

Analyzing the Turbo Decoder Using the Gaussian Approximation

Hesham El Gamal, *Member, IEEE*, and A. Roger Hammons, Jr., *Member, IEEE*

Abstract—In this paper, we introduce a simple technique for analyzing the iterative decoder that is broadly applicable to different classes of codes defined over graphs in certain fading as well as additive white Gaussian noise (AWGN) channels. The technique is based on the observation that the extrinsic information from constituent maximum *a posteriori* (MAP) decoders is well approximated by Gaussian random variables when the inputs to the decoders are Gaussian. The independent Gaussian model implies the existence of an iterative decoder threshold that statistically characterizes the convergence of the iterative decoder. Specifically, the iterative decoder converges to zero probability of error as the number of iterations increases if and only if the channel E_b/N_0 exceeds the threshold. Despite the idealization of the model and the simplicity of the analysis technique, the predicted threshold values are in excellent agreement with the waterfall regions observed experimentally in the literature when the codeword lengths are large. Examples are given for parallel concatenated convolutional codes, serially concatenated convolutional codes, and the generalized low-density parity-check (LDPC) codes of Gallager and Cheng–McEliece. Convergence-based design of asymmetric parallel concatenated convolutional codes (PCCC) is also discussed.

Index Terms—Convergence-based design, graphical codes, iterative decoding, low-density parity-check (LDPC) codes, turbo codes.

I. INTRODUCTION

ONE of the main reasons behind the impressive performance achieved by graphical codes such as parallel concatenated convolutional code(s) (PCCC) is the elegant iterative decoding algorithm with the exchange of soft information between successive iterations. Recently, it has been shown that this iterative decoding algorithm is an instance of Pearl belief propagation in Bayesian networks [1]. Whereas belief propagation is known to converge to the maximum *a posteriori* (MAP) solution for graphs without loops, relatively little progress has been achieved to date in understanding the theoretical behavior of the algorithm on graphs with loops, especially as a suboptimal decoder for codes having such graphical representation [1].

The rediscovery of Gallager’s low-density parity-check (LDPC) codes [2] and Wiberg’s work on the graphical repre-

sentation of convolutional codes [3] were two important steps toward generalizing the structure behind PCCC to what is now known as graphical codes. Over the past five years, different constructions for regular graphical codes have been proposed including serially concatenated convolutional codes(s) (SCCC) [4], trellis constrained codes [5], and the high-rate graphical codes proposed by Cheng and McEliece [6]. Interestingly, none of these codes succeeded in providing performance superior to the originally proposed turbo construction of Berrou *et al.* [7] in the low signal-to-noise ratio (SNR) region. This result is somehow at odds with the fact that some of these codes (e.g., SCCC) offer better distance spectrum than turbo codes [4]. Quite recently, a tighter upper bound on the probability of error of PCCC and SCCC at SNR below the channel cutoff rate was developed [8]. The new bound confirmed that SCCC would achieve superior performance to PCCC at low SNRs if maximum-likelihood decoding were possible. In this paper, we develop an idealized mathematical model of the iterative decoder, representative of the case in which codeword block sizes are large, that predicts a limit on performance in the low SNR regime due to convergence issues; empirically, this limit is an accurate indication of the “waterfall” region observed for large block lengths.

This paper is based on a simple but powerful technique originally developed by the first author in his Ph.D. dissertation [9] to evaluate the convergence characteristics of the iterative decoder for various graphical codes. The mathematical treatment given here is new as is the use of the tool as an aid to convergence-based code design. Independently and at roughly the same time as [9], Richardson and Urbanke [10] developed a rigorous method of analysis for iterative decoding of LDPC codes. Their approach entails computation of density functions as they evolve from one iteration to the next and makes use of the concentration theorem of Luby *et al.* [11] to make rigorous the asymptotic assumption of loop-free operation. The density evolution approach was later argued to be applicable to turbo codes [12], [13]. The main thrust of our work is the analysis and design of general graphical codes with emphasis on PCCC. The analysis technique proposed in this paper is simpler to evaluate than the technique of Richardson and Urbanke and, as we will show later, provides insights into the decoder operation that we believe would be difficult to extract using the density evolution approach. Furthermore, despite the idealization of the mathematical model and the simplicity of the analysis technique, the close agreement between its predictions and the simulation results available in the literature, including [10], is striking. As part of the review process for the initial submission of this paper, one of the reviewers pointed us to Stephan ten Brink’s work

Manuscript received January 7, 2000; revised July 10, 2000. The material in this paper was presented in part at the IEEE International Symposium on Information Theory, Sorrento, Italy, June 2000.

H. El Gamal was with the Department of Electrical and Computer Engineering, University of Maryland, College Park, MD 20742 USA. He is now with the Department of Electrical Engineering, Ohio State University, Columbus, OH 43210 USA (e-mail: helgamal@ee.eng.ohio-state.edu).

A. R. Hammons, Jr. was with Hughes Network Systems, Germantown, MD 20876. He is now with Corvis Corporation, Columbia, MD 21046 USA (e-mail: rhammons@corvis.com).

Communicated by B. J. Frey, Guest Editor.

Publisher Item Identifier S 0018-9448(01)00728-3.

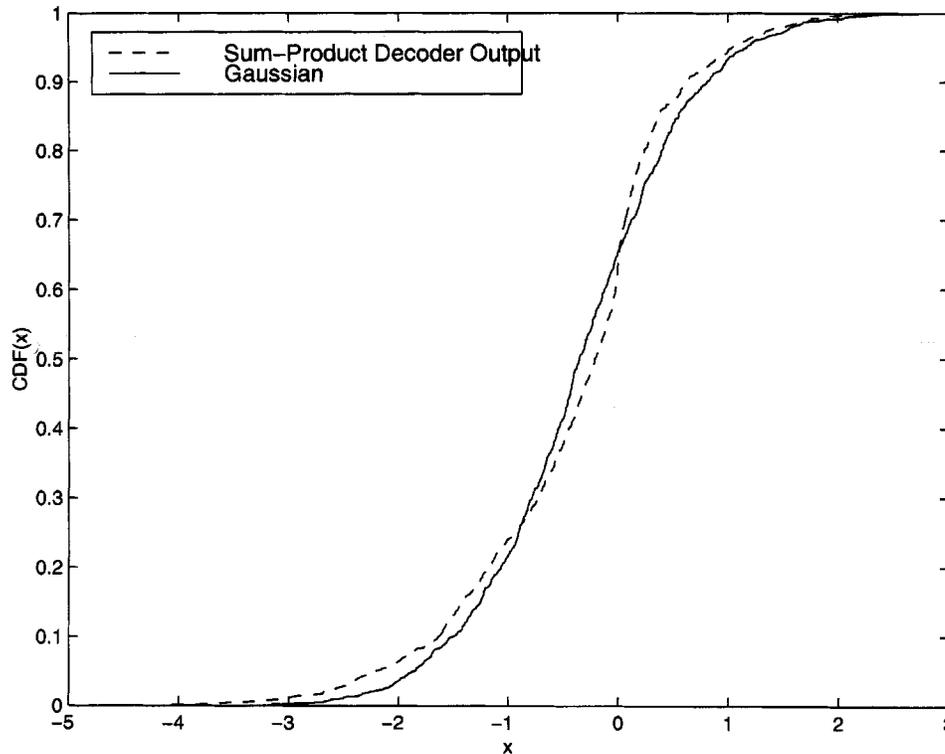


Fig. 1. Cumulative density function of the sum-product algorithm output.

[14], where a similar approach, based on mutual information to analyze the turbo decoder convergence, was used. In contrast to our work, no attempt was made in [14] to prove the existence of convergence thresholds.

The rest of this paper is organized as follows. In Section II, the mathematical model for the iterative decoder is developed and basic theorems regarding convergence characteristics of the idealized decoder are proven. The application of the proposed convergence analysis technique to various graphical codes is discussed in Section III. In addition to an analysis of code designs available in the literature, which demonstrates the utility and accuracy of the new technique, we also present in this section an interesting new code designed based on convergence considerations that improves upon prior state of the art. Section IV discusses extensions to slow and fast-fading channels. The last section presents a few concluding remarks.

II. MATHEMATICAL MODEL FOR DECODER CONVERGENCE

Iterative decoding on graphs can be viewed as a multistage decoding operation where soft information is exchanged between the different stages. The algorithm performed in each iteration can be either the sum-product or the min-sum algorithm [3]. It was shown in [3] that the soft-output Viterbi (SOVA) and the maximum MAP decoding algorithms are direct applications of the min-sum and the sum-product algorithms to trellis-based codes. It was observed in [3] and [15] that, if inputs to either decoding algorithm are independent Gaussian random variables, then the output can be tightly approximated by a Gaussian random variable. In fact, it was pointed out in [3] that the Gaussian approximation can be helpful in analyzing

the performance of turbo codes. As shown in this section, the independent Gaussian approximation allows for complete characterization of the turbo decoder convergence in terms of a single parameter: the extrinsic information SNR.

The accuracy of the Gaussian approximation has been validated experimentally for different classes of graphical codes. Fig. 1 compares the cumulative density function (cdf) of the extrinsic information produced by the sum-product algorithm for a check node of an LDPC code with four independent Gaussian inputs and the cdf of a Gaussian random variable with the same mean and variance. Figs. 2 and 3 show bit-error rate (BER) scatter diagrams (predicted versus measured) for the extrinsic information produced by the ensemble of all rate-2/3 eight-state and 16-state constituent MAP decoders, respectively, under the independent Gaussian assumption. These scattering diagrams were generated by feeding independent Gaussian intrinsic and extrinsic information with different SNRs to the decoders. The predicted BERs were computed from the output extrinsic information's measured SNR assuming a Gaussian distribution [i.e., $P_e = Q(\sqrt{2\text{SNR}})$]. The average relative errors in the predicted BERs are 1.4% and 1.8% for the eight-state and 16-state decoders, respectively.

