Two Simple Stopping Criteria for Turbo Decoding

Rose Y. Shao, *Student Member, IEEE*, Shu Lin, *Fellow, IEEE*, and Marc P. C. Fossorier, *Member, IEEE*

***Abstract*—This paper presents two simple and effective criteria for stopping the iteration process in turbo decoding with a negligible degradation of the error performance. Both criteria are devised based on the cross-entropy (CE) concept. They are as efficient as the CE criterion, but require much less and simpler computations.**

***Index Terms*— Iterative decoding, stopping criterion, turbo-codes.**

I. Introduction

TURBO (or iterative) decoding [1], [2] achieves an error performance close to the Shannon limit through decoding iterations and by using a large interleaver to provide sufficient randomness. Each decoding iteration results in additional computations and decoding delay. As the decoding approaches the performance limit of a given turbo-code, any further iteration results in very little improvement. Therefore, it is important to devise an efficient criterion to stop the iteration process and prevent unnecessary computations and decoding delay. One such stopping criterion has been devised based on the cross entropy (CE) between the distributions of the estimates at the outputs of the decoders at each iteration [2], [3]. This criterion is known as CE criterion. It effectively stops the iteration process with very little performance degradation. In this paper, we present two new stopping criteria which are simpler and computationally more efficient than the CE criterion. Both are devised based on the CE concept. Just like the CE criterion, they effectively stop the iteration process with very little performance degradation.

II. Turbo Decoding and Hagenauer's Stopping Criterion

For simplicity, we consider a turbo-code that consists of two rate-$1/n$ systematic convolutional codes with feedback. A two-dimensional iterative decoder is shown in Fig. 1. Let $\boldsymbol{u} = (u_1, u_2, \cdots, u_N)$ be an information block of length N and $\boldsymbol{v} = (\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_N)$ be the corresponding coded sequence, where $\boldsymbol{v}_k = (v_{k,1}, v_{k,2}, \cdots, v_{k,n})$, for $k \in \Lambda \triangleq \{1, 2, \cdots, N\}$ is the output code block at time k. Assuming BPSK transmission over an AWGN channel, u_k and $v_{k,l}$ all take values in $+1, -1$ for $k \in \Lambda$ and $1 \le l \le n$. Let $\boldsymbol{y} = (\boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_N)$ be the received sequence, where $\boldsymbol{y}_k = (y_{k,1}, y_{k,2}, \cdots, y_{k,n})$ is the received block at time k. Then $y_{k,l} = v_{k,l} + n_{k,l}$, where $n_{k,l}$ is a Gaussian random variable with zero mean and variance σ^2. Since the component convolutional codes are systematic, $v_{k,1}$ and $y_{k,1}$ correspond to the information bit u_k. In the following, we use $\hat{\boldsymbol{u}} = \{\hat{u}_1, \hat{u}_2, \cdots, \hat{u}_N\}$ to denote the estimate of $\boldsymbol{u}$.

At the ith iteration, let $L_m^{(i)}(\hat{u}_k)$ and $Le_m^{(i)}(\hat{u}_k)$ denote the log-likelihood-ratio (LLR) and the extrinsic values of the estimated information bit $\hat{u}_k$ delivered by decoder m, respectively, with $m = 1, 2$. It is shown in [2] that

$$L_1^{(i)}(\hat{u}_k) = Le_2^{(i-1)}(\hat{u}_k) + \frac{2}{\sigma^2}\, y_{k,1} + Le_1^{(i)}(\hat{u}_k) \tag{1}$$

$$L_2^{(i)}(\hat{u}_k) = Le_1^{(i)}(\hat{u}_k) + \frac{2}{\sigma^2}\, y_{k,1} + Le_2^{(i)}(\hat{u}_k) \tag{2}$$

where $(2/\sigma^2)y_{k,1}$ is the channel soft value. The probability distribution $p_m^{(i)}(\hat{u}_k)$ at the output of the mth decoder is given by [2]

$$p_m^{(i)}(\hat{u}_k = \pm 1) = \frac{e^{\pm L_m^{(i)}(\hat{u}_k)}}{1 + e^{\pm L_m^{(i)}(\hat{u}_k)}}. \tag{3}$$

