A New Technique To Determine The Upper Threshold for Finite Length Turbo Codes

A.Rajesh and A.K.Chaturvedi

Department of of Electrical Engineering

Indian Institute of Technology Kanpur

INDIA-208016

Phone: +91-512-259 7897, Fax: +91-512-259 0063

email: rajesha@iitk.ac.in

ABSTRACT

There exist two SNR thresholds in finite frame length turbo codes. These thresholds depend on the component encoder as well as the frame length. A simple technique to find the upper threshold using the EXIT chart method is proposed.

INTRODUCTION

The existence of a threshold of SNR in iterative decoding was established by Richardson *et al* [1]. Agrawal *et al* [2] demonstrated the existence of two threshold points S_l and S_h that correspond to indecisive and unequivocal fixed points respectively, for finite frame length turbo codes. Concurrently, Gamal *et al* [5] and Brink [4] computed the threshold for turbo codes for infinite frame length. However, this value of threshold is not useful for finite frame length turbo codes as it falls in the indecisive fixed point region. In this paper, a model has been developed to find the S_h of a given turbo code using EXIT chart [4].

THE ERROR FLOOR AND THE S_h OF TURBO CODE

The error floor of the probability of bit error rate P_b^{EF} is a figure of merit of the turbo code performance. The computation of the same is given in [6] as

$$P_b^{EF} = \frac{N_{free}\tilde{w}_{free}}{N}\mathrm{erfc}\left(\sqrt{d_{free}R\frac{E_b}{N_0}}\right) \qquad (1)$$

where N is the interleaver size, d_{free} is the free distance, $\tilde{w}_{free}$ is the average weight of the information sequences causing free-distance code words, N_{free} is the multiplicity of free-distance codewords and R is the rate of the code.

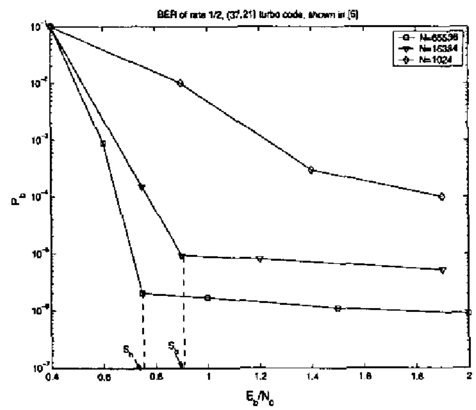

Fig. 1. Simulation results for (37,21) turbo code [6]

From Fig.1 the S_h can be seen as abscissa of the origin of error floor in the BER curve. Hence the problem of finding the upper threshold reduces to the computation of the abscissa of this origin. shown in Fig.4(a).

Crux of the turbo decoding

The turbo decoding can be characterised as [3]

(i) Increasing of extrinsic information by exchange of soft information through iteration.

(ii) The stopping of iteration and making the hard decision of log-likelihood value (L-value) after the cross entropy between the two *a posteriori* distributions of decoding operation crosses the earlier set threshold.

0-7803-8964-6/05/$20.00 ©2005 IEEE

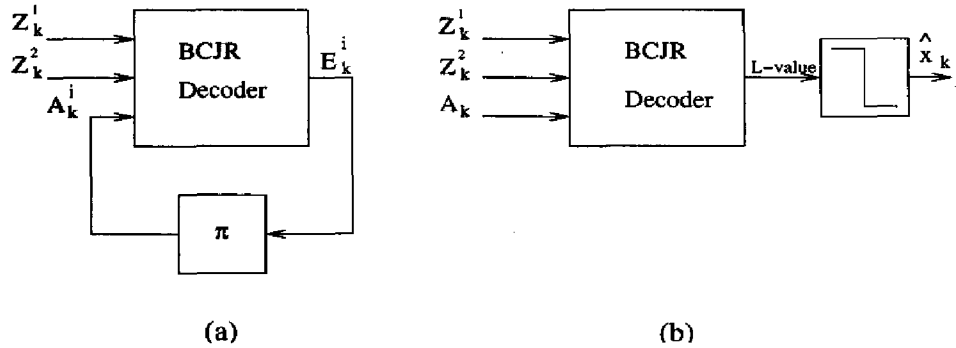

Fig. 2. BCJR decoder in feedback mode and open loop mode

Consider a hypothetical situation where we can decouple these two operations and treat them separately.

The feedback mode of the BCJR decoder

The feedback mode of the BCJR encoder is depicted in Fig.2(a). The Z_1^k and Z_2^k are the additive white Gaussian noise (AWGN) affected k^{th} systematic and parity bit respectively. The extrinsic information for k^{th} systematic bit at i^{th} iteration is denoted by E_i^k and it is fed back to the input along with channel symbols as the *a priori* value A_i^k after scrambling through interleaver π. This process of iteration will continue till the value $I(X;E)$ is saturated to I_{max} where $x \in \{\pm 1\}$ is the systematic bit at the recursive systematic convolutional (RSC) encoder output. A typical curve of I_{max} versus SNR is plotted in Fig.3. The computation of $I(X;E)$ has been done by taking the time average as shown in [7].

$$I(X;E) \simeq 1 - \frac{1}{N}\sum_{k=1}^{N} \log_2\left(1 + e^{-x_k E_k}\right) \qquad (2)$$

The open loop mode of BCJR decoder

The BCJR decoder is kept in the open loop mode with a hard decision making as shown in Fig.2(b). The channel symbols are fed along with the *a priori* value, which is a symmetric Gaussian ($\mu = \frac{\sigma^2}{2}$) distributed random variable. The variance of the same will be gradually increased for a given SNR to attain the error floor that can be derived from (1). The value of required variance σ_{ef}^2 is noted.