In this paper, we only consider the sum-product algorithm. This choice is motivated by the equivalence of the sum-product algorithm to MAP decoding for codes with loop-free graphical representations [3]. In this context, we define a constituent code as one having a loop-free graphical representation and, therefore, we assume that the constituent codes are decoded by a soft-input/soft-output (SISO) MAP decoder. The model developed in this section is intended to cover graphical codes that enjoy some symmetry in their structure (e.g., regular LDPC

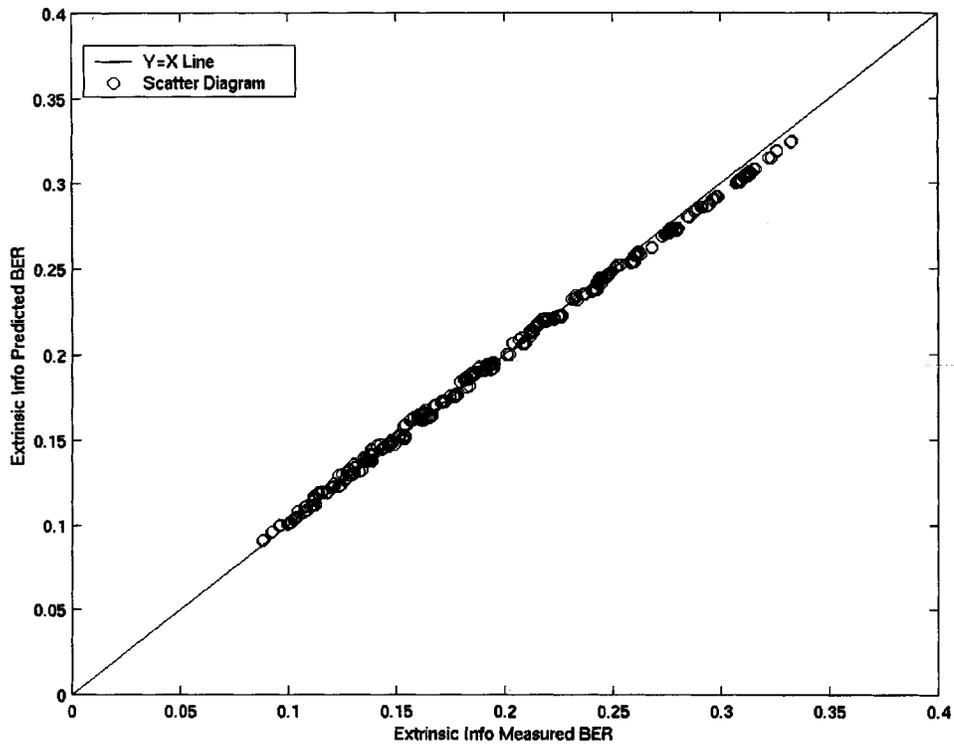


Fig. 2. Scattering diagram of the Gaussian approximation predicted BERs versus measured BERs for eight-state codes.

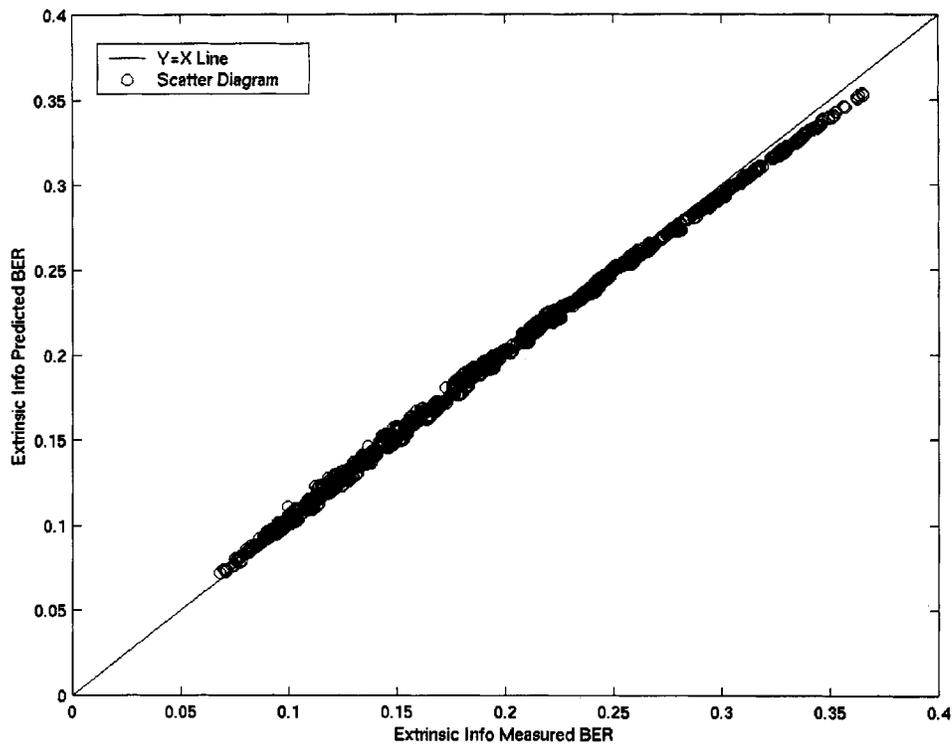


Fig. 3. Scattering diagram of the Gaussian approximation predicted BERs versus measured BERs for 16-state codes.

codes, symmetrical turbo codes, symmetrical trellis-constrained codes); however, with minor modifications, the proposed technique can be extended to handle certain irregular codes. In Section III, the necessary modifications required to model the iterative decoding of asymmetric parallel concatenated convolutional codes and serially concatenated convolutional codes are described. No attempt is made to analyze irregular LDPC codes since these codes are relatively well understood due to the recent work by Richardson *et al.* [10], [16].

In our model, each symbol is processed by M identical constituent decoders at decoding step i and each constituent decoder processes the information for n symbols. The m th constituent decoder accepts intrinsic information

$$\mathcal{Y} = \{y_1, y_2, \dots, y_n\}$$

(log-likelihood values associated with the channel variables) and extrinsic information

$$\mathcal{E}_m^{(i-1)} = \{\bar{\xi}_{1,m}^{(i-1)}, \bar{\xi}_{2,m}^{(i-1)}, \dots, \bar{\xi}_{n,m}^{(i-1)}\}$$

(log-likelihood values supplied by the other constituent decoders at the previous decoding step for use as *a priori* information) and produces updated extrinsic information for the n symbols. The vector

$$\bar{\xi}_{\ell,m}^{(i)} = (\xi_{\ell,1}^{(i)}, \xi_{\ell,2}^{(i)}, \dots, \xi_{\ell,m-1}^{(i)}, \xi_{\ell,m+1}^{(i)}, \dots, \xi_{\ell,M}^{(i)})$$

comprises the extrinsic information produced by the other $M-1$ constituent decoders for the binary data variable d_ℓ during the i th decoding step.

The extrinsic information $\xi_{\ell,m}^{(i)}$ is the additional likelihood information produced by the m th constituent decoder beyond that provided by the channel measurement y_ℓ and *a priori* information $\bar{\xi}_{\ell,m}^{(i-1)}$. In particular

$$\xi_{\ell,m}^{(i)} = \log \frac{P(d_\ell = 1 | \mathcal{Y} - \{y_\ell\}, \mathcal{E}_m^{(i-1)} - \{\bar{\xi}_{\ell,m}^{(i-1)}\})}{P(d_\ell = -1 | \mathcal{Y} - \{y_\ell\}, \mathcal{E}_m^{(i-1)} - \{\bar{\xi}_{\ell,m}^{(i-1)}\})}.$$

After the i th decoding step, the best estimate for d_ℓ is given by

$$\hat{d}_\ell^{(i)} = \text{sgn}(x_\ell^{(i)})$$

where

$$x_\ell^{(i)} = y_\ell + \xi_{\ell,m}^{(i)} + \sum_{j \neq m} \xi_{\ell,j}^{(i)}. \quad (1)$$

The BER $P_\ell^{(i)}$ associated with d_ℓ is the probability that $\hat{d}_\ell^{(i)} \neq d_\ell$. The constituent MAP decoders are optimal in that no decoding rule for the constituent decoders is capable of producing a smaller $P_\ell^{(i)}$.

The statistical behavior of the MAP decoder is determined by the statistical distribution of the input variables. For the additive white Gaussian noise (AWGN) channel, the intrinsic information $y \in \mathcal{Y}$ are Gaussian random variables. Also for

the AWGN channel, it has been observed experimentally that the extrinsic information variables are also approximately Gaussian. For randomly interleaved codes of sufficiently long codeword length, it is reasonable to assume that the loops in the bipartite graph describing the code can be made relatively long, so that the extrinsic information is approximately independent over some number of decoding steps. In the case of LDPC codes, Richardson and Urbanke [10] (following work by Luby *et al.* [11]) prove that the loop-free assumption holds in an asymptotic sense that enables a rigorous convergence analysis for these codes. In the case of turbo codes, however, the loop-free assumption has only heuristic justification.

Motivated by the fact that SNR is a simple sufficient statistic for Gaussian random variables, we formalize the Gaussian and loop-free assumptions as follows.

Independent Gaussian Assumption: The random sequences

$$\{y_\ell\}_{\ell=1}^n, \{\bar{\xi}_{\ell,m}^{(0)}\}_{\ell=1}^n, \{\bar{\xi}_{\ell,m}^{(1)}\}_{\ell=1}^n, \dots, \{\bar{\xi}_{\ell,m}^{(i)}\}_{\ell=1}^n, \dots$$

are jointly Gaussian and statistically independent in the sense that any finite collection of the y_ℓ and $\xi_{\ell,m}^{(i)}$ are jointly Gaussian and pairwise independent.

For each Gaussian random variable r with mean μ and variance σ^2 , we associate a SNR $\text{SNR}(r) = \mu^2/\sigma^2$. Under the independent Gaussian assumption, the behavior of the constituent MAP decoder is completely determined by the SNRs of the input variables; and the bit-error probability $P_\ell^{(i)}$ is completely determined by the SNR of the output log-likelihood ratio $x_\ell^{(i)}$. By symmetry of the general decoder model, the SNR of the extrinsic information $\xi_{\ell,j}^{(i)}$ is the same for all j . Since the M constituent encoders are statistically equivalent, we will simplify the notation hereafter by dropping the subscript m in reference to the extrinsic information.

The following proposition follows from the fact that the constituent MAP decoder is optimal.

Proposition 1: Under the independent Gaussian assumption, the SNR $(x_\ell^{(i)})$ and SNR $(\xi_\ell^{(i)})$ are nondecreasing functions of the input SNRs $\text{SNR}(y)$ and $\text{SNR}(\xi)$ for $y \in \mathcal{Y} - \{y_\ell\}$ and $\xi \in \mathcal{E}^{(i-1)} - \{\bar{\xi}_{\ell,m}^{(i-1)}\}$. Furthermore, $\text{SNR}(x_\ell^{(i)}) \rightarrow \infty$ (and, thus, $P_\ell^{(i)} \rightarrow 0$) as $i \rightarrow \infty$ if and only if $\text{SNR}(\xi_\ell^{(i)}) \rightarrow \infty$ as $i \rightarrow \infty$.

Proof: Consider two scenarios in which only the SNR for input variable r is different. Let $\text{SNR}(r') < \text{SNR}(r)$ denote the two different values of signal-to-noise ratio for r with corresponding output $\text{SNR}'(x_\ell^{(i)})$ and $\text{SNR}(x_\ell^{(i)})$, respectively.

