WATERFALL REGION ANALYSIS FOR ITERATIVE DECODING

Ali Özgür Yılmaz

Middle East Technical University, Dept. of Electrical and Electronics Eng., Ankara, Turkey, aoyilmaz@metu.edu.tr

Abstract - Finite length analysis of iterative decoders can be done by using probabilistic models based on EXIT charts. The validity of these models will be investigated by checking the performance of iterative decoding under various scenarios.

Keywords - turbo, iterative, log-MAP,max-log-MAP, EXIT, finite.

I. Introduction

For over a decade turbo codes [1] and turbo decoding attracted much attention from the channel coding community. Turbo codes achieve error rate performance close to the limits predicted by Shannon. Moreover, these codes achieve such good performance with reasonable complexity by the use of a subtoptimal iterative decoding scheme. An exact analysis on the performance of turbo codes is currently out of reach due to the interleaving operation and the *ad hoc* nature of iterative decoding. Therefore, various researchers offered classifications on the convergence characteristics of turbo decoders [2], [3]. These attempts to classify different modes of convergence brought out two basic modes of erroneous operation of a turbo decoder [4]: Maximum likelihood (ML) decoding errors and decoding failures. It is conjectured in the literature that turbo decoders operate optimally at high signal-to-noise ratio (SNR) region. ML decoding errors happen in this region and cause an error floor in the error rate curves. The probability of ML decoding errors can be found by determining the distance spectrum of a code [5], [6]. Decoding failures occur when a turbo decoder fails to provide the optimal result. These errors occur prevalently at lower SNR where a rapid improvement in the turbo code performance is observed (waterfall region). The likelihoods of these two types of errors are combined to obtain the error rate of a turbo code.

In this paper, our focus is on the probability of decoding failures in the case of finite block length turbo codes. When investigated independently from other component decoders, the isolated behavior of a component decoder help model the turbo decoder based on the component decoders [7], [8]. Once the isolated behavior of component decoders are obtained, the EXIT chart method [8] can be used to obtain a threshold above which the considered system allows error-free communications for infinite block lengths. We will extend the use of EXIT charts to predicting the finite length performance of iterative decoders. The method that uses EXIT chart for finite length analysis will be referred to as finite-EXIT.

Finite-EXIT method is contingent on the wide-sense stationarity (WSS) property of extrinsic information. The regular Gaussian approximation to extrinsic information was modified in [9] to incorporate the WSS property. In this paper, we will present some examples of decoders that can produce WSS extrinsic information sequences. A probabilistic model of iterative decoding will be proposed. Also, an algorithm to obtain packet error rate (PER) approximations will be derived. We will verify the validity of this approach by studying different decoding algorithms and concatenation scenarios. Bit error rate approximations will not be pursued in this paper but can be easily obtained with a small modification from PER plots.

The outline of the paper is as follows. The basic properties on which the finite-EXIT method is based will be briefly explained in Section II. The error rate approximation method will be presented in Sections III and IV. Section V will be devoted to numerical results and the paper will be concluded with Section VI. Convolutional component codes and additive white Gaussian noise (AWGN) channel model are used in this study. The original turbo code structure (overall rate-1/3, two component codes of rate-1/2) is utilized unless noted otherwise.

II. WSS Property of Extrinsic Information

Finite length decoding causes a random variation in the input/output relation of component decoders which are used in the EXIT chart method. This variation of the IO relation can be accurately modelled by studying the statistical properties of extrinsic information. When the extrinsic information sequences are Gaussian and wide-sense stationary, the IO relation can be modelled with a Gaussian random process. Due to space limitations, we cannot provide more information on this and thus refer interested readers to [4]. Conditioned on an I_{in} value, the information content of extrinsic information produced by a decoder is Gaussian with some mean and variance. The mentioned mean and variance can be obtained from practically large packets, say $100,000$ data bits. The random model for iterative decoding process will be used in Section IV in order to obtain error rate approximations.

