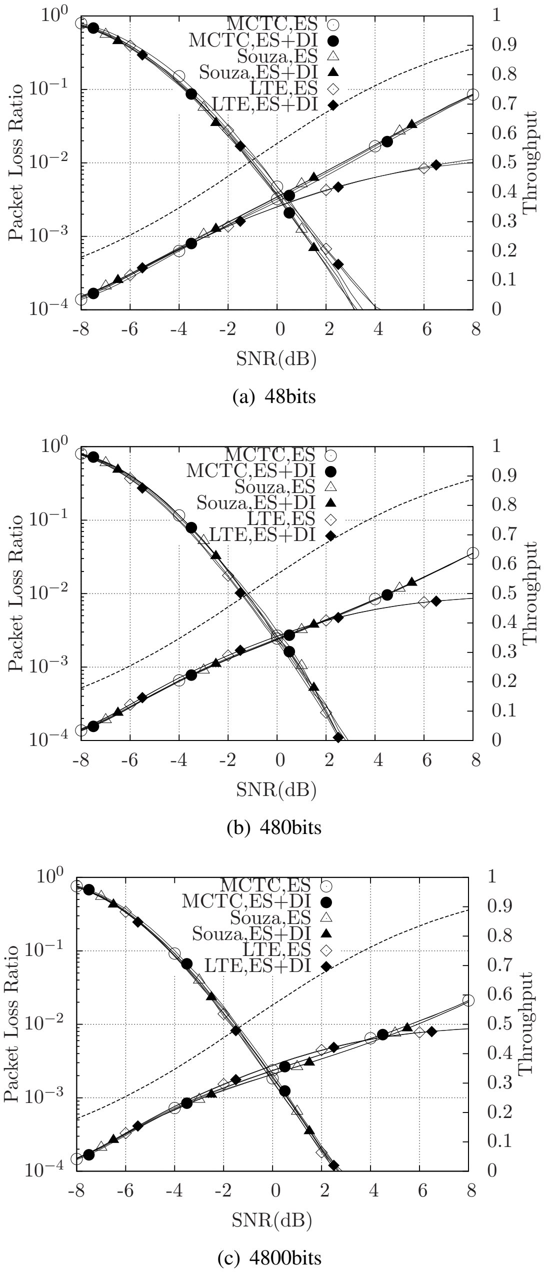
(a) 48bits

(b) 480bits

(c) 4800bits

Fig. 20. Complexity versus the quasi-static Rayleigh fading channel SNR for message packets of length a) 48 bits, b) 480 bits and c) 4800 bits.

'MCTC,ES' scheme. However, when the LUT based DI is applied to Souza's systematic TCTC HARQ, the complexity reductions become about 35%, 32% and 30% for the 48, 480 and 4800-bit packet lengths, since Souza's scheme [67] only relied on the ES strategy. Furthermore, the 'LTE,ES+DI' arrangement achieved the highest complexity reductions of up to 50% for all three packet lengths, since the LTE HARQ scheme activates the turbo decoding right away after the first transmission. The LUT-based DI aided MCTC HARQ scheme shows the lowest complexity among all HARQ schemes.



(a) 48bits

(b) 480bits

(c) 4800bits

Fig. 21. PLR and throughput versus the quasi-static Rayleigh fading channel SNR for message packets of length a) 48 bits, b) 480 bits and c) 4800 bits. The dashed line represents the DCMC capacity.

The left and right axes of Figure 21, respectively, illustrate the PLR and throughput performances, which are similar, regardless of which turbo HARQ scheme is used and whether the DI is employed, for all the three packet lengths considered. Nonetheless, there is one exception, where the throughput of the LTE HARQ scheme becomes significantly lower than that of the other two HARQ schemes, namely at high SNRs[1].

[1] In our simulations, the LTE HARQ has been implemented without the aid of other schemes specified in the LTE standard, for example the adaptive modulations.