For either scenario, the constituent MAP decoder must produce the smallest possible value of BER P_ℓ . Thus, we must have $\text{SNR}'(x_\ell^{(i)}) \leq \text{SNR}(x_\ell^{(i)})$. Otherwise, for the scenario with $\text{SNR}(r)$, a decoder strategy in which noise is injected to reduce the SNR to $\text{SNR}(r')$ before performing MAP decoding would yield a smaller BER than optimal MAP decoding. This contradiction shows that $\text{SNR}(x_\ell^{(i)})$ is a nondecreasing function of the input SNRs as claimed.

The constituent MAP decoder forms the linear combination (1) of the log-likelihood information y_ℓ , $\xi_\ell^{(i)}$, and $\bar{\xi}_{\ell,j}^{(i)}$ to produce the decision statistic $x_\ell^{(i)}$ yielding the smallest BER. For

Gaussian random variables, the optimal linear combination producing the highest SNR (and hence lowest BER) is (up to a proportionality constant) the weighted sum

$$z_\ell^{(i)} = \lambda(y_\ell) \cdot y_\ell + \lambda(\xi_\ell^{(i)}) \cdot \xi_\ell^{(i)} + \sum_{j \neq m} \lambda(\xi_{\ell,j}^{(i)}) \cdot \xi_{\ell,j}^{(i)} \quad (2)$$

where the coefficient $\lambda(r) = E\{r\}/\text{Var}\{r\}$ is the mean-to-variance ratio of random variable r . Thus, under the independent Gaussian assumption, the decision statistic $x_\ell^{(i)}$ is a scaled version of $z_\ell^{(i)}$. [Note that this implies that the $\lambda(r)$ values are equal. This is similar to [16] in which for Gaussian log-likelihood ratios, one has $\lambda(r) = 1/2$ as a consequence of the consistency condition that the probability density function (pdf) of log-likelihood ratio messages must satisfy $f(x) = e^{-x}f(-x)$. Hence, $x_\ell^{(i)} = 2z_\ell^{(i)}$.]

It is well known that the SNR of $z_\ell^{(i)}$ is the sum of the SNRs of each term. We therefore conclude that the value of SNR ($\xi_\ell^{(i)}$) changes in the same direction as the value of SNR ($x_\ell^{(i)}$). Hence, SNR ($\xi_\ell^{(i)}$) is also a nondecreasing function of the SNRs of the other input variables. Likewise, SNR ($x_\ell^{(i)}$) $\rightarrow \infty$ if and only if SNR ($\xi_\ell^{(i)}$) $\rightarrow \infty$. \square

Given intrinsic information having a common SNR_{in}, we can now view the essential action of the constituent decoders as enhancing the SNR of the extrinsic information. In this regard, let $S_\ell^{(i)}(v)$ denote the SNR of the extrinsic information $\xi_\ell^{(i)}$ associated with d_ℓ after the i th decoding step, where $v = \text{SNR}_{\text{in}}$ represents the decoder initial condition. Then, the sequence $\{S_\ell^{(i)}(v)\}_{i=0}^\infty$ evolves recursively

$$\begin{aligned} S_\ell^{(0)}(v) &= 0 \\ S_\ell^{(i)}(v) &= f_\ell(S_\ell^{(i-1)}(v), v). \end{aligned}$$

By the independence assumption, the function f_ℓ is the same for all decoding steps.

By the Heine–Borel property, the sequence $\{S_\ell^{(i)}(v)\}_{i=0}^\infty$ either has an accumulation point $\tau(v) < \infty$ or is unbounded ($\tau(v) = \infty$). Since, by the previous proposition, the sequence is nondecreasing, it must in fact converge to $\tau(v)$ from below. Thus, $\tau(v) = \sup_{i \geq 0} \{S_\ell^{(i)}(v)\}$.

Proposition 2: $\tau(v)$ is a nondecreasing function of v . Thus, if $\tau(v) < \infty$ for some v , then $\tau(s) < \infty$ for all $s \leq v$; and, conversely, if $\tau(v) = \infty$ for some v , then $\tau(s) = \infty$ for all $s \geq v$.

Proof: Let $v' < v$. Since f_ℓ is a nondecreasing function of both arguments, we have

$$f_\ell(S_\ell^{(i)}(v'), v') \leq f_\ell(S_\ell^{(i)}(v'), v) \leq f_\ell(S_\ell^{(i)}(v), v).$$

Thus,

$$S_\ell^{(i+1)}(v') \leq S_\ell^{(i+1)}(v).$$

Taking the supremum over i yields the desired result

$$\tau(v') = \sup_{i \geq 0} \{S_\ell^{(i)}(v')\} \leq \sup_{i \geq 0} \{S_\ell^{(i)}(v)\} = \tau(v). \quad \square$$

If the threshold $\tau(v)$ is finite for some input SNR_{in} = v , then, by Proposition 1, the bit-error probability for d_ℓ is bounded away from 0 no matter how many decoding iterations are performed. Conversely, if $\tau(v)$ is infinite, Proposition 1 implies that the bit-error probability converges to 0 as the number of iterations goes to infinity.

Thus, the behavior at one input SNR determines the behavior for all smaller or larger values of input SNR [depending on whether $\tau(v) < \infty$ or $= \infty$]. Since $\tau(0) = 0$, the supremum

$$\nu = \sup \{v: \tau(v) < \infty\} \quad (3)$$

is well-defined and serves as a fundamental threshold determining the convergence of the iterative decoder under the independent Gaussian assumption.

Summarizing these results, we have the following proposition.

Proposition 3: If $v < \nu$, then $\tau(v) < \infty$. If $v > \nu$, then $\tau(v) = \infty$. Thus, under the independent Gaussian assumption, the iterative decoder converges with zero error probability if the input SNR is greater than ν , but converges with nonzero error probability if the input SNR is less than ν .

Proposition 3 allows for a complete characterization of the turbo decoder convergence properties using a simple technique. It is sufficient to characterize the extrinsic information SNR input/output relation of the basic constituent decoder(s) to determine if the turbo decoder will converge or not at any E_b/N_0 . The characterization is possible either through approximate numerical analysis or, more accurately, through simple Monte Carlo simulation. In this paper, we will rely on the simulation approach to benefit from its accuracy. The simulation required in this case is much simpler than the iterative decoder simulation. Here, to obtain the input/output relation at each intrinsic SNR, we only need to simulate one constituent decoder, assuming symmetry, with Gaussian extrinsic and intrinsic inputs and measure a *single parameter*: the output extrinsic information SNR. Then, the convergence threshold is evaluated from the input/output characteristics of the extrinsic information SNR as described in Section III. This demonstrates the simplicity of our technique compared to the density evolution approach where the pdf is estimated at each step. As shown in Section III, for all of the major classes of codes admitting iterative decoding via constituent MAP decoders, the threshold ν is found experimentally to be finite and greater than zero. Indeed, for almost all known graphical codes of suitably large codeword length, ν is found to correspond to the sharp “waterfall region” of the performance characteristic.

It is worth noting that the threshold ν depends on the structure of the *constituent codes* rather than that of the composite code. Thus, for the class of graphical codes obtained by certain concatenation of interleaved convolutional codes, analysis based on the independent Gaussian assumption indicates that the waterfall region determined by ν is not a function of the code weight spectrum and thus cannot be improved by turbo code interleaver optimizations aimed at improving the weight spectrum. Rather, the waterfall region is determined more by how amenable the constituent codes are to iterative decoding.

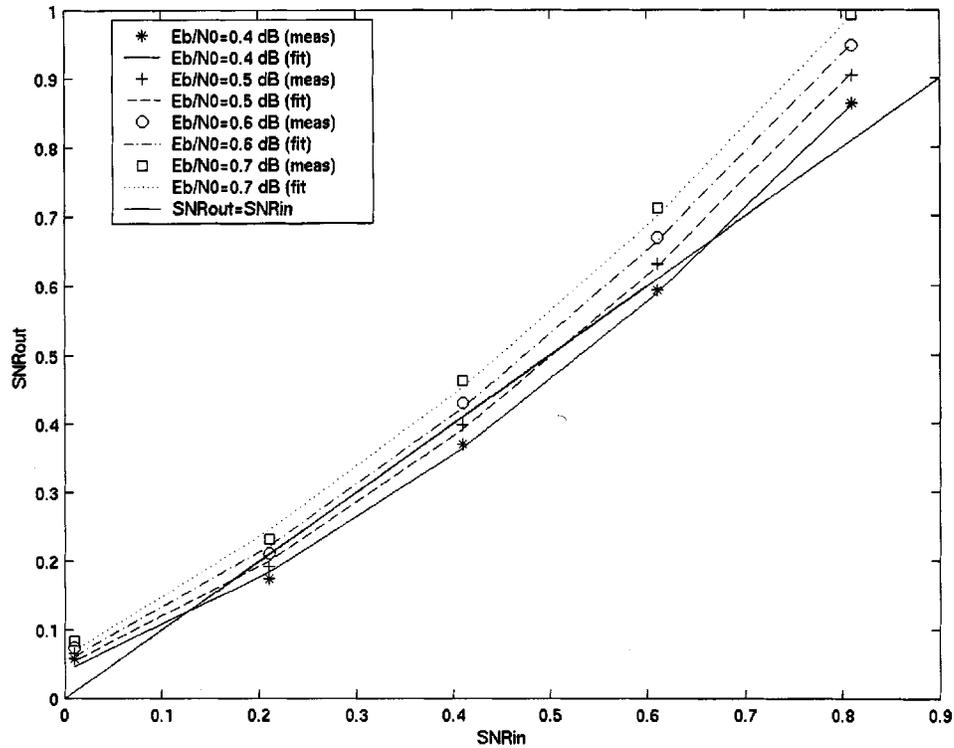


Fig. 4. Extrinsic information SNR input/output relations for Berrou's 16-state.

III. APPLICATION TO DIFFERENT CODE CONSTRUCTIONS

In this section, we analyze in detail the effect of the iterative decoder convergence characteristics on the performance of various graphical codes. Our primary focus will be PCCC, although representative examples for SCCC and regular LDPC codes will also be discussed to emphasize the broad applicability of the proposed technique. For all of the cases considered, the convergence results predicted by the proposed technique are within a very small fraction of a decibel from the simulation results reported in the literature [4], [17], [6], [2]. The PCCC results constitute an exhaustive study of rate-1/2 and -1/3 symmetric codes with four, eight, and 16-state constituent codes. Based on these convergence results, we design an interesting asymmetric PCCC that outperforms the asymmetric code proposed by Takeshita *et al.* [18] in the waterfall region.

A. Symmetric PCCC

For simplicity, we consider the classical turbo code [7] consisting of two identical constituent encoders. The extension to multiple PCCC is straightforward.