At iteration i, the CE between the distributions $p_1^{(i)}(\hat{\boldsymbol{u}})$ and $p_2^{(i)}(\hat{\boldsymbol{u}})$ of the outputs of decoders one and two for an independently, identically distributed source $\boldsymbol{u}$ is defined as [2]

$$T(i) \triangleq E_{\boldsymbol{p}_2^{(i)}}\left\{\log \frac{p_2^{(i)}(\hat{\boldsymbol{u}})}{p_1^{(i)}(\hat{\boldsymbol{u}})}\right\} = \sum_{k=1}^{N} E_{p_2^{(i)}}\left\{\log \frac{p_2^{(i)}(\hat{u}_k)}{p_1^{(i)}(\hat{u}_k)}\right\} \tag{4}$$

where $E[X]$ denotes the expectation of a random variable X. This CE can be used to stop the iteration process in turbo decoding.

Let

$$\Delta Le_2^{(i)}(\hat{u}_k) = L_2^{(i)}(\hat{u}_k) - L_1^{(i)}(\hat{u}_k) = Le_2^{(i)}(\hat{u}_k) - Le_2^{(i-1)}(\hat{u}_k). \tag{5}$$

Suppose that the decoding iteration converges, and at iteration i the decoding process can be terminated. Then the following assumptions on the LLR's and the extrinsic values at the outputs of the two decoders can be made [2].

1) Hard decisions of the information bits based on their LLR values do not change anymore, i.e., $\operatorname{sign}(L_1^{(i)}(\hat{u}_k)) = \operatorname{sign}(L_2^{(i)}(\hat{u}_k)) = \hat{u}_k^{(i)} = \pm 1$.
2) The magnitudes of $L_1^{(i)}(\hat{u}_k)$ and $L_2^{(i)}(\hat{u}_k)$ are very large so that by (3) either $p_m^{(i)}(\hat{u}_k = 1) \approx 1.0$ or $p_m^{(i)}(\hat{u}_k = -1) \approx 1.0$, for $m = 1, 2$.
3) $\Delta Le_2^{(i)}(\hat{u}_k)$ has the same sign as $\hat{u}_k^{(i)}$.
4) The difference between the magnitudes of $Le_2^{(i-1)}(\hat{u}_k)$ and $Le_2^{(i)}(\hat{u}_k)$ are very small and less than 1.0. Hence, when there is no sign change between them,

Paper approved by S. S. Pietrobon, the Editor for Coding Theory and Techniques of the IEEE Communications Society. Manuscript received August 4, 1998; revised January 6, 1999. This work was supported by NSF under Grant NCR-94-15374 and Grant CCR-97-32959 and NASA under Grant NAG 5-931. This paper was presented at the 1998 IEEE International Symposium on Information Theory (ISIT), Cambridge, MA, August 1998.

The authors are with the Department of Electrical Engineering, University of Hawaii at Manoa, Honolulu, HI 96822 USA (e-mail: marc@aravis.eng.hawaii.edu).

Publisher Item Identifier S 0090-6778(99)06286-8.

