Multi-Level Turbo Decoding Assisted Soft Combining Aided Hybrid ARQ

H. Chen, R. G. Maunder and L. Hanzo
School of ECS, University of Southampton, SO17 1BJ, UK.
Tel: +44-23-8059 3125, Fax: +44-23-8059 4508
Email: {hc07r,rm,lh}@ecs.soton.ac.uk; http://www-mobile.ecs.soton.ac.uk

***Abstract*—[1]Hybrid Automatic Repeat reQuest (ARQ) plays an essential role in error control. Combining the incorrectly received packet replicas in hybrid ARQ has been shown to reduce the resultant error probability, while improving the achievable throughput. Hence, in this contribution, multi-level turbo codes have been amalgamated both with hybrid ARQ and efficient soft combining techniques for taking into account the Log-Likelihood Ratios (LLRs) of retransmitted packet replicas. In this paper, we present a soft combining aided hybrid ARQ scheme based on multi-level turbo codes, which avoid the capacity loss of the twin-level turbo codes that are typically employed in hybrid ARQ schemes. More specifically, the proposed receiver dynamically appends an additional parallel concatenated Bahl, Cocke, Jelinek and Raviv (BCJR) algorithm based decoder in order to fully exploit each retransmission, thereby forming a multi-level turbo decoder. Therefore, all the extrinsic information acquired during the previous BCJR operations will be used as *a priori* information by the additional BCJR decoders, whilst their soft output iteratively enhances the *a posteriori* information generated by the previous decoding stages. We also present link-level Packet Loss Ratio (PLR) and throughput results, which demonstrate that our scheme outperforms some of the previously proposed benchmarks.**

I. Introduction

Hybrid Automatic Repeat reQuest (ARQ) [1], [2], [3] plays an essential role in data communication systems, incorporating Forward Error Correction (FEC). Type-I Hybrid ARQ (HARQ) is based on the straightforward retransmission of the FEC coded packets that cannot be perfectly recovered, while type-II HARQ is capable of achieving an increased throughput by transmitting more and more of the previously punctured parity information during each retransmission. When using turbo codes [4], the concept of type-III HARQ was created. Like in type-II HARQ, type-III HARQ uses different redundant information during each transmission attempt, but each of them is self-decodable. The authors of [5], [6] characterized the attainable performance of type-III HARQ schemes. During the evolution of HARQ schemes, minimizing the required number of retransmissions has received a significant research attention, because unnecessary retransmissions reduce the effective throughput. A typical approach has been that of combining the various corrupted retransmission components in order to provide a more reliable decision for the original bits. Two main combining strategies have been proposed, namely Chase combining [7] and the transmission of incremental redundancy [8]. Chase combining achieves diversity gain by beneficially combining the identical data replicas conveyed during the different retransmissions. By contrast, incremental redundancy 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 [9], [10].

Soft decision aided Chase combining and incremental redundancy based techniques have been proposed for HARQ schemes that use iterative soft-decision-based FEC decoders [11], [12], like turbo codes. For example, the High Speed Downlink Packet Access (HSDPA) protocol [13] uses a punctured $R = 1/3$-rate turbo code as the basis of its HARQ scheme. Here, whenever a retransmission is received, the corresponding Logarithmic Likelihood Ratios (LLRs) will be added to those that were recovered from previous transmissions or used to provide soft information for bits that were punctured during previous transmissions. Similarly, [2], [11] and [12] used the LLRs obtained from previous transmissions as *a priori* values during the decoding of the retransmissions. In a recent paper by Souza *et al.* [3] proposed a HARQ scheme that integrates Chase combining and incremental redundancy in a twin-level turbo code. Here the LLRs obtained from each replica of a packet are added and iterative decoding is employed to recover the transmitted data.

