On the suboptimality of turbo decoding in the case of short frame turbo codes and a way to improve the performance

Arnaud GUEGUEN
Mitsubishi Electric ITE
80 av. des Buttes de Coësmes, Immeuble Germanium, 35700 Rennes, France
Phone : +33 2 99 84 11 21, Fax : +33 2 99 84 21 15, email : gueguen@tcl.ite.mee.com

Damien CASTELAIN
Mitsubishi Electric ITE
80 av. des Buttes de Coësmes, Immeuble Germanium, 35700 Rennes, France
Phone : +33 2 99 84 11 26, Fax : +33 2 99 84 21 15, email : castelain@tcl.ite.mee.com

Abstract

Turbo codes are becoming a widespread and mature coding scheme, as they are included in the standards of the third generation of mobile communication systems [1]. They were originally shown to perform very well from long to medium sizes of blocks [2] thanks to their good codewords weight distribution – i.e. good free distance and low multiplicity of low weight codewords - and to the ability of iterative decoding to perform near optimum decoding in the sense of Maximum Likelihood (ML). However in recent releases of the standards they are also intended to be used for short sizes of blocks, down to 40 bits, and in those cases turbo decoding no longer provides ML decoding. The aim of this paper is first to quantify the suboptimality of turbo decoding when applied to short Parallel Concatenated Convolutional Codes as compared to ML decoding, and then to provide a way to improve the turbo decoding process in those cases. In this view the ML bounds of the considered short turbo codes are derived according to their measured truncated weight distribution. It is recalled that, unlike average bounds that are derived for a so-called uniform interleaver [3,4,5] and that only reflect an average behaviour, the resulting bound predicts the ML performance with a specific interleaver. Whereas for medium to long interleavers the performance of turbo codes stick to this bound, it is shown that they can be around 0.5 dB away from the bound in the case of short turbo codes. At last, a scheme that enables to partly overcome this loss is presented and evaluated.

1 Deriving the ML bound for a particular turbo code

Beyond their theoretical interest, error bounds enable to predict the performance of a code at high Signal to Noise Ratios (SNR), at Bit Error Rates (BER) that cannot be reached through simulations, and also to account for the efficiency of the decoding scheme. In the case of turbo codes the error bounds should also help predicting the so-called error floor that shows up at low BER. However most of the existing bounds for turbo codes have been derived considering a statistical interleaver, also called uniform interleaver [3,4,5], and they only provide the performance of a turbo code after averaging on all possible interleavers. Although it is theoretically interesting, such statistical approach is not well suited here, as we want to analyse the performance with one specific interleaver. In the rest of the article the bound we choose to use is the ML Union Bound as it is the most straightforward. Considering this bound, we first show that the average weight distribution obtained with the uniform interleaver does not provide sufficiently accurate results, especially when compared to the performance obtained with an optimised interleaver. Thus, we recall that it is necessary to use the measured weight distribution of the code. All results are presented on AWGN channel.

Considering a PCCC obtained by concatenation of two Recursive Systematic Convolutional codes (RSC) denoted C_1 and C_2, separated by an interleaver of length N, the turbo code weight enumerating polynomial using the uniform interleaver concept is given by equation (1) :

$$A^{C_p}(W,Z) = \sum_{w=1}^{N} W^{w} \frac{A_w^{C_1}(Z)\cdot A_w^{C_2}(Z)}{\binom{N}{w}} \overset{\Delta}{=} \sum_{w,j} A_{w,j} W^{w} Z^{j} \qquad (1)$$

where $A_{w,j}$ is the number of codewords with information weight w and redundancy weight j, and $A_w^{C1}(Z)$ and $A_w^{C2}(Z)$ are the conditional weight enumerating polynomials of the two elementary codes as defined in [3]. The obtained weight distribution is used in the ML Union Bound equation for the BER :

$$P_b(e) \approx \frac{1}{2}\sum_m D_m \operatorname{erfc}\left(\sqrt{m\frac{R_c E_b}{N_0}}\right) \qquad (2)$$

where the coefficients D_m are given by :

$$D_m \overset{\Delta}{=} \sum_{j+w=m} \frac{w}{N} A_{wj} \qquad (3)$$

R_c is the code rate and E_b/N_0 is the SNR per information bit.

