Information Theoretic Criterion for Stopping Turbo Iteration

Lei Huang, Q. T. Zhang, *Fellow, IEEE*, and L. L. Cheng

***Abstract*—Most existing stopping criteria for turbo decoding have their root in hypothesis test, requiring a *subjective* threshold for decision making. A consequence is that the turbo decoding receiver so-constructed can converge at high signal-to-noise ratios (SNRs) but fails at low SNRs, thereby calling for a new design philosophy for stopping criteria. In this correspondence, the problem is tackled in the framework of information theoretic criterion, which enables the turbo decoding to properly work in a changing SNR environment. Numerical results are presented for illustrating the good performance of the proposed method.**

***Index Terms*—MDL, stopping criterion, turbo decoding.**

I. Introduction

By iteratively exchanging soft extrinsic information between two component decoders, a turbo decoding receiver [1] is capable of closely approaching the theoretic performance predicted by Shannon. Usually, the bit error rate (BER) performance of a turbo decoder improves rapidly at the early stages of iteration; but gradually slows down and eventually vanishes as the iteration stages increase. Further iteration usually only incurs extra computations and decoding delay. Clearly, early termination is necessary for the operation of turbo decoding, and has been widely studied in the literature [2]–[5].

Existing early termination techniques range from the cross-entropy (CE) based criterion [2] to its various modified versions [3]–[5]. All these techniques have their root in the framework of hypothesis test, requiring a subjective threshold for decision making. The operational environment, on the other hand, can vary from frame to frame in terms of signal-to-noise ratio (SNR), further aggravating the difficulty in threshold determination. If the SNR is known *a priori*, one might calculate different thresholds for different SNR intervals, enabling the iterative decoding to properly work in a varying SNR environment. In practice, however, the SNR is generally unknown to the receiver and needs to be estimated, thereby incurring additional complexity to the system. Furthermore, even though the SNR could be correctly estimated, it is still difficult to accurately determine the thresholds at low SNRs since, under the low SNR environments, the thresholds might change not only with the SNR but also with the frame size and code structure. With this in mind, it is easy to understand why the threshold-like criteria can only early terminate the iterative decoding at high SNRs but fails at low SNRs. Therefore, adaptive termination according to a changing SNR environment becomes an issue of practical importance.

Manuscript received November 24, 2009; revised March 26, 2010, June 15, 2010; accepted August 21, 2010. Date of publication September 16, 2010; date of current version January 12, 2011. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Olgica Milenkovic. This work was supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project CityU 124509), by the Natural Science Foundation of China (NSFC) under Grant 60702068, and by the program for New Century Excellent Talents in University under Grant NCET-09-0039.

L. Huang is with the Department of Electronic Engineering, Beijing Institute of Technology, Beijing 100081, China (e-mail: lhuang8sasp@hotmail.com; lhuang@cuhk.edu.hk).

Q. T. Zhang and L. L. Cheng are with the Department of Electronic Engineering, City University of Hong Kong, Kowloon, Hong Kong, China (e-mail: eekzhang@cityu.edu.hk; itacheng@cityu.edu.hk).

Digital Object Identifier 10.1109/TSP.2010.2077284

In this correspondence, we observe that a turbo decoder exploits the code/interleave structure to calculate the extrinsic value which is then fed back as an *a priori* value for the next iteration. As the iteration approaches convergence, the extrinsic information becomes more and more reliable, having the effect of forcing the variance of the normalized output extrinsic sequence [2] to converge to its minimum value. Hence, if we model this normalized extrinsic sequence as a statistically independent Gaussian vector, the number of free parameters required for its characterization depends on how fast the iteration converges. This observation motivates us to formulate the termination problem of turbo decoding in the framework of information theoretic criterion. In particular, we will derive a convergence-detection technique based on the minimum description length (MDL) criterion. At each step of the MDL criterion, we only need to examine the variances of the most updated normalized extrinsic data, thereby making the computation very simple. However, the MDL criterion needs an additional iteration to complete the detection and this extra iteration usually cannot further improve the BER performance. To circumvent this problem, we propose an enhanced scheme by combining the predicted BER examination with the MDL detection. The improved MDL criterion can terminate the iterative process earlier than the threshold-like criteria, in particular at low SNRs, thereby avoiding more unwanted computations and decoding delay.

II. MDL-Based Stopping Criterion

A. Turbo Decoding

Consider a turbo code with two identical constituent recursive systematic convolutional (RSC) codes whereby an information block of N bits, $\{x_n\}_{n=1}^N$, is encoded, BPSK modulated, and then transmitted over an additive white Gaussian noise (AWGN) channel. At the receiver, the noisy data samples corresponding to the information bit x_n are represented by $\{y_n, y_{p_1,n}, y_{p_2,n}\}$ which are the systematic bit, and the parity bits from the two RSC codes, respectively. Thus, we can write

$$y_n = x_n + w_n \tag{1}$$

where $\{w_n\}$ denotes an AWGN random process with mean zero and variance σ_w^2, briefly represented by $w_n \sim \mathcal{N}(0, \sigma_w^2)$. Therefore, the conditional probability density function (PDF) of the received systematic bits can be given by

$$p(y_n|x_n) = \frac{1}{\sqrt{2\pi}\sigma_w} \exp\left(-\frac{(y_n - x_n)^2}{2\sigma_w^2}\right). \tag{2}$$