This is because many packets may be successfully received after the first transmission attempt by the other two HARQ schemes, while in the LTE HARQ scheme, a minimum of two transmissions are needed for recovering the source packet. Furthermore, the dashed curve seen in Figure 21 reveals the gap between the DCMC capacity and the throughput that these three HARQ schemes can achieve.

VI. Conclusions

HARQ schemes constitute efficient means of enhancing the transmission reliability of communication networks. When turbo codes are combined with HARQ, the achieved throughput may be significantly improved due to their near-capacity performance. However, the complexity of turbo coded HARQ schemes becomes a critical issue, since iterative decoding may be performed following each transmission. Observe from the expression of $Complexity = 2^m \cdot K$ that the memory length of the polynomial m has an exponential impact on the complexity. Therefore, the lowest-memory $m = 1$ polynomial $(2, 3)_o$ becomes the best potential choice for low-complexity HARQ schemes. In this paper, MCTCs have been demonstrated to attain the desirable performance based on this $(2, 3)_o$ minimum-memory polynomial. We also considered the impact of the other complexity related factor, namely that of K in order to reduce the complexity of MCTC aided HARQ schemes.

Two complexity reduction strategies have been proposed. The ES strategy of Figure 16 may curtail the iterative decoding process after each transmission attempt, depending on the MI increment provided. By contrast, the DI strategy delays the commencement of iterative decoding until sufficient MI has been received for ensuring successful decoding convergence.

More specifically, the ES strategy introduces two stopping thresholds, namely T_{dump} and T_{conv} of Figure 16 which were determined by an offline training process. Based on the thresholds found by the offline training, the simulation results have shown that the proposed ES strategy is capable of reducing the complexity by as much as 85% compared to Souza's systematic TCTC aided HARQ scheme using a fixed number of 10 BCJR operations, which was achieved without degrading the PLR and throughput performance.

The DI strategy of Figure 16 defers the commencement of iterations by estimating the emergence of an open EXIT tunnel in the i-component MCTC decoder, which is constructed for all i received replicas by the i^{th} transmission. This led to the creation of the LUT seen in Figure 19 for storing the MI thresholds for all possible combinations of received MIs. We conceived an efficient LUT structure and employed multiple-dimensional linear interpolation for minimizing the storage requirements. Then, an efficient training process was proposed for generating the LUT of MCTC HARQ schemes. The resultant LUT associated with the step size of $G = 0.1$ only has a few hundred rows for all $T_2, T_3, \cdots, T_7$ sub-tables. The simulation results revealed that the DI strategy of Figure 16 is capable of further reducing the complexity imposed by as much as 10% to 50%, yet maintaining a nearly unaffected PLR and throughput performance.

In this paper, the simulations characterizing our ES and DI strategies were based on the assumption of BPSK modulation and a quasi-static channel. However, an attractive performance may be expected both for arbitrary modulation schemes and for arbitrary channels, since both the ES and DI strategies exploit the MI, which can be calculated regardless of the specific modulation schemes and channel types. For example, the received MIs may be $\{0.15, 0.15, 0.16, 0.17\}$ during four consecutive transmissions, which have similar MIs for transmissions over a Rician slow fading channel. The threshold MI corresponding to the previous three MIs of $\{0.15, 0.15, 0.16\}$ is 0.68. According to our DI strategy, using the condition of $0.17 < 0.68$ suggests that we expect a closed EXIT tunnel for these four transmissions.

Furthermore, the implementation of our ES and DI strategies based on the MCTCs requires a modest amount of extra storage and additional calculations, such as for example, more interleaving operations, the calculations of MI and the storage of the LUT. However, they impose a lower complexity than the recursive calculations required for the BCJR decoder. The complexity reductions become especially significant, when the packet length N is high.

Although the ES and DI strategies were invoked in the context of MCTC aided HARQ schemes in this paper, both of them are general concepts. They may be extended into arbitrary scenarios employing turbo codes. For example, cooperative relay networks relying on distributed turbo codes may adopt the ES and DI strategies for improving their power efficiency.