Computation of the threshold value $\hat{S}_h$

From I_{max} it is possible to compute the value of σ_{max}^2 using $I(X;A)$ (where A is symmetric Gaussian distributed random variable) which is computed in [4] as

$$I(X;A) = I_A(\sigma_A) = 1 - \int_{-\infty}^{\infty} e^{-((\xi-\sigma_A^2/2)^2/2\sigma_A^2)} \times \log_2(1+e^{-\xi})d\xi \qquad (3)$$

and

$$J(\sigma_{max}) := I_{max}(\sigma = \sigma_{max}) \qquad (4a)$$

$$\sigma_{max} = J^{-1}(I_{max}). \qquad (4b)$$

The computed σ_{max}^2 (monotonously increasing) and σ_{EF}^2 (monotonously decreasing) are plotted against E_b/N_0. The abscissa of intersection of these two curves will give the required $\hat{S}_h$ as shown in Fig.4(a).

SIMULATION AND DISCUSSION

Here, we have considered $(7,5)$ turbo code of rate $1/3$ with 1024 frame size. The σ_{max}^2 and σ_{ef}^2 are computed for different SNRs and plotted as shown in Fig.4(a). The intersection of these two curves gives the estimated value of $\hat{S}_h = 1.135$. Secondly the turbo code has been simulated with (7,5) as the component encoder. From the BER curve of the same, it is possible to find the actual value of $S_h = 1.28$ (as shown in Fig.4(b)) which is reasonably close to the computed earlier $\hat{S}_h$.

The advantage of the proposed technique is the lesser number of frames required to estimate the value of S_h. For elucidation let us take the example of $(7,5)$ code. Through simulation it is found that for the above turbo code the minimum number of errors in an erroneous frame with high SNR is 23. This implies that to get the error floor of $5\times10^{-6}(P_b^{EF})$ one has to run the simulation for at least 4500 frames, whereas

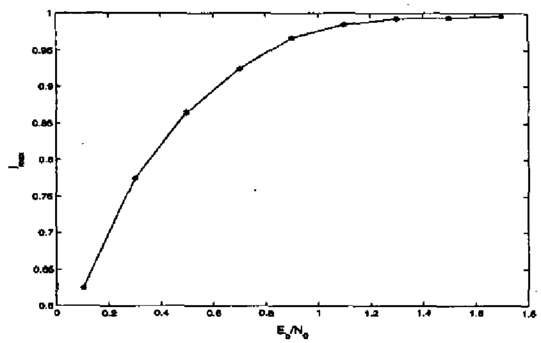

Fig. 3. I_{max} versus E_b/N_0 for (7,5) turbo code

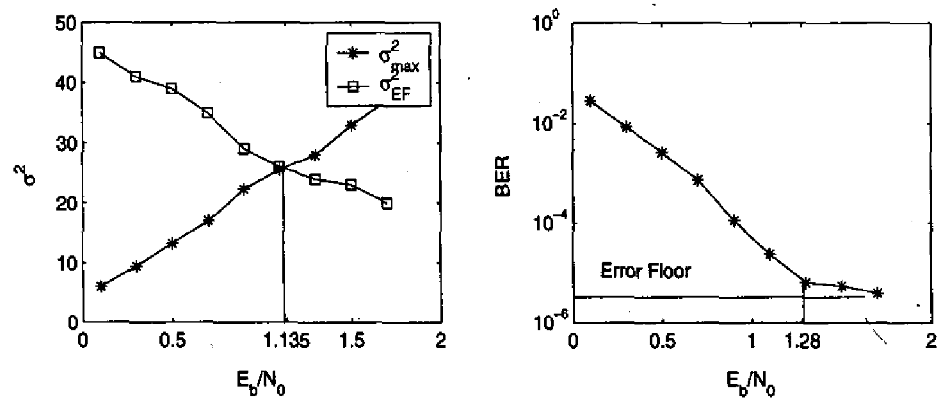

Fig. 4. The $\hat{S_h}$ computation and the actual S_h of (7,5) turbo code

with the proposed technique 700 frames suffice to get a value close to S_h.

CONCLUSION

A simple technique to find the upper threshold of finite frame length turbo codes has been proposed. It is based on the computation of mutual information of the exchanged soft information between successive iterations.

REFERENCES

[1] T. Richardson and R. Urbanke, "The capacity of low-density paritycheck codes under message-passing decoding", *IEEE Transactions on Information Theory* Vol. 47,pp. 619-637, 2001

[2] H. Agrawal and A. Vardy, "The turbo decoding and its phase trajectories", *IEEE Transactions on Information Theory* Vol. 47,No. 2,pp. 699-722, 2001

[3] J. Hagenauer and E. Offer, "Iterative decoding of binary block and convolutional codes", *IEEE Transactions on Information Theory* Vol. 42,No. 1,pp. 429-431, 1996

[4] S.t. Brink, "Convergence behavior of iterative decoded parallel concatenated codes", *IEEE Transactions on Communications* Vol. 49,No. 10,pp. 1727-686, 2001

[5] H.E. Gamal and A.R. Hammons, "Analyzing the turbo decoder using the Gaussian approximation", *IEEE Transactions on Information Theory* Vol. 47,No. 2,pp. 671-686, 2001

[6] J. Seghers, L.C. Perez and D.J. Costello jr, "A distance spectrum interpretation of turbo codes", *IEEE Transactions on Information Theory* Vol. 47,No. 6,pp. 1698-1709, 1996

[7] M. Tuchler, "Convergence prediction for iterative decoding of threefold concatenated systems", *Proceedings of the Globecom* 2002