In contrast to the twin-level turbo code of [3], multi-level turbo codes employ a parallel concatenation of more than two component codes, which are combined by iterative decoding at the receiver. In general, an N-level turbo code having an overall rate R can be interpreted as a parallel concatenation of N component codes, each having a coding rate of RN. The area properties[2] of EXtrinsic Information Transfer (EXIT) charts [14] suggest that a turbo code will suffer a capacity loss, if the coding rate RN of the component codes is less than unity [14, Section VIII]. **Therefore, a turbo code having an overall coding rate of R should employ $N = 1/R$ levels of coding in order to achieve $RN = 1$ and hence to avoid the above-mentioned capacity loss. It is this principle that motivates the design of our HARQ scheme, which constitutes the novel contribution of the paper.** As in Souza's scheme, the overall rate R of our scheme decreases to $1/2$, $1/3$, $1/4$ and so on with each subsequent retransmission. However, in our scheme the number of component codes N combined by the iterative decoder is accordingly increased to 2, 3, 4 and so on with each retransmission, therefore allowing us to maintain $RN = 1$ and hence avoiding the above-mentioned capacity loss. This is in contrast to Souza's scheme, which only ever employs $N = 2$ levels and therefore suffers from a capacity loss, when R drops below $1/2$.

The rest of this paper is organized as follows. The system model and the proposed HARQ scheme are presented in Section II, where several different benchmarker HARQ schemes

[1]The financial support of the China-UK Scholarship Council, of the EPSRC, UK and of the EU under the auspices of the Optimix Project is gratefully acknowledgement.

[2]In simple terms, the area property states that the area under the outer decoder's EXIT curve is given by the code-rate, while the area between the outer and inner decoder's curves is related to the distance from the channel capacity.

978-1-4244-2519-8/10/$26.00 ©2010 IEEE

TABLE I
SUMMARY OF THE INFORMATION TRANSMITTED BY THE FOUR HYBRID ARQ SCHEMES.

Tx Number	Souza's scheme	Puncturing-aided Souza-scheme	Fixed 3-level scheme	Proposed scheme
$1st$ Tx	x_n	x_n	x_n	x_n
$2st$ Tx	x_{p1}	punc(x_{p1}, x_{p2})	punc(x_{p1}, x_{p2})	punc(x_{p1}, x_{p2})
$3st$ Tx	x_{p2}	x_{p1}	x_{p3}	x_{p3}
$4st$ Tx	x_n	x_{p2}	x_{p1}	x_{p4}
$5st$ Tx	x_{p1}	x_{p1}	x_{p2}	x_{p5}
$6st$ Tx	x_{p2}	x_{p2}	x_{p3}	x_{p6}

are also highlighted. Section III discusses our Packet Loss Ratio (PLR), throughput and complexity results. Drawing on these results, Section IV concludes the paper.

II. SYSTEM MODEL

In this section, we will describe four different HARQ schemes. Before introducing the proposed multi-level turbo code scheme, we describe both the original and a puncturing-aided version of Souza's scheme[3] [3], both of which use a twin-level turbo code. We also describe a third benchmarker, which employs a three-level turbo code in order to allow the investigation of an intermediate design between Souza's scheme and our own.

Each of the four HARQ schemes employs a simple stop-and-wait ARQ protocol, which appends an $(n-k)$-bit Cyclic Redundancy Check (CRC) to the k-bit message $\mathbf{u} = [u_1, u_2, ..., u_k]$ in order to facilitate reliable error detection. The resultant n-bit packet $\mathbf{u_n}$ is input to the corresponding FEC scheme. Each of the four HARQ schemes conveys the systematic bits $\mathbf{x_n} = \mathbf{u_n}$ during the first transmission, in order to achieve the maximal throughput, when the channel is benign. In all cases, Binary Phase Shift Keying (BPSK) is used. If the hard-decision CRC detection fails at the receiver, it will send back a Negative ACKnowledgement (NACK) or will simply wait for the transmitter's timeout to trigger a retransmission. In this case, the FEC scheme generates the information to be retransmitted using a particular parallel concatenated Unity Rate Code (URC) [15] having the octally represented generator polynomials of $(2, 3)$, as it will be detailed in the context of Figures 1 and 3. The BCJR algorithm [16] is employed to facilitate iterative decoding. Retransmissions are continually generated, until the hard-decision based CRC suggests error-free detection or until a maximum of $M = 6$ transmissions have been sent, at which point, the packet will be discarded.