The finite-EXIT method is based on the wide-sense stationarity of the extrinsic information from component decoders. Most trellis-based soft decoding algorithms produce extrinsic information with this property. The log-MAP and max-log-MAP algorithms [10] are considered in this study. Extension to other algorithms are possible.

III. Decoding Failures at Finite Lengths

As it was mentioned in Section II, the IO relation of a decoder is probabilistic. Although the EXIT chart method cannot be of direct help for finite block lengths, the basic idea can still be applied. The basic idea is to extract some property from the IO relations of component decoders that can be used to predict the behavior of iterative decoding. For the infinite length case, this property is stated as the intersection rule or "$I_{i+2} > I_i$" [8], where

$$I_j = \frac{1}{K}\sum_{i=1}^{K} 1 - \log_2(1 + e^{z_i^j}) \tag{1}$$

denotes the information content of the extrinsic information sequence at iteration j. Through experimentation, we will try to decide on such a rule to be used for finite block lengths that is both practical and has good prediction capability.

In order to approximate the probability of decoding failure in some way, a rule similar to "$I_{i+2} > I_i$" is necessary. In the case of incorrect decoding (with regard to decoding failure), the iterative process cannot go beyond some value of $I(E)$ (information content of extrinsic information sequence) away from $I(E) \approx 1$. It is either stuck at a fixed point or oscillates around some point. In other words, $I(E)$ takes values around a point. Therefore, any rule, be it "$I_{i+1} > I_i$", "$I_{i+2} > I_i$" etc., can detect a decoding failure. Then, the accuracy of a rule is contingent on the frequency that the rule can predict the correct decoding of a packet. For this purpose, we ran simulations to follow iterative decoding trajectories and evaluate how frequently some given rules can predict correct decoding.

In Fig. 1 the percentage of correctly decoded packets which obey a particular rule is plotted as a function of SNR. The rules are k-step rules such that decoding is deemed correct whenever $I(E)$ keeps on increasing in every k iterations. The turbo code has rate-1/3 and asymmetric. The generator matrices of the component codes are $[1, 5/7]$ and $[1, 33/23]$, both in octal form.In general, it should be expected that rules that can allow more steps permit more failures to recover and thus should perform better. This is almost what is observed in Fig. 1. The 1-step rule is quite bad. The 2-step rule has a considerable performance improvement in comparison to the 1-step rule. The 3-step rule performs worse than the 2-step. (This is due to the asymmetry of IO relations of the component decoders and happens for every consecutive even to odd number of iterations.) The 4-step rule is better than the 2-step. However, the difference is not as striking as the difference between the 1-step rule and the 2-step.

As seen in Fig. 1 all the rules perform better as the SNR increases. The 2-step rule "$I_{i+2} > I_i$" is the rule by which the decoding threshold for infinitely large block lengths can be determined. The 2-step rule also performs quite well in the finite length case. The simplicity of this rule compared to higher order rules should be kept in mind as well. Based on all these arguments, we choose this rule to be used for evaluating the probability of iterative decoding failure.

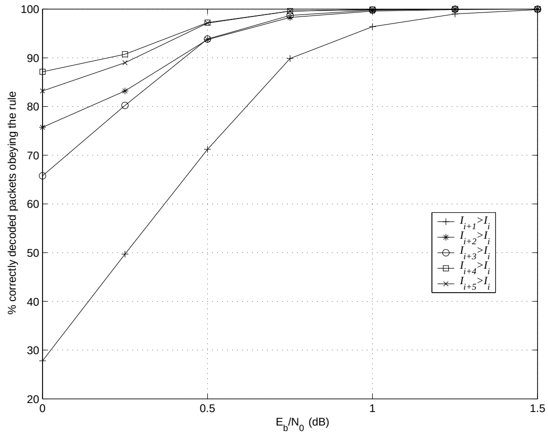

Fig. 1
The percentage of correctly decoded packets that obey a certain rule.