The analysis technique is best illustrated through a detailed example. Consider the rate-1/3 PCCC built from the 16-state convolutional code with transfer function

$$G(D) = \left(1, \frac{1+D+D^3+D^4}{1+D+D^2+D^3+D^4}\right)$$

originally introduced by Berrou [7]. Due to symmetry of the code construction, it is sufficient to study the input/output characteristic of the constituent decoder. Fig. 4 shows the SNR input/output relation for the extrinsic information from the MAP decoder for this constituent code as a function of different bit-energy-to-noise ratio E_b/N_0 . In the figure, the symbols

denote measured SNRs obtained via simulation by inverting the observed BER using the Gaussian Q -function, whereas the various dashed and solid lines represent curve fits. Since the input extrinsic information to one constituent decoder is an interleaved version of the other decoder output, the turbo decoder will converge asymptotically to $P_\ell^{(i)} \rightarrow 0$ after a sufficient number of iterations if and only if the constituent decoder characteristic *does not* intersect the (In = Out) line; the intersection point is the finite accumulation point τ , and hence, the nonexistence of the intersection point implies that $\tau = \infty$. From the figure, it is clear that the decoder will converge to $P_\ell^{(i)} = 0$ for $E_b/N_0 = 0.6$ and 0.7 dB; but, for $E_b/N_0 = 0.4$ and 0.5 dB the probability of error will be bounded away from zero even if the number of iterations is unbounded. Therefore, a coarse estimate for the convergence threshold predicted by the independent Gaussian assumption is that it lies between 0.5 and 0.6 dB.

A more precise threshold can be calculated from the observation that the extrinsic information SNR input/output relation is locally quadratic (true to a high degree for all PCCC that we have studied), allowing the minimum signed distance between the SNR characteristic and (In = Out) line to be accurately determined analytically. The minimum signed distances (so-called H -metrics) are plotted in Fig. 5 as a function of input E_b/N_0 . The convergence threshold is the E_b/N_0 value for which the H -metric is 0 and is easily determined by interpolation (the curves are again locally quadratic). From Fig. 5, a more precise threshold for the Berrou *et al.* code is $E_b/N_0 = 0.57$ dB.

Using our analysis technique, we conducted an exhaustive search for symmetric rate-1/3 and -1/2 PCCC with convergence thresholds less than or equal to 0 and 0.7 dB, respectively (i.e., within 0.5 dB from the Shannon limit for binary signaling).

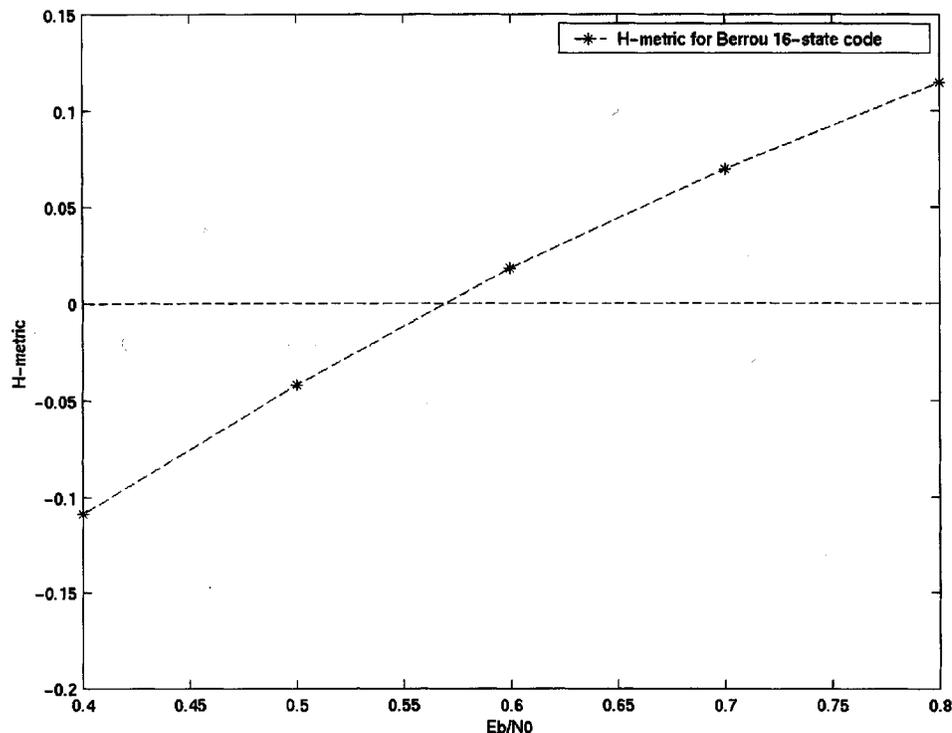
Fig. 5. The H -metric for Berrou's 16-state code.

TABLE I
 CONSTITUENT CODES FOR SYMMETRIC RATE-1/3 16-STATE PCCC WITH CONVERGENCE THRESHOLDS ≤ 0 dB

| $G(D)$ in octal representation | Threshold in dB | Remarks |
|--------------------------------|-----------------|--|
| 37/25 | -0.15 | 16-state, non-primitive |
| 27/25 | -0.13 | 16-state, non-primitive |
| 35/25 | -0.13 | 16-state, non-primitive |
| 31/37 | -0.081 | 16-state, non-primitive |
| 23/37 | -0.079 | 16-state, non-primitive |
| 33/37 | -0.061 | 16-state, non-primitive |
| 35/37 | -0.060 | 16-state, non-primitive |
| 27/37 | -0.057 | 16-state, non-primitive |
| 21/37 | -0.055 | 16-state, non-primitive, Berrou's code |
| 27/21 | -0.040 | 16-state, non-primitive |
| 35/21 | -0.027 | 16-state, non-primitive |

We restricted our search to constituent codes having at most $2^v \leq 16$ states and feedback polynomial of the form

$$1 + a_1D + a_2D^2 + \dots + D^n.$$

The rate-1/2 PCCC are obtained from the rate-1/3 PCCC by periodically puncturing the output of the constituent codes as described in [7]. Tables I and II report the octal representation of the generator polynomials for best constituent codes, from a convergence point of view, found in our search. For rate-1/3 PCCC, all the constituent codes are 16-state with nonprimitive

feedback polynomials. On the other hand, for the rate-1/2 case, two eight-state codes with primitive feedback polynomial exhibit convergence thresholds within 0.5 dB from the capacity limit. Since primitive feedback polynomials are advantageous with respect to high SNR asymptotic performance [18], [19], Table III provides a survey of the convergence thresholds for the best rate-1/3 primitive PCCC available in the literature. These results suggest that the eight-state code 15/13 chosen for the third-generation wireless standards achieves the best performance tradeoff both for rate-1/2 and -1/3 codes [20] (i.e., the

TABLE II
CONSTITUENT CODES FOR SYMMETRIC RATE-1/2 16-STATE PCCC WITH CONVERGENCE THRESHOLDS ≤ 0.7 dB

| $G(D)$ in octal representation | Threshold in dB | Remarks |
|--------------------------------|-----------------|--|
| 27/37 | 0.56 | 16-state, non-primitive |
| 21/37 | 0.57 | 16-state, non-primitive, Berrou's code |
| 35/37 | 0.58 | 16-state, non-primitive |
| 31/37 | 0.59 | 16-state, non-primitive |
| 31/27 | 0.60 | 16-state, non-primitive |
| 33/37 | 0.61 | 16-state, non-primitive |
| 37/27 | 0.62 | 16-state, non-primitive |
| 23/35 | 0.62 | 16-state, non-primitive |
| 26/37 | 0.64 | 16-state, non-primitive |
| 25/35 | 0.64 | 16-state, non-primitive |
| 25/27 | 0.64 | 16-state, non-primitive |
| 37/35 | 0.65 | 16-state, non-primitive |
| 32/37 | 0.65 | 16-state, non-primitive |
| 22/37 | 0.66 | 16-state, non-primitive |
| 31/33 | 0.67 | 16-state, non-primitive |
| 13/15 | 0.68 | 8-state, primitive |
| 15/13 | 0.70 | 8-state, primitive, CDMA2000 |

TABLE III
CONVERGENCE THRESHOLDS FOR BEST PRIMITIVE SYMMETRIC RATE-1/3 PCCC

| $G(D)$ in octal representation | Threshold in dB | Remarks |
|--------------------------------|-----------------|---------------------------------------|
| 35/23 | 0.096 | 16-state Takeshita <i>et al</i> [18] |
| 37/23 | 0.102 | 16-state, Benedetto-Montorsi [21] |
| 33/23 | 0.11 | 16-state, Divsalar <i>et al</i> [17] |
| 15/13 | 0.02 | 8-state, primitive, Eroz-Hammons [20] |
| 5/7 | 0.134 | 4-state |

code has a primitive feedback polynomial and its convergence thresholds are among the best for both rate-1/2 and -1/3 codes).

Figs. 6 and 7 compare the convergence thresholds obtained by our technique with simulation results for rate-1/3 PCCC with various block lengths.¹ In Fig. 6, we report simulation results for the primitive four-state and eight-state codes in Table III, whereas in Fig. 7 we compare the 16-state PCCC with the best convergence threshold (37/25) with the primitive 16-state PCCC with the best distance spectrum (37/23) [21]. For the 65 536 block lengths, all codes achieve 10^{-5} BER within less than 0.1 dB from the predicted convergence thresholds. This small offset is similar to that shown in [16] for the irregular LDPC codes with 1 000 000 block length. It is also shown that the relative performance between the different

codes in the waterfall region is very close to the convergence analysis prediction even for relatively small block lengths.

The extrinsic information SNR transfer characteristic also provides useful information regarding rates of convergence (improvement in BER as a function of the number of iterations). In [7], Berrou *et al.* observed that the performance of a certain four-state PCCC was better than their powerful 16-state PCCC in the first few iterations but ultimately failed to converge to small probability of error with increasing numbers of iterations. This behavior is readily explained by the SNR input/output characteristics for the two codes shown in Fig. 8. From the figure, it is clear that the iterative decoder will converge at $E_b/N_0 = 0.7$ dB only for the 16-state code since the SNR input/output relation of the four-state code intersects the (In = Out) line. Note that the improvement in the extrinsic information output SNR is higher for the four-state decoder at relatively small input extrinsic information SNRs, while the situation is reversed in favor

¹In all our simulations we assumed random interleaving and 18 decoding iterations.