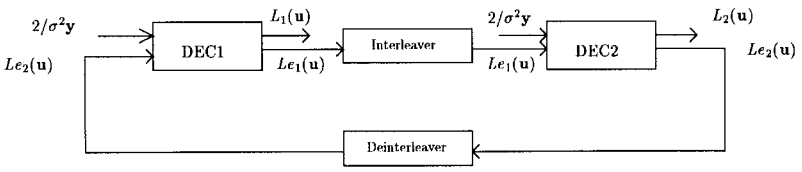

Fig. 1. Decoder for a two-dimensional turbo-code.

the values of $\Delta Le_2^{(i)}(\hat{u}_k)$ are negligible compared with those with sign changes, i.e., when $k \in \Lambda_s \triangleq \{j: \text{sign}(Le_2^{(i-1)}(\hat{u}_j)) = \text{sign}(Le_2^{(i)}(\hat{u}_j)), j \in \Lambda\}$, $\Delta Le_2^{(i)}(\hat{u}_k) \approx 0$.

Based on these assumptions, the CE in (4) can be approximated as follows [2]:

$$T(i) \approx \sum_k \frac{|\Delta Le_2^{(i)}(\hat{u}_k)|^2}{e^{\left|L_1^{(i)}(\hat{u}_k)\right|}}. \tag{6}$$

In [2], it is shown that when $T(i)$ drops to a value of $(10^{-2} \sim 10^{-4})T(1)$, the distributions $p_1^{(i)}(\hat{\boldsymbol{u}})$ and $p_2^{(i)}(\hat{\boldsymbol{u}})$ are "close enough" to terminate the iterative process with very little performance degradation. This is known as the CE criterion.

III. Two New Stopping Criteria

The SCR Criterion

Let $C(i)$ denote the number of sign changes in $Le_2(\hat{\boldsymbol{u}}) \triangleq (Le_2(\hat{u}_1), Le_2(\hat{u}_2), \cdots, Le_2(\hat{u}_N))$ from iteration $(i-1)$ to iteration i. The following criterion is a direct result of the CE criterion.

The approximation of $T(i)$ given by (6) can be written as the sum of two parts

$$\begin{aligned} T(i) &\approx \sum_{k\in\Lambda_s} \frac{\left|\Delta Le_2^{(i)}(\hat{u}_k)\right|^2}{e^{|L_1^{(i)}(\hat{u}_k)|}} + \sum_{k\in\Lambda\setminus\Lambda_s} \frac{\left|\Delta Le_2^{(i)}(\hat{u}_k)\right|^2}{e^{|L_1^{(i)}(\hat{u}_k)|}} \\ &= T_1(i) + T_2(i). \end{aligned} \tag{7}$$

Based on assumption 4), the values $|\Delta Le_2^{(i)}(\hat{u}_k)|^2$ which contribute to $T_1(i)$ are much smaller than those that contribute to $T_2(i)$. Furthermore, $|L_1^{(i)}(\hat{u}_k)|$ in $T_1(i)$ have much larger average values than those in $T_2(i)$. Therefore, $T_1(i)$ is assumed negligible compared with $T_2(i)$, and we have

$$T(i) \approx \sum_{k\in\Lambda\setminus\Lambda_s} \frac{\left|\Delta Le_2^{(i)}(\hat{u}_k)\right|^2}{e^{|L_1^{(i)}(\hat{u}_k)|}} \approx \delta_i C(i) \tag{8}$$

where δ_i is defined as the average value of $|\Delta Le_2^{(i)}(\hat{u}_k)|^2 / e^{|L_1^{(i)}(\hat{u}_k)|}$ for $k \in \Lambda \setminus \Lambda_s$.

Equation (8) shows that the number of sign changes in $Le_2(\hat{\boldsymbol{u}})$ between two consecutive iterations directly relates to the CE between distributions $p_1^{(i)}(\hat{\boldsymbol{u}})$ and $p_2^{(i)}(\hat{\boldsymbol{u}})$. This relationship provides a stopping criterion for iterative decoding based on the sign changes $C(i)$ in $Le_2(\hat{\boldsymbol{u}})$. Simulation shows that if $C(i) \le (0.005 \sim 0.03)N$, iterative decoding can be stopped with about the same performance degradation as the CE criterion used in [2]. The ratio $C(i)/N$ is called the ratio of sign changes. This stopping criterion is referred to as the sign-change-ratio (SCR) criterion.