IV. A Markov Model for Iterative Decoding

The extrinsic information produced by one decoder is fed as the *a priori* information to the other decoder in the next iteration. The Gaussian assumption asserts that *a priori* information sequences that are fed into a component decoder at different iterations are independent. This suggests that the extrinsic information produced by a component decoder at any iteration depends only on the observation and the *a priori* information at that iteration. It does not depend on the *a priori* information at earlier iterations. Consequently, the Gaussian assumption implies a memoryless property in iterative decoding so that past *a priori* information can be ignored. This further implies that I_{k+1}, as defined in the last section, only depends on I_k when conditioned on a given observation. We will make use of this Markov property in order to approximate the probability of decoding failure.

Definition 1: Let $\{X_k, k = 1, 2, 3, \ldots\}$ be a discrete-time stochastic process with state space S. The conditional density function of X_{t_n} is denoted by $f_{X_{t_n}|X_{t_{n-1}},\ldots,X_{t_1}}(\cdot|\cdot,\ldots,\cdot)$. $\{X_k, k = 1, 2, 3, \ldots\}$ is said to be a Markov process when, for any collection of $t_1 < t_2 < \ldots < t_n \in T$ and $x_1, x_2, \ldots, x_n \in S$,

$$f_{X_{t_n}|X_{t_{n-1}},\ldots,X_{t_1}}(x_n|x_{n-1},\ldots,x_1) = f_{X_{t_n}|X_{t_{n-1}}}(x_n|x_{n-1}),$$

provided that both probability density functions exist.

By the above definition of a Markov process, the iterative decoding process represented by the $\{I_k\}$ sequence forms a Markov process. The transition probability function of this chain ($f_{I_{k+1}|I_k}(\cdot|\cdot)$) is time-invariant and given by the IO relation. As discussed in Section II, the probability density function $f_{I_{k+1}|I_k}(\cdot|\cdot)$ can be approximated as Gaussian. By this approximation, the state space S of this process is the range of normal distribution. The state space S is continuous and thus hard to work with. A Markov chain is a Markov process with finite state space and easier to work with. It is shown in [12] that continuous state space Markov processes can be discretized into Markov chains. We will simply use this result without much elaboration. The range of interest in our case is $[0,1]$ since that is the domain of mutual information function from which the information content function is derived. Thus, we will take S to be bounded from below and above by 0 and 1 respectively. Bounding S in this manner does not significantly affect the results since information content of extrinsic information sequences seldom lies out of $[0,1]$.

The state space S will be discretized and taken to be a finite set. The effect of discretization levels will be investigated later. The following form of the partitions of the interval $[0,1]$ will be considered,

$$S = \bigcup_{i=1}^{n} q_i, \tag{2}$$

where q_i's are continuous intervals and have the property that

$$x \in q_i, y \in q_{i+1} \Rightarrow x < y, \tag{3}$$

for all x, y and i. The set q_i will be referred to as the quantization level q_i from this point on. We have the following approximation to the transition probability from level q_j to q_i

$$p_{I_{k+1}|I_k}(q_i|q_j) = \begin{cases} \int_{-\infty}^{\sup_{x\in q_i} x} f_{I_{k+1}|I_k}(x|\bar{q}_j)dx, & i = 1 \\ \int_{\inf_{x\in q_i} x}^{\sup_{x\in q_i} x} f_{I_{k+1}|I_k}(x|\bar{q}_j)dx, & 1 < i < n \\ \int_{\inf_{x\in q_i} x}^{\infty} f_{I_{k+1}|I_k}(x|\bar{q}_j)dx, & i = n \end{cases} \tag{4}$$

for any k, q_i, and q_j where $\bar{q}_j = \frac{\inf_{x\in q_j} x + \sup_{x\in q_j} x}{2}$. If the time-invariant nature of the chain is considered, then we can simplify notation as

$$p_{ji} = p_{I_{k+1}|I_k}(q_i|q_j) = p(q_i|q_j). \tag{5}$$

We will evaluate the likelihood of first decoding failure at the i^{th} iteration $P_{df}(i)$. By the "$I_{i+2} > I_i$" rule, a decoding failure occurs at the i^{th} iteration if $I_{i+2} < I_i$. Define an indicator function that determines whether any decoding failure occurred up to iteration i by

$$N_{df}(i) = \begin{cases} 1, & \text{if } \exists j : I_j < I_{j-2}, j \le i \\ 0, & \text{otherwise .} \end{cases} \tag{6}$$