If $x_n \in \{\pm 1\}$ is equally likely, and its estimate is denoted by $\hat{x}_n$, the *a posteriori* probability (APP) log-likelihood ratio (LLR) (or L-value [2]) of $\hat{x}_n$ is calculated as

$$\begin{aligned} z_{CH}(\hat{x}_n) &= z_{CH}(x_n|y_n) \\ &= \log \frac{p(x_n = +1|y_n)}{p(x_n = -1|y_n)} \\ &= \log \frac{p(y_n|x_n = +1)}{p(y_n|x_n = -1)} + \log \frac{p(x_n = +1)}{p(x_n = -1)} \\ &= \frac{2}{\sigma_w^2} \cdot y_n \\ &= \frac{2}{\sigma_w^2} \cdot (x_n + w_n). \end{aligned} \tag{3}$$

which, in a slightly different manner, can be written as

$$z_{CH}(\hat{x}_n) = \mu_{CH} \cdot x_n + w_{CH,n}. \tag{4}$$

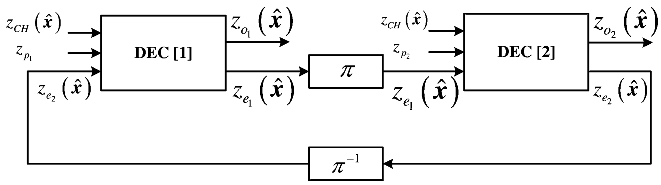

Fig. 1. Turbo decoder.

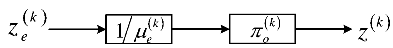

Fig. 2. Observed data generation.

Here, $\mu_{CH} = 2/\sigma_w^2$ and $w_{CH,n} = (2/\sigma_w^2)w_n$. Clearly, we have $w_{CH,n} \sim \mathcal{N}(0, \sigma_{CH}^2)$ with $\sigma_{CH}^2 = 4/\sigma_w^2$.

Fig. 1 depicts a decoder for the two-dimensional turbo code, where π (π^{-1}) denotes the interleaver (deinterleaver), $\hat{\boldsymbol{x}} = [\hat{x}_1, \ldots, \hat{x}_N]$ is the estimate of $\boldsymbol{x} = [x_1, \ldots, x_N]$ whereas z_{p_i}, $z_{o_i}(\hat{\boldsymbol{x}})$ and $z_{e_i}(\hat{\boldsymbol{x}})$ are the parity bit LLRs, output LLRs and extrinsic values of the ith decoder, respectively, for $i = 1,\ 2$. At iteration k and time n, it follows from [2] that

$$z_{o_1}^{(k)}(\hat{x}_n) = z_{CH}(\hat{x}_n) + z_{e_2}^{(k-1)}(\hat{x}_n) + z_{e_1}^{(k)}(\hat{x}_n) \tag{5a}$$
$$z_{o_2}^{(k)}(\hat{x}_n) = z_{CH}(\hat{x}_n) + z_{e_1}^{(k)}(\hat{x}_n) + z_{e_2}^{(k)}(\hat{x}_n). \tag{5b}$$

The interleaver's size is usually quite large. It is, therefore, reasonable to assume that $z_{e_i}^{(k)}(\hat{x}_n)$ at the output of the interleaver (or deinterleaver) is an independent process which approaches a Gaussian-like distribution. The rationale of Gaussian assumption has been discussed by Brink in [6] and by Gamal *et al.* in [7]. Additionally, we assume that d is the true iteration number required by the turbo decoding to converge and m $(m > d)$ is the final iteration number. Ideally, the turbo decoding should be terminated at iteration d which, however, is usually unknown. In what follows, we will show how to use the MDL criterion for its estimation.

B. Observed Data Model

For brevity of notation, we drop the time index n by simply writing $z_{e_2}^{(k)}(\hat{x}_n)$ as $z_e^{(k)}$ unless otherwise specified. Following the technique widely adopted in the literature [2], [6], we model the evolution of $z_e^{(k)}$ with iteration, in the form of (4), to obtain

$$z_e^{(k)} = \mu_e^{(k)} \cdot x + w_e^{(k)}, \quad (k = 1, \ldots, m) \tag{6}$$

where $\mu_e^{(k)} = \sigma_e^{(k),2}/2$ and $w_e^{(k)} \sim \mathcal{N}(0, \sigma_e^{(k),2})$ is an independent Gaussian random process. To generate data for signal processing, we normalize $z_e^{(k)}$ with $\mu_e^{(k)}$ and pass it through a random interleaver $\pi_o^{(k)}$ for decorrelation, as sketched in Fig. 2. The resulting sequence is then given by

$$z^{(k)} = x^{(k)} + w_z^{(k)} \tag{7}$$

where $x^{(k)}$ denotes the interleaved x from the kth interleaver, $w_z^{(k)} = w_e^{\prime(k)}/\mu_e^{(k)}$ is the noise term of the kth observed data and $w_e^{\prime(k)}$ is the interleaved $w_e^{(k)}$. It follows that $w_z^{(k)} \sim \mathcal{N}(0, \sigma_z^{(k),2})$. The square root of $\sigma_z^{(k),2}$, denoted by $\sigma_z^{(k)}$ for subsequent use, is calculated as

$$\sigma_z^{(k)} = \frac{\sigma_e^{(k)}}{\mu_e^{(k)}} = \frac{2}{\sigma_e^{(k)}}. \tag{8}$$