The main difference between the four HARQ schemes is in the choice of the particular parallel concatenation of the component URCs that they employ. Table I summarizes the information conveyed by each transmission in these four HARQ schemes, where x_{pi} refers to the parity bits generated by the ith parallel concatenated URC encoder. Note that independent pseudo-random interleavers are employed to ensure that each parallel concatenated URC encoder considers a different ordering of the bit sequence $\mathbf{u_n}$.

A. Souza's scheme

Figure 1 depicts the configuration of the parallel concatenated URC codes employed by Souza's scheme [3]. In simple terms, Souza's scheme transmits x_n, x_{p1}, x_{p2}, x_n, x_{p1} and x_{p2} during the $M = 6$ transmissions, as contrasted to the other schemes in Table I. In Souza's original proposal, (x_n, x_{p1}) are transmitted together during the first transmission. However, in order to maintain the same coding rate in the four HARQ schemes, we separate this first copy into two transmissions in order to attain the maximal throughput, when the channel Signal Noise Ratio (SNR) is sufficiently high.

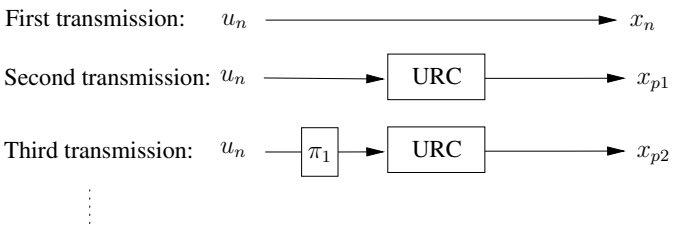

Fig. 1. The configuration of the parallel concatenated URC codes in Souza's scheme.

At the receiver, we let $y_n^{(j)}$ and $y_{pi}^{(j)}$ represent the LLRs of the received packets corresponding to the transmitted x_n and x_{pi}, where the superscript j denotes the jth repetition packet, for example, y_n^1 and y_n^2 indicate the systematic LLRs during the first and fourth transmission, respectively. Hard-decision based error detection is applied to the systematic bits recovered from the systematic LLRs $y_n^{(1)}$ during the first transmission. For the parity LLRs $y_{p1}^{(1)}$ gleaned from the second transmission, only one BCJR operation is carried out, using the previous systematic LLRs $y_n^{(1)}$ as the *a priori* input. Once the third transmission has been received, a twin-level turbo decoder is constructed and iterative decoding starts. From then on, the newly received repetitions of the packets owing to later retransmissions are added, as seen in Figure 2. More specifically, the fourth received y_n^2 is added to y_n^1, the fifth received y_{p1}^2 is added to y_{p1}^1 and so on. Once the turbo decoder has been constructed, up to five decoding iterations are performed following the reception of each packet, each comprising two BCJR URC decoding operations, as it will be detailed in the next section. By contrast, a reduced number of decoding iterations are performed, if the process converges or if a BCJR operation produces *a posteriori* LLRs that result in a legitimate codeword, hence satisfying the CRC, in which case the hard decision bits are output and an Acknowledgement (ACK) flag is returned to the transmitter. The extrinsic information obtained following the reception of the previous retransmission is used as *a priori* information in order to initialize the iterative decoding process.

[3]This puncturing-aided scheme is introduced for the sake of direct comparability with the proposed design.