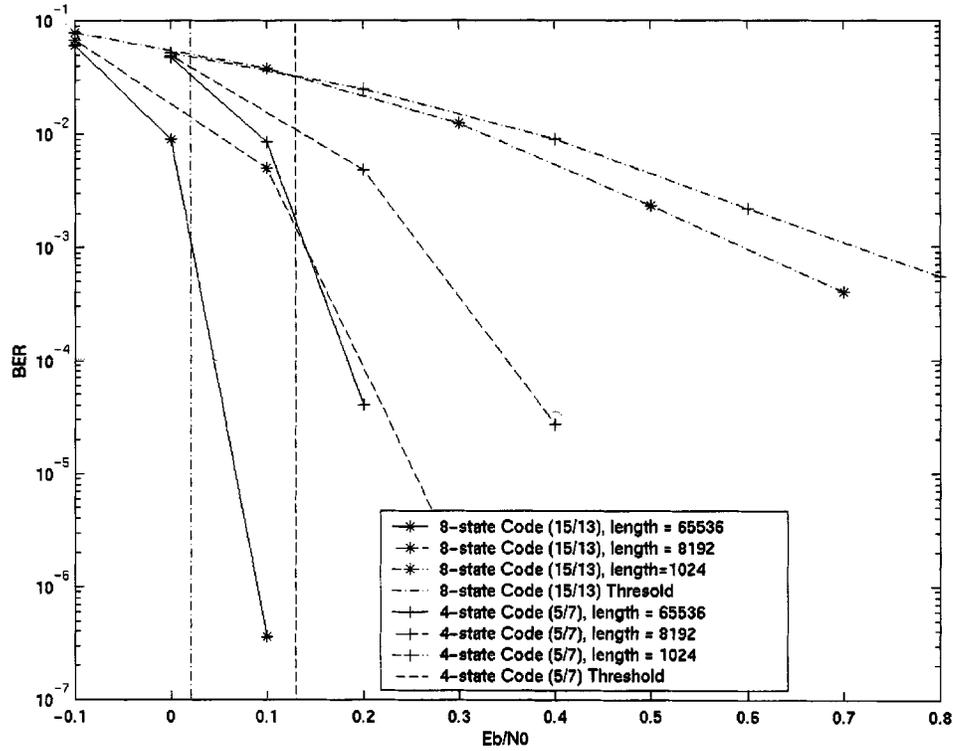


Fig. 6. Comparison between simulation results and convergence thresholds for one primitive four-state and one primitive eight-state codes.

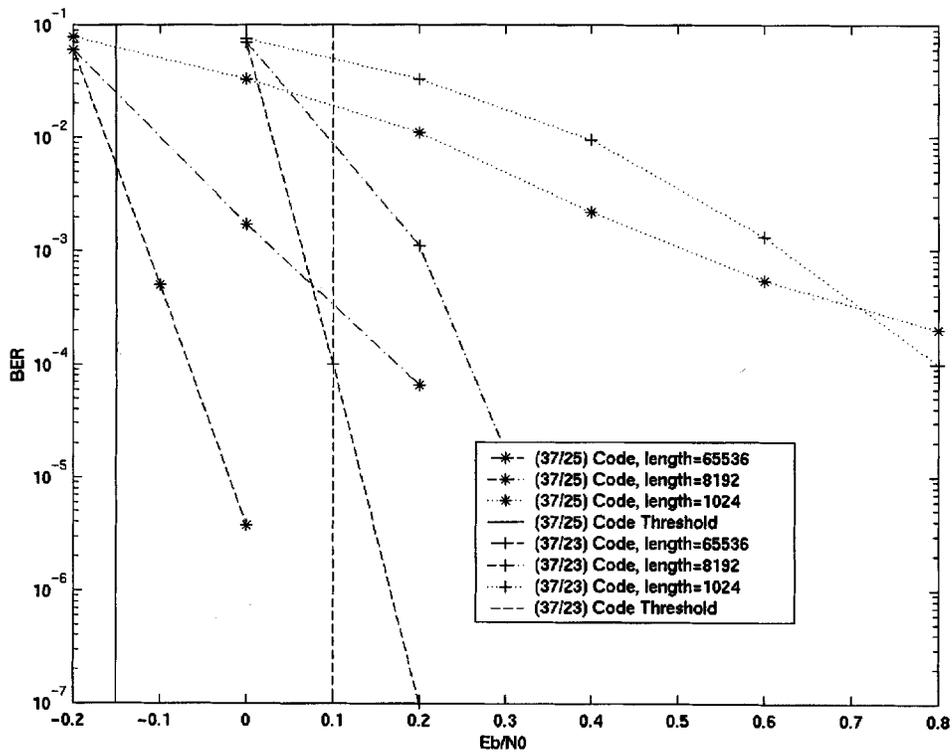


Fig. 7. Comparison between simulation results and convergence thresholds for the best waterfall 16-state code and the best error floor 16-state code.

of the 16-state decoder at higher input extrinsic information SNRs. Thus, there will be more improvement in performance

in the early decoding iterations of the four-state PCCC. However, this code will ultimately fail to converge to $P_\ell^{(i)} = 0$.

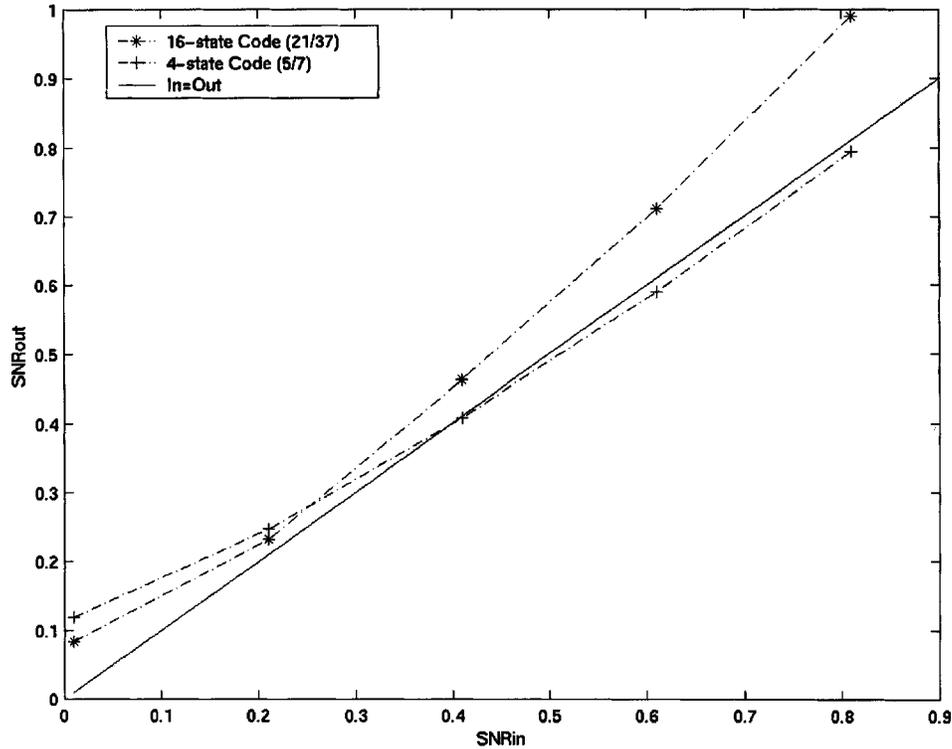


Fig. 8. Extrinsic information SNR input/output relations for the 16-state and four-state codes at $E_b/N_0 = 0.7$ dB.

B. Asymmetric and Irregular PCCC

In [19], it was argued that symmetric PCCC should be built using constituent codes with primitive feedback polynomials because of their favorable asymptotic performance in the “error floor” region. However, it was observed through simulation that primitive PCCC exhibit relatively poorer performance in the “waterfall” region compared to other known nonprimitive PCCC [18]. This phenomenon was validated by our results in the previous section which show that the best symmetric PCCC from a convergence point of view have nonprimitive feedback polynomials. In [18], Takeshita *et al.* proposed a new construction for asymmetric PCCC that uses one primitive and one nonprimitive constituent code. Their proposed code was shown through simulation to “split” the performance of the corresponding codes both in the waterfall and error floor regions [18].

It is not necessary to make this compromise, however, as it is possible to design better asymmetric codes using convergence threshold analysis. For example, let us take the primitive 16-state code with best convergence threshold as the first of the constituent codes. This is the 35/23 code from Table III with a convergence threshold of 0.096 dB (the same primitive code used by Takeshita *et al.*). Fig. 9 shows the extrinsic information SNR input/output relation for this code at $E_b/N_0 = 0$ dB. From the figure, it is clear that if a genie were to provide the iterative decoder with extrinsic information having $\text{SNR} \geq 0.4$ in the first iteration only, the symmetric PCCC built from this 16-state

constituent code would converge to the desired $P_\ell^{(i)} = 0$. This is predicted by the extrinsic information characteristic of Fig. 9 and can be easily verified through simulation. This says that, if there were a way in the initial iterations to aid this 16-state in avoiding the intersection point, the PCCC would eventually reach the desired zero error convergence point. In the previous section, we noted that carefully chosen codes with different memory size exhibit different rates of convergence. This observation suggests that the second constituent code should be chosen to yield high-quality extrinsic information in the input SNR range where the 16-state is close to the (In = Out) line. This second code will then help bootstrap the convergence of the iterative decoder. The best eight-state code found in our search that suits this requirement has the transfer function 15/11.

A slightly different analytical approach is required to find the convergence threshold for asymmetric PCCC. It is no longer sufficient to compare the extrinsic information SNR input/output relation of one constituent code with the (In = Out) line. To analyze asymmetric PCCC convergence, the extrinsic information SNR input/output relations of both decoders are evaluated. To determine whether the turbo decoder will converge or not at any particular E_b/N_0 , the extrinsic information SNR input/output relations of the two decoders are plotted on reverse axes (i.e., the input axis of one decoder is the output axis of the other decoder and vice versa). Based on the independent Gaussian model, it is easy to see that the asymmetric PCCC decoder will converge to $P_\ell^{(i)} = 0$ with sufficient number of iterations if and only if the two characteristics do

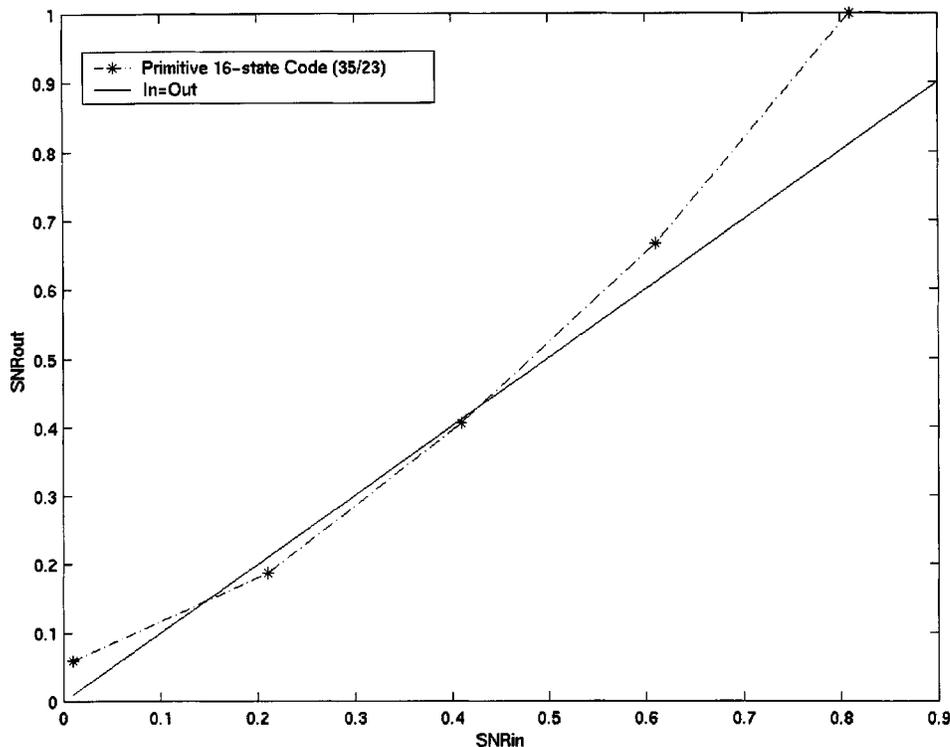


Fig. 9. Extrinsic information SNR input/output relation for the primitive 16-state code at $E_b/N_0 = 0$ dB.

not intersect. The rest of the procedure is the same as that used for symmetric PCCC.

