Analysis of the Iterative Decoding of LDPC and Product Codes Using the Gaussian Approximation

Frederic Lehmann and Gian Mario Maggio, *Member, IEEE*

***Abstract*—We propose a novel density evolution approach to analyze the iterative decoding algorithms of low-density parity-check (LDPC) codes and product codes, based on Gaussian densities. Namely, for these classes of codes we derive a one–dimensional (1-D) map whose iterates directly represent the error probability both for the additive white Gaussian noise (AWGN) and the Rayleigh-fading channel. These simple models allow a qualitative analysis of the nonlinear dynamics of the decoding algorithm. As an application, we compute the decoding thresholds and show that they are consistent with the simulation results available in the literature.**

***Index Terms*—Density evolution, Gaussian approximation, low-density parity-check (LDPC) codes, nonlinear dynamics, threshold computation.**

I. Introduction

Recently, it has been demonstrated that iterative-decoding algorithms can perform at rates extremely close to the Shannon limit imposed by the noisy channel coding theorem [1], yet with reasonable complexity. In particular, irregular *low-density parity-check* (LDPC) *codes* and *product codes* are among the most promising candidates for future applications. LDPC codes were originally introduced by Gallager [2] in 1962, and rediscovered by MacKay*et al.* [3] in 1996. The crucial innovation of LDPC codes being the introduction of iterative decoding algorithms. Recently, it has been recognized that the various message-passing decoding algorithms, which provide good decoding performances for these codes, can be formulated in terms of a general framework, namely, the sum-product algorithm [4]. On the other hand, product codes were introduced by Elias [5] as a class of powerful concatenated block codes with high minimum distance. Elias proposed a suboptimal decoding algorithm based on the sequential algebraic decoding of the constituent codes to keep the decoding complexity relatively low. The introduction of turbo codes [6] later showed that performance near capacity can be obtained by decoding constituent codes with an iterative soft-input soft-output (SISO) decoder. Using the same idea, Pyndiah [7], [8], improved the decoding scheme of Elias by replacing the algebraic decoders by iterative SISO decoding based on the Chase algorithm [9].

Recently, several techniques have been proposed in the literature to analyze iterative decoding by tracking the density of the information exchanged in the decoder. This idea was originally introduced for LDPC codes [10], [11], under the name of *density evolution*. For these particular codes, the exact densities of the messages exchanged in the decoder are available because the extrinsic information admits a closed-form representation given by the “tanh rule. ” Density evolution, though, requires a numerical evaluation of the densities of the messages used by the decoder and is usually computationally intensive. Often, the density of the extrinsic information is approximated by a Gaussian either to simplify the analysis or when a closed-form expression of the extrinsic information is not available. The validity of this assumption was first recognized for LDPC and turbo codes by Wiberg [12] and used in [13] to perform an approximate analysis of LDPC codes. The Gaussian approximation in conjunction with Monte Carlo simulations has also been proposed to analyze the performances of the turbo decoding algorithm [14], [15]. Previous work concerning Gaussian approximations has relied on different parameters in order to obtain a one-dimensional (1–D) model, namely, the mean value in [13], the signal-to-noise ratio (SNR) in [14], [15], the mutual information in [16], [17], and the bit-error rate (BER) [17], [18]. Another method based on matching mean and covariance was presented in [19].

In this correspondence, we propose a model of the iterative decoding of LDPC and product codes based on the BER, using the Gaussian approximation. Under the Gaussian approximation, the mean, SNR, and mutual information can be easily converted to the corresponding BER as shown in [13], [14], and [17], respectively. In contrast with this approach, our expression of the BER is not merely a byproduct, but the parameter by which we identify the tail of the Gaussian distribution. For LDPC codes, we follow a method somewhat similar to the one suggested in [13] in order to analyze the message-passing decoder. However, our method is based on a closed-form expression in terms of error probabilities. By error probability we mean here the probability that variable nodes are sending incorrect messages. Moreover, we show that our approach leads to the correct stability condition, that is consistent with density evolution. On the other hand, for product codes our starting point is [8]. For this class of codes, we introduce a novel density evolution approach based upon the evaluation of the extrinsic information exchanged by the constituent decoders. In both cases, despite the simplicity of the model, it is possible to predict the thresholds of the decoder with acceptable accuracy, when compared to simulation results.