As an example, the PCCC made of two identical $(5,7)_{oct}$ RSC codes with interleaver length N=100 is considered. The average distribution D_m and the associated ML bound obtained with the uniform interleaver, are represented respectively on figures 1 and 2. They are consistent with those presented in [3].

The simulated performance of this PCCC using an optimised interleaver (i.e. yielding a good minimum distance) is plotted together with the average ML bound on figure 2. It shows that the bound is quite far away from the simulation curve and that it fails to predict the error floor effect as it is above the simulation curve. Indeed, at high SNR the first term of the summation in equation (2), corresponding to the minimum distance of the code, becomes the most significant and it turns out to be very different from one particular interleaver to another. In particular, the minimum distance with an optimised interleaver is bigger than the one given by the uniform interleaver.

Therefore, for the ML bound to reflect the behaviour of the turbo code with one particular interleaver, at least the first terms of the summation in (2) should be the effective ones, instead of the terms given by equation (1). The effective first values of D_m are obtained by feeding the turbo encoder with all possible sequences of information weight below or equal to 5 and measuring the output codewords weights. Low weight codewords are given by low weight information sequences [4] and it is statistically unlikely that the free distance is produced by an input sequence of weight above 5. Indeed, low weight codewords are produced by sequences that terminate both trellises, that is by sequences that terminate the first trellis and that are interleaved in sequences that terminate the second one. Such mapping is all the more unlikely as the input weight is high. The obtained measured truncated distribution is drawn together with the average distribution on figure 1 and the corresponding ML bound is plotted on figure 2.

The simulation curve converges very close to this second bound at high SNR and the error floor occurrence at a BER around 10^{-4} is obviously related to this bound and, through it, to the first values of D_m. The BER saturation phenomenon observed on the average ML bound at low SNR is not observed on this second bound because the weight distribution is truncated. However none of the two bounds is really valid at low SNR.

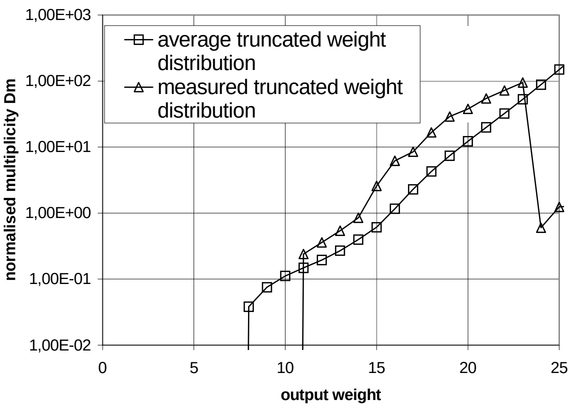

Figure 1 average and measured truncated weight distributions D_m, interleaver length N = 100

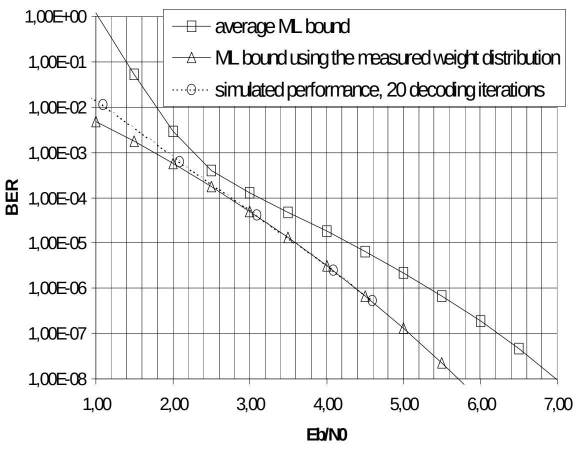

Figure 2 ML bounds using the average and measured weight distributions, and simulation results (N = 100)

