Convolutional/Single Parity Check Turbo Codes for Wireless Multimedia Communications

Nikolay KOSTOV

Dept. of Radio Engineering, Technical University of Varna, Student 1, 9010 Varna, Bulgaria

n_kostov@mail.bg

Abstract. *Error correction codes are widely used in digital communications to improve the Quality of Service. The Quality of Service is typically expressed in terms of maximum acceptable frame error rate and bit error rate. The key implementation issues for most powerful error correction codes are the complexity and overall encoding/decoding latency. In this paper, short-frame turbo product codes for real-time wireless multimedia communications are proposed. Performance of the proposed turbo codes is studied through simulations on an additive white Gaussian noise (AWGN) channel. The obtained results indicate that the performance of these codes is quite exceptional given their decoding complexity.*

Keywords

Turbo codes, iterative decoding, simulation.

1. Introduction

Turbo codes are the recent state-of-the-art development in the field of error control coding. The rate $R_C = 0.5$ convolutional turbo code proposed by Berrou et. al. in [1] has been acknowledged as an extremely powerful coding scheme that achieves a bit error rate (BER) of 10^{-5} at a signal-to-noise ratio (SNR) E_b/N_0 only 0.7 dB above the Shannon capacity limit. However, the excellent performance of the original turbo code was achieved for a data frame size of 65536 bits and 18 decoding iterations per data frame. Because of prohibitively long latency involved with such large frame sizes, the original turbo code is not well suited for real-time multimedia communications.

The discovery of turbo codes [1] has stimulated a lot of research efforts in the area of concatenated codes with soft-in/soft-out (SISO) iterative decoding. A lot of results have been published on turbo codes, but most of the authors have focused either on convolutional turbo codes (CTC's) or on block turbo codes (BTC's) and very few have considered other approaches. In this paper, short-frame hybrid turbo code (HTC) schemes based on convolutional/single parity check codes concatenation are proposed. As will be shown, the performance of these codes is quite exceptional given their decoding complexity.

The paper is organized as follows. In Section 2, a brief discussion of tradeoffs in turbo codes is given. Section 3 presents the proposed hybrid turbo code schemes. Simulation results along with discussions are presented in Section 4. Finally, the concluding remarks are given in Section 5.

2. Tradeoffs in Turbo Codes

There are many factors that affect the performance of turbo codes. Convolutional turbo codes (CTC's) with rates $R_C \leq 0.5$ have been shown to exhibit outstanding performance for large interleaver sizes and a sufficient number of decoding iterations (typically 8 iterations or more) [1], [2], [3]. The most important parameter on CTC's performance is the interleaver size. As the interleaver size increases, performance improves. For example, the original CTC [1] gains more than 1 dB at a bit error rate (BER) of 10^{-5} as the interleaver size increases from 1024 bits to 4096 bits. However, as the interleaver size increases so does the overall encoding/decoding latency. Thus, CTC's possess an inherent tradeoff between performance and latency. It should be mentioned that if the interleaver size is below a threshold value of (approximately) 200 bits then a convolutional code will outperform the CTC of comparable complexity [4]. An additional tradeoff between performance and complexity is embedded in the choice of decoding algorithm. Turbo decoders fall into two general categories [3]: maximum *a posteriori* (MAP)-based decoders and soft output Viterbi algorithm (SOVA)-based decoders. MAP-based turbo decoders have the best performance, while their complexity is normally large and the speed is relatively low. SOVA-based turbo decoders are attractive for practical implementations due to their low complexity and relatively high speed. However, there is a gap of 0.5 dB or even higher between their performance and that of MAP-based turbo decoders. Another important issue regarding the performance of CTC's is the so-called "error floor", e.g., the flattening of the BER curve for moderate to high signal-to-noise ratios (SNR's). The "error floor" observed with CTC's is due to the fact that they have a relatively small free code distance. Decreasing the interleaver size results in the "error floor" being raised and the BER curve flattens at lower SNR's.

The SISO iterative decoding of two or more concatenated block codes is known as block turbo code (BTC) or turbo product code [5]. The BTC's are suitable for high code rates, typically with rates greater than 0.7, for systems that require high spectral efficiencies. Their performance does not depend significantly on the interleaver design. The main advantage of the BTC's with respect to the CTC's is the better asymptotic BER performance due to the larger minimum code distance. Another advantage is the possibility for fast parallel decoding of the rows/columns of the BTC since they are independent.

Although both the CTC's and BTC's have excellent performance for a wide range of code rates and data frame sizes designing low complexity turbo codes for short data frames (<200 bits) is still an open research area. An interesting approach is the concatenation of convolutional/single parity check codes with SISO iterative decoding. Details for this type of turbo processing are given in the next section.