References

[1] L. Hanzo, T. H. Liew, B. L. Yeap, R. Tee, and S. X. Ng, *Turbo coding, turbo equalisation and space-time coding for transmission over fading channels, second edition*. New York, USA: John Wiley & Sons, 2011.

[2] R. Comroe and J. Costello, D., "Arq schemes for data transmission in mobile radio systems," *IEEE J. Sel. Areas Commun.*, vol. 2, no. 4, pp. 472 –481, July 1984.

[3] R. Hamming, "Error dectecing and error correcting codes," *Bell System Technical Journal*, vol. 29, pp. 147–160, April 1950.

[4] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near shannon limit error-correcting coding and decoding: turbo-codes. 1," *Proc. IEEE International Conference on Communications (ICC 1993)*, vol. 2, pp. 1064–1072, May 1993.

[5] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate (Corresp.)," *IEEE Trans. Inf. Theory*, no. 2, pp. 284–287, March 1974.

[6] L. Hanzo, J. P. Woodard, and P. Robertson, "Turbo Decoding and Detection for Wireless Applications," *Proc. IEEE*, vol. 95, no. 6, June 2007.

[7] P. Robertson, E. Villebrun, and P. Hoeher, "A comparison of optimal and sub-optimal MAP decoding algorithms operating in the log domain," in *IEEE International Conference on Communications, 1995. ICC'95 Seattle*, vol. 2, June 1995, pp. 1009 –1013 vol.2.

[8] S.-J. Park, "Combined Max-Log-MAP and Log-MAP of turbo codes," *Electronics Letters*, vol. 40, no. 4, pp. 251 – 252, February 2004.

[9] L. Hanzo, R. G. Maunder, J. Wang, and L. L. Yang, *Near-capacity variable-length coding: regular and EXIT-chart-aided irregular designs*. John Wiley & Sons, 2011.

[10] S. Kim, J. Chang, and M. H. Lee, "Simple iterative decoding stop criterion for wireless packet transmission," *Electronics Letters*, vol. 36, no. 24, pp. 2026 –2027, November 2000.

[11] F.-M. Li and A.-Y. Wu, "On the new stopping criteria of iterative turbo decoding by using decoding threshold," *IEEE Trans. Signal Process.*, vol. 55, no. 11, pp. 5506 –5516, November 2007.

[12] M. Moher, "Decoding via cross-entropy minimization," in *Technical Program Conference Record of the IEEE Global Telecommunications Conference, including a Communications Theory Mini-Conference. (GLOBECOM 1993)*, vol. 2, November 1993, pp. 809–813.

[13] J. Hagenauer, E. Offer, and L. Papke, "Iterative decoding of binary block and convolutional codes," *IEEE Trans. Inf. Theory*, vol. 42, no. 2, pp. 429 –445, March 1996.

[14] J. Chen, L. Zhang, and J. Qin, “Average-entropy variation in iterative decoding of turbo codes and its application,” *Electronics Letters*, vol. 44, no. 22, pp. 1314 –1315, 23 2008.

[15] R. Y. Shao, S. Lin, and M. P. C. Fossorier, “Two simple stopping criteria for turbo decoding,” *IEEE Trans. Commun.*, vol. 47, no. 8, pp. 1117 – 1120, August 1999.

[16] Y. Wu, B. Woerner, and W. Ebel, “A simple stopping criterion for turbo decoding,” *IEEE Commun. Lett.*, vol. 4, no. 8, pp. 258 –260, August 2000.

[17] N. Y. Yu, M. G. Kim, Y. S. Kim, and S. U. Chung, “Efficient stopping criterion for iterative decoding of turbo codes,” *Electronics Letters*, vol. 39, no. 1, pp. 73 – 75, January 2003.

[18] F. Q. Zhai and I. J. Fair, “Techniques for early stopping and error detection in turbo decoding,” *IEEE Trans. Commun.*, vol. 51, no. 10, pp. 1617 – 1623, October 2003.