In [2], $T(i)$ is divided by $T(1)$ to be normalized with respect to the signal-to-noise ratio (SNR). However this normalization no longer applies to our case. For a given SNR, we can normalize $C(i)$ by N, but the value of $C(i)/N$ takes a relatively wider range than $T(i)/T(1)$. Consequently, it has to be determined precisely from simulations. The basic ideas behind choosing a threshold are: 1) the smaller the threshold, the smaller the performance degradation, but a smaller number of iterations can be saved; 2) the threshold of SCR needs to be decreased when the interleaver size increases; 3) at the error floor region of turbo-codes, the threshold can be loosened up due to the fact that more oscillating patterns appear in iterative decoding.

Simulation results show that CE and SCR both save about the same number of iterations when the thresholds are properly chosen. However, the implementation of the SCR criterion is simpler than that of the CE criterion in terms of computational complexity and memory space requirement. At iteration i, the CE criterion requires a total of $(5N-1)$ real number operations and $(N+2)$ real number memory units for storing $L_1^{(i)}(\hat{\boldsymbol{u}})$, $T(1)$, and $T(i)$. However, the SCR criterion only needs $(3N-1)$ integer operations and $(N+2)$ integer memory units for storing the signs of $Le_2^{(i)}(\hat{\boldsymbol{u}})$, $C(1)$, and $C(i)$. The SCR criterion only needs the extrinsic values from the component decoders, but the CE criterion has to deal with both the extrinsic values and the LLR values. For large N, the SCR criterion results in large savings in computation and memory space.

The HDA Criterion

Although iterative decoding improves the LLR value for each information bit through iterations, the hard decision of the information bit is ultimately made based on the sign of its LLR value. The hard decisions of the information sequence at the end of each iteration provide information on the convergence of the iterative decoding process. As the decoding iteration converges to the final stage, we can modify the first assumption as follows.

1′) $\text{sign}\left(L_2^{(i-1)}(\hat{u}_k)\right) = \text{sign}\left(L_2^{(i)}(\hat{u}_k)\right) = \hat{u}_k^{(i)} = \pm 1.$

Based on assumptions 1′) and 2), the CE between $p_2^{(i-1)}(\hat{\boldsymbol{u}})$ and $p_2^{(i)}(\hat{\boldsymbol{u}})$ becomes

$$\begin{aligned} T'(i) &= \sum_{k\in\Lambda} E_{p_2^{(i)}}\left\{\log \frac{p_2^{(i)}(\hat{u}_k)}{p_2^{(i-1)}(\hat{u}_k)}\right\} \\ &\approx \sum_{k\in\Lambda} \left\{\hat{u}_k \Delta L_2^{(i)}(\hat{u}_k) + \log\frac{1+e^{|L_2^{(i-1)}(\hat{u}_k)|}}{1+e^{|L_2^{(i)}(\hat{u}_k)|}}\right\} \end{aligned} \tag{9}$$

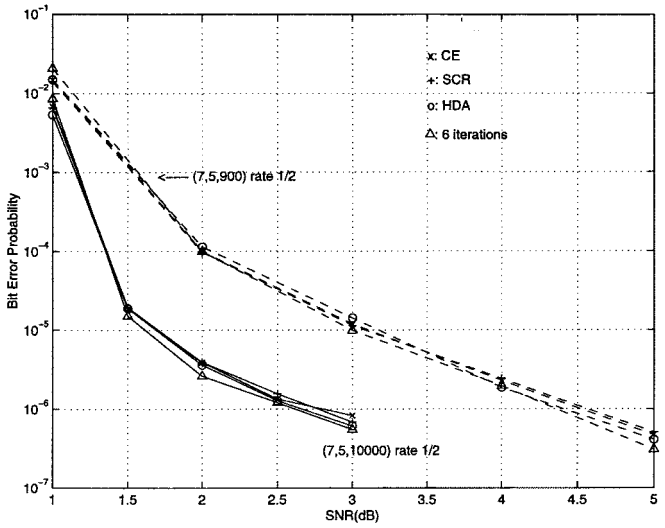

Fig. 2. BER of the (7, 5, N) codes with iterative MAP decoding using CE, SCR, HDA ($M = 6$).