As the interleaver changes randomly for $k = 1, \ldots, m$, we obtain $E[x^{(k)}x^{(\ell)}] = \delta_{k,\ell}$ and $E[w_z^{(k)}w_z^{(\ell)}] = \delta_{k,\ell}$. Here, $\delta_{k,\ell}$ denotes a Kronecker delta function that equals one for $k = \ell$ and zero for $k \neq \ell$. As a result, the observed data $z^{(k)}$ is statistically uncorrelated with $z^{(\ell)}$ for $k \neq \ell$. For each time index, say n, the evolution of $z^{(k)}$ with iteration forms an $m \times 1$ observed vector, as shown by $\boldsymbol{z}_n = [z^{(1)}(\hat{x}_n), z^{(2)}(\hat{x}_n), \ldots, z^{(m)}(\hat{x}_n)]^T$. Then, for $n = 1, \ldots, N$, we have an $m \times N$ data matrix:

$$\boldsymbol{Z} \triangleq [\boldsymbol{z}_1, \ldots, \boldsymbol{z}_N] = \begin{pmatrix} z^{(1)}(\hat{x}_1) & z^{(1)}(\hat{x}_2) & \cdots & z^{(1)}(\hat{x}_N) \\ z^{(2)}(\hat{x}_1) & z^{(2)}(\hat{x}_2) & \cdots & z^{(2)}(\hat{x}_N) \\ \vdots & \vdots & \ddots & \vdots \\ z^{(m)}(\hat{x}_1) & z^{(m)}(\hat{x}_2) & \cdots & z^{(m)}(\hat{x}_N) \end{pmatrix}. \tag{9}$$

With the random interleavers, we may assume that $\boldsymbol{z}_n$ are i.i.d Gaussian vectors. Given the mean $\boldsymbol{x} = [x^{(1)}, \ldots, x^{(m)}]^T$, we assert that each $\boldsymbol{z}_n$ has the distribution of

$$\boldsymbol{z}_n | \boldsymbol{x} \sim \mathcal{N}(\boldsymbol{x}, \boldsymbol{R}_z) \tag{10}$$

where $\boldsymbol{R}_z = E\left[(\boldsymbol{z}_n - \boldsymbol{x})(\boldsymbol{z}_n - \boldsymbol{x})^T\right]$ is the covariance matrix, which can be calculated as

$$\boldsymbol{R}_z = \operatorname{diag}\left(\sigma_z^{(1),2}, \ldots, \sigma_z^{(m),2}\right). \tag{11}$$

Equation (11) indicates that the covariance matrix $\boldsymbol{R}_z$ can be obtained by calculating $\sigma_z^{(k)}$ $(k = 1, \ldots, m)$. Specifically, using the Gaussian assumption of $z_e^{(k)}$, $\sigma_z^{(k)}$ can be directly calculated by the following formula:

$$p_b^{(k)} = \frac{1}{1 + e^{\gamma^{(k)}}} \tag{12a}$$
$$I_e^{(k)} = 1 - E\left[H_b(p_b^{(k)})\right] \tag{12b}$$
$$\sigma_z^{(k)} = \frac{2}{J^{-1}(I_e^{(k)})} \tag{12c}$$

where $\gamma^{(k)} = |z_e^{(k)}|$, $H_b(p) = -p \cdot \log_2 p - (1-p) \cdot \log_2(1-p)$ is the binary entropy function, and $J^{-1}(\cdot)$ is defined in (A3b). The derivation of (12) is done in the Appendix.

C. Adaptive Stopping Criterion Based on MDL Principle

In the iterative decoding, as the iteration number k increases, the extrinsic information becomes more and more reliable. That is to say, $\gamma^{(k)}$ becomes larger, making the error probability in (12a) quickly tend to zero. Consequently, it follows from (12a) and (12b) that as k increases from 1 to m, $I_e^{(k)}$ monotonically increases and eventually converges to a constant number less than one; namely,

$$I_e^{(1)} < \cdots < I_e^{(d-1)} < I_e^{(d)} = \cdots = I_e^{(m)}. \tag{13}$$

The true value d varies with SNR. For the case of low SNR, $d = 1$ and (13) reduces to $I_e^{(1)} = I_e^{(2)} = \cdots = I_e^{(m)}$, implying the failure of the iterative decoding. This expression, when used along with (12c) and the monotonic property of $J^{-1}(\cdot)$ given in (A3b), enables us to assert that

$$\sigma_z^{(1),2} > \cdots > \sigma_z^{(d-1),2} > \sigma_z^{(d),2} = \cdots = \sigma_z^{(m),2} \triangleq \epsilon^2 \tag{14}$$

where ϵ^2 is a constant number. In other words, each extrinsic vector, say $\boldsymbol{z}_n$, is an independent variance-decreasing process which can be characterized by d free parameters. The existence of such $(d-1)$ distinct variances forms the foundation for various threshold-like stopping criteria. However, the true value d varies with the SNR of a given operational environment and therefore, the threshold must be adjusted adaptively according to the SNR variation, making the detection difficult to be implemented in practice. For example, at low SNRs, d takes on the value of unity for which $\sigma_z^{(1),2} = \cdots = \sigma_z^{(m),2}$ and the iteration fails to converge, calling for the immediate termination of turbo decoding to avoid useless calculation. In such a low-SNR environment, many threshold-like criteria cannot terminate the iterative process until the iteration number increases to the presetting maximum number, which results in the unnecessary computations and decoding delay. In what follows, we show how to implement adaptive termination in the framework of information theoretic criterion to reduce the unwanted computations and decoding delay.