[19] A. Taffin, “Generalised stopping criterion for iterative decoders,” *Electronics Letters*, vol. 39, no. 13, pp. 993 – 994, June 2003.

[20] T. Ngatched and F. Takawira, “Simple stopping criterion for turbo decoding,” *Electronics Letters*, vol. 37, no. 22, pp. 1350 –1351, October 2001.

[21] A. Shibutani, H. Suda, and F. Adachi, “Reducing average number of turbo decoding iterations,” *Electronics Letters*, vol. 35, no. 9, pp. 701 –702, April 1999.

[22] Z. Ma, W. H. Mow, and P. Fan, “On the complexity reduction of turbo decoding for wideband cdma,” *IEEE Trans. Wireless Commun.,*, vol. 4, no. 2, pp. 353 – 356, March 2005.

[23] G. T. Chen, L. Cao, L. Yu, and C. W. Chen, “An efficient stopping criterion for turbo product codes,” *IEEE Commun. Lett.*, vol. 11, no. 6, pp. 525 –527, June 2007.

[24] W. Zhanji, P. Mugen, and W. Wenbo, “A new parity-check stopping criterion for turbo decoding,” *IEEE Commun. Lett.*, vol. 12, no. 4, pp. 304 –306, April 2008.

[25] M. Rovini and A. Martinez, “Efficient stopping rule for turbo decoders,” *Electronics Letters*, vol. 42, no. 4, pp. 235 – 236, February 2006.

[26] D. H. Kim and S. Kim, “Bit-level stopping of turbo decoding,” *IEEE Commun. Lett.*, vol. 10, no. 3, pp. 183 – 185, March 2006.

[27] J. Wu, Z. Wang, and B. Vojcic, “Partial iterative decoding for binary turbo codes via cross-entropy based bit selection,” *IEEE Trans. Communications*, vol. 57, no. 11, pp. 3298 –3306, November 2009.

[28] W. Shao and L. Brackenbury, “Early stopping turbo decoders: a high-throughput, low-energy bit-level approach and implementation,” *Communications, IET*, vol. 4, no. 17, pp. 2115 –2124, November 2010.

[29] M. W. Williard, “Introduction to redundancy coding,” *IEEE Trans. Veh. Technol.*, vol. 27, no. 3, pp. 86–98, August 1978.

[30] E. Y. Rocher and R. L. Pickholtz, “An analysis of the effectiveness of hybrid transmission schemes,” *IBM Journal of Research and Development*, vol. 14, no. 4, pp. 426 –433, July 1970.

[31] H. O. Burton and D. D. Sullivan, “Errors and error control,” *Proc. IEEE*, vol. 60, no. 11, pp. 1293–1301, November 1972.

[32] S. Lin and P. Yu, “A hybrid ARQ scheme with parity retransmission for error control of satellite channels,” *IEEE Trans. Commun.*, vol. 30, no. 7, pp. 1701–1719, July 1982.

[33] A. Sastry, “Performance of hybrid error control schemes of satellite channels,” *IEEE Trans. Commun.*, vol. 23, no. 7, pp. 689–694, July 1975.

[34] A. Sastry and L. Kanal, “Hybrid error control using retransmission and generalized burst-trapping codes,” *IEEE Trans. Commun.*, vol. 24, no. 4, pp. 385–393, April 1976.

[35] C. Fujiwara, M. Kasahara, K. Yamashita, and T. Namekawa, “Evaluations of error control techniques in both independent-error and dependent-error channels,” *IEEE Trans. Commun.*, vol. 26, no. 6, pp. 785 – 794, June 1978.

[36] A. Drukarev and J. D. J. Costello, “A comparison of block and convolutional codes in ARQ error control schemes,” *IEEE Trans. Commun.*, vol. 30, no. 11, pp. 2449–2455, November 1982.