2 On the suboptimality of turbo decoding for short turbo codes

The ML bound based on the measured truncated weight distribution is accurate at high SNR as it takes the effective weight distribution of the code in consideration. Therefore the simulation curve of a turbo code after a large number of decoding iterations shall stick to this bound if the decoding process is optimal in the sense of ML, possibly showing the so-called error floor as described above. However, as mentioned in [6,7,8], the graph theory predicts that the forward-backward algorithm used in turbo decoding converges towards ML decoding provided that the graph representing the code is cycle-free. Turbo codes graphs are never cycle free, nevertheless reference [8] explained that, as an extension to graph theory, turbo decoding converges towards ML decoding provided that the cycles on the graph are long enough for the correlation on extrinsic information to completely vanish along them. Therefore, when designing a turbo code interleaver, one should both maximise the free distance of the code and the size of cycles. The minimum distance of the code will condition the error floor while long cycles will guarantee that the turbo decoder indeed converges towards the ML bound. Both criteria are easily achievable when the interleaver size is sufficiently large but they become hard to fulfil in the case of short interleavers.

To illustrate this, figure 3 and 4 show turbo codes performance in two different cases, as well as the corresponding ML bounds obtained with the measured truncated weight distribution. Both turbo codes use the $(13,15)_{oct}$ RSC codes. 50 decoding iterations are performed to guarantee optimal performance. The first turbo code uses an interleaver of length N = 424 that has been designed to yield long cycles but the code is punctured to R_c = ¾ so that the free distance is only 3 : the bound is relatively high but the turbo decoding is optimal as the simulation curve sticks to it. The second turbo code uses an interleaver of length N = 80 that has been designed to yield a large free distance (d_{min} = 15, R_c = 1/3) but the cycles are short because the interleaver is small : the bound is very low but the simulation curve does not reach it as the turbo decoding process is suboptimal. Besides, considering the distance between the simulation curve and the bound, one can predict the potential performance gain to achieve by making the decoding process closer to ML decoding : in the case of figure 2(b), the potential gain is approximately 0.5dB at a BER of 10^{-6}.

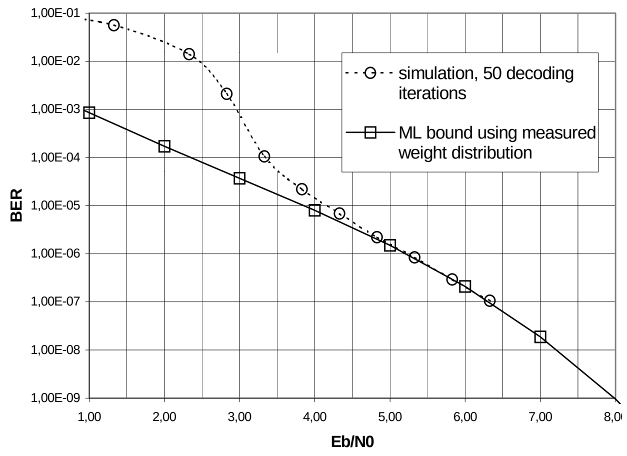

Figure 3 simulated performance vs. with ML decoding : N = 424, R_c = ¾, d_{min} = 3, long cycles

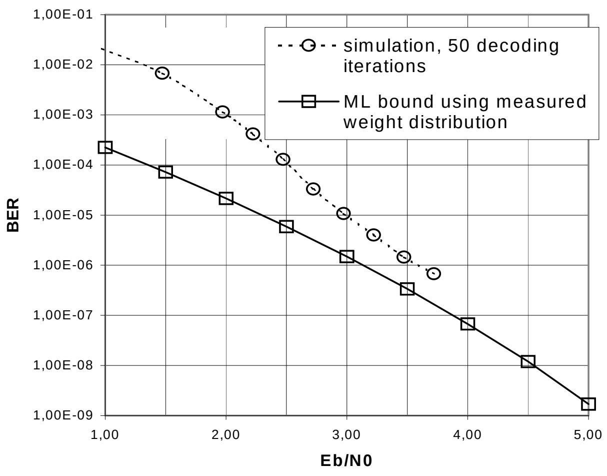

Figure 4 simulated performance vs. with ML decoding : N = 80, R_c = 1/3, d_{min} = 15, short cycles