Having modeled the extrinsic vector process in (9) and (10), we employ the MDL principle [10] to derive the MDL criterion for stopping the iterative decoding in terms of the maximum likelihood function (MLF) of data set $\boldsymbol{Z} = \{z_n\}_{n=1}^N$ and a penalty term compatible with the free parameters used for their parametrization. It turns out that the iteration number d is chosen such that the MDL function

$$\mathrm{MDL}(\kappa) = -\max_{\boldsymbol{\Theta}} \log p(\boldsymbol{Z}|\boldsymbol{\Theta}_\kappa) + \frac{1}{2}\kappa \log N \tag{15}$$

is minimized. Here, $\boldsymbol{\Theta}_\kappa = [\sigma_z^{(1),2} \ldots, \sigma_z^{(\kappa-1),2}, \epsilon^2]$ is the free parameter vector compatible with the assumption that the turbo decoding converges at iteration κ. Denoting $\boldsymbol{R}_z^{(\kappa)} = \mathrm{diag}(\sigma_z^{(1),2} \ldots, \sigma_z^{(\kappa-1),2}, \epsilon^2, \ldots, \epsilon^2)$ and using the notation in (9) and (10), the likelihood function $p(\boldsymbol{Z}|\boldsymbol{\Theta}_\kappa, \boldsymbol{x})$ can be written as

$$\begin{aligned} p(\boldsymbol{Z}|\boldsymbol{\Theta}_\kappa, \boldsymbol{x}) &= \prod_{n=1}^N \frac{\det\left[\boldsymbol{R}_z^{(\kappa)}\right]^{-1/2}}{(2\pi)^{m/2}} \exp\left(-\frac{1}{2}(z_n - \boldsymbol{x})^T [\boldsymbol{R}_z^{(\kappa)}]^{-1}(z_n - \boldsymbol{x})\right) \\ &= \frac{\left(\epsilon^2\right)^{-(m-\kappa+1)N/2}}{(2\pi)^{(m-\kappa+1)N/2}} \exp\left(-\frac{1}{2\epsilon^2}\sum_{i=\kappa}^m \sum_{n=1}^N [z^{(i)}(\hat{x}_n) - x^{(i)}]^2\right) \\ &\quad \times \prod_{i=1}^{\kappa-1}\left(\frac{\left(\sigma_z^{(i),2}\right)^{-N/2}}{(2\pi)^{N/2}} \exp\left(-\frac{1}{2\sigma_z^{(i),2}}\sum_{n=1}^N [z^{(i)}(\hat{x}_n) - x^{(i)}]^2\right)\right) \end{aligned} \tag{16}$$

which, when maximized with respect to $\boldsymbol{\Theta}_\kappa$, leads to the ML estimations

$$\hat{\sigma}_z^{(i),2}|\boldsymbol{x} = \frac{1}{N}\sum_{n=1}^N [z^{(i)}(\hat{x}_n)]^2 - 1 = \hat{\sigma}_z^{(i),2}, \ (i = 1, \ldots, \kappa - 1) \tag{17a}$$

$$\hat{\epsilon}^2|\boldsymbol{x} = \frac{1}{(m-\kappa+1)N}\sum_{i=\kappa}^m \sum_{n=1}^N [z^{(i)}(\hat{x}_n)]^2 - 1 = \hat{\epsilon}^2 \tag{17b}$$

where we have used the fact that $x^{(i)} \in \{\pm 1\} (i = 1, \ldots, m)$ are equally likely, thereby indicating that the *a priori* knowledge of $\boldsymbol{x}$ is unnecessary for calculating the MLF. Thus, inserting (17) into (16) and omitting the factors independent of κ enable us to represent the MLF as

$$\begin{aligned} p(\boldsymbol{Z}|\hat{\boldsymbol{\Theta}}_\kappa) &= \left[\hat{\epsilon}^2\right]^{-(m-\kappa+1)N/2} \prod_{i=1}^{\kappa-1}\left[\hat{\sigma}_z^{(i),2}\right]^{-N/2} \\ &= \alpha \cdot \frac{\left[\prod_{i=\kappa}^m \hat{\sigma}_z^{(i),2}\right]^{N/2}}{\left[\frac{1}{m-\kappa+1}\sum_{i=\kappa}^m \hat{\sigma}_z^{(i),2}\right]^{(m-\kappa+1)N/2}} \end{aligned} \tag{18}$$

where the factor

$$\alpha = \prod_{i=1}^m \left(\hat{\sigma}_z^{(i),2}\right)^{-N/2} \tag{19}$$

is independent of κ and we have extended the notation $\hat{\sigma}_z^{(i),2}$ to $i = \kappa, \ldots, m$. Inserting (18) into (15) and omitting the term independent of κ yield

$$\begin{aligned} \mathrm{MDL}(\kappa) &= N(m-\kappa+1) \\ &\quad \times \log\left(\frac{\frac{1}{m-\kappa+1}\sum_{i=\kappa}^m \hat{\sigma}_z^{(i),2}}{\left(\prod_{i=\kappa}^m \hat{\sigma}_z^{(i),2}\right)^{1/m-\kappa+1}}\right) + \kappa \log N \end{aligned} \tag{20}$$

minimizing which results in the estimate of d, i.e., $\hat{d} = \arg\min_{\{\kappa=1,\ldots,m\}} \mathrm{MDL}(\kappa)$. Actually, at iteration m, we only need to minimize $\mathrm{MDL}(\kappa)$ for $\kappa = m-1, m$ to determine d as the former $(m-1)$ iterations cannot yield the estimate of d. Therefore, for m iterations, we estimate d as

$$\hat{d} = \arg\min_{\kappa=m-1,m} \mathrm{MDL}(\kappa) \tag{21}$$

and terminate the iterative process provided that $\hat{d} = m - 1$. Let us summarize the iterative stopping algorithm as follows.

a) Calculate $\hat{\sigma}_z^{(i),2}$ for $i = 1, 2$. Start from $m = 2$ and use the MDL criterion given in (20) to determine $\hat{d}$. If $\hat{d} = 1$, terminate the iteration at $m = 2$; otherwise set $m = m + 1$ and go to step b).

b) Compare $\mathrm{MDL}(\kappa)$ for $\kappa = m - 1$ and $\kappa = m$. If $\mathrm{MDL}(m - 1) < \mathrm{MDL}(m)$, estimate $\hat{d} = m - 1$ and stop.

c) Otherwise set $m = m + 1$ and go to b).

