A Simple Stopping Criterion for Turbo Decoding

Yufei Wu, Brian D. Woerner, and William J. Ebel

Abstract—**This paper proposes a simple stopping criterion for turbo decoding that extends the existing sign change ratio (SCR) technique. The new sign difference ratio (SDR) criterion counts the sign differences between the *a priori* information and the extrinsic information. Unlike the SCR, it requires no extra data storage. Simulations comparing the new technique with other well known stopping criteria show that the proposed SDR scheme achieves similar performance in terms of BER, FER, and the average number of iterations, while requiring lower complexity. A GENIE scheme is included as the limit of all possible stopping criteria.**

Keywords— **Iterative decoding, stopping criteria, turbo codes.**

I. Introduction

Iterative decoding is a key feature of turbo codes. As the number of iterations increases, the bit error rate (BER) and frame error rate (FER) of the decoder decrease and the incremental improvement gradually diminishes. Often, a fixed number M is chosen and each frame is decoded for M iterations (called "FIXED" in the following). Usually M is set with the worst corrupted frames in mind. Most frames need fewer iterations to converge. It would reduce the average computation substantially without performance degradation if the decoder terminated the iterations for each individual frame immediately after the bits are correctly estimated.

While this is unrealistic when the transmitted bits are unknown, several schemes have been proposed to control the termination. For purpose of comparison, three existing terminating schemes are briefly reviewed in Section II. Then a new stopping criterion is proposed in Section III which requires no storage and minimal calculation. Simulation results are presented in Section IV to compare the new criterion with the existing schemes.

II. Existing Stopping Criteria Review

Three known dynamic stopping criteria are reviewed in the following. A maximum of M iterations are performed if the dynamic stopping criteria are not satisfied within M iterations.

1. **Cyclic Redundancy Check (CRC)** [1]: Under this approach, n_c CRC bits are appended to the end of each information frame and then the expanded frame is sent to the turbo encoder. In the decoder, following each iteration, the decoder makes hard decisions and the CRC bits are used to check for errors. Iteration is stopped when the CRC detects no error. Note that an outer code different than CRC, e.g., BCH code [2], could also be used.

The authors are with MPRG Lab at Virginia Tech. This work has been supported by the MPRG Industrial Affiliates Foundation and the Defense Advanced Research Projects Agency (DARPA) through the GloMo program.

2. **Cross Entropy (CE)** [3]: After each iteration i, the CE technique computes the approximate cross entropy $T(i)$ between log-likelihood ratios (LLRs) of the component decoders, where

$$T(i) = \sum_k \frac{|\Lambda_{2e}^{(i)}(u_k) - \Lambda_{2e}^{(i-1)}(u_k)|^2}{\exp(|\Lambda_1^{(i)}(u_k)|)},$$

$\Lambda_{2e}^{(i)}(u_k)$ is the extrinsic information of information bit u_k produced by component decoder two at iteration i, and $\Lambda_1^{(i)}(u_k)$ is the complete information of u_k produced by component decoder one. Iteration is discontinued if $T(i) < (10^{-2} \sim 10^{-4})T(1)$.

3. **Sign Change Ratio (SCR)** [4]: This technique is related to the CE technique [4]. It computes $C(i)$, which is the number of sign changes of the extrinsic information between iteration $(i-1)$ and i. Decoding is terminated when $C(i) \leq qN$, where q is a constant usually chosen to be $0.005 \leq q \leq 0.03$, and N is the frame size.

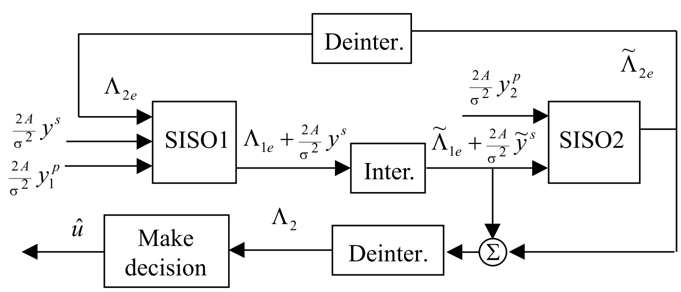

Fig. 1. A simplified turbo decoder.