The convergence threshold for the new asymmetric code is found to be -0.066 dB. This threshold is superior to that of both the symmetric 16-state and the symmetric eight-state PCCC built from the two constituent codes. The simulation results in Fig. 10 confirm that the asymmetric code is superior to both 16-state and eight-state symmetric codes in terms of performance in the waterfall region. This result is rather unexpected in that it indicates that good asymmetric codes can be obtained by mixing the constituent codes from two inferior symmetric codes.

Fig. 11 compares the performance of three codes: the new asymmetric code, the asymmetric code of Takeshita *et al.*, and the symmetric code with the best convergence threshold from Table I. The relative performance of the three codes in the waterfall region is in excellent agreement with the convergence threshold analysis. The three codes converge to 10^{-5} BER within 0.2 dB from their convergence thresholds, respectively, at this particular frame size (16384). The new asymmetric code offers performance improvement over the current state of the art. The new code outperforms the asymmetric code of Takeshita *et al.* in the waterfall region. Its convergence threshold is 0.162 dB lower than that of the best symmetric primitive 16-state PCCC, listed in Table III, and is within 0.086 dB of the convergence threshold of the best symmetric PCCC in Table I. Due to the strength of the primitive constituent code, the new asymmetric code also avoids the very poor asymptotic performance of the symmetric nonprimitive codes as shown in Fig. 11.

A first step toward the design of irregular PCCC was taken by Frey and MacKay [22] in which they allowed the symbol nodes to have variable degrees. However, their construction was limited by having a single constituent code [22]. Allowing more than one constituent code, with possibly different constraint lengths, would provide for additional degrees of freedom in the design, perhaps making it possible to optimize the degree sequences of both the symbol and check nodes to yield better convergence characteristic in the waterfall region. We are currently in the process of investigating irregular codes designed in this fashion.

As a final note, it is clear from Fig. 12 that the same relative performance predicted by the convergence threshold analysis holds in the waterfall region for a relatively small block size 1024. This observation contrasts the case of irregular LDPC codes constructed by optimizing the degree sequences for both the symbol and check nodes to allow for better convergence characteristics of the iterative decoder [16]. The performance of an exemplary rate-1/2 code constructed using this technique was shown to be within 0.1 dB from the Shannon limit with a block length of 1 000 000. For small block lengths, however, the performance of regular PCCC was shown to be still superior to irregular LDPC codes [16]. Therefore, irregular PCCC hold the promise of achieving superior performance to irregular LDPC codes for small and large block lengths.

C. SCCC

For PCCC, the constituent codes produce parity symbols that are not shared or exchanged; hence, the constituent decoders

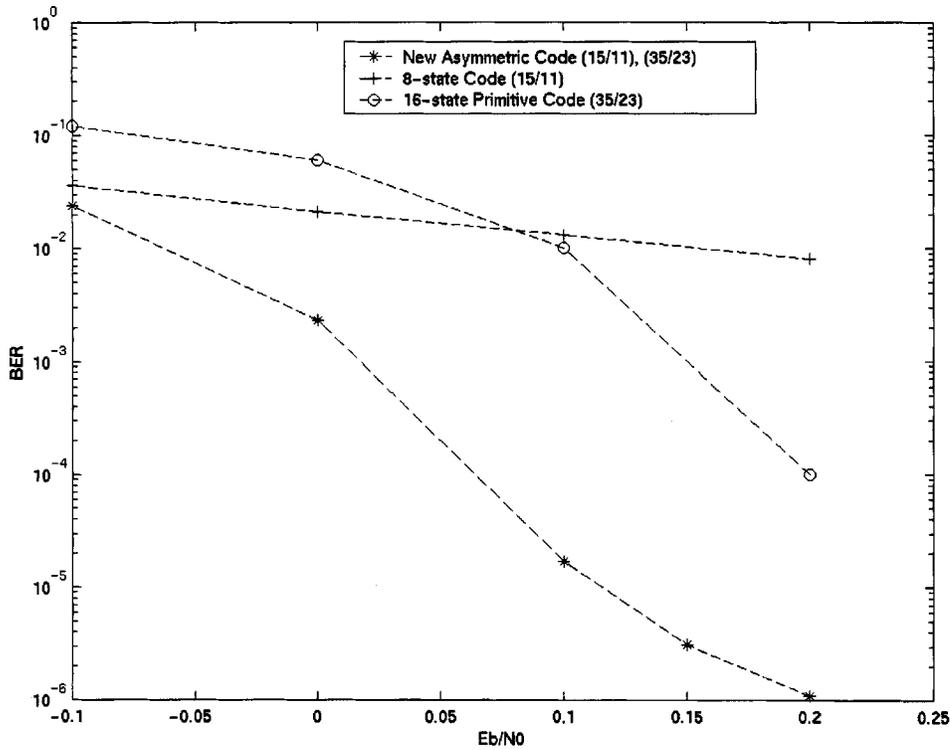


Fig. 10. Simulation results for the new asymmetric code and the two symmetric codes built from the eight-state and 16-state constituent codes.

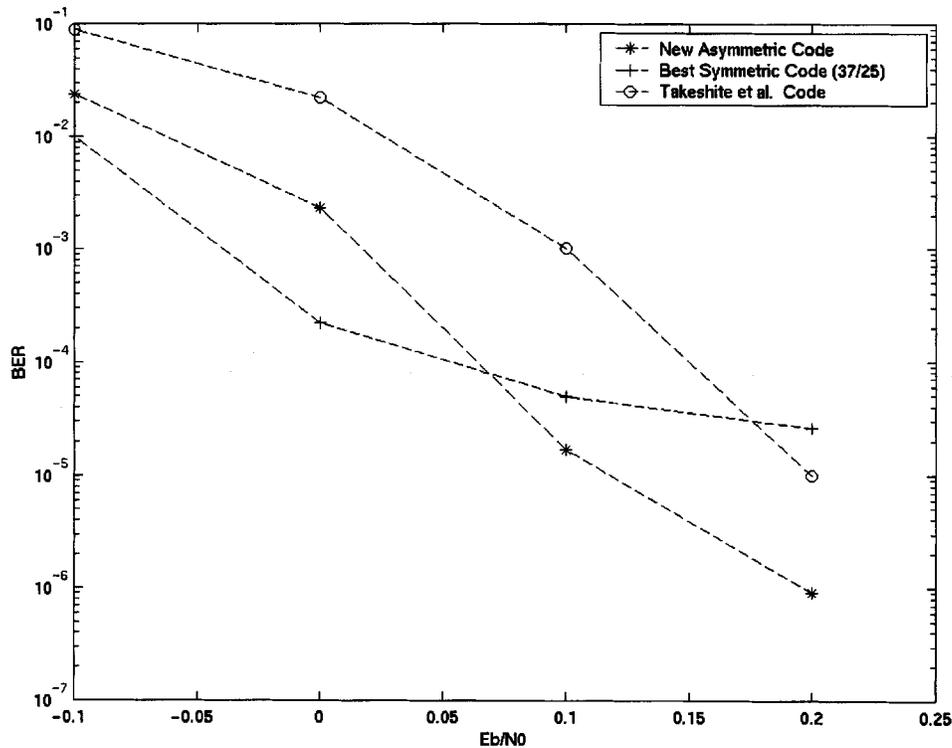


Fig. 11. Simulation results for the new asymmetric code, Takeshita *et al.* asymmetric code, and the 16-state code with the best convergence threshold with 16384 frame length.

produce extrinsic information for the data symbols only. The situation is different in the SCCC case. In the SCCC encoder, the parity symbols added by the outer code are multiplexed with

the data symbols, and the output stream is then interleaved and encoded by the inner code. At the receiver, the inner constituent decoder provides extrinsic information for both the data and the

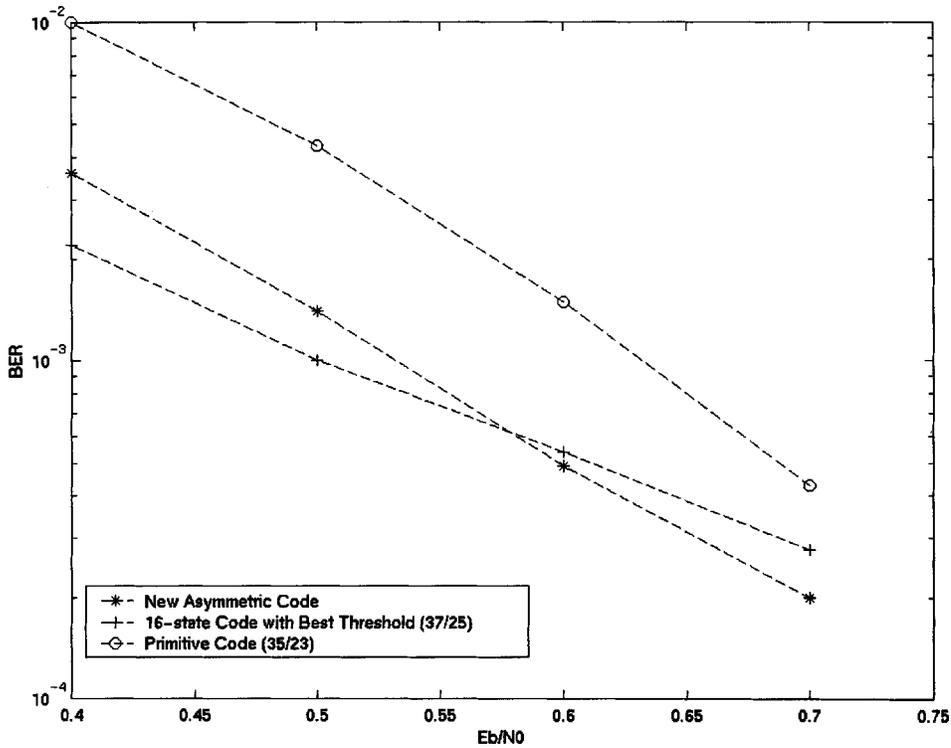


Fig. 12. Simulation results for frame length 1024.