We will slightly change the rule so that it will be assumed convergence to correct decoding occurs when $I_k = q_n$, i.e., I_k is large, and $N_{df}(k-1) = 0$ for some k. By this change, q_n becomes an absorbing state of the Markov chain, i.e., $p_{nn} = 1$, which means that the chain stays indefinitely in state q_n whenever the process enters it. By the above definitions and explanations,

$$P_{df}(i) = P(I_i < I_{i-2}, N_{df}(i-1) = 0) \tag{7}$$

and

$$P_{df} = \sum_{i=1}^{\infty} P_{df}(i). \tag{8}$$

These values can be effectively found by using matrix operations as described in [9].

V. Numerical Results

For all the results that will be presented in this section, the properties of extrinsic information sequences are obtained at various SNR values with various values of I_{in}. Based on the properties of the extrinsic information, the IO relation for each SNR and I_{in} value is found. In most cases, the I_{in} values in steps of 0.1 are used and the IO relation is obtained by interpolation. Since the IO relation is smooth, this interpolation does not pose any threat and it effectively reduces the complexity of the method.

A. Quantization

We will first explain the quantization method employed and then present results on the effect of quantization. For any quantization step q_{step} we will have the following quantization levels:

$$\begin{aligned} q_1 &= [0, \frac{q_{step}}{2}) \\ q_2 &= [\frac{q_{step}}{2}, \frac{3}{2}q_{step}) \\ &\dots \\ q_{n-1} &= [I_{max} - \frac{3}{2}q_{step}, I_{max} - \frac{q_{step}}{2}) \\ q_n &= [I_{max} - \frac{q_{step}}{2}, 1]. \end{aligned}$$

We will set I_{max} value to 0.95 (rather than 1) in this study. This value of I_{max} was determined based on the simulation results for determining the correct decoding rule. It was observed that when a packet reaches $I(E) \approx 0.95$ it is almost always correctly decoded. It follows from the description of quantization that the smaller the number q_{step} is, the finer the quantization gets.

The effect of quantization is shown in Fig. 2. A symmetric turbo code with component code generator matrix $[1, \frac{5}{7}]$ is used. The three plots correspond to the P_{df} estimates for three different block lengths $K = 256, 1024$, and 4096. As seen in all the plots, finer quantization increases the P_{df} estimate since more states will be deemed as failed in that case. However, the estimate differences between different values of q_{step} are not very significant. The differences are also decreasing so that there is sign of a convergent behavior. We will choose $q_{step} = 0.005$ to be used for this paper. This value of q_{step} seems to be sufficient since the error rates of interest for the codes we investigate is up to packet error rate $PER = 10^{-5}$.