Consider the simplified turbo decoder structure in Fig. 1. This form of the decoder minimizes the required storage of SISO input vectors, and reduces the computation inside SISO. The CRC method requires the transmission of n_c extra bits for error detection. For each iteration, the CRC method requires LLR computation(N real number additions) first and then CRC decoding using the hard decisions (at least $3N$ binary additions for commonly used CRC generator polynomials). The CE method requires $(6N-1)$ real number operations (including N additions to get $\Lambda_1^{(i)}(u_k)$) and $(N+2)$ real number memory units for storage [4]. The SCR technique requires only N binary additions, a counter no greater than N, and N bits to store the sign bits of the extrinsic information.

Although the SCR method is very simple, we show in the following a modified scheme which obviates the need for storage of values from the previous iteration. For notational convenience, we call this new technique the sign difference ratio (SDR).

III. A New Stopping Criterion

Consider a turbo code with two identical recursive systematic convolutional (RSC) codes. Let u_k, $k \in \{1, ..., N\}$, be the information bits which are BPSK modulated and transmitted through an $\mathcal{N}(0, \sigma^2)$ AWGN channel. At the receiver, $(y_k^s, y_{1k}^p, y_{2k}^p)$ are signals corresponding to u_k, where y_k^s is the systematic signal, y_{1k}^p and y_{2k}^p are parity signals for RSC1 and RSC2 respectively. They are sent to the soft-input soft-output (SISO) [5] Log-MAP decoders SISO1 and SISO2 to produce estimates $\hat{u}_k$.

At the decoder input $Y_k^s = (2/\sigma^2)y_k^s$ is the LLR representation of the measured information corresponding to u_k. In the i-th iteration, let $\Lambda_{1a}^{(i)}(u_k)$ and $\Lambda_{1e}^{(i)}(u_k)$ be the *a priori* information and the extrinsic information of SISO1 respectively; let $\Lambda_{2a}^{(i)}(u_k)$ and $\Lambda_{2e}^{(i)}(u)$ be those of SISO2. Then the LLRs at the output of SISO1 and SISO2 are $\Lambda_1^{(i)}(u_k)$ and $\Lambda_2^{(i)}(u_k)$:

$$\begin{cases} \Lambda_1^{(i)}(u_k) & = Y_k^s + \Lambda_{1a}^{(i)}(u_k) + \Lambda_{1e}^{(i)}(u_k) \\ \Lambda_2^{(i)}(u_k) & = Y_k^s + \Lambda_{2a}^{(i)}(u_k) + \Lambda_{2e}^{(i)}(u_k) \end{cases} \tag{1}$$

The iterative process is implemented by setting:

$$\begin{cases} \Lambda_{1a}^{(i)}(u_k) & = \Lambda_{2e}^{(i-1)}(u_k) \\ \Lambda_{2a}^{(i)}(u_k) & = \Lambda_{1e}^{(i)}(u_k) \end{cases} \tag{2}$$

According to (1), for $j = 1, 2$, the j-th LLR of u_k is composed of three estimates: Y_k^s from the channel directly, the *a priori* value $\Lambda_{ja}^{(i)}(u_k)$, and $\Lambda_{je}^{(i)}(u_k)$ obtained based on the code constraint. Among the three estimates, Y_k^s is fixed for every iteration, while $\Lambda_{ja}^{(i)}(u_k)$ and $\Lambda_{je}^{(i)}(u_k)$ are updated from iteration to iteration. Since the extrinsic information is used as the *a priori* information for the next decoder as specified by (2), $\Lambda_{ja}^{(i)}(u_k)$ and $\Lambda_{je}^{(i)}(u_k)$ are correlated for $i \geq 2$.