Although the convergence can be detected at iteration $\hat{d}$, we need an additional iteration to complete the detection due to $m = \hat{d} + 1$. As a result, we can only stop the iterative process at iteration m instead of $\hat{d}$. To circumvent this problem, an improved scheme will be devised in the next subsection.

D. Enhanced Scheme for MDL Criterion

To circumvent the shortcoming pointed out above, we propose an improved MDL (iMDL) criterion in this subsection. The iMDL scheme is motivated from the fact that the BER can be predicted quite accurately without any *a priori* information of the transmitted bits. Notice that $\hat{x} = \mathrm{sign}(z_{o_2}^{(k)})$ is the estimates of the transmitted bits and $\gamma'^{(k)} = |z_{o_2}^{(k)}|$ is the decision reliability, indicating that

$$z_{o_2}^{(k)} = \hat{x} \cdot \gamma'^{(k)}. \tag{22}$$

With the similar arguments used in the Appendix, we can obtain the probability of the error event $\hat{x} \cdot x = -1$ as

$$p_b^{(k)} \triangleq p(\hat{x} = \pm 1 | x = \mp 1) = \frac{1}{1 + e^{\gamma'^{(k)}}}. \tag{23}$$

Similar to [9], we can therefore exploit the mean of the error probability to predict the BER at iteration k, i.e.,

$$\bar{p}_b^{(k)} = E\left[p_b^{(k)}\right] \approx \frac{1}{N}\sum_{n=1}^N \frac{1}{1 + e^{\gamma_n'^{(k)}}}. \tag{24}$$

The comparison between the numerical and predicted BERs is demonstrated in Fig. 3, implying the predicted BER is quite close to its numerical counterpart. As a result, we can examine the predicted BER before the MDL detection. The pseudocode of the iMDL criterion is

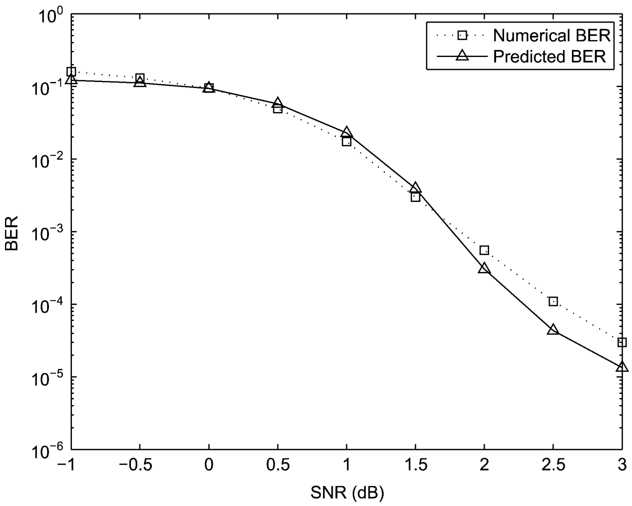

Fig. 3. Numerical BER and predicted BER of the $(11, 15, 1000)$ code of rate $R = 1/2$ with iterative MAP decoding using 8 iterations and 200 frames.

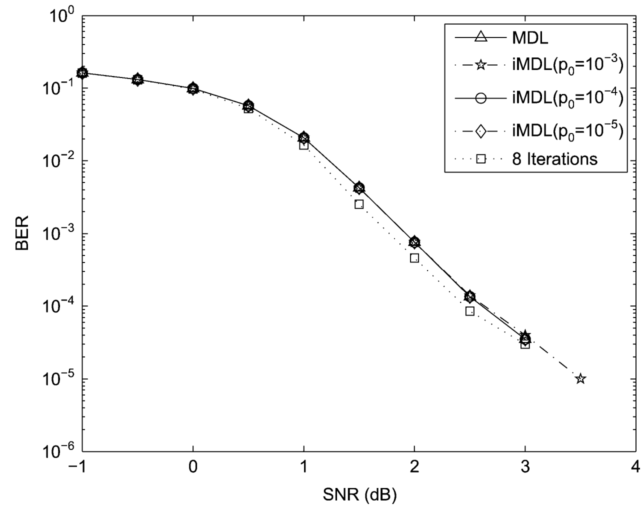

Fig. 4. BER of the $(11, 15, 1000)$ code with iterative MAP decoding using MDL and iMDL criteria, 200 frames.

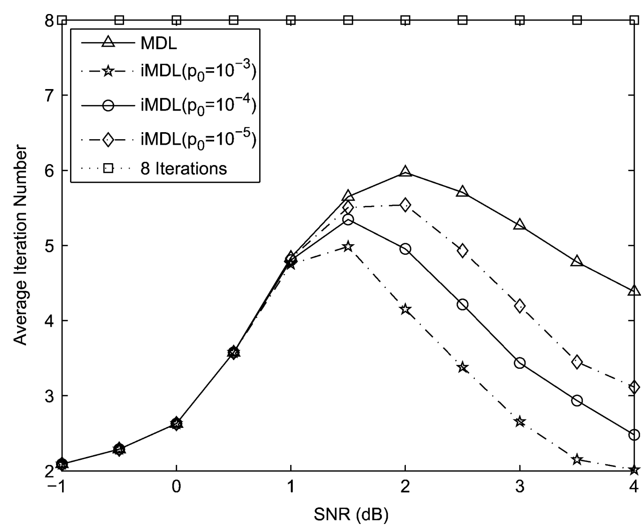

Fig. 5. Average iteration number of the $(11, 15, 1000)$ code with iterative MAP decoding using MDL and iMDL criteria, 200 frames.

