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

A.Rajesh and A.K.Chaturvedi
Department of Electrical Engineering
Indian Institute of Technology Kanpur
INDIA-208016
Phone: +91-512-259 7897, Fax: +91-512-259 0063
e-mail: 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 [3] 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 \( P_{BF} \) is a figure of merit of the turbo code performance. The computation of the same is given in [6] as

\[
P_{BF} = \frac{N_{free} \bar{d}_{free} \text{erfc} \left( \sqrt{\frac{d_{free} R E_b}{N_0}} \right)}{N}
\]

(1)

where \( N \) is the interleaver size, \( d_{free} \) is the free distance, \( \bar{d}_{free} \) 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.

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.
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 Zf and Zj are the additive white Gaussian noise (AWGN) affected kth systematic and parity bit respectively. The extrinsic information for bth systematic bit at fth iteration is denoted by E and it is fed back to the input along with channel symbols as the a priori value Af after scrambling through interleaver .

This process of iteration will continue till the value $I(X;E)$ is saturated to $I_{max}$ where $zf \in \{1\}$ 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].

$$I(X;E) \simeq 1 - \frac{1}{N} \sum_{k=1}^{N} \log_{2} \left(1 + e^{-2 \sigma^{2}E_{k}} \right)$$  \hspace{1cm} (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 $A_{f}^{k}$ after scrambling through interleaver $\pi$. This process of iteration will continue till the value $I(X;E)$ is saturated to $I_{max}$ where $x \in \{1\}$ 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].

$$I(X;A) = I_{A}(\sigma_{A}) = 1 - \int_{-\infty}^{\infty} e^{-((k-\sigma^{2}/2)/2\sigma^{2})} \times \log_{2}(1 + e^{-\xi}) \, d\xi$$  \hspace{1cm} (3)

and

$$J(\sigma_{\text{max}}) := I_{\text{max}}(\sigma = \sigma_{\text{max}})$$  \hspace{1cm} (4a)

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

The computed $\sigma_{\text{max}}^{2}$ (monotonously increasing) and $\sigma_{\text{EF}}^{2}$ (monotonously decreasing) are plotted against $E_{b}/N_{0}$. The abscissa of intersection of these two curves will give the required $\tilde{S}_h$ 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 $\sigma_{\text{max}}^{2}$ and $\sigma_{\text{EF}}^{2}$ are computed for different SNRs and plotted as shown in Fig.4(a). The intersection of these two curves gives the estimated value of $\tilde{S}_h = 1.35$. 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 $\tilde{S}_h$.

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 $5 \times 10^{-5}$($P_{\text{EF}}$) one has to run the simulation for at least 4500 frames, whereas
with the proposed technique 700 frames suffice to get a value close to $S_b$.

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