3. Hybrid Turbo Codes

Hybrid concatenation of convolutional/single parity check codes with SISO decoding was considered for the first time by Hagenauer and Hoeher in [6] along with the discovery of the soft output Viterbi algorithm (SOVA). The authors showed that the use of non-iterative SISO decoding would allow the hybrid product code with a memory three (v=3), rate one-half (R_C=0.5) inner convolutional code and the (9,8) single parity check (SPC) outer code to equal the performance of the more complex v=6, R_C=0.5 convolutional code. This approach was followed by Freemen and Michelson [7] for more powerful component codes of the product code. In the present work, further study of the hybrid convolutional/single parity check codes with SISO iterative (turbo) decoding will be considered. Our aim is to build simple and yet powerful turbo codes with low overall encoding/decoding latency.

Let us now consider the encoding/decoding processes associated with the hybrid turbo code (HTC). The encoding process is quite straightforward. The data bits a_i, i=1,2,..,M are first arranged in a rectangular array. Then, all columns are encoded with an (n, n-1) single parity check (SPC) code. Finally, the rows, including those containing the parity bits, are encoded with a v=2 or v=3, R_C=0.5 convolutional code. Thus, the overall code rate R of the hybrid turbo code will be $R=R_C.R_S$, where $R_S=(n-1)/n$ is the rate of the SPC code. For an additive white Gaussian noise (AWGN) channel model the input/output relationship can be expressed as

$$r_i = \sqrt{E_s}\,(2b_i - 1) + n_i \ , \tag{1}$$

where r_i is the received symbol, E_S is the energy per code symbol, ($2b_i$-1) is the binary phase shift keying modulated code symbol and n_i is a zero-mean Gaussian variable with variance $\sigma^2=N_0/2$. A simplified diagram of the HPC decoder is shown in Fig.1.

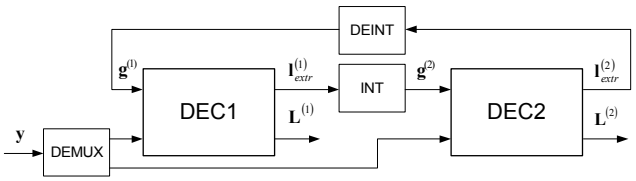

Fig. 1. Turbo decoder block diagram.

The log-likelihood ratio (LLR) L_i at the output of a SISO decoder can be represented in general as [3]

$$L_i = y_i + g_i + l_i \ , \tag{2}$$

where $y_i=4E_Sr_i/N_0$ is the weighted channel observation, g_i is the *a priori* information and l_i is the so-called extrinsic information gained by the current stage of decoding. The first elementary decoder (DEC1) in Fig.1 uses SOVA to form an estimate of the LLR of each bit encoded by the convolutional code (e.g., the bits of the SPC code). The essence of SOVA is finding the most likely transmitted sequence of bits along with reliability values for the bits [3], [7]. Let us define the likelihood ratio or "soft" value of the binary path decision at time i as

$$\Delta_i^0 = \frac{1}{2}\left(M_i^{m_0} - M_i^{\tilde{m}_0}\right), \tag{3}$$

where $M_i^{m_0}$ and $M_i^{\tilde{m}_0}$ are the path metrics of the survivor and competitor path, respectively. Now, the SOVA output LLR of the δ-delayed decision $\hat{b}_{i-\delta}$ can be expressed as

$$L\left(\hat{b}_{i-\delta}\right) \approx \hat{b}_{i-\delta} \cdot \min_{l=0,\ldots,\delta} \Delta_i^l \ . \tag{4}$$

For detailed explanation of SOVA see [3], [7]. Once the LLR's are obtained, the corresponding extrinsic information $\mathbf{l}_{extr}^{(1)}$ is used as *a priori* input to the second elementary decoder (DEC2). The extrinsic information $\mathbf{l}_{extr}^{(2)}$ associated with DEC2 (the SPC code decoder) can be computed according to [3]

$$l_i^{(2)} = (-1)\cdot\left(\min_{\substack{j=1,\ldots,n \\ j\neq i}} |L_j|\right)\cdot\prod_{\substack{j=1 \\ j\neq i}}^{n} \operatorname{sign}\left(L_j\right), \tag{5}$$

where now $L_j=y_j+l_j^{(1)}$. According to (5) the magnitude of the extrinsic information for a particular code element is equal to the minimum magnitude of all of the other parity elements. The sign of the extrinsic information for a particular code element is equal to the sign of the element itself, if the parity of the overall equation is satisfied, and opposite to the sign of the element, if the overall parity fails. The extrinsic information $\mathbf{l}_{extr}^{(2)}$ is used as *a priori* information by the DEC1 during the next iteration as shown in Fig.1. After a predetermined number of iterations, the final estimate of the message bits $\hat{a}_i, i=1,\ldots,M$, is found by hard-limiting the output of the DEC2:

$$\hat{a}_i = \begin{cases} 1 \ if \ L_i^2 \geq 0 \\ 0 \ if \ L_i^2 < 0 \end{cases}. \tag{6}$$