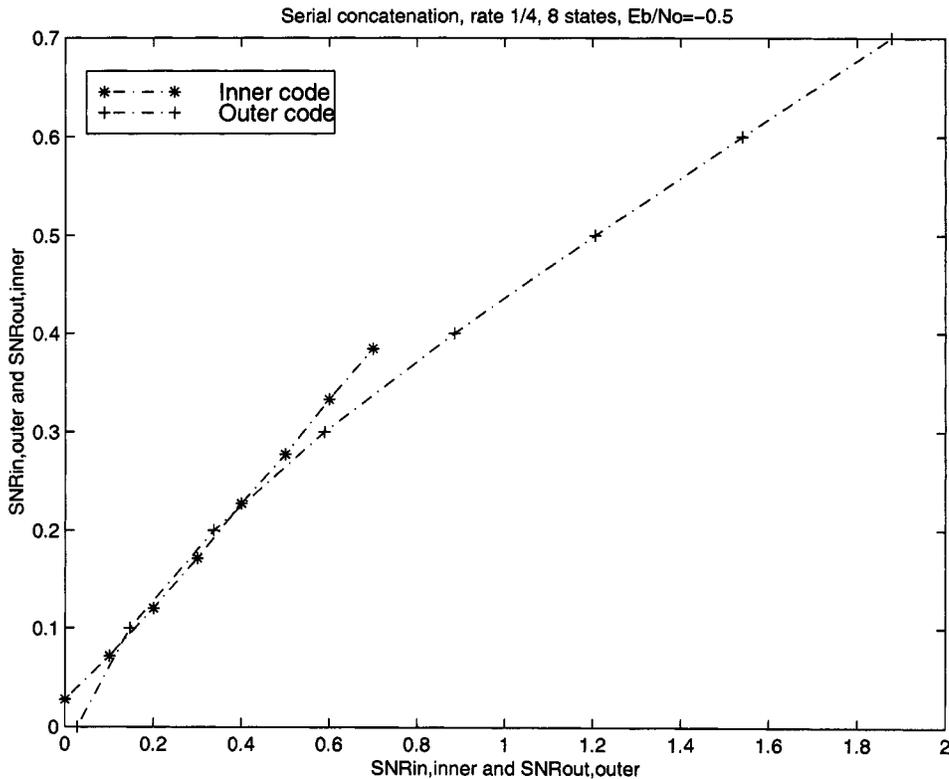


Fig. 13. Eight-state SCCC extrinsic information SNR characteristics: $E_b/N_0 = -0.5$ dB.

outer code's parity symbols each iteration, while the outer constituent decoder provides extrinsic information for the data symbols only.

To account for the asymmetry in the operation of the two constituent decoders, the analytical approach used for asymmetric

PCCC is required to analyze the SCCC convergence. The extrinsic information SNR input/output relations of both the inner and outer decoders are evaluated first. For the outer decoder, the input extrinsic information is added to the intrinsic information of both the data and parity symbols, whereas the input

TABLE IV
CONVERGENCE THRESHOLDS FOR REGULAR LDPC CODES

| M | rate | Gaussian Approximation Threshold in dB | Density Evolution Threshold in dB |
|-----|-------|--|-----------------------------------|
| 3 | 0.5 | 1.21 | 1.11 |
| 4 | 0.5 | 1.69 | 1.618 |
| 5 | 0.5 | 2.13 | 2.04 |
| 3 | 0.4 | 1.10 | 0.969 |
| 4 | 0.333 | 1.80 | 1.6744 |
| 3 | 0.25 | 1.15 | 1 |

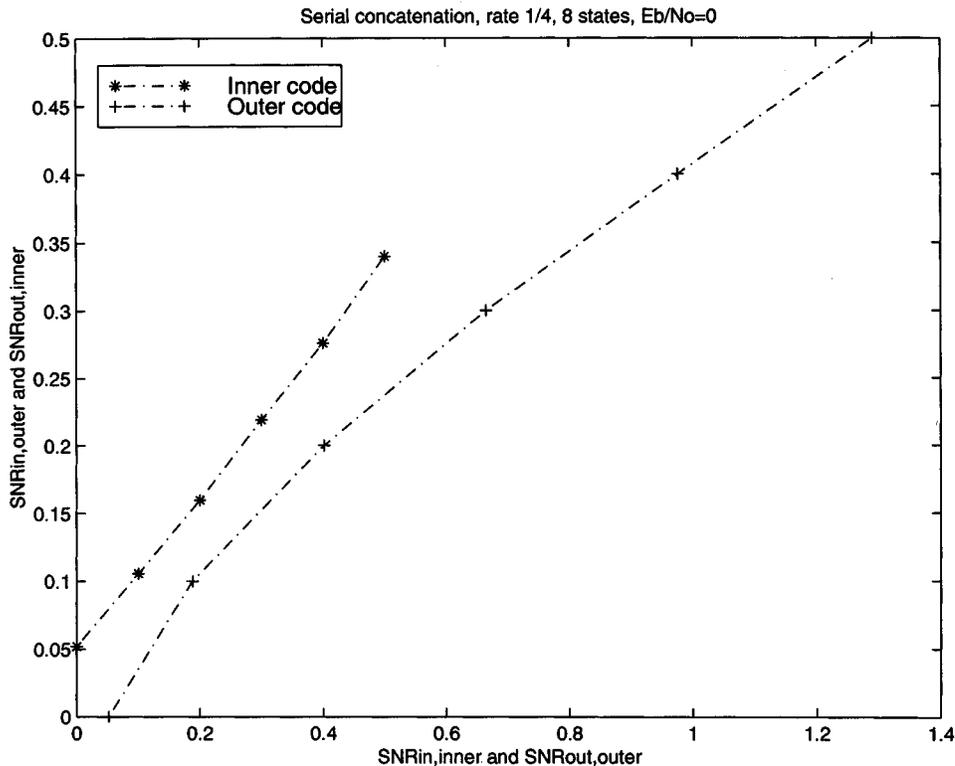


Fig. 14. Eight-state serially concatenated convolutional code extrinsic information SNR characteristics: $E_b/N_0 = 0$ dB.

extrinsic information to the inner decoder is added only to the intrinsic information of the data symbols. To determine whether the turbo decoder will converge or not at any particular E_b/N_0 , the extrinsic information SNR input/output relations of the inner and outer decoders are plotted on reverse axes. As an example, we consider the rate-1/4 SCCC using the eight-state constituent codes studied in [17]. Figs. 13 and 14 show the extrinsic information SNR input/output relations of the inner and outer decoders at $E_b/N_0 = 0$ dB and $E_b/N_0 = -0.5$ dB, respectively. From these diagrams, it is clear that the decoder will converge at $E_b/N_0 = 0$ dB and will not converge at $E_b/N_0 = -0.5$ dB. The result is in excellent agreement with the simulation results presented in [17], where the decoder was shown to converge at -0.25 dB. It is also clear from the figures that the 0.25-dB margin was only chosen for demonstration purposes.

D. Gallager LDPC Codes

In Gallager's regular LDPC codes, each data or parity symbol has the same degree M (i.e., connected to exactly M check nodes in the bipartite graph). For a rate r code, the degree of each check node is $\frac{M}{1-r}$. Therefore, each constituent decoder has $n = \frac{M}{1-r}$ inputs. In the iterative decoder, each symbol is estimated by M different and independent decoders. Each check node calculates each symbol output extrinsic information from the other $\frac{M}{1-r} - 1$ inputs. The input extrinsic information to any decoder is obtained from the other $M - 1$ outputs. Table IV shows the convergence thresholds for several LDPC codes obtained through our technique and the density evolution technique [10], respectively. In all the cases considered, the difference between the two estimated is within 0.15 dB.

TABLE V
CONVERGENCE THRESHOLDS FOR CHENG AND MC ELIECE HIGH-RATE CODES

| M | Threshold in dB |
|-----|-----------------|
| 6 | 1.8 |
| 7 | 2 |
| 8 | 2.3 |

Cheng and McEliece introduced their higher rate version of Gallager's LDPC codes in [6]. The difference between this coding scheme and the LDPC codes is that each parity symbol is involved in only one check equation. Hence, in the bipartite graph, the parity nodes and the check nodes are the same. The authors' expectation in [6] was that increasing the connectivity degree would improve the distance spectrum and, hence, the performance. They found that the effect of increasing the connectivity degree is to lower the error floor and shift the waterfall threshold to a higher SNR. While the improvement in the weight spectrum is supported by the improved asymptotic performance at high SNRs, the degradation in the waterfall region with increasing connectivity was not explained.

The iterative decoder for Cheng–McEliece codes is easily accommodated in the general decoder model and analyzed via the independent Gaussian assumption technique. Assuming that the degree of each data node is M , and the code rate is r , then the degree of each parity/check node is $\frac{Mr}{1-r} + 1$ and each constituent decoder will have $\frac{Mr}{1-r} + 1$ inputs. The iterative decoder in this scheme is different from Gallager's LDPC decoder in one major aspect. Unlike the former codes case, the constituent decoders provide extrinsic information for the data symbols only. Hence, the parity symbols will be coupled with intrinsic information only. However, similar to Gallager's LDPC codes, the extrinsic information for the information symbols is obtained from the other $(M - 1)$ decoders. Taking that into account, the convergence points predicted under the independent Gaussian assumption for codes with different connectivity degrees M are reported in Table V.

The detrimental effect of increasing the connectivity degree on the ability of the turbo decoder to converge to the ML solution is clear from the table. Since the performance in the low SNR region is largely determined by the decoder convergence, the predicted convergence points explain the degradation in performance observed in [6].

IV. EXTENSION TO FADING CHANNELS

Available research [10] on the behavior of iterative decoding thus far has not considered the fading channel scenario. In this section, we extend our convergence analysis to certain frequency nonselective slow and fast Rayleigh fading models; the extension to Rician channels is straightforward. Perfect channel state information is assumed to be available at the receiver.

A. Quasi-Static Fading Channels

In this model, the fading amplitude is constant across one code word and is independent from one codeword to the next.