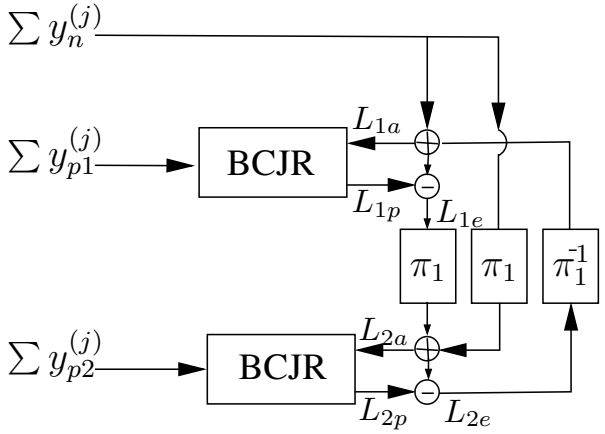

Fig. 2. The decoder structure of Souza's scheme after six transmissions.

B. Our proposed scheme

Figure 3 illustrates the different transmissions generated by our proposed scheme. In contrast to Souza's original scheme, where the bits encoded by a rate-$1/2$ Recursive Systematic Convolutional (RSC) code are transmitted during the first transmission, our scheme initially transmits only the systematic bits x_n in order to achieve the maximal throughput, when the channel SNR is sufficiently high. The second transmission is generated by puncturing the encoded output bits of $N = 2$ URC encoders to generate exactly the same number of bits, as during the first transmission, as seen in Figure 3. This approach achieves a coding rate of $R = 1/2$ after two transmissions, maintaining $RN = 1$ and facilitating iterative decoding. Similarly, during the subsequent retransmissions, different interleavers and additional URCs are employed to incrementally generate further URC-encoded bits and to maintain $RN = 1$.

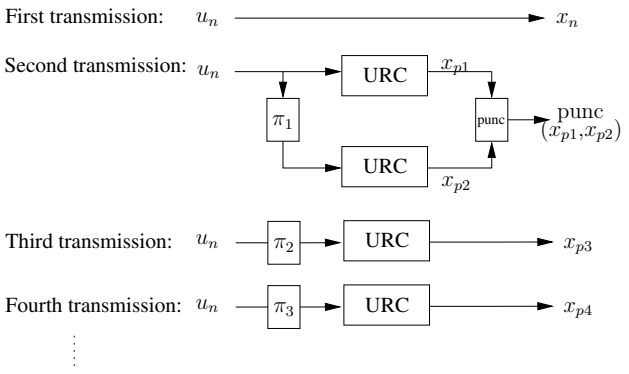

Fig. 3. Encoder in different retransmissions.

At the receiver, a multi-level turbo code is constructed. Following the second transmission characterized in Figure 3, an initial twin-level turbo code is formed and iterative decoding commences. Thereafter, a BCJR decoder is activated upon the reception of each transmitted packet, hence increasing the number of levels in the turbo decoder. In our HARQ scheme, the *a priori* input provided for each newly activated BCJR decoder is generated as the sum of all the extrinsic information contributions obtained from decoding the previous transmissions, as well as the interleaved systematic information obtained during the first transmission. In return, the extrinsic LLRs L_e of the new BCJR decoder are passed back to aid the other BCJR decoders. Figure 4 shows the decoder's structure. As in Souza's scheme seen in Figure 2, up to ten BCJR operations are performed following the reception of each transmission, ensuring that both schemes have the same complexity.