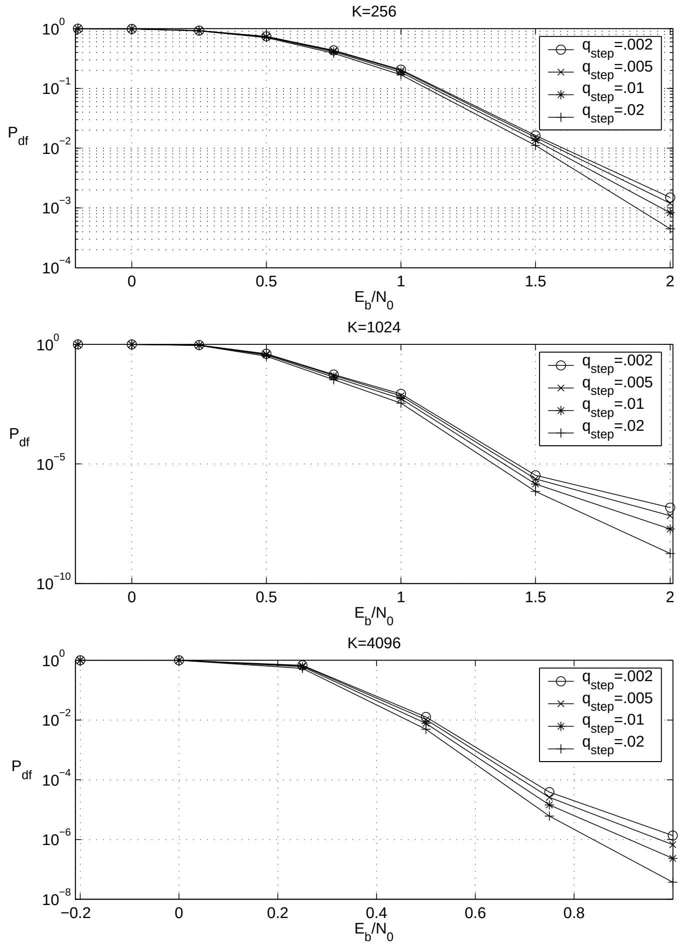

Fig. 2

The effect of quantization on P_{df} estimation.

B. Error rate approximations

In all the simulations in this section, the turbo codes were run for 100 iterations (each decoder is run 50 times). At least 50 packet errors have been collected in all cases. We will first consider a symmetric turbo code with component code generator matrix $[1, 33/23]$, 1024 data bits and S-random interleaving. Performance with two different decoding algorithms are shown in Fig. 3. Simulation and P_{df} estimate curves are plotted with solid and dashed lines respectively. The lines marked with '*' correspond to decoding with max-log-MAP algorithm. The log-MAP algorithm performs around 0.5dB better compared to the max-log-MAP algorithm. The P_{df} estimates are found by obtaining the IO relations for the used component decoder. In both cases, the P_{df} estimates closely follow the packet error rate curves in the waterfall region. The 0.5dB performance difference is also seen in the P_{df} estimates.

As observed in the figure, the P_{df} estimate behaves as an upper bound to the actual P_{df}. In the model we defined, it was said that the rule "$I_{i+2} > I_i$" does not perfectly predict the correct decoding of a packet. Thus, the developed model used with this rule offers a worst-case scenario so that its

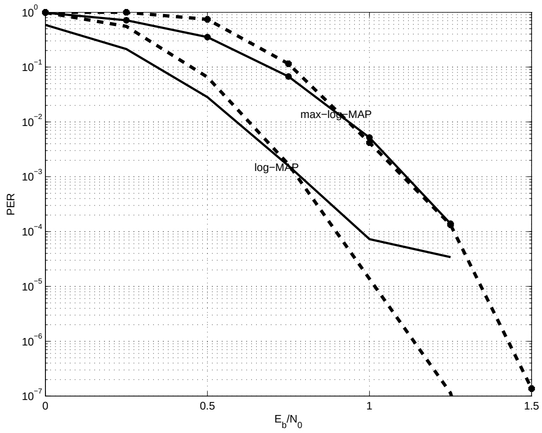

Fig. 3

Packet error rate estimates and simulation results of a symmetric turbo code with two different decoding algorithms.

estimate behaves like an upper bound.

We also consider serial concatenation of two convolutional codes. The serial concatenation algorithm and the application of EXIT chart method have been discussed in [13]. We will directly apply our method following the studies for asymptotically long serially concatenated codes. The serially concatenated code of Fig. 4 is constructed by using the same memory-2 code with generator matrix $[1, \frac{1+D^2}{1+D+D^2}]$. 128 data bits are fed into the first encoder and 256 coded bits are generated. These 256 bits are fed into the second decoder to generate 512 bits in total. The overall rate of the code is $1/4$. No error floor is observed until $PER = 10^{-4}$ for this code. The P_{df} estimate is about 0.15dB off the simulation curve for PER$< 10^{-1}$.

