Decreasing of the Turbo MAP Decoding Time by Using an Iterations Stopping Criterion

L. Trifina[1], H.G.Balta[2], A. Rusinaru[1]
[1]Electronics and Telecommunications Faculty, Bd. Carol I, no. 11, Iasi, Romania,
e-mail: lucianetc@yahoo.com, ancarushi@yahoo.com
[2]Electronics and Telecommunications Faculty, 2 Bd. V. Parvan, 1900 Timisoara, Romania,
e-mail: horiabalta@etc.utt.ro

***Abstract* – In the paper an iterations stopping criterion in iterative turbo decoding, based on the LLR of the decoded bit, by using a comparison with a threshold, is proposed. We present the performances of a turbo code, both in the case of a fixed number of iterations and in the case in which the stopping criterion is used. We compare the decoding times. An analysis of the influence of the threshold on the turbo code bit error rate (BER) performance, using the iterations stopping criterion, is presented.**
Keywords: turbo codes, LLR, stopping criterion, MAP decoder.

I. Introduction

The constitutive decoders „Dec1 (2)" of a turbo decoder (Fig. 1) implementing the MAP algorithm, compute the logarithm likelihood ratio (LLR) per bit, LLR_k, in the form, [1]:

$$L\left(u_k \left| \underline{y}\right.\right) = \ln\left(\frac{\sum\limits_{\substack{(\hat{s},s)\Rightarrow \\ u_k=+1}} \alpha_{k-1}(\hat{s}) \cdot \gamma_k(\hat{s},s) \cdot \beta_k(s)}{\sum\limits_{\substack{(\hat{s},s)\Rightarrow \\ u_k=-1}} \alpha_{k-1}(\hat{s}) \cdot \gamma_k(\hat{s},s) \cdot \beta_k(s)} \right), \quad (1)$$

where $\alpha_{k-1}(\hat{s})$ (the probability that the encoder trellis was in $\hat{s}$ state at instant $k-1$ and the received channel sequence, before this moment, is $\underline{y}_{j<k}$) and $\beta_k(s)$ (the probability that, having been given the trellis state s at instant k, the received channel sequence, after this moment, to be $\underline{y}_{j>k}$) are computed by recurrence using the branches' metrics, $\gamma_k(\hat{s},s)$ (the probability that the encoder trellis took the transition from state $\hat{s}$ to state s and the received channel sequence for this transition is $\underline{y}_k$). The LLR of the bit u_k can be expressed as a function of three terms, [2]:

$$LLR_k = L_c(u_k) + L_a(u_k) + L_e(u_k), \quad (2)$$

The first term, $L_c(u_k)$, depends on the received sequence, $L_a(u_k)$ is a priori information (resulted from the other decoder) and $L_e(u_k)$ represents the extrinsic information (output of the decoder) that will be the a priori information for the pair decoder.

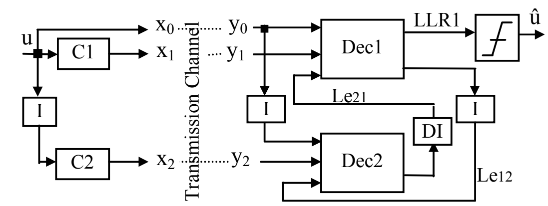

Figure.1. Parallel Concatenated Convolutional Code (PCCC)

In the AWGN noise hypothesis, LLR_k is a Gaussian random variable having a positive mean if the bit u_k has 1^L value and a negative mean for the 0^L value.

The output of the turbo decoder is obtained by a hard decision on the LLR_k values, generated from the one of the two constitutive decoders (Dec1 in Fig. 1), after performing the number of iterations. This number can be fixed, or variable, in which case there is a stopping criterion.

II. The Influence of the Number of Iterations on the Decoding Gain

As the iterative decoding process advances, the set of values $LLR_{k=1\div N}$ (N is the length of a data block), gives, by their sign, more accurate information about the polarity of the bits from the $u_{k=1\div N}$ transmitted sequence. From an iteration to other (see III.) the LLR values of the bits from the $u_{k=1\div N}$ sequence, which have 1^L value, are moving to positive values, as the LLR values of the bits from the $u_{k=1\div N}$ sequence, which have 0^L value, are moving to negative values; hence the errors are fewer and fewer.