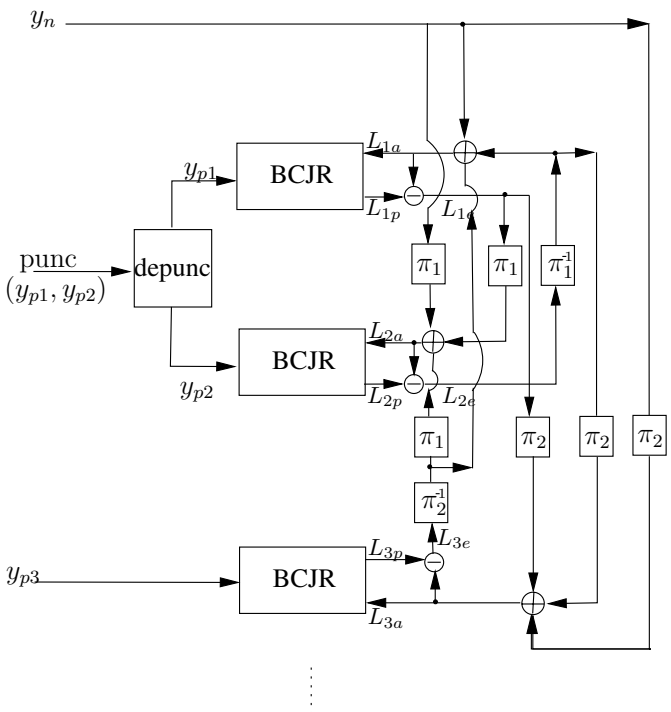

Fig. 4. Decoder structure after six transmissions.

In detail, for the systematic bits received during the first transmission, only CRC decoding is carried out. If any bit errors have been detected, the receiver requests the second transmission. The second transmission contains the punctured encoded bits of the first two URC codes, hence depuncturing is employed to reinsert the punctured bits and therefore to provide the soft input for the two BCJR decoders respectively, while the systematic LLRs obtained during the first transmission are employed as the *a priori* input. Again, the depuncturing operation reinserts the punctured bits of y_{p1} and y_{p2}, both of which initially have zero-valued LLRs. At this point, iterative decoding commences. If this iterative decoding process fails to obtain an *a posteriori* output that satisfies the CRC, the third transmission will be requested. In Figure 4, L_{1e}, L_{2e} represent the extrinsic information obtained by the first two BCJR decoders. These are added to the systematic LLRs received from the first transmission and employed as the *a priori* input for the third BCJR decoder. At this point, iterative decoding recommences, starting with the third BCJR decoder, whose extrinsic LLRs are represented by L_{3e}. This is added to the extrinsic LLRs obtained by the second BCJR decoder and the systematic LLRs in order to provide an *a priori* input for the first BCJR decoder. In this way, iterative decoding continues by exchanging extrinsic information among the three BCJR decoders. Likewise, if subsequent retransmissions are requested, further URC-encoded segments are appended and the LLR-addition operations will continue, until the packet becomes error-free or until the retransmission limit is reached.

C. The puncturing-aided Souza scheme and the fixed three-level scheme

Since there are a number of differences between the operation of Souza's scheme and our own, we additionally consider two further benchmarkers, which represent the intermediate steps between the two schemes. In contrast to Souza's scheme, our approach employs puncturing. For this reason, our first additional benchmarker resembles Souza's scheme, but with the addition of puncturing. More specifically, this benchmarker adopts a twin-level turbo code and Chase combining of the packet replicas, as in Souza's scheme, but the second transmission uses the punctured URC-encoded bits of the two URC encoders. From the third transmission on, the parity bits x_{p1}, x_{p2} are alternately transmitted, as shown in Table I. Correspondingly, depuncturing is employed at the receiver similar to our proposed scheme.

We additionally consider a benchmarker that employs a three-level turbo code, in order to investigate the intermediate solution between a twin- and a multi-level turbo code. As shown in Table I, this benchmarker consecutively transmits the parity bits x_{p1}, x_{p2}, x_{p3}. The schematic of the decoder designed for this benchmarker can be created by simply concatenating no more than three BCJR decoders in Figure 4 and including the sum of the LLRs extracted from the repeated packets.