It is reasonable to expect that for a "good" (easy to decode) frame, $\Lambda_{je}^{(i)}(u_k)$ will agree with $\Lambda_{ja}^{(i)}(u_k)$ on the hard estimation $\hat{u}_k$ as the iteration converges. In other words, $\text{sign}(\Lambda_{je}^{(i)}(u_k))$ converges to $\text{sign}(\Lambda_{ja}^{(i)}(u_k))$ as the decoding proceeds, since the sign of the soft value results in a hard estimate of the desired bit. If $\text{sign}(\Lambda_{je}^{(i)}(u_k)) = \text{sign}(\Lambda_{ja}^{(i)}(u_k))$, then they are correlated. The *a priori* information will be a positive excitation to the extrinsic information and as a result $|\Lambda_{ja}^{(i)}(u_k)|$ increases with the sign intact. This implies that the *a priori* information of the next decoder is enhanced, and the extrinsic information of the next SISO will tend to have the same sign.

Let N_{ber} be the number of bit errors in a frame and let D_{ji} be the number of sign differences between $\Lambda_{ja}^{(i)}(u_k)$ and $\Lambda_{je}^{(i)}(u_k)$. The above speculations are supported by the following observations made from repeated simulations:

1. For a "bad" (hard to decode) frame, both $E[|\Lambda_{ja}^{(i)}(u_k)|]$ and $E[|\Lambda_{je}^{(i)}(u_k)|]$ do not increase significantly, but stay close to or lower than $E[Y_k^s]$.
2. For a "good" frame, both $E[|\Lambda_{ja}^{(i)}(u_k)|]$ and $E[|\Lambda_{je}^{(i)}(u_k)|]$ increase as i increases. When N_{ber} gets close to 0, $E[|\Lambda_{ja}^{(i)}(u_k)|]$ and $E[|\Lambda_{je}^{(i)}(u_k)|]$ are significantly ($5 \sim 10$ times) larger than $E[|Y_k^s|]$. Consequently, $\Lambda_1(u_k)$ is determined primarily by $\Lambda_{ja}^{(i)}(u_k) + \Lambda_{je}^{(i)}(u_k)$.
3. For a "bad" frame, D_{ji} stays high as i increases.
4. For a "good" frame, D_{ji} tends towards 0 as i increases, similar to N_{ber}. Usually, N_{ber} reaches 0 about ($0.5 \sim 1$) iteration earlier than D_{ji}.
5. Even after N_{ber} drops to 0, $|\Lambda_{ja}^{(i)}(u_k)|$, $|\Lambda_{je}^{(i)}(u_k)|$, and

The performance for $N = 200$ with code rate $r = 1/3$ is shown in Fig. 2 and 3, while that for $N = 5120$ and $r = 1/2$ is shown in Fig. 4 and 5. In both cases, all six schemes exhibit similar BER and FER performance. The simple SDR technique is as efficient as CE and SCR methods in terms of BER, FER, and the average number of iterations.

It is observed that the CRC method uses almost the same average number of iterations as the GENIE scheme, while the CE, SCR and SDR methods all require about one more iteration on average. However, the CRC method transmits extra bits (16 in this case) which compromises the bandwidth efficiency. The CRC technique is also more computationally expensive than the SCR and SDR schemes. In addition, it is observed that the BER of CRC at $E_b/N_0 = 1.2$ dB in Fig. 4 is significantly higher than that of the other techniques, which implies the occurrence of detection failure.

Fig. 5 shows that for the high BER region (BER$>$ 5×10^{-2}), the CE and SCR methods require fewer iterations than the others. This happens because at low BER, $\Lambda_{ja}^{(i)}$ is usually small, thus Y_k^s and Y_{jk}^p dominate in the computation. Since Y_k^s and Y_{jk}^p are invariant from iteration to iteration, the extrinsic information stabilizes quickly. This leads to premature termination, although the frame may be still in error. Since turbo codes are usually used in the low BER region (BER$<10^{-3}$), this effect can be ignored.