In Fig. 2, Fig. 3 and Fig. 4 the results of turbo MAP decoding simulation, with 15 iterations, over AWGN channel, for a parallel turbo code, with a S- random interleaver ($S=29$) having a length of $N=1784$ bits, without puncturing (with a nominal coding rate 1/3) and BPSK modulation, are presented. We used the following constitutive convolutional codes: $G=[1,5/7]$ (in octal) for $m=2$ memory, $G=[1,15/13]$ for $m=3$ and $G=[1,25/31]$ for $m=4$. The simulations were performed with a step of 0.1 dB and the used number of blocks increases, up to $2 \cdot 10^6$ for $BER<10^{-6}$.

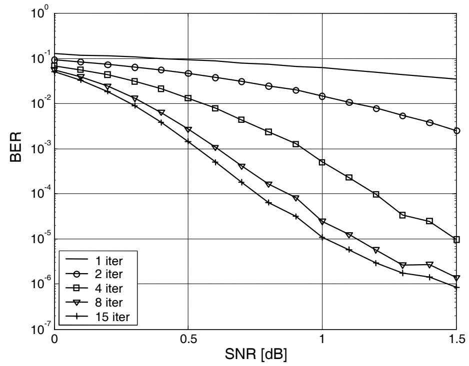

Figure.2. PCCC with 1/3 rate and 5/7 constitutive code

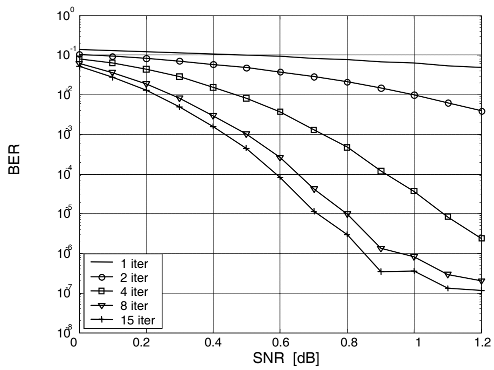

Figure.3. PCCC with 1/3 rate and 15/13 constitutive code

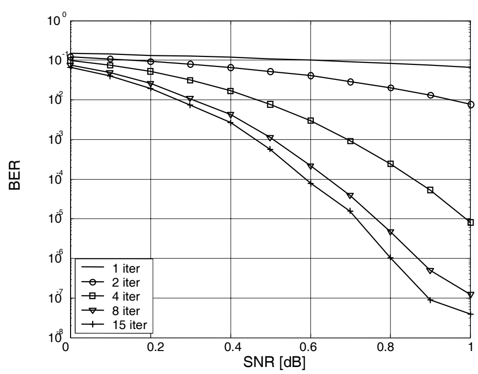

Figure.4. PCCC with 1/3 rate and 25/31 constitutive code

Table I shows the difference between the SNR of previous iteration and the SNR of current iteration as a function of the number of iteration. The small value of the accumulated gain after the 10^{th} iteration does not justify the performing of the iterations from *11* to *15*. Since the results of the repeated simulations were mediated, the following aspect is imperceptible, but should not be neglected: there are blocks for which the turbo code generates more errors after the 15^{th} iteration (for example) than after the 10^{th} iteration. This "rebound" phenomenon determines that the turbo code performances to decrease dramatically at a small *BER*. Hence, the strategy of the maximum number of iterations (*15*) with

TABLE I.