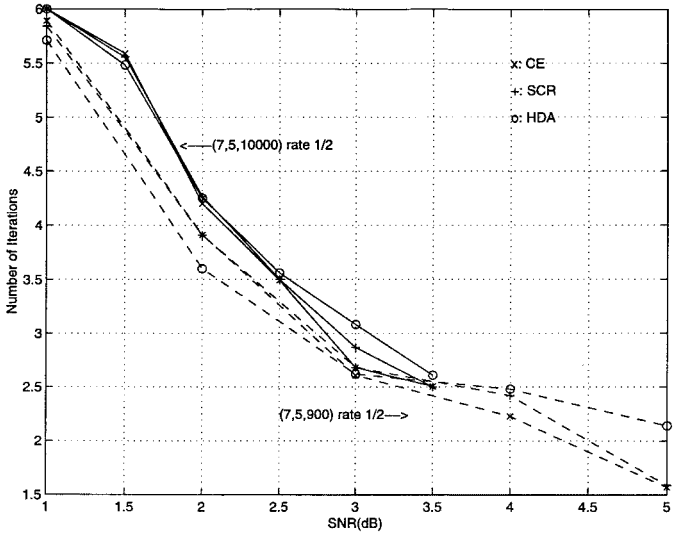

Fig. 3. Average number of iterations of the (7, 5, N) codes with iterative MAP decoding using CE, SCR, HDA ($M = 6$).

where $\Delta L_2^{(i)}(\hat{u}_k) \triangleq L_2^{(i)}(\hat{u}_k) - L_2^{(i-1)}(\hat{u}_k)$. Simulation shows that when $\text{sign}(L_2^{(i)}(\hat{u}_k)) = \text{sign}(L_2^{(i-1)}(\hat{u}_k))$ for all $k \in \Lambda$, $T'(i)$ is small enough for terminating the iterative process.

At iteration $(i-1)$, we store the hard decisions of the information bits based on $L_2^{(i-1)}(\hat{\boldsymbol{u}})$ and check the hard decisions based on $L_2^{(i)}(\hat{\boldsymbol{u}})$ at iteration i. If they agree with each other for the entire block, we simply terminate the iterative process at iteration i. This stopping criterion is called the hard-decision-aided (HDA) criterion. At each iteration, the HDA criterion requires N binary operations to obtain the signs of $L_2^{(i)}(\hat{\boldsymbol{u}})$ and at most N logic operations to check the sign changes (whenever a sign change happens, the HDA criterion is violated and the iterative process continues). It needs only N integer memory units for storing the signs of $L_2^{(i)}(\hat{\boldsymbol{u}})$.

Simulation shows that the HDA criterion saves more iterations at low to medium SNR's for small to medium interleavers than both CE and SCR criteria for similar bit-error-rate (BER) performances. However, at high SNR's, the HDA criterion is not as efficient as the CE and SCR criteria.