3 Improving the performance of turbo decoding

The scheme that is proposed below aims at improving the turbo decoding towards ML decoding when the interleaver design or size prevent it from converging properly. The basic principle is the following : when errors are detected in the decoded block after a large number of iterations, the decoded erroneous binary sequence is turbo encoded and modulated. The resulting modulated bits sequence is then multiplied by a coefficient α, which is in the order of magnitude of 10^{-2}, and subtracted from the corresponding input sequence. The turbo decoding process is then applied over to this modified input sequence. Again, if residual errors are found in the decoded block, the same "post processing" scheme of re-encoding, re-modulation and subtraction is applied, and so on until there is no error left or until a maximum number of iterations is reached. The proposed scheme is depicted on figure 3. The switch on the left hand side is down when the received sequence is going to be turbo decoded for the first time, and up when post processing is performed.

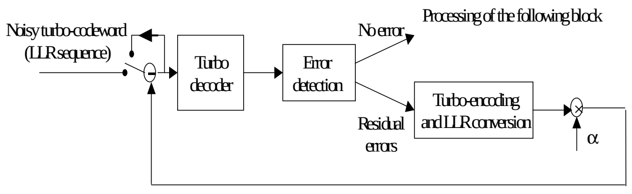
Noisy turbo-codeword (LLR sequence)
-
Turbo decoder
Error detection
No error
Processing of the following block
Residual errors
Turbo-encoding and LLR conversion
α

Figure 5 iterative post processing scheme

The underlying idea is that the contribution of the decoded sequence is subtracted from the received input

sequence if the turbo decoder fails to correct all errors. Consequently, when turbo decoding is performed on the modified sequence, the path metric of the erroneous sequence – i.e. that was previously produced – in the turbo code trellis is reduced so that a neighbouring sequence will be produced. If the residual errors were due to the suboptimality of the turbo decoding, the probability of error free convergence at the next step is increased.
Error detection can be implemented in various ways : an error detection code such as Cyclic Redundancy Check (CRC) may be concatenated to the data sequence before turbo encoding and used at the receiver side to check the integrity of the decoded data (CRC are envisaged in the UMTS standard [1]). Convergence detection based on the cross entropy criterion [9,11] or derived criteria [10] may also be used. Performance evaluation is presented below with various detection schemes.

3.1 Ideal error detection

The proposed scheme is applied to the case of figure 4 with the same turbo decoder, with a post processing coefficient $\alpha = 10^{-2}$, and a maximum of 20 post processing iterations. Perfect error detection is assumed in this first case. The results are plotted on figure 6 together with the previous curves of figure 4.
The global complexity may be excessive for a system implementation (although the average complexity on a large number of blocks is not much affected because at high SNR very few blocks use the post processing scheme), but the simulation indicates that the performance loss compared to ML decoding, observed on figure 4, may effectively be due to the weakness of the iterative decoding and that it can be partially overcome with a simple scheme.

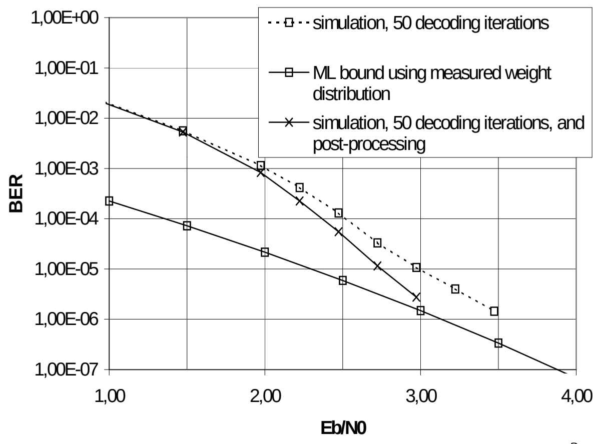

Figure 6 performance with post processing, $\alpha = 10^{-2}$, until 20 post processing iterations, perfect error detection

Ideal error detection is not realistic because, provided a very long decoding time, all possible codewords could be tried until the right one is found. However, because of the low number of post-processing iterations, the search space of the turbo decoder is several orders of magnitude below the total number of possible codewords.
Further results using a more realistic error detection scheme are presented below.