TABLE I
IMPROVED SCHEME FOR MDL DETECTION

Initialization: $m := 0$, STOP $:= 1$

Iteration & Detection:
WHILE (STOP)
$m := m + 1$
Perform the mth iteration as (5)
Calculate $\bar{p}_b^{(m)}$ by (24)
IF $\bar{p}_b^{(m)} < p_0$
STOP $:= 0$
ELSE
MDL detection by (21) for $m \geq 2$
IF $\hat{d} = m - 1$
STOP $:= 0$
END
END
END

Final Output: $\hat{x} = \mathrm{sign}(z_{o_2}^{(m)})$, m.

TABLE II
COMPUTATION ANALYSIS OF THE MDL-BASED STOPPING CRITERION

Operation	Computational cost	
	Multiplication	Addition
$\hat{p}_{b,n}^{(m)} = (1+e^{\gamma_n^{(m)}})^{-1}, n=1,\cdots,N$	N	N
$\hat{I}_e^{(m)} = 1-\sum_{n=1}^{N} H_b(\hat{p}_{b,n}^{(m)})/N$	$2N+1$	$3N$
$\hat{\sigma}_z^{(m),2} = 4/(J^{-1}(\hat{I}_e^{(m)}))^2$	5	2
MDL calculated by (20)	12	6
Total computational cost	$3N+18$	$4N+8$

described in Table I, where p_0 denotes a prescribed BER level of the system.

E. Computational Cost and Memory Requirement

The computational cost of the MDL method at iteration m results from the calculations of $\hat{\sigma}_z^{(m),2}$ and MDL, which is around $(3N + 18)$ real number multiplications and $(4N+8)$ real number additions, as analyzed in Table II. Consequently, the MDL method approaches the CE method [2] in complexity as the latter requires about $3N$ real number multiplications and $(3N - 1)$ real number additions for each iteration [4]. Note that the predicted BER calculation only needs $(N + 1)$ real number multiplications and $(2N - 1)$ real number additions, making the iMDL method computationally simpler at high SNRs. On the other hand, the iMDL method uses $\hat{\sigma}_z^{(m-1),2}$ and $\hat{\sigma}_z^{(m),2}$ for detection at iteration m, thereby only requiring one real number memory unite to store $\hat{\sigma}_z^{(m-1),2}$. However, the CE approach [2] needs $(N + 2)$ real number memory unites. Therefore, the MDL method requires much less memory space than the CE method.

III. NUMERICAL RESULTS

A. Experiment 1

This experiment studies the behaviors of the MDL and iMDL criteria for stopping the iterative maximum *a posteriori* probability (MAP) decoding. The turbo codes is given as $(11, 15, 1000)$ with rate $R = 1/2$, where 11 and 15 are the octal representation of the forward and backward RSC code generator and 1000 is the frame size. The numerical results are depicted in Figs. 4 and 5. For comparison, the results of an eight-iteration method are given as well. Observe that, although the MDL criterion can correctly detect the convergence, it involves an extra iteration to finish the detection and thereby needs more iterations. This result coincides with the analysis in Section II-C. The iMDL criterion,

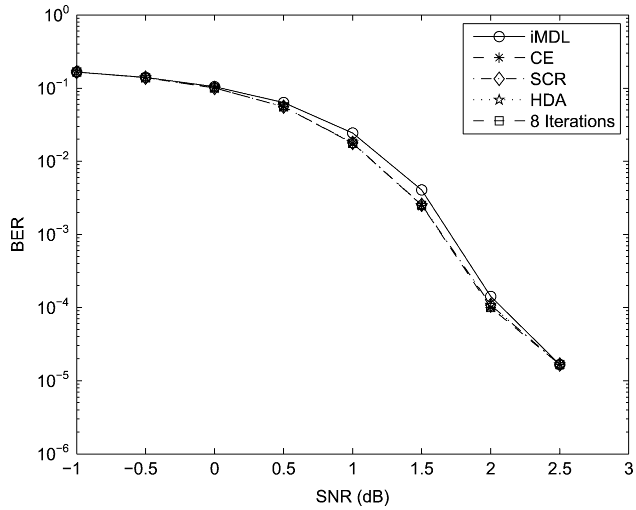

Fig. 6. BER of the $(7, 5, 600)$ code with iterative MAP decoding using iMDL, CE, SCR and HDA criteria, 200 frames.

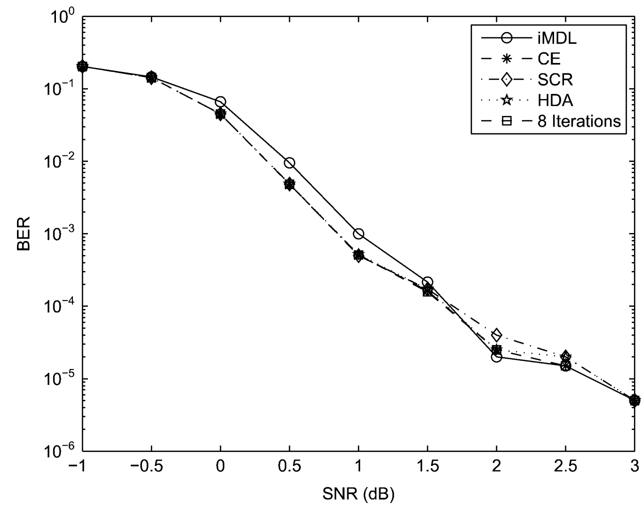

Fig. 8. BER of the $(11, 15, 2000)$ code with iterative MAP decoding using iMDL, CE, SCR and HDA criteria, 200 frames.