Iter	***SNR* (iter-1) – *SNR* (iter) [dB] (*BER*=10^{-5})**		
	5/7	***15/13***	***25/31***
1	4.871	5.474	5.798
2	2.176	2.064	1.962
3	0.740	0.689	0.618
4	0.305	0.273	0.223
5	0.224	0.148	0.102
6	0.066	0.085	0.059
7	0.042	0.034	0.037
8	0.036	0.024	0.026
9	0.033	0.029	0.015
10	0.027	0.020	0.011
11	0.021	0.013	0.007
12	0.014	0.008	0.004
13	0.008	0.008	0.004
14	0.005	0.002	0.004
15	0.006	0.005	0.003

TABLE II.

m	Code	Time per 1000 blocks (sec)	
		1.6 GHz	***533 MHz***
2	5/7	280	460
3	15/13	585	915
4	25/31	1207	1946

stop after a certain criterion (described in the next paragraph) will permit, on one hand, no performing of the remaining iterations up to *15*, on a corrected block, and the other hand, the avoidance of the "rebound" phenomenon on a corrected block. Using a stopping criterion the average decoding time will decrease implicitly. For comparison, in the table II we evaluated the decoding time for *1000* transmitted blocks on a Pentium 4, at *1.6* GHz, processor and a Pentium 3, at *533* MHz, processor. The decoding time is constant for a code with a certain memory and is doubling when the order of memory increases by one unity due to doubling of the number of states in the encoder trellis.

III. STOPPING CRITERION WITH LLR

In the hypothesis of an AWGN channel and using BPSK modulation, LLR_k is a random variable with normal distribution:

$$f_{LLR}\left(\Lambda\right)=\frac{1}{\sigma_{\Lambda}\sqrt{2\pi}}\cdot\exp\left\{\frac{\left(\Lambda-m_{\Lambda}\right)^{2}}{2\sigma_{\Lambda}^{2}}\right\}, \qquad (3)$$

where the mean m_{Λ} has positive or negative values depending on the sign of the transmitted information bit. Hence, the histogram of the set of $LLR_{k=1\div N}$ values, when the data sequence is random, includes two subsets, that have the shape of a two Gaussians, centered on $m_{\Lambda 0}$ and $m_{\Lambda 1} \approx -m_{\Lambda 0}$ respectively, with a approximately equal dispersion.

For the same number of iterations the LLR values (the two Gaussians) are moving away from the origin and are more

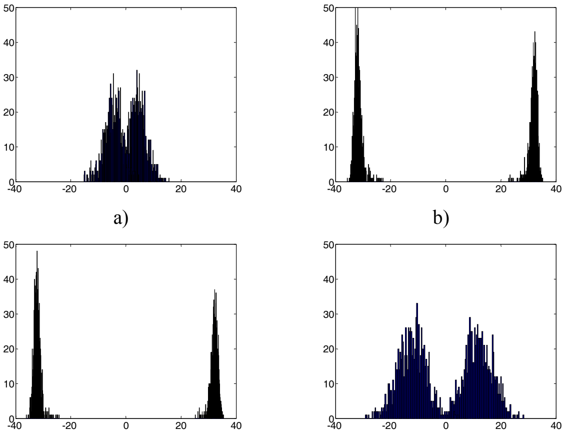

Figure.5. The LLR histograms for the 5/7 code: a) SNR=0.1dB (15 decoding iterations); b) SNR=0.5dB (15 decoding iterations); c) SNR=1dB (15 decoding iterations), d) SNR=1dB (3 decoding iterations)

grouped when SNR increases (i.e. they have bigger absolute mean value and a smaller dispersion). This is observed from the histograms presented in Fig. 5, a), b), c), performed for the same transmitted data block, with *15* decoding iterations of the code with $G=[1,5/7]$ and three SNR values.

The fact that, for $SNR=0.1$ *dB*, the two Gaussian are imbedded, shows that there are errors in decoding (there are k indexes, so that $LLR_k \cdot u_k<0$). The same thing happens at bigger values of the SNR, but for smaller number of iterations (see Fig. 5, d), for $SNR=1$ *dB* and *3* decoding iterations).

Table III shows the mean and the dispersion for the positive and negative values respectively, in the three considered cases.