The performance of the SDR method is sensitive to p in the low BER region. As an example, Fig. 6 illustrates that BER and FER performance degrades significantly when p increases from 10^{-4} to 5×10^{-3} when BER could reach 4×10^{-7}, although the performance is not affected by p in the BER$> 10^{-4}$ region. Thus p should be chosen carefully at low BER region in order not to compromise the performance. Fig. 7 illustrates the increase of average number of iterations when p decreases. It is also shown that the SDR criterion can be used after both SISOs to save roughly 0.25 iteration on average without BER or FER degradation.

V. Conclusion

In this paper, we presented a new stopping criterion, the sign difference ratio, and compared it with four existing stopping schemes. It has been shown by simulations that the SDR criterion performs with negligible difference from CE and SCR methods. Like the previously reported SCR method, the new SDR method requires significantly less computation than the CE method, with the additional advantage of reduced storage requirement. Although the CRC method requires the fewest iterations, it results in much more computation than the SDR and SCR methods, and its potential failures at low BER (BER$\leq 10^{-6}$) result in degraded performance.

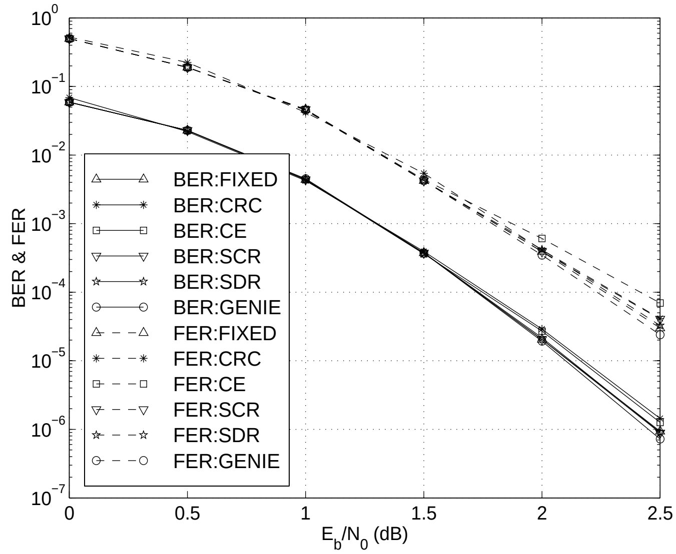

Fig. 2. BER and FER vs. E_b/N_0 for six stopping schemes: FIXED, CRC, CE, SCR($q = 10^{-2}$), SDR($p = 10^{-3}$) and GENIE (rate 1/3 (13,15,200) code, 8 maximum iterations).

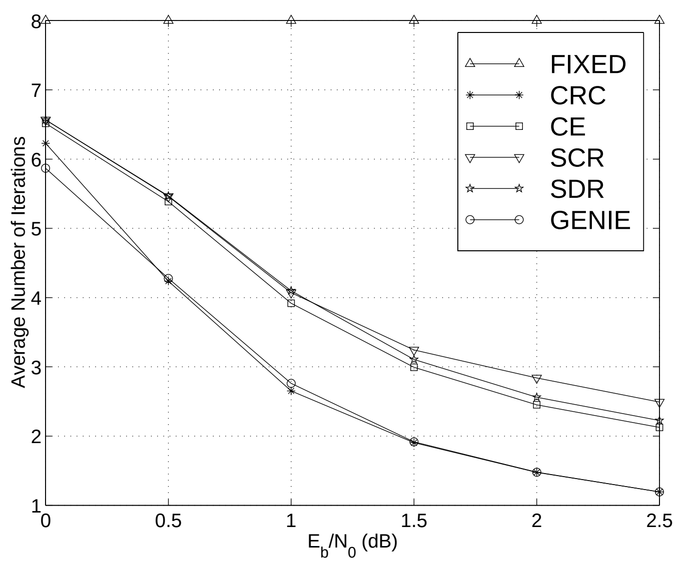

Fig. 3. Average number of iterations vs. E_b/N_0 for six stopping schemes: FIXED, CRC, CE, SCR($q = 10^{-2}$), SDR($p = 10^{-3}$) and GENIE (rate 1/3 (13,15,200) code, 8 maximum iterations).