[37] S. Kallel and D. Haccoun, “Generalized type II hybrid ARQ scheme using punctured convolutional coding,” *IEEE Trans. Commun.*, vol. 38, no. 11, pp. 1938 –1946, November 1990.

[38] Q. Zhang and S. Kassam, “Hybrid ARQ with selective combining for fading channels,” *IEEE J. Sel. Areas Commun.*, vol. 17, no. 5, pp. 867 –880, May 1999.

[39] J. Hagenauer, “Rate-compatible punctured convolutional codes (RCPC codes) and their applications,” *IEEE Trans. Commun.*, vol. 36, no. 4, pp. 389 –400, April 1988.

[40] S. Kallel, “Complementary punctured convolutional (CPC) codes and their applications,” *IEEE Trans. Commun.*, vol. 43, no. 6, pp. 2005–2009, June 1995.

[41] Q. C. Chen and P. Z. Fan, “On the performance of type-III hybrid ARQ with RCPC codes,” in *Proc. 14th IEEE on Personal, Indoor and Mobile Radio Communications (PIMRC 2003)*, vol. 2, September 2003, pp. 1297–1301.

[42] Y. F. Wang, L. Zhang, and D. C. Yang, “Performance analysis of type III HARQ with turbo codes,” in *Proc. IEEE Vehicular Technology Conference (VTC 2003-Spring)*, vol. 4. IEEE, April 2003, pp. 2740–2744.

[43] C. X. Wang, H. G. Ryu, H. H. Chen, and Y. He, “Hybrid ARQ with rate adaptation in multiband OFDM UWB systems,” *Proc. IEEE International Conference on Communications (ICC 2009)*, pp. 1–5, June 2009.

[44] H. Krishna and S. Morgera, “A new error control scheme for hybrid ARQ systems,” *IEEE Trans. Commun.*, vol. 35, no. 10, pp. 981 – 990, October 1987.

[45] S. Morgera and V. Oduol, “Soft-decision decoding applied to the generalized type-II hybrid ARQ scheme,” *IEEE Trans. Commun.*, vol. 37, no. 4, pp. 393 –396, April 1989.

[46] M. Kousa and M. Rahman, “An adaptive error control system using hybrid ARQ schemes,” *IEEE Trans. Commun.*, vol. 39, no. 7, pp. 1049 –1057, July 1991.

[47] M. Rice, “Comparative analysis of two realizations for hybrid-ARQ error control,” in *Proc. IEEE Global Communications Conference, (GLOBECOM 1994)*, December 1994, pp. 115 –119.

[48] P. Coulton, C. Tanriover, B. Wright, and B. Honary, “Simple hybrid type II ARQ technique using soft output information,” *Electronics Letters*, vol. 36, no. 20, pp. 1716 –1717, September 2000.

[49] M. Buckley and S. Wicker, “The design and performance of a neural network for predicting turbo decoding error with application to hybrid ARQ protocols,” *IEEE Trans. Commun.*, vol. 48, no. 4, pp. 566 –576, April 2000.

[50] S. Choi and K. Shin, “A class of adaptive hybrid ARQ schemes for wireless links,” *IEEE Trans. Veh. Technol.*, vol. 50, no. 3, pp. 777 –790, May 2001.

[51] B. Harvey, “Adaptive viterbi algorithm with ARQ for bounded complexity decoding,” *IEEE Trans. Wireless Commun.*, vol. 3, no. 6, pp. 1948 – 1952, November 2004.

[52] L. Cao and T. Shi, “Turbo codes based hybrid ARQ with segment selective repeat,” *Electronics Letters*, vol. 40, no. 18, pp. 1140 – 1141, September 2004.

[53] I. D. Holland, H. J. Zepernick, and M. Caldera, “Soft combining for hybrid ARQ,” *IET Electronics Letters*, vol. 41, no. 22, pp. 1230–1231, October 2005.