After *10* iterations (on an average) the two Gaussians are totally separated from a symmetrical interval with respect to the origin that, with scarce cases, contains any LLR values. From this moment *BER* is not improved significantly and the remaining iterations must not be performed. We can achieve this by the next stopping criterion:

The iterative decoding process of a block is stopped if:

$$|LLR_k| > \mu, \quad k = 1 \div N, \tag{4}$$

If condition (4) is accomplished the iterations are not performed and the $u_{k=1\div N}$ binary sequence is reconstructed by a hard decision on the corresponding $LLR_{k=1\div N}$ values.

The turbo code with stop (TCS) makes an extra error compared the one without stop if at least one LLR_k value from the set corresponding to the positive bits ($u_k=1$) is smaller than $-\mu$, and all the other values are outside the $[-\mu,\mu]$ interval and if at least one LLR_k value from the set corresponding to the

TABLE III.

SNR [dB]	Mean (LLR)		Var (LLR)	
	LLR_pos	***LLR_neg***	***LLR_pos***	***LLR_neg***
0.1	4.409	-4.642	10.269	11.360
0.5	31.797	-31.815	2.510	2.282
1	32.161	-32.162	1.813	1.748

negative bits ($u_k=-1$) is bigger than μ and all the other values are outside of the $[-\mu,\mu]$ interval.

Assuming the following:

– the previous mentioned events are independent;

– the transmitted sequence is random ($p(0_{emitted})=p(1_{emitted})$);

– $m_{\Lambda 1}= -m_{\Lambda 0}= m_\Lambda$ and $\sigma_{\Lambda 0}= \sigma_{\Lambda 1}= \sigma_\Lambda$,

the probability that TCS to make an extra error than TC is:

$$P(\mu)= C^{1}_{N/2} \cdot a^{N/2-1} \cdot b, \tag{5}$$

where:

$$a=0.5\cdot erfc[(\mu - m_\Lambda)/(\sqrt{2}\cdot\sigma_\Lambda)], \tag{6}$$

$$b=1-0.5\cdot erfc[(-\mu - m_\Lambda)/(\sqrt{2}\cdot\sigma_\Lambda)], \tag{7}$$

and *erfc(.)* is the error function.

IV. The Performance of TCS

Relying on the analysis of the LLR histograms of the 5/7, 15/13 and 25/31 codes, three thresholds were chosen: $\mu_1=5$, $\mu_2=10$ and $\mu_3=20$. Fig. 6, Fig. 7 and Fig. 8 present the simulations for the three codes using the three thresholds and a maximum number of iterations.

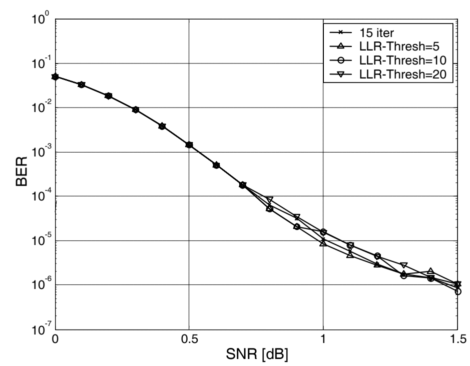

Figure.6. PCCC 5/7 with LLR Stop and a maximum number of iterations of 15

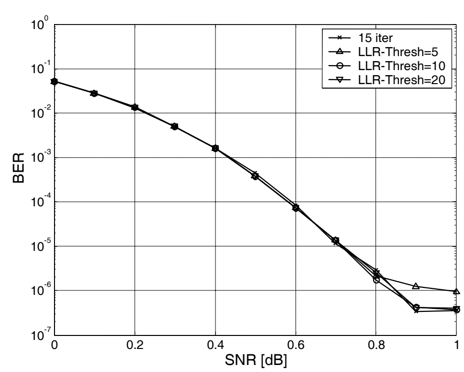

Figure.7. PCCC 15/13 with LLR Stop and a maximum number of iterations of 15

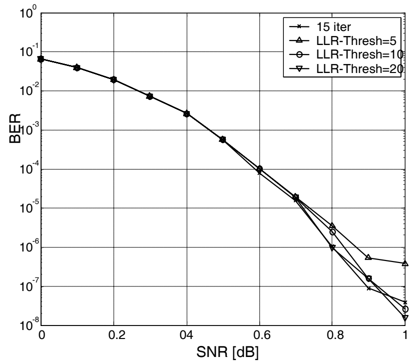

Figure.8. PCCC 25/31 with LLR Stop and a maximum number of iterations of 15

The tables IV, V and VI correspond to the three codes, where:

– r.e.i. = rate of effectuated iterations (per block, from 15),

– r.s.b. = rate of stopped blocks (percent of blocks for which the stopping criterion was applied),

– t.s.b = total simulated blocks.