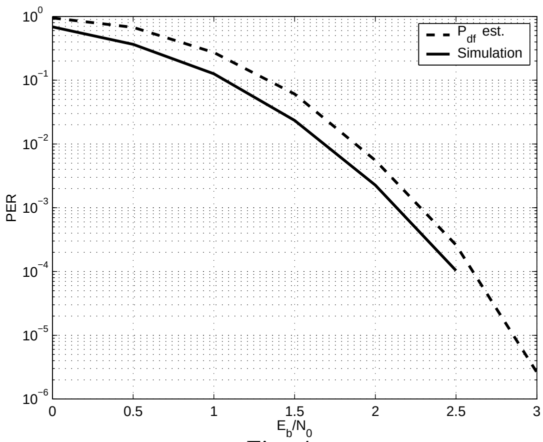

Fig. 4

Packet error rate estimates and simulation results of a serially concatenated code ($K = 128$).

The estimates are within $0.05 - 0.3$dB of the simulation results depending on the operation SNR of the code. There are methods to predict the error floor of serially concatenated codes [5]. However, use of almost any random interleaver in a serially concatenated code results in a very low error floor which makes the error floor and its estimate irrelevant for most practical situations. In this respect, the result with the serially concatenated code is especially important. The serial concatenation example also supports the claim that our P_{df} calculation method corresponds to a worst-case scenario and thus acts as an upper bound.

VI. Conclusions

We developed a probabilistic model in order to characterize the behavior of component decoders of an iterative decoder. This characterization is based on the statistical properties of the extrinsic information produced by component decoders. We extended the EXIT chart method to studying the performance of finite length iterative decoding by introducing a Markov model of the iterative process. The numerical results showed that performance of iterative decoders can be accurately predicted by the introduced model. We believe that this methodology can be applied to many other iterative decoding systems in various scenarios depending on the properties of the extrinsic information sequences produced within the system.

References

[1] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo Codes," *Proc. IEEE International Conference on Communications*, pp. 1064–1070, May 1993.

[2] O. Y. Takeshita, O. M. Collins, P. C. Massey, and D. J. Costello, "On the frame-error rate of concatenated turbo codes," *IEEE Transactions on Communications*, vol. 49, pp. 602–608, April 2001.

[3] A. C. Reid, T. A. Gulliver, and D. P. Taylor, "Convergence and errors in turbo-decoding," *IEEE Transactions on Communications*, vol. 49, pp. 2045–2051, December 2001.

[4] A. O. Yilmaz, *Performance of Turbo Codes: The Finite Length Case*. PhD thesis, University of Michigan, Ann Arbor, 2003.

[5] R. Garello, P. Pierleoni, and S. Benedetto, "Computing the free distance of turbo codes and serially concatenated codes with interleavers: Algorithms and applications," *IEEE Journal on Selected Areas in Communications*, vol. 19, pp. 800–812, May 2001.

[6] P. C. Yeh, A. O. Yilmaz, and W. E. Stark, "On the error floor analysis of turbo codes: weight spectrum estimation (wse) scheme," *Proc. IEEE International Symposium on Information Theory*, p. 439, 2003.

[7] H. E. Gamal and A. R. Hammons, "Analyzing the turbo decoder using the Gaussian approximation," *IEEE Transactions on Information Theory*, vol. 47, pp. 671–686, February 2001.

[8] S. ten Brink, "Convergence behavior of iteratively decoded parallel concatenated codes," *IEEE Transactions on Communications*, vol. 49, pp. 1727–1737, October 2001.

[9] A. O. Yilmaz and W. E. Stark, "Application of gaussian approximation to iterative decoding with finite blocklengths," *IEEE Transactions on Information Theory*, submitted.

[10] P. Robertson, E. Villebrun, and P. Hoeher, "A comparison of optimal and suboptimal MAP decoding algorithms operating in the log domain," *Proc. IEEE International Conference on Communications*, pp. 1009–1013, 1995.

[11] C. Guihenneuc-Jouyaux and C. P. Robert, "Discretization of continuous Markov chains and Markov chain Monte Carlo convergence assessment," *Journal of American Statistical Association*, vol. 93, pp. 1055–1067, September 1998.

[12] D. Divsalar, S. Dolinar, and F. Pollara, "Iterative turbo decoder analysis based on density evolution," *IEEE Journal on Selected Areas in Communications*, vol. 19, pp. 891–907, May 2001.