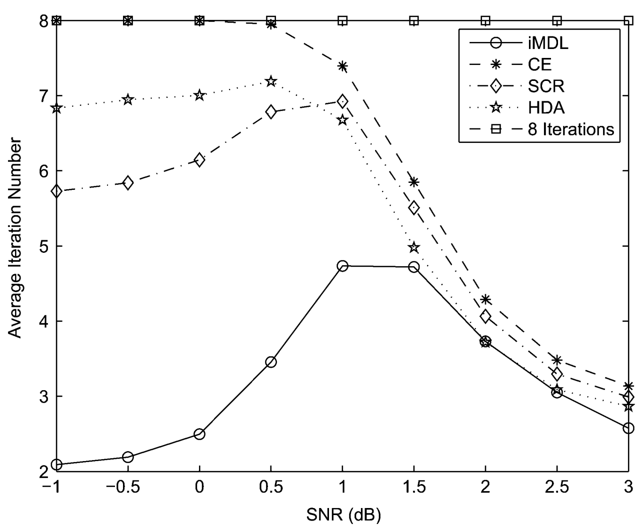

Fig. 7. Average iteration number of the $(7, 5, 600)$ code with iterative MAP decoding using iMDL, CE, SCR and HDA criteria, 200 frames.

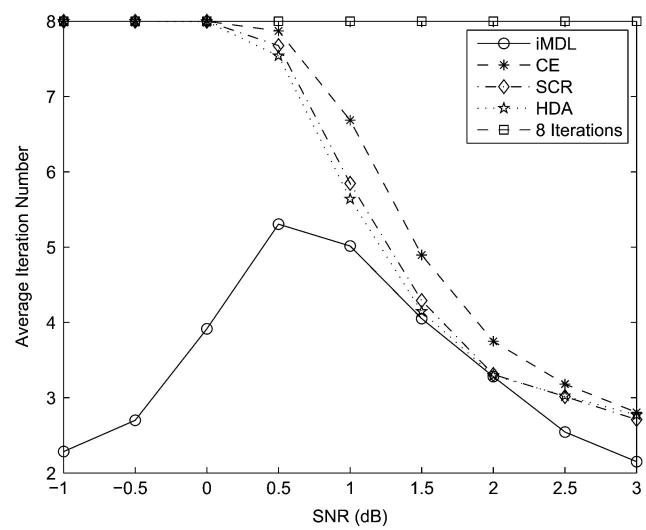

Fig. 9. Average iteration number of the $(11, 15, 2000)$ code with iterative MAP decoding using iMDL, CE, SCR and HDA criteria, 200 frames.

however, can terminate the iterative decoding earlier than the MDL criterion. Moreover, as the prescribed BER p_0 increases, the iteration number decreases. However, the decreased iterative number might lead to a large penalty in BER performance, especially when $p_0 \geq 10^{-3}$. Actually, as the turbo codes are generally used in the low-BER scenarios $(\mathrm{BER} < 10^{-3})$ [4], the prescribed BER should be lower than 10^{-3}.

B. Experiment 2

In this experiment, we compare the performance of the iMDL criterion with that of the classical threshold-like criteria, such as the CE, sign-change-ratio (SCR) and hard-decision-aided (HDA) criteria. Consider first the $(7, 5, 600)$ turbo codes with $R = 1/2$. The BER of the turbo code with iterative MAP decoding is shown in Fig. 6, whereas the average iteration number is depicted in Fig. 7. Here, the thresholds of the CE and SCR criteria are set to 10^{-2} and $0.01N$, respectively, and the prescribed BER is set as 10^{-4}. Observe that the iMDL criterion yields the similar BER as the threshold-like criteria but requires far fewer iterations, especially for $\mathrm{SNR} < 1.5$ dB. Actually, Fig. 7 also indicates that the iMDL criterion is able to adaptively adjust the iteration number in a varying SNR environment. For example, when the SNR is low, representing a severe scenario encountered, the iterative process fails to work even if the iteration number increases infinitely. In such a scenario, the turbo decoding should be terminated as early as possible to save computational cost and avoid unnecessary decoding delay. After comparing the first two $\hat{\sigma}_z^{(k),2}$, the iMDL criterion can terminate the iteration at the second stage. However, the threshold-like methods cannot terminate the iteration until they are forced to stop at the maximum iteration number. This is due to the fact that the threshold-like criteria can only provide one threshold for decision making, which is efficient at high SNRs but inefficient at low SNRs. As the SNR becomes higher, the iMDL method detects a larger iteration number as the iteration requires more stages to converge.

Consider now the turbo codes $(11, 15, 2000)$ with $R = 1/3$. In this setting, the thresholds of the CE and SCR are 10^{-3} and $0.008N$, respectively, and the prescribed BER is 10^{-5}. Fig. 8 implies that the penalty in BER performance is negligible for all the stopping criteria. However, the iMDL criterion requires far fewer iterations than the threshold-like criteria, especially for SNR < 1 dB, as is indicated by Fig. 9.

$$J^{-1}(I_e) \approx \begin{cases} 1.09542 \times I_e^2 + 0.214217 \times I_e + 2.33727 \times \sqrt{I_e} & (0 \le I_e \le I_{tp}) \\ -0.706692 \times \log(0.386013 \times (1 - I_e)) + 1.75017 \times I_e & (I_{tp} < I_e < 1) \end{cases} \tag{A3b}$$

IV. Conclusion

We have devised an information theoretic criterion to early terminate the iterative decoding. By minimizing the MDL, we can adaptively stop the iterative process. However, although the MDL criterion can work well at low SNRs, it involves an extra iteration to complete the detection, requiring more iterations at high SNRs. To circumvent this problem, an improved MDL scheme is proposed. The enhanced MDL method surpasses the threshold-like methods in terms of early terminating the iterative process, particularly at low SNRs.