Both the Fig. 6, Fig. 7, Fig. 8 and $P(\mu_i)$ from (5) show that, TCS actually does not make an extra error than TC ($P(\mu_i)$ has values bellow 10^{-10} for any (μ, m_Λ, σ_Λ) set of the of ones discussed). There is the possibility that the performances TCS to exceed on the ones of TC, due to diminishing of the "rebound" effect.

With respect to decoding times, the discussion must take into account both *SNR* and μ. Hence, for small *SNR* the stopping rate of iterations is smaller for μ_2 and μ_3 and there are no differences between the performances of TCS(μ_1), TCS(μ_2) or TCS(μ_3). Hence, from the point of view of BER, for small *SNR* is not justified to choose a big threshold.

For big values of *SNR* (where *BER* is smaller than 10^{-5}) the performances of TCS(μ_1) are a little inferior than the ones of TCS(μ_2) and TCS(μ_3). The identical performances for TCS(μ_2) and TCS(μ_3) do not justify the choosing of a threshold bigger than 10, that will lead to a bigger decoding time. In conclusion, a solution could be μ_2, but we can also choose a threshold varying with *SNR*.

TABLE IV. PCCC 5/7

LLR – Thresh (μ)	r.e.i	r.s.b	t.s.b
5	26.3 %	99.4 %	2373000
10	30.2 %	99.2 %	2558000
20	42.3 %	94.6 %	3661000

TABLE V. PCCC 15/13

LLR – Thresh (μ)	r.e.i	r.s.b	t.s.b
5	25 %	99.8 %	2092000
10	27.9 %	99.7 %	1742000
20	31.8 %	99.3 %	2242000

TABLE VI. PCCC 25/31

LLR – Thresh (μ)	r.e.i	r.s.b	t.s.b
5	22.4 %	99.8 %	2368700
10	24 %	99.8 %	2379900
20	27.1 %	99.6 %	2562700

V. CONCLUSIONS

In this paper, we presented an iterations stopping criterion using LLR and an analysis of the turbo MAP decoding performance depending on the number of iterations and on the chosen LLR threshold.

The iterations stopping criterion uses a threshold in the range of the LLR values which must be exceeded (in absolute value) for all the bits of a data block, in order not to effectuate the remaining iterations.

From the simulations performed for the three LLR thresholds we noticed that the number of effectuated iterations decreased on the whole (with approximately 70 %) and also that the decoding time was reduced, depending on the *SNR* value.

ACKNOWLEDGMENT

L. Trifina, A. Rusinaru: We would like to thank Prof. dr. ing. Valeriu Munteanu and Prof. dr. ing. Liviu Goras for their given support in our difficult moments.

H.G. Balta: I would like to thanks to the one who opened my way in the turbo-codes area, Mrs. Prof. dr. ing. Miranda Nafornită.

REFERENCES

[1] C. Berrou, A. Glavieux, P. Thitimajshima - "*Near Shannon limit error-correcting coding and decoding: Turbo-codes*", *Proc.ICC'93,* Geneva, Switzerland, May 1993, pp. 1064 – 1070.

[2] L.Hanzo, T.H.Liew, B.L.Yeap, "Turbo Coding, Turbo Equalisation and Space-Time Coding for Transmission over Fading Channels", John Wiley & Sons Ltd, England, 2002

[3] Horia Baltă, Maria Kovaci - *„A study on turbo decoding iterative algorithm"*, Scientific Bulletin of the "Politehnica" University of Timişoara, România, Oct 2004, pp. 33 –37