3.2 Error detection with CRC

Residual errors can be detected with Cyclic Redundancy Check bits (CRC) added at the end of the useful data sequence. In the case of UMTS, CRC are envisaged for any size of block and they were originally introduced for ARQ. When used as the error detection scheme within the post processing scheme they provide a performance gain, as shown on figure 7. The turbo code parameters are the same as on figure 4. 16 CRC bits are added before turbo encoding. At the receiver side, 20 turbo decoding iterations are performed. When using post processing, at most 10 post processing iterations are performed. Error detection is used both to interrupt the turbo decoding process if there is no error left before the last iteration, and in the post processing loop.

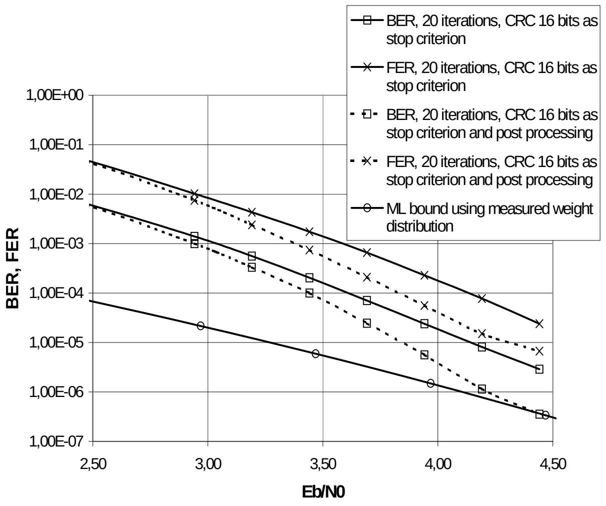

Figure 7 performance with post processing, $\alpha = 10^{-2}$, until 10 post processing iterations, error detection with 16 CRC bits

The ML bound is the same as on figures 4 and 6, shifted by 0.96dB = 10log(80/(80-16)) to account for the E_b/N_0 penalty due to the 16 CRC bits. The performance gain is around 0.5 dB at a BER of 10^{-6} and the simulation curve is close to the ML bound.
In this particular case, post processing can be seen as an alternative to ARQ with very small latency as no retransmission is needed.

3.3 Convergence detection based on cross entropy

Stop criteria are used in turbo decoding to interrupt the iterative decoding process when a sequence is properly decoded before the last iteration, or, more precisely, when further iterations will provide no additional performance gain. Indeed, one wishes to avoid unnecessary computations, i.e. unnecessary latency and energy consumption. Various stop criteria proposed in the literature [9,10,11] are based on the Cross Entropy (CE) between the distributions of the estimates of the decoders outputs at each iteration.
In this part we evaluate the performance of the post processing scheme when the simplified CE criterion proposed in [9] is used as a stop criterion and as an error detection scheme : if the CE ratio between the current iteration and the first one drops below a threshold of 10^{-4} the sequence is assumed error free. On the contrary, if the ratio remains above this threshold until the last turbo decoding iteration, the sequence is considered erroneous and post processing is performed. The turbo coding and turbo decoding parameters are the same as in part 3.2.

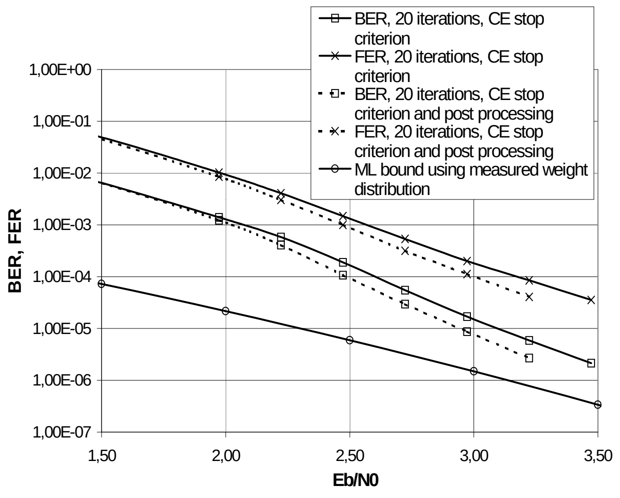

Figure 8 performance with post processing, $\alpha = 10^{-2}$, until 10 post processing iterations, error detection using cross entropy

The performance gain is rather small but the BER curve is closer to the ML bound. The convergence detection scheme used may not provide accurate enough error detection. More sophisticated convergence detection may produce better results.