It should be mentioned that the structure of Fig.1 could be used for non-iterative decoding of the hybrid product code with a slight modification of the decoding algorithm. In this case the soft output $\mathbf{L}^{(1)}$ of DEC1 is used for maximum likelihood decoding of the SPC code by the DEC2. Now DEC2 computes all parity checks and if a parity check fails the sign of the corresponding code element with the lowest magnitude is flipped. The final hard decisions are obtained again according to (6).

Some important notes are in order here. First, the turbo decoder operation could be improved by scaling the extrinsic information of both elementary decoders. In the present work the so-called improved SOVA will be employed in which the performance of the SOVA decoder is enhanced by scaling the extrinsic information with a factor $c=2\mu_S/\sigma_S^2$, where μ_S and σ_S^2 are the mean and variance of the absolute value of the SOVA output, respectively. The extrinsic information of the SPC code decoder could be normalized (in the simplest way) by using a fixed set of scaling factors obtained through simulation. Second, it is straightforward to decode the HTC with a variable number of iterations using a predetermined "early stopping" rule. A simple hard-decision stopping rule is to check whether identical tentative bit decisions are made at successive iterations or half-iterations. Another approach is based on comparing a metric on bit reliabilities (soft bit decisions) with a threshold.

4. Performance Results

Performance of various HTC's with information frame sizes between 64 bits and 200 bits was studied through simulations according to the following setup. The information bits are obtained using uniformly distributed pseudorandom data. Maximum free distance memory v=2 or v=3, rate R_C=0.5 convolutional codes are used as inner codes in the HTC scheme. The SOVA decoding window is set to 24 bits. The channel output is unquantized throughout and obtained according to (1). Iterative decoding with up to six iterations and a hard-decision stopping rule is used to decode the HTC scheme. The "early stopping" rule is as follows: stop iterations if both elementary decoders output identical sets of hard-limited extrinsic values at a given full iteration.

Fig. 1 shows simulation results of the considered HTC schemes for the case of 144 bits data frame along with performance of two reference schemes. In Fig. 1 HTC1 denotes a HTC with a (13, 12) SPC outer code and a v=3 inner convolutional code, and HTC2 denotes a HTC with a (13, 12) SPC outer code and a v=2 inner convolutional code. Reference schemes in Fig. 1 are the powerful v=8 convolutional code (denoted as CC8) and a hybrid product code with non-iterative (message passing) SISO decoding (denoted as HPC1), which uses the same component codes as the above HTC1.

As can be observed from Fig. 2 both HTC schemes outperform the HPC1 scheme. The performance gain of the HTC1 over the HPC1 is above 0.5 dB for BER's of practical importance. Furthermore, the BER of the HTC1 is within 0.25 dB from that of the CC8. Also, the performance of the HPC1 is comparable to that of the rate R_C=0.5 CTC's with interleaver sizes below 150 bits and SOVA decoding. Presented results of the HTC schemes are typical for data frame sizes between 140 bits and 200 bits. However, significant performance degradation is observed for frame sizes below 100 bits. It should be mentioned that the performance of the HTC's can be improved in two ways. First, an improvement of at least 0.2 dB could be obtained if optimal MAP decoding of both component codes of the HTC is used. Second, a slight improvement could be expected for high SNR's using more complex "early stopping" rules with a larger maximum number of iterations.

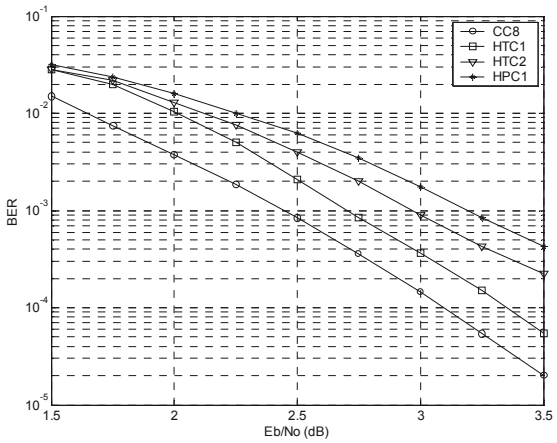

Fig. 2. BER performance of various error correction codes.

Another important issue, regarding the turbo decoder efficiency, is the number of decoding iterations. It is possible to improve both the average decoding speed and power consumption of the turbo decoder if a stopping rule is applied. Fig. 3 shows the simulation results for the turbo decoders of HTC1 and HTC2 (denoted as D1 and D2, respectively) wherein the average number of iterations per decoded frame versus the SNR is plotted. For comparisons, performance results of the corresponding "genius" decoders (denoted as D1genius and D2genius) are also shown. The "genius" turbo decoder operates as follows: stop iterations as soon as the correct transmitted codeword is observed.