This correspondence is organized as follows. In Section II, we recall the basic principles of LDPC codes and derive a 1-D model of the message-passing decoder based on Gaussian densities, both for the additive white Gaussian noise (AWGN) and the Rayleigh-fading channel models. Section III is devoted to the iterative decoding of product codes. Namely, for both the AWGN and the Rayleigh-fading channel, we derive a 1-D map based on the Gaussian approximation. Section IV shows an application of the models derived for the iterative decoding of LDPC and product codes, for threshold computation purposes. Specifically, we illustrate the qualitative nonlinear dynamics of the iterative decoding process and explain the mechanisms underlying the existence of the threshold.

II. Model of the Message-Passing Decoding of LDPC Codes

A. Preliminaries on LDPC Codes

An LDPC code is defined by a bipartite graph [10] formed by variable and check nodes appropriately related by edges. Assume d_v (resp., d_c) is the maximum variable (resp., check) node degree; we denote the variable (resp., check) degree distribution polynomial of the graph by $\lambda(x) = \sum_{i=2}^{d_v} \lambda_i x^{i-1}$ (resp., $\rho(x) = \sum_{i=2}^{d_c} \rho_i x^{i-1}$) [11]. In particular, if $\lambda(x) = x^{d_v-1}$ and $\rho(x) = x^{d_c-1}$, the code is said to be *regular*, otherwise, the code is said to be *irregular.*

The message-passing algorithm used to decode LDPC codes implements the sum-product algorithm applied to the bipartite graph of the code. The message v sent by a variable node to a check node on edge e

Manuscript received September 13, 2002; revised April 28, 2003. This work is sponsored by STMicroelectronics, Inc., Advanced System Technology (AST) group and supported by the IEEE.

F. Lehmann is with the Laboratory for Analysis and Architecture of Systems (LAAS), Toulouse, France (e-mail: flehmann@laas.fr).

G. M. Maggio is with STMicroelectronics, Inc., Advanced System Technology (AST), San Diego, CA USA, and with the Center for Wireless Communications, University of California, San Diego, La Jolla, CA 92093–0407 USA (e-mail: gian-mario.maggio@st.com; gmaggio@ucsd.edu).

Communicated by K. Abdel-Ghaffar, Associate Editor for Coding Theory.

Digital Object Identifier 10.1109/TIT.2003.819335

is the log-likelihood ratio of that variable node, given the log-likelihood ratios of the check nodes u_i received on all incoming edges except e and given the channel log-likelihood ratio u_0 [10]

$$v = u_0 + \sum_i u_i. \tag{1}$$

The message u sent by a check node to a variable node on edge e is the log-likelihood ratio of that check node, given the log-likelihood ratios of the variable nodes v_i received on all incoming edges except e [10]

$$\tanh\left(\frac{u}{2}\right) = \prod_i \tanh\left(\frac{v_i}{2}\right). \tag{2}$$

Equations (1) and (2) constitute one decoding iteration and each variable node message is initialized with the channel log-likelihood ratio u_0 of the corresponding bit [11], [13].

B. Gaussian Density Approximation

As already stated in [13], the density of variable and check node messages is close to a Gaussian, although this is less obvious for check node messages. This is especially true when the right degree distribution polynomial $\rho(x)$ is concentrated on a few degrees, which is verified for regular codes and for almost all good irregular codes as well [13]. Moreover, the analysis is greatly simplified if a density, call it f, verifies the so-called symmetry condition: $f(x) = e^x f(-x)$. It was shown by Richardson *et al.* [11] that the densities of u_0, v, and u in (1) and (2) satisfy the symmetry condition. Therefore, we can assume that all the messages involved in the decoding process have a symmetric Gaussian distribution of the form

$$f_m(x) = \frac{1}{\sqrt{4\pi m}} e^{-\frac{(x-m)^2}{4m}},$$

where the parameter m is the mean.

Throughout this analysis, we will restrict ourselves to binary phase-shift keying (BPSK) modulation (binary $0 \to +1$, binary $1 \to -1$). The message-passing algorithm can be analyzed with the following assumption. If the block length of the code tends to infinity, the concentration theorem [10] ensures that the performance of a particular bipartite graph chosen at random can be assimilated to the average performance of the cycle-free graph, i.e., the messages received by every node at every iteration are independent and identically distributed (i.i.d.) random variables. In the remainder of the present correspondence, this assumption is supposed to be valid. Without loss of generality, we will also suppose that the all-zero codeword is sent, therefore, the error probability $P_e^l(\sigma)$ at iteration l is simply the average probability that the variable node messages are negative [11].