Despite the above-mentioned differences, there are a number of similarities between the four HARQ schemes. Firstly, there is no need to restart the iterative decoding process for each new retransmission in any of the schemes. Instead, the process continues from the state reached during the iterative decoding process employed after the previous transmission was received. Secondly, all four schemes employ the same iterative decoding stopping criteria. Namely, as soon as any BCJR operation produces *a posteriori* information that satisfies the CRC, decoding may be concluded and the successful detection of the packet can be acknowledged. Furthermore, whenever the mutual information associated with the extrinsic LLRs fails to increase by more than 0.001 between two consecutive operations of the same BCJR decoder, the same action of curtailing further iterations is carried out. Finally, the maximum number of BCJR operations that is performed following the reception of each transmission is limited to ten, like in Souza's original scheme, regardless of the number of component decoders. Owing to this measure, all of our schemes are associated with the same computational complexity, allowing their fair and equitable comparison.

III. Performance Results

In this section, we compare the link-level PLR, throughput and complexity characteristics of the four schemes introduced in the Section II. Our simulations considered the transmission of a statistically relevant number of packets, each comprising 256 bytes, over an uncorrelated Rayleigh fading channel. This packet length is more appropriate in network applications than the eight-times shorter 256-*bit* packets considered by Souza [3], which are disproportionately small compared to the length of the headers that are appended by the network protocols.

The link-level PLR versus SNR characteristics of the four schemes are shown in Figure 5. Here, a packet loss event occurs, whenever six transmissions are insufficient for the iterative decoding process to generate *a posteriori* information that satisfies the CRC. The results of Figure 5 show that for SNRs below -6.5dB, six transmissions are insufficient to allow packet reconstruction in any of the four schemes considered. However, the proposed scheme exhibits a significantly better performance than the three benchmarkers, when the channel SNR exceeds -6.5dB. More specifically, observed in Figure 5 that our scheme offers a steep turbo cliff at an SNR of -5.5dB, facilitating low PLRs in excess of this SNR. By contrast, the PLRs of the twin-level turbo code benchmarkers decrease more gradually, offering a near-unity PLR at an SNR of -6dB and a PLR of approximately 10^{-3} at an SNR of 2dB. As shown in Figure 5, the three-level turbo code benchmarker offers a PLR performance, which approaches that of our proposed scheme. This demonstrates that increasing the number of concatenated codes by even one may significantly enhance the attainable performance.

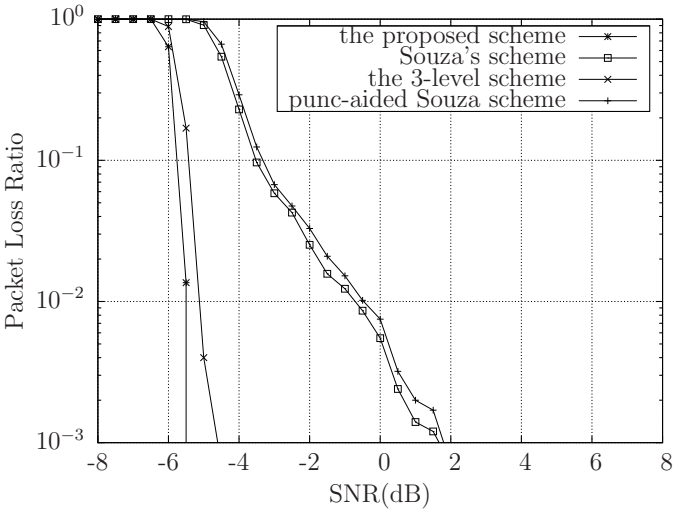

Fig. 5. Link-level packet loss ratio versus SNR for transmission over uncorrelated Rayleigh channels. The coding rate becomes $1/6$ after six transmissions. The packet length is 256 bytes.