[54] X. Zhang and Q. Du, “Adaptive low-complexity erasure-correcting code-based protocols for QoS-driven mobile multicast services over wireless networks,” *IEEE Trans. Veh. Technol.*, vol. 55, no. 5, pp. 1633 –1647, September 2006.

[55] K. Oteng-Amoako and S. Nooshabadi, “Asymmetric rate compatible turbo codes in hybrid automatic repeat request schemes,” *IEE Proc. Commun.,*, vol. 153, no. 5, pp. 603 –610, October 2006.

[56] F. Chiti and R. Fantacci, “A soft combining hybrid-ARQ technique applied to throughput maximization within 3g satellite ip networks,” *IEEE Trans. Veh. Technol.*, vol. 56, no. 2, pp. 594 –604, March 2007.

[57] J. Heo, S. Kim, J. Kim, and J. Kim, “Low complexity decoding for raptor codes for hybrid-ARQ systems,” *IEEE Trans. Consum. Electron.*, vol. 54, no. 2, pp. 390 –395, May 2008.

[58] B. Mielczarek and W. Krzymien, “Adaptive hybrid ARQ systems with BCJR decoding,” *IEEE Trans. Veh. Technol.*, vol. 57, no. 3, pp. 1606 –1619, May 2008.

[59] J. Fricke and P. Hoeher, “Reliability-based retransmission criteria for hybrid ARQ,” *IEEE Trans. Commun.*, vol. 57, no. 8, pp. 2181 –2184, August 2009.

[60] D. Chase, “A combined coding and modulation approach for communication over dispersive channels,” *IEEE Trans. Commun.*, vol. 21, no. 3, pp. 159–174, March 1973.

[61] D. Mandelbaum, “An adaptive-feedback coding scheme using incremental redundancy (corresp.),” *IEEE Trans. Inf. Theory*, vol. 20, no. 3, pp. 388–389, May 1974.

[62] J. N. Laneman, D. N. C. Tse, and G. W. Wornell, “Cooperative diversity in wireless networks: Efficient protocols and outage behavior,” *IEEE Trans. Inf. Theory*, vol. 50, no. 12, pp. 3062–3080, Dec. 2004.

[63] R. H. Liu, P. Spasojevic, and E. Soljanin, “Incremental redundancy cooperative coding for wireless networks: Cooperative diversity, coding, and transmission energy gains,” *IEEE Trans. Inf. Theory*, vol. 54, no. 3, pp. 1207–1224, March 2008.

[64] E. Uhlemann, T. M. Aulin, L. K. Rasmussen, and P. A. Wiberg, “Packet combining and doping in concatenated hybrid ARQ schemes

using iterative decoding," in *Proc. IEEE Wireless Communications and Networking (WCNC 2003)*, vol. 2, March 2003, pp. 849–854.

[65] 3GPP, "High speed downlink packet access: Physical layer aspects," TR 25.858 V5.0.0, Tech. Rep., March 2002.

[66] K. R. Narayanan and G. L. Stuber, "A novel ARQ technique using the turbo coding principle," *IEEE Commun. Lett.*, vol. 1, no. 2, pp. 49–51, March 1997.

[67] R. D. Souza, M. E. Pellenz, and T. Rodrigues, "Hybrid ARQ scheme based on recursive convolutional codes and turbo decoding," *IEEE Trans. Commun.*, vol. 57, no. 2, pp. 315–318, February 2009.

[68] T. Ait-Idir, H. Chafnaji, and S. Saoudi, "Turbo packet combining for broadband space-time BICM hybrid-ARQ systems with co-channel interference," *IEEE Trans. Wireless Commun.*, vol. 9, no. 5, pp. 1686 –1697, May 2010.

[69] D. Divsalar, S. Dolinar, and F. Pollara, "Serial concatenated trellis coded modulation with rate-1 inner code," in *Proceeding of the IEEE Global Telecommunications Conference (GLOBECOM 2000)*, vol. 2, November 2000, pp. 777–782.