References

[1] A. Shibutani, H. Suda, and F. Adachi, "Reducing average number of tubo decoding iterations," *Electronics Letters*, vol. 35, no. 9, pp. 701–702, Apr. 1999.

[2] O. Y. Takeshita, O. M. Collins, P. C. Massey, and D. J. Costello, "Concatenated codes that achieve 10^{-6} frame error rates at SNR's 0.6dB from the sphere packing bound," in *Proc., 36th Annual Allerton Conf. on Communication, Control, and Computing*, Sept. 1998.

[3] J. Hagenauer, E. Offer, and L. Papke, "Iterative decoding of binary block and convolutional codes," *IEEE Trans. Inform. Theory*, vol. IT-42, no. 2, pp. 429–445, Mar. 1996.

[4] R. Y. Shao, S. Lin, and M. P. C. Fossorier, "Two simple stopping criteria for turbo decoding," *IEEE Trans. Commun.*, vol. 47, no. 8, pp. 1117–1120, Aug. 1999.

[5] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara, "A soft-input soft-output maximum a posteriori (MAP) module to decode parallel and serial concatenated codes," *JPL TDA Progress Report*, vol. 42-127, Nov. 15, 1996.

[6] D. Agrawal and A. Vardy, "On the phase trajectories of the turbo-decoding algorithm," in *IMA Summer Program*, Aug. 1999.

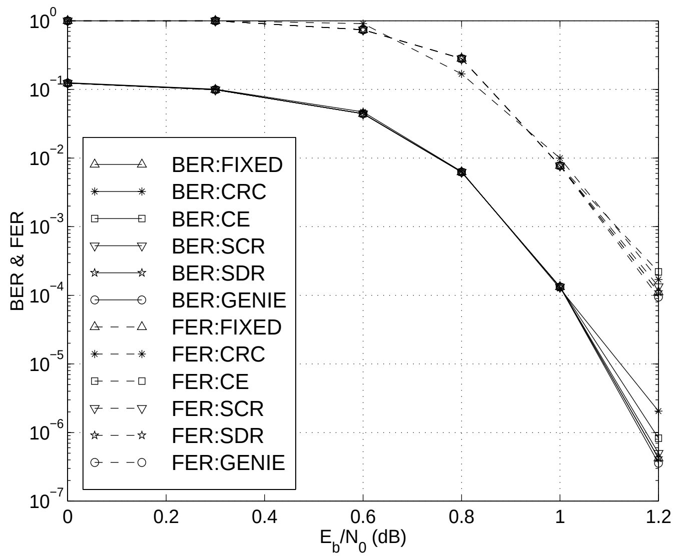

Fig. 4. BER and FER vs. E_b/N_0 for six stopping schemes: FIXED, CRC, CE, SCR($q = 10^{-3}$), SDR($p = 10^{-4}$) and GENIE (rate 1/2 (13,15,5120) code, 10 maximum iterations).