Figure 6 shows the four schemes' throughput versus the SNR. Here, the normalized throughput is defined as the ratio of successfully recovered packets to the total number of transmitted packets. When the SNR is lower than -6dB, all schemes have a zero throughput, indicating that no messages are successfully recovered and that the PLR is 1, as shown in Figure 5. In the SNR range between -6dB and 0dB, our proposed scheme offers a 1.5dB to 2dB gain over the twin-level turbo code benchmarkers. There is however a significant throughput increase for SNRs between 1.5dB and 4dB for both our scheme and for the three-level scheme, both of which offer a normalized throughput of about 0.5 in this region, which is significantly higher than the 0.33 throughput offered by Souza's scheme. A similar trend may be observed for the puncturing-aided Souza scheme in this region, which in fact requires a 0.5dB lower SNR than our scheme and than the three-level scheme. While we do not show simulation results for very high SNR values, all schemes are capable of approaching the normalized throughput of 1 when the channel

is benign, because their initial transmissions are constituted by the original systematic information bits.

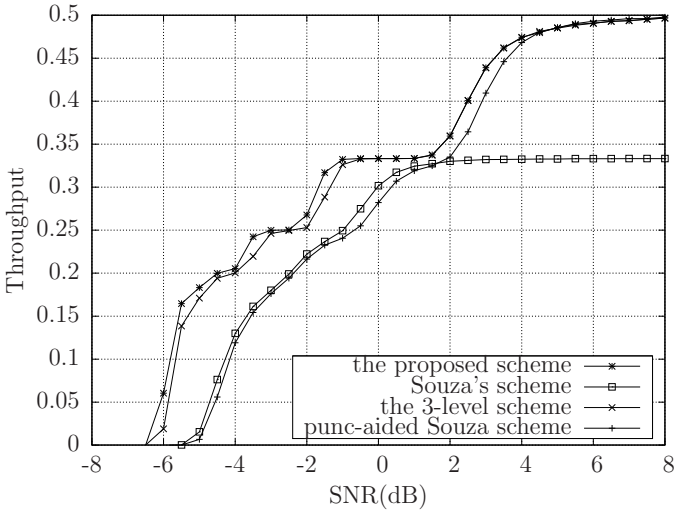

Fig. 6. Throughput versus SNR for transmission over uncorrelated Rayleigh channels. The packet length is 256 bytes.

Finally, we consider how the iterative decoding complexity of the four schemes varies with the channel SNR. In Figure 7, this complexity is quantified in terms of the average number of BCJR operations performed during the reconstruction of each original message. In general, the complexity of all of the four schemes peaks in the SNR region that corresponds to the 'turbo cliff'. For SNRs below -8dB, the reduced complexity is explained by the rapid convergence of the iterative decoding process, when the amount of information received is low. Similarly, the low complexity at high SNRs is explained by the rapid acquisition of *a posteriori* information that satisfies the CRC. Observe that for SNRs in the range of [-5dB, -2dB] and for SNRs in excess of 4dB, our proposed scheme offers the lowest complexity. In the other SNR regions, the proposed scheme does not have a significantly higher complexity than the benchmarkers. For this reason, the proposed scheme offers PLR and throughput advantages without any significant complexity increase.

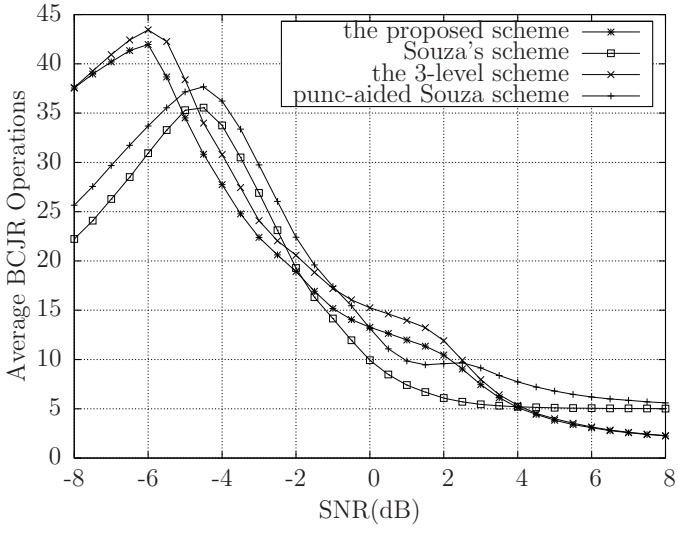

Fig. 7. Average number of BCJR operations versus SNR for transmission over uncorrelated Rayleigh channels. The packet length is 256 bytes.