Hence, fading causes the effective SNR to vary from one codeword to the next according to the distribution of the square of the fading amplitude. From the AWGN analysis, it is clear that the iterative decoder will converge for a particular codeword with a fixed fading amplitude if and only if the effective SNR for this codeword exceeds the convergence threshold. Let a be the fading amplitude, E_s/N_0 be the transmitted symbol energy-to-white-noise ratio, and E_{Th}/N_0 be the minimum symbol energy-to-white-noise ratio required for iterative decoder convergence in the AWGN channel. Then, the frame error probability for the quasi-static Rayleigh fading channel is given by

$$P_f = P\left(a < \sqrt{\frac{E_{Th}}{E_s}}\right) = 1 - e^{-\frac{E_{Th}}{E_s}}. \quad (4)$$

The importance of this result is twofold. First, it provides a simple and general technique of characterizing the performance of graphical codes over slow-fading channels. Second, it shows that the performance of graphical codes in slow-fading channels is largely determined by the iterative decoder convergence characteristics at high as well as low SNRs. This suggests that, for slow Rayleigh-fading channels, it is more important to optimize for decoder convergence than code weight spectrum.

B. Fast-Fading Channels

For the fast-fading channel, each symbol is multiplied by a different and statistically independent Rayleigh fading amplitude. In this case, we modify the independent Gaussian assumption so that, even though the channel variables are no longer Gaussian, the extrinsic information variables are still assumed to be Gaussian. Under this model, Proposition 1 is still true. Essentially, the same proof shows that the SNRs of the decision statistic $x_\ell^{(i)}$ and the extrinsic information $\xi_\ell^{(i)}$ are both non-decreasing functions of the input SNRs, and the one SNR increases to infinity if and only if the other does. But, since the decision statistic $x_\ell^{(i)}$ is no longer Gaussian, it no longer immediately follows as in the AWGN case that $P(\hat{d}_\ell^{(i)} \neq d_\ell) \rightarrow 0$ as $\text{SNR}(x_\ell^{(i)}) \rightarrow \infty$ or $\text{SNR}(\xi_\ell^{(i)}) \rightarrow \infty$.

To repair that part of the proof, we note that

$$\begin{aligned} P(\hat{d}_\ell^{(i)} \neq d_\ell) &= \int_0^\infty P(\hat{d}_\ell^{(i)} \neq d_\ell | a_\ell) P(a_\ell) da_\ell \\ &\leq I(a^*) + P(\hat{d}_\ell^{(i)} \neq d_\ell | a_\ell = a^*) (1 - I(a^*)) \end{aligned}$$

where a^* is arbitrary and

$$I(a^*) = \int_0^{a^*} P(a) da.$$

By conditioning on a_ℓ , we can apply Proposition 2.1 to show that $P(\hat{d}_\ell^{(i)} \neq d_\ell | a^*) \rightarrow 0$ as the extrinsic SNR $\text{SNR}(\xi_\ell^{(i)}) \rightarrow \infty$. Thus, we can make $P(\hat{d}_\ell^{(i)} \neq d_\ell) < \epsilon$ by first choosing a^* so that $I(a^*) < \epsilon/2$ and then choosing $\text{SNR}(\xi_\ell^{(i)})$ sufficiently large so that $P(\hat{d}_\ell^{(i)} \neq d_\ell | a^*) < \epsilon/2$. This establishes the desired result that $P(\hat{d}_\ell^{(i)} \neq d_\ell) \rightarrow 0$ as $\text{SNR}(\xi_\ell^{(i)}) \rightarrow \infty$ [and hence as $\text{SNR}(x_\ell^{(i)}) \rightarrow \infty$].

The rest of the analysis in Section II regarding the existence of thresholds $\tau(v)$ and ν then follows exactly as in the AWGN case. Empirically, it is found that the Rayleigh-fading convergence threshold is also finite and nonzero. For example, in [23], Hall *et al.* noted that the performance of PCCC in the fast

Rayleigh-fading channel exhibits a waterfall behavior similar to that in the AWGN case but with a larger threshold value. Our analysis based on the independent Gaussian assumption as modified for the Rayleigh-fading channel predicts threshold values in excellent agreement with the simulation results of [23].

V. CONCLUSION

The main result established in this paper is that the performance of graphical codes in the low SNR region is governed by the convergence characteristics of the iterative decoder independent of the distance spectrum of the code. Thus, traditional optimization of the code parameters with respect to the distance spectrum will not in general improve the performance in the low SNR region. To improve the performance in this region, different optimization should be considered to allow for better convergence characteristics of the turbo decoder. This explains, for example, why more sophisticated code constructions such as SCCC which aim to improve asymptotic code performance do not outperform Berrou's original construction in the low SNR region. The simple method developed in this paper to analyze the iterative decoder convergence is based on the Gaussian approximation and yields very accurate results compared with the literature. Based on the insights obtained from such analysis, we presented an interesting construction for asymmetric PCCC that enjoys a favorable performance in the waterfall region. This construction provides a new ingredient to add to Frey and MacKay irregular PCCC design [22] and improves upon the asymmetric code proposed by Takeshita *et al.* [18]. Finally, we extended our analysis to fast- and slow-fading channels. The fast-fading scenario turns out to be very similar to the AWGN case in which decoder convergence characteristics determine the waterfall threshold, while the performance at high SNR is governed by the code weight spectrum. In contrast, for slow-fading channels, the performance of graphical codes is largely determined by the decoder convergence characteristics for all input SNRs.

ACKNOWLEDGMENT

The authors wish to thank the anonymous reviewers for this careful review and the reference to S. ten Brink's work [14]. They believe that their comments have significantly helped in improving the presentation of this paper.

REFERENCES

- [1] R. J. McEliece, D. J. C. MacKay, and J. F. Cheng, "Turbo decoding as an instance of Pearl's 'belief propagation' algorithm," *IEEE J. Select. Areas Commun.*, vol. 16, pp. 297–302, Feb. 1998.

- [2] D. J. C. MacKay. (1996) Near Shannon limit performance of low density parity check codes. *IEEE Commun. Lett.* [Online] Available: <http://131.111.48.24/mackay/homepage.html>
- [3] N. Wiberg, "Codes and decoding on general graphs," Linköping Studies in Sci. and Technol., Linköping, Sweden, Ph.D. dissertation 440, 1996.
- [4] S. Benedetto, G. Montorsi, D. Divsalar, and F. Pollara, "Serial concatenation of interleaved codes: Performance analysis, design, and iterative decoding," *Telecom. and Data Acquisition Progress Repts., Jet Propulsion Lab.*, pp. 42–126, Aug. 1996.
- [5] B. Frey and D. MacKay, "Trellis constrained codes," in *Proc. 35th Allerton Conf. Communications, Control and Computing*, Oct. 1997. Available: [Online] <http://131.111.48.24/mackay/homepage.html>.
- [6] J. F. Cheng and R. J. McEliece. (1996, Oct.) Some high rate near capacity codes for the Gaussian channel. *Proc. 34th Allerton Conf. Communications, Control and Computing* [Online] Available: <http://www.systems.caltech.edu/systems/jfc/index.html>
- [7] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding," in *IEEE Int. Conf. Communications*, May 1993, pp. 1064–1070.
- [8] I. Sason and S. Shamai, "Improved upper bounds on the performance of parallel and serial concatenated turbo codes via their ensemble distance spectrum," in *Proc. IEEE Int. Symp. Information Theory*, Cambridge, MA, Aug. 1998, p. 30.
- [9] H. El Gamal, "On the theory and applications of space-time and graph based codes," Ph.D. dissertation, Univ. Maryland, College Park, May 1999.
- [10] T. Richardson and R. Urbanke, "The capacity of low-density parity-check codes under message passing decoding," *IEEE Trans. Inform. Theory*, vol. 47, pp. 599–618, Feb. 2001.
- [11] M. Luby, M. Mitzenmacher, A. Shokrollahi, and D. Spielman, "Analysis of low density codes and improved designs using irregular graphs," in *Proc. 30th Annual ACM Symp. Theory of Computation*, 1998, pp. 249–258.
- [12] T. Richardson and R. Urbanke. (1999, Aug.) An introduction to the analysis of iterative coding systems. *IMA Proc.* [Online] Available: <http://icavwww.epfl.ch/ruediger/publications.html>
- [13] T. Richardson and R. Urbanke, "Thresholds for turbo codes," presented at the IEEE Int. Symp. Information Theory, Cambridge, MA, June 2000.
- [14] S. ten Brink, "Trajectories of PCCC decoders," presented at the 3rd IEEE/ITG Symp. Source and Channel Coding, Munich, Germany, Jan. 2000.
- [15] F. Malardel. (1996, November) Simulation and optimization of the turbo decoding algorithm. Final year project report. [Online] Available: <http://www.itr.unisa.edu.au/steven/turbo/fm.ps.gz>
- [16] T. Richardson, A. Shokrollahi, and R. Urbanke, "Design of provably good low density parity check codes," *IEEE Trans. Inform. Theory*, to be published.
- [17] S. Benedetto, G. Montorsi, D. Divsalar, and F. Pollara, "A soft-input soft-output maximum *a posteriori* module to decode parallel and serially concatenated codes," *Telecom. and Data Acquisition Progress Repts., Jet Propulsion Lab.*, pp. 42–127, Nov. 1996.
- [18] O. Takeshita, O. M. Collins, P. C. Massey, and J. D. J. Costello, "A note on asymmetric turbo codes," *IEEE Commun. Lett.*, vol. 3, pp. 69–71, Mar. 1999.
- [19] L. C. Perez, J. Seghers, and J. D. J. Costello, "A distance spectrum interpretation of turbo codes," *IEEE Trans. Inform. Theory*, vol. 42, pp. 1698–1709, Nov. 1996.
- [20] L. Lee, F. Sun, K. Karimullah, M. Eroz, and A. R. Hammons, "Third generation wireless communication technologies," in *Wireless Multimedia Network Technologies*, R. Ganesh, K. Pahlavan, and Z. Zvonar, Eds. Norwell, MA: Kluwer, 2000, pp. 23–44.
- [21] S. Benedetto, R. Garello, and G. Montorsi, "A search for good convolutional codes to be used in the construction of turbo codes," *IEEE Trans. Commun.*, vol. 46, pp. 1101–1105, Sept. 1998.
- [22] B. Frey and D. MacKay. Irregular turbo codes. presented at 37th Allerton Conf. Communications, Control and Computing. [Online] Available: <http://131.111.48.24/mackay/homepage.html>. <http://icavwww.epfl.ch/>
- [23] E. Hall and S. Wilson, "Design and analysis of turbo codes on rayleigh fading channels," *IEEE J. Select. Areas Commun.*, vol. 23, pp. 371–377, May 1998.