It can be observed from Fig. 3 that a significant improvement in (average) turbo decoding speed is obtained when using the "early stopping" rule compared to the conventional turbo processing. Less than two iterations on average are required for both turbo decoders (the decoders D1 end D2) to converge at typical operational BER's below $1\cdot10^{-4}$. Also, the average number of iterations for the decoders D1 and D2 is less than one iteration away from that of the corresponding "genius" decoders. It was observed through simulations that no practical performance degradation is introduced with the considered stopping rule

compared to the fixed turbo decoding. This is because in fixed turbo processing with a relatively large number of iterations performed the errors often grow up with the number of processed iterations if no convergence is to be achieved.

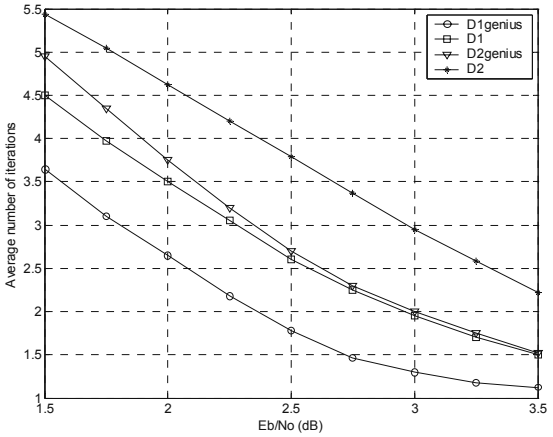

Fig. 3. Average number of iterations in turbo decoding the HTC1 and HTC2 schemes versus the SNR.

5. Conclusion

In this paper, hybrid convolutional/single parity check turbo codes (HTC's) for real-time wireless multimedia communications are proposed. Performance of various HTC's is studied through simulations on an additive white Gaussian noise (AWGN) channel. The obtained results indicate that the performance of these codes is quite exceptional given their decoding complexity. In fact, the best performance results are comparable to that of the powerful convolutional codes ($v \geq 7$) and, also, to that of the short-frame CTC's with SOVA decoding. Further performance improvement is possible if better decoding algorithms are used to decode the component codes of the HTC. For example, a simplified SISO decoding algorithm (based on the Viterbi algorithm) for convolutional codes exist [8] that significantly outperforms SOVA without increase in complexity.

References

[1] BERROU, C., GLAVIEUX, A., THITIMASJSHIMA, P. Near Shannon limit error-correcting coding and decoding: Turbo-codes. In *Proceedings of IEEE International Conference on Communications*, Geneva (Switzerland), 1993, p. 1064-1070.

[2] BENEDETTO, S., MONTORSI, G. Unveiling turbo codes: Some results on parallel concatenated coding schemes. *IEEE Transactions on Information Theory*, 1996, vol. 42, no. 2, p. 409–428.

[3] HAGENAUER, J., OFFER, E., PAPKE, L. Iterative decoding of binary block and convolutional codes. *IEEE Transactions on Information Theory*, 1996, vol. 42, no. 2, p. 429–445.

[4] KOORAPATY, H., WANG, Y., BALACHANDRAN, K. Performance of turbo codes with short frame sizes. In *Proceedings of IEEE Vehicular Technology Conference*, Phoenix (USA), 1997, p. 329-333.

[5] PYNDIAH, R. Near-optimum decoding of product codes: Block turbo codes. *IEEE Transactions on Communications*, 1998, vol. 46, no. 8, p. 1003-1010.

[6] HAGENAUER, J., HOEHER, P. A Viterbi algorithm with soft-decision outputs and its applications. In *Proceedings of IEEE Global Communications Conference*. Dallas (USA), 1989, p. 47.1.1-47.1.7.

[7] FREEMAN, D., MICHELSON, A. A two-dimensional product code with robust soft-decision decoding. *IEEE Transactions on Communications*, 1996, vol. 44, no. 10, p. 1222-1226.

[8] TAN, J., STUBER, G. New SISO decoding algorithms. *IEEE Transactions on Communications*, 2003, vol. 51, no. 6, p. 845-848.

About Author…

Nikolay KOSTOV was born in 1969 in Varna, Bulgaria. He received Dipl. Eng. and Dr. Eng. degrees from the National Military University of Veliko Turnovo, Bulgaria, in 1993 and 1999, respectively. He is an Assistant Professor at the Radio Engineering Department of the Technical University of Varna, Bulgaria. His research interests include source/channel coding, modulation techniques, and communications over fading channels.