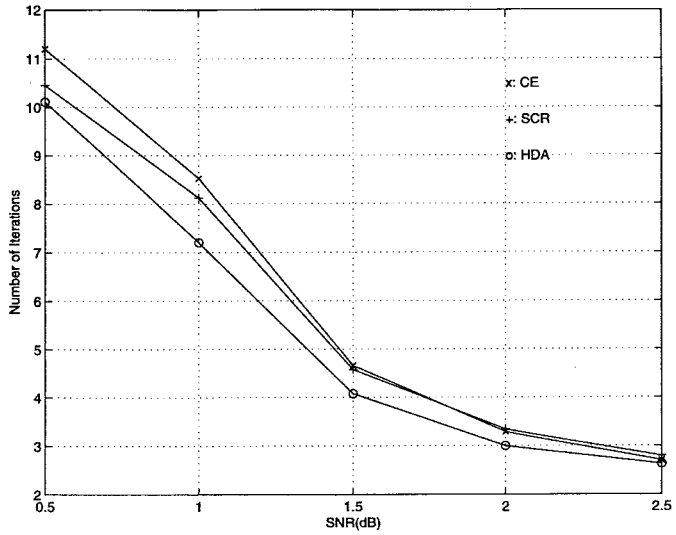

Fig. 4. Average number of iterations of the (37, 21, 400) code by iterative MAP decoding using CE, SCR, HDA ($M = 18$).

IV. Simulation Results and Comparisons

Simulations for turbo-codes using both convolutional and block component codes have been conducted, and only the results for convolutional component codes are presented in this section. The performance degradations and the reductions of decoding iterations using different stopping criteria for the iterative maximum *a posteriori* probability (MAP) probability decoding scheme are considered.

Simulation results for the 4-state (7, 5, N) and 16-state (37, 21, N) convolutional-component turbo-codes (expressed in octal form) are obtained for different interleaver sizes N and an overall rate $R = 1/2$. Fig. 2 depicts the bit-error performances of iterative MAP decoding for the (7, 5, 900) and (7, 5, 10^4) codes with the CE, SCR, HDA criteria and without stopping criterion when the maximum iteration number is set to $M = 6$. For both turbo-codes, the error performances with and without any stopping criterion are about the same. The corresponding average numbers of iterations required by the CE, SCR, and HDA criteria are shown in Fig. 3. For the CE criterion, the threshold $T(i)$ is set to $10^{-4}T(1)$. For the SCR criterion, when $N = 900$, the threshold $C(i)$ is set to $0.01N$ for SNR $\leq$ 3.0 dB and $0.03N$ for SNR $\geq$ 4.0 dB; when $N = 10^4$, $C(i)$ is set to $0.005N$. Fig. 4 shows the iteration reduction effects by the three stopping criteria for the (37, 21, 400) code with $M = 18$. The threshold of the CE criterion is set to $10^{-4}T(1)$, while that of the SCR criterion is $0.01N$.

V. Conclusion

In this paper, two new stopping criteria, the SCR and HDA criteria, for iterative decoding have been presented. They both

efficiently reduce the number of iterations with about the same error performance degradation as the CE criterion. Both the SCR and CE criteria follow the CE concept and save about the same number of iterations. However, the SCR criterion simplifies the CE in terms of computational complexity and memory space requirement. The HDA criterion uses the hard-decision information at the end of each iteration to stop the decoding iteration. It saves more iterations than the SCR and CE criteria for small to medium SNR's and small to medium interleaver sizes. Both the SCR and HDA criteria require only integer operations, while the CE criterion needs complex operations with real numbers. The SCR and HDA criteria also require less memory space than the CE criterion. Both the HDA and SCR criteria are simple enough for hardware implementation.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for improving the presentation of the results and Dr. S. S. Pietrobon, Editor, for pointing out a mistake in the original derivation of the SCR criterion and suggesting an alternative solution.

REFERENCES

[1] C. Berrou and A. Glavieux, "Near-optimum error-correcting coding and decoding: Turbo-codes," *IEEE Trans. Commun.*, vol. 44, pp. 1261–1271, Oct. 1996.

[2] J. Hagenauer, E. Offer, and L. Papke, "Iterative decoding of binary block and convolutional codes," *IEEE Trans. Inform. Theory*, vol. 42, pp. 429–445, Mar. 1996.

[3] M. Moher, "Decoding via cross entropy minimization," in *Proc. IEEE Globecom Conf.*, Houston, TX, Dec. 1993, pp. 809–813.