IV. Conclusions

In this paper, we have proposed a HARQ scheme based on multi-level turbo codes. Our design was motivated by the area properties of EXIT charts. More specifically, this approach avoids the capacity loss that is associated with using HARQ combined with twin-level rather than multi-level turbo codes. Indeed, our simulation results have shown that the proposed approach outperforms Souza's scheme [3] in two aspects, namely in terms of its improved PLR and throughput, without having an increased computational complexity. However, our scheme requires the implementation of several interleavers. In situations where this is unattractive, our results have demonstrated that a three-level turbo code offers a significant gain over the existing techniques, at the cost of requiring only a single additional interleaver.

References

[1] S. Lin and P. Yu, "A hybrid ARQ scheme with parity retransmission for error control of satellite channels," *IEEE Transactions on Communications*, vol. 30, no. 7, pp. 1701–1719, Jul 1982.

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

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

[4] L. Hanzo, T. H. Liew, and B. Yeap, *Turbo Coding, Turbo Equalisation and Space-Time Coding for Transmission over Fading Channels*. JOHN WILEY & SONS, 2002.

[5] S. Kallel, "Complementary punctured convolutional (CPC) codes and their applications," *IEEE Transactions on Communications*, vol. 43, no. 6, pp. 2005–2009, June 1995.

[6] Q. Chen and P. 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, 7–10 Sept. 2003, pp. 1297–1301.

[7] D. Chase, "A combined coding and modulation approach for communication over dispersive channels," *IEEE Transactions on Communications*, vol. 21, no. 3, pp. 159–174, Mar 1973.

[8] D. Mandelbaum, "An adaptive-feedback coding scheme using incremental redundancy (corresp.)," *IEEE Transactions on Information Theory*, vol. 20, no. 3, pp. 388–389, May 1974.

[9] J. N. Laneman, D. N. C. Tse, and G. W. Wornell, "Cooperative diversity in wireless networks: Efficient protocols and outage behavior," *IEEE Transactions on Information Theory*, vol. 50, no. 12, pp. 3062–3080, Dec. 2004.

[10] R. Liu, P. Spasojevic, and E. Soljanin, "Incremental redundancy cooperative coding for wireless networks: Cooperative diversity, coding, and transmission energy gains," *IEEE Transactions on Information Theory*, vol. 54, no. 3, pp. 1207–1224, March 2008.

[11] 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, 20–20 March 2003, pp. 849–854.

[12] I. D. Holland, H. J. Zepernick, and M. Caldera, "Soft combining for hybrid ARQ," *Electronics Letters*, vol. 41, no. 22, pp. 1230–1231, 27 Oct. 2005.

[13] 3GPP, "High speed downlink packet access: Physical layer aspects," TR 25.858 V5.0.0, Tech. Rep., March 2002.

[14] A. Ashikhmin, G. Kramer, and S. ten Brink, "Extrinsic information transfer functions: model and erasure channel properties," *IEEE Transactions on Information Theory*, vol. 50, no. 11, pp. 2657–2673, Nov. 2004.

[15] D. Divsalar, S. Dolinar, and F. Pollara, "Serial concatenated trellis coded modulation with rate-1 inner code," in *Proc. IEEE Global Telecommunications Conference GLOBECOM '00*, vol. 2, 27 Nov.–1 Dec. 2000, pp. 777–782.

[16] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate (corresp.)," *IEEE Transactions on Information Theory*, vol. 20, no. 2, pp. 284–287, Mar 1974.