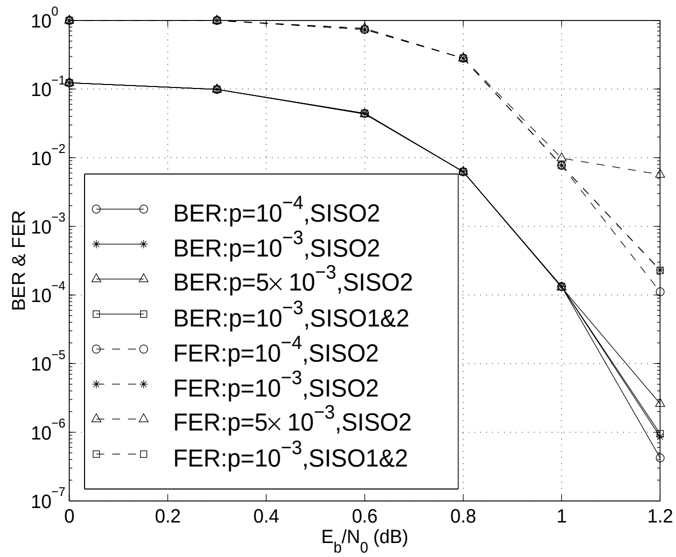

Fig. 6. BER and FER vs. E_b/N_0 for four SDR cases (rate 1/2 (13,15,5120) code, 10 maximum iterations). "SISO2" indicates that SDR criterion is used only on SISO2, while "SISO1&2" implies both SISO1 and SISO2.

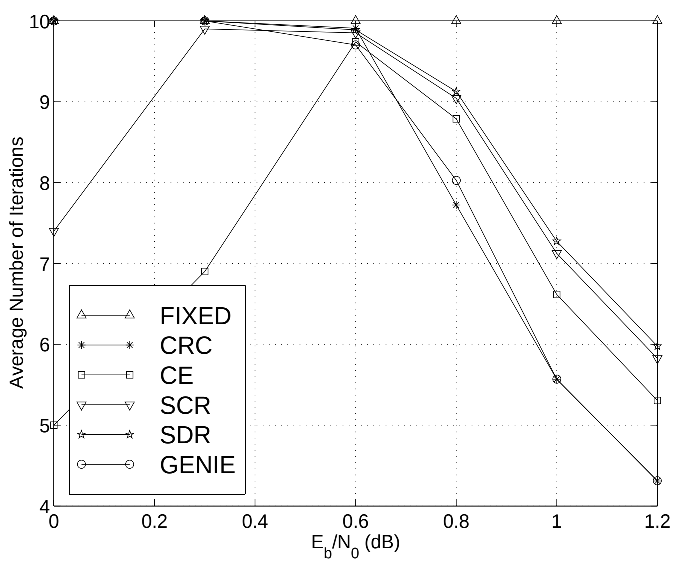

Fig. 5. Average number of iterations vs. E_b/N_0 for six stopping schemes: FIXED, CRC, CE, SCR($q = 10^{-3}$), SDR($p = 10^{-4}$) and GENIE (rate 1/2 (13,15,5120) code, 10 maximum iterations).

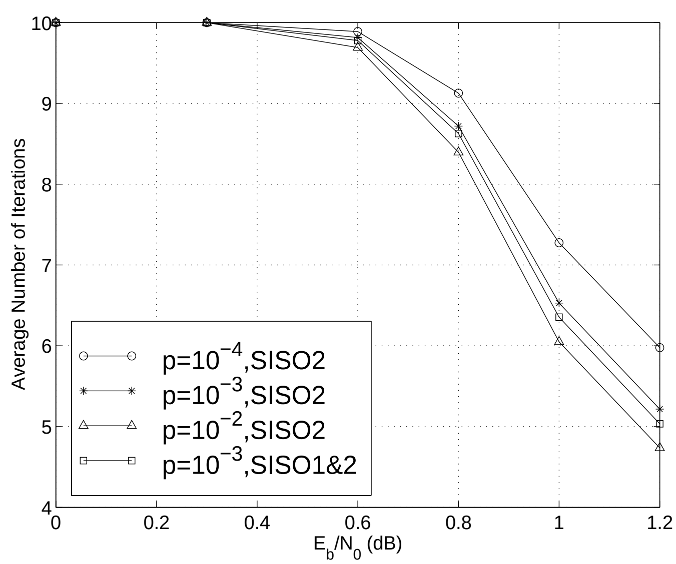

Fig. 7. Average number of iterations vs. E_b/N_0 for four SDR cases (rate 1/2 (13,15,5120) code, 10 maximum iterations). "SISO2" and "SISO1&2" are used as in Fig. 6.