Appendix
Derivation of (12)

It follows from (6) that the conditional PDF of z_e is

$$p(z_e|x) = \frac{1}{\sqrt{2\pi}\sigma_e} \exp\left(-\frac{\left(z_e - \left(\frac{\sigma_e^2}{2}\right)\cdot x\right)^2}{2\sigma_e^2}\right). \tag{A1}$$

In this Appendix, we will suppress the superscript $(\cdot)^{(k)}$ for brevity unless otherwise specified. The mutual information between the information bits x and the extrinsic values z_e can be computed as [6]

$$\begin{aligned} &I(x;z_e) \\ &= \frac{1}{2}\sum_{x=\pm 1}\int_{-\infty}^{+\infty} p(z_e|x)\log_2 \frac{2p(z_e|x)}{(p(z_e|x=+1)+p(z_e|x=-1))}dz_e \\ &= 1 - \int_{-\infty}^{+\infty} \frac{1}{\sqrt{2\pi}\sigma_e}\exp\left(-\frac{\left(z_e - \left(\frac{\sigma_e^2}{2}\right)\cdot(+1)\right)^2}{2\sigma_e^2}\right)\log_2(1+e^{-z_e})dz_e \\ &\triangleq J(\sigma_e) \end{aligned} \tag{A2}$$

where we have used the fact that the L-value is of symmetric distribution $p(z_e|x) = p(-z_e| - x)$ and satisfies the consistency condition $p(-z_e|x) = p(z_e|x)\cdot e^{-x\cdot z_e}$. As $J(\sigma_e)$ is a monotonically increasing function of σ_e, letting $J^{-1}(\cdot)$ be the inverse function of $J(\cdot)$, σ_e can be concisely represented as

$$\sigma_e = J^{-1}(I_e) \tag{A3a}$$

where $I_e \triangleq I(x;z_e)$. The integral in (A2) can only be evaluated numerically. By curve fitting, the function $J^{-1}(\cdot)$ can be approximated as [8, cf. eq. (B.12)] (A3b), shown at the top of the page, where $I_{tp} = 0.3646$ denotes the turning point.

On the other hand, the extrinsic value can be expressed as $z_e = \hat{x}\cdot\gamma$ with $\hat{x} = \mathrm{sign}(z_e)$ and $\gamma = |z_e|$. Actually, $\hat{x}$ can be regarded as the phase of z_e which is independent of γ. Therefore, $p(-z_e|x) = p(z_e|x)e^{-z_e\cdot x}$ is simplified as

$$p(-\hat{x}|x) = p(\hat{x}|x)e^{-\hat{x}\cdot x\cdot\gamma}$$

which, when combined with $p(-\hat{x}|x) + p(\hat{x}|x) = 1$, results in

$$p(\hat{x}|x) = \frac{1}{1+e^{-\hat{x}\cdot x\cdot\gamma}}.$$

It follows that the probability of the error event $\hat{x}\cdot x = -1$ is

$$p_b \triangleq p(\hat{x} = \pm 1|x = \mp 1) = \frac{1}{1+e^{\gamma}}. \tag{A3c}$$

With these results, the mutual information in (A2) can also be calculated as

$$\begin{aligned} I_e &= 1 - \int_{-\infty}^{+\infty} p(z_e|x=+1)\log_2(1+e^{-z_e})dz_e \\ &= 1 - \int_{-\infty}^{+\infty}\left(\sum_{\hat{x}=\pm 1} p(\hat{x}|x=+1)\log_2(1+e^{-\hat{x}\cdot\gamma})\right)p(\gamma)d\gamma \\ &= 1 - \int_{-\infty}^{+\infty}\left(-\frac{1}{1+e^{\gamma}}\log_2\frac{1}{1+e^{\gamma}} - \frac{1}{1+e^{-\gamma}}\log_2\frac{1}{1+e^{-\gamma}}\right)p(\gamma)d\gamma \\ &= 1 - \int_{-\infty}^{+\infty}\left(-p_b\log_2 p_b - (1-p_b)\log_2(1-p_b)\right)p(\gamma)d\gamma \\ &\triangleq 1 - E\left[H_b(p_b)\right] \end{aligned} \tag{A3d}$$

where $p(\gamma)$ is the PDF of γ. Thus, inserting (A3) into (8) yields (12).

Acknowledgment

The authors would like to thank the associate editor and the anonymous referees for their careful review and constructive suggestions that helped to improve the manuscript considerably.

References

[1] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo codes," in *Proc. IEEE Int. Conf. Commun.*, May 1993, pp. 1064–1070.

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

[3] 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.

[4] Y. Wu, D. Woerner, and J. Ebel, "A simple stopping criteria for turbo decoding," *IEEE Commun. Lett.*, vol. 4, pp. 258–260, Aug. 2000.

[5] F. M. Li and A. Y. Wu, "On the new stopping criteria of iterative turbo decoding by using decoding threshold," *IEEE Trans. Signal Process.*, vol. 55, no. 11, pp. 5506–5516, Nov. 2007.

[6] S. ten Brink, "Convergence behavior of iteratively decoded parallel concatenated codes," *IEEE Trans. Commun.*, vol. 49, no. 10, pp. 1727–1737, Oct. 2001.

[7] H. E. Gamal and A. R. Hammons, "Analyzing the turbo decoder using the Gaussian approximation," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 671–686, Feb. 2001.

[8] S. ten Brink, "Design of concatenated coding schemes based on iterative decoding convergence," Ph.D. dissertation, Univ. at Stuttgart, Shaker Verlag, Aachen, Germany, 2002.

[9] I. Land, P. A. Hoeher, and S. Gligorevic, "Computation of symbol-wise mutual information in transmission systems with LogAAP decoders and application to exit charts," in *Proc. Int. ITG Conf. Source Channel Coding (SCC)*, Erlangen, Germany, Jan. 2004, pp. 195–202.

[10] M. Wax and T. Kailath, "Detection of signals by information theoretic criteria," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 33, no. 2, pp. 387–392, Apr. 1985.