4 Conclusion

The ML bound using the measured truncated weight distribution of a PCCC with a given interleaver is used to investigate the optimality of turbo decoding at high SNR in the sense of ML sequence decoding. It is observed through simulations that turbo decoding is suboptimal for interleavers yielding short cycles in the graph of the turbo code, and in particular for short turbo codes. The performance loss is quantified in a particular case and a scheme based on error detection and re-encoding is proposed to partially overcome this loss. Depending on the robustness of the error detection scheme, a performance gain between 0.1 dB and 0.5 dB is observed at a BER of 10^{-6}. For instance, when convergence detection of the turbo decoder is used as the error detection scheme, little performance gain is achieved. Thus, although the scheme may not be suitable in its current state for a practical use, it shows that some gain is still to be expected in the case of short turbo codes.

5 Literature

[1] 3GPP TSG RAN WG1 25.212, v3.1.0 : “Multiplexing and Channel Coding”
[2] C. Berrou, A. Glavieux, P. Thitimajshima : “Near Shannon limit error-correction coding : Turbo codes”, in Proc. IEEE ICC'93, Geneva, Switzerland, pp. 1064-1070, May 1993
[3] S. Benedetto, G. Montorsi : “Unveiling Turbo Codes : Some Results on Parallel Concatenated Coding Schemes”, IEEE Transactions on Information Theory, September 28, 1995
[4] S. Benedetto, G. Montorsi : “Design of Parallel Concatenated Convolutional Codes”, IEEE Transactions on Communications, vol. 44, pp. 591-600, May 1996
[5] T.M. Duman, M. Salehi : “New Performance Bounds for Turbo Codes”, IEEE Transactions on Communications, vol. 46, No.6, June 1997
[6] N. Wiberg : “Codes and decoding on general graphs”, PhD thesis no; 440, Dept. Elect. Eng., Linköping Univ., Sweden, 1996
[7] R.J. McElliece, D.J.C. McKay, J.F. Cheng : “Turbo Decoding as an Instance of Pearls’s ‚Belief Propagation‘ Algorithm”, IEEE J. on Selected Areas in Communications, V. 16 Number 2, February 1998
[8] E. Fabre, A. Guyader : “Dealing with short cycles in graphical codes”, paper submitted to ISIT 2000
[9] J. Hagenhauer, E. Offer, L. Papke : “Iterative decoding of binary block and convolutional codes”, IEEE Transactions on Information Theory, vol. 42, pp.429-445, March 1996
[10] R.Y. Shao, S. Lin, M.P.C. Fossorier : “Two Simple Stopping Criteria for Turbo Decoding”, IEEE Transactions on Communications, vol. 47, No.8, August 1998

[11] M. Moher : “Decoding via cross entropy minimization”, in Proc. IEEE Globecom Conf., Houston, TX, December 1993, pp.809-813

6 Biography

Arnaud Guéguen graduated from the Ecole Centrale Paris in 1996 and from the Royal Institute of Technology KTH in Sweden in 1997. He joined Mitsubishi Electric ITE in 1998 as a research engineer and his main interest is currently in turbo-codes. He was involved in the standardisation process of channel coding in ETSI UMTS.

Damien Castelain graduated from the Ecole Nationale Supérieure des Télécommunications in 1982, and joined CIT-Alcatel in 1983 to participate in the development of videophone (COST211) and satellite equipment. He joined CNET-CCETT in 1988 and became head of the ‘Modulation and Channel Coding’ department in 1995. From 1988 to 1994, he was involved in Digital Audio Broadcasting developments within EU-147 DAB and JESSI/DAB AE14 projects, where his main interests were in channel coding theory and VLSI implementation. From 1990 to 1998, he was involved in the development of digital terrestrial television, in the framework of European projects dTTb, VALIDATE, DVBIRD, INTERACT. He joined Mitsubishi Electric ITE in 1998, to co-ordinate research activities in digital communications applied to next generation systems (UMTS, BRAN, and DAWS).