[70] "3rd Generation Partnership Project; Technical specification group radio access network; Evolved Universal Terrestrial Radio Access (E-UTRA); Multiplexing and channel coding (Release 10)," 3GPP TS 36.212, December, 2010, Downloadable at http://www.3gpp.org/FTP/Specs/html-info/26346.htm.

[71] H. Chen, R. G. Maunder, and L. Hanzo, "Multi-level turbo decoding assisted soft combining aided hybrid ARQ," in *Proc. IEEE Vehicular Technology Conference (VTC 2010-Spring)*, May 2010.

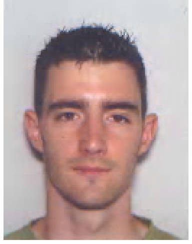
[72] S. ten Brink, "Convergence behavior of iteratively decoded parallel concatenated codes," *IEEE Trans. Commun.*, vol. 49, no. 10, pp. 1727–1737, October 2001.

[73] J. Hagenauer, "The EXIT chart - introduction to extrinsic information transfer in iterative processing," in *Proc. 12th European Signal Processing Conference (EUSIPCO)*, September 2004, pp. 1541–1548.

[74] S. ten Brink, G. Kramer, and A. Ashikhmin, "Design of low-density parity-check codes for modulation and detection," *IEEE Trans. Commun.*, vol. 52, no. 4, pp. 670 – 678, April 2004.

[75] J. W. Lee and R. E. Blahut, "Generalized EXIT chart and BER analysis of finite-length turbo codes," in *Proc. IEEE Global Communications Conference, (GLOBECOM 2003)*, vol. 4, December 2003, pp. 2067 – 2072.

Hong Chen received her Master Degree in computer science from University of Electronic Science and Technology, China in 2000. Since then, she has been a lecturer at UESTC. In November 2007, she started her PhD study in the Communications Research Group, School of Electronics and Computer Science, University of Southampton, U.K. Her current research interests include cross-layer optimization of wireless networks, turbo coding aided hybrid ARQ. Currently she is a postdoctoral research fellow at the University of Surrey, UK.



Robert G. Maunder (MIEEE'03) has studied with the School of Electronics and Computer Science, University of Southampton, UK, since October 2000. He was awarded a first class honours BEng in Electronic Engineering in July 2003, as well as a Ph.D. in Wireless Communications and a lectureship in December 2007, all from University of Southampton, UK. His research interests include video coding, joint source/channel coding and iterative decoding. He has published a number of IEEE papers in these areas.

Lajos Hanzo (http://www-mobile.ecs.soton.ac.uk) FREng, FIEEE, FIET, Fellow of EURASIP, DSc received his degree in electronics in 1976 and his doctorate in 1983. In 2009 he was awarded the honorary doctorate "Doctor Honaris Causa" by the Technical University of Budapest. During his 35-year career in telecommunications he has held various research and academic posts in Hungary, Germany and the UK. Since 1986 he has been with the School of Electronics and Computer Science, University of Southampton, UK, where he holds the chair in telecommunications. He has successfully supervised in excess of 70 PhD students, co-authored 20 John Wiley/IEEE Press books on mobile radio communications totalling in excess of 10 000 pages, published 1250+ research entries at IEEE Xplore, acted both as TPC and General Chair of IEEE conferences, presented keynote lectures and has been awarded a number of distinctions. Currently he is directing an academic research team, working on a range of research projects in the field of wireless multimedia communications sponsored by industry, the Engineering and Physical Sciences Research Council (EPSRC) UK, the European IST Programme and the Mobile Virtual Centre of Excellence (VCE), UK. He is an enthusiastic supporter of industrial and academic liaison and he offers a range of industrial courses. He is also a Governor of the IEEE VTS. Since 2008 he has been the Editor-in-Chief of the IEEE Press and since 2009 a Chaired Professor also at Tsinghua University, Beijing. For further information on research in progress and associated publications please refer to http://www-mobile.ecs.soton.ac.uk