1) AWGN Channel: We consider here an AWGN channel and denote by σ the noise standard deviation. Let $m_{u_0} = \frac{2}{\sigma^2}$ be the mean of u_0, and m_u^l and m_v^l be the mean of u and v at iteration l, respectively. Our goal is to find an expression of $P_e^{l+1}(\sigma)$, the error probability at iteration $l+1$, as a nonlinear function of $P_e^l(\sigma)$, the error probability at iteration l. In order to achieve this, consider a check node u of degree j at iteration $l+1$; then it follows from (2) that

$$\operatorname{sign}(u) = \sum_i \operatorname{sign}(v_i) \mod 2$$

where $\operatorname{sign}(x)$ is 0 if $x > 0$, and 1 otherwise. It was shown by Gallager [2] that the probability of $u < 0$ for degree j nodes is

$$\frac{1}{2}\left[1 - \left(1 - 2P_e^l(\sigma)\right)^{(j-1)}\right].$$

Now, by averaging over all the possible check node degrees and keeping in mind that the density of u is approximately a symmetric Gaussian density, we obtain the average probability of $u < 0$ as

$$Q\left(\sqrt{\frac{m_u^{l+1}}{2}}\right) = \frac{1}{2}\sum_{j=2}^{d_c} \rho_j \left[1 - \left(1 - 2P_e^l(\sigma)\right)^{(j-1)}\right] \tag{3}$$

where

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^{+\infty} e^{-\frac{t^2}{2}}\, dt.$$

Similarly, consider (1) in the case of a variable node v of degree i, then

$$m_v^{l+1} = m_{u_0} + (i-1) m_u^{l+1}$$

and the density of v is a symmetric Gaussian since v is a sum of random variables whose density is Gaussian and symmetric; therefore, the probability of $v < 0$ is

$$Q\left(\sqrt{\frac{m_v^{l+1}}{2}}\right) = Q\left(\sqrt{\frac{1}{\sigma^2} + (i-1)\frac{m_u^{l+1}}{2}}\right).$$

Now, by averaging over all the possible variable node degrees, we obtain

$$P_e^{l+1}(\sigma) = \sum_{i=2}^{d_v} \lambda_i Q\left(\sqrt{\frac{1}{\sigma^2} + (i-1)\frac{m_u^{l+1}}{2}}\right). \tag{4}$$

Then by combining (3) and (4), and defining the polynomial $s(x)$ as

$$s(x) = \frac{1}{2}\sum_{j=2}^{d_c} \rho_j \left[1 - (1-2x)^{(j-1)}\right]$$

we obtain the expression of the error probability at iteration $l+1$ as

$$P_e^{l+1}(\sigma) = \sum_{i=2}^{d_v} \lambda_i Q\left(\sqrt{\frac{1}{\sigma^2} + (i-1)\left\{Q^{-1}\left(s\left(P_e^l(\sigma)\right)\right)\right\}^2}\right). \tag{5}$$

In other words, (5) represents a nonlinear 1-D map of the form

$$P_e^{l+1}(\sigma) = f\left(P_e^l(\sigma), \sigma\right), \qquad l \geq 1 \tag{6}$$

describing the dynamics of the message-passing algorithm in terms of error probability, with $P_e^0(\sigma) = Q\left(\frac{1}{\sigma}\right)$. The nonlinear map $f(x;\sigma)$ is defined as

$$\boxed{f(x;\sigma) = \sum_{i=2}^{d_v} \lambda_i Q\left(\sqrt{\frac{1}{\sigma^2} + (i-1)\left\{Q^{-1}\left(s(x)\right)\right\}^2}\right)} \tag{7}$$

where σ acts as a control parameter.

We now show that the map $f(x;\sigma)$ admits the same stability condition derived in [11] by using density evolution.

Theorem 2.1: $x = 0$ is a stable fixed point of the map $f(x;\sigma)$ if and only if $\lambda'(0)\rho'(1) < e^{\frac{1}{2\sigma^2}}$.

Proof: From (7), we have [20]

$$\lim_{x \to 0} f(x;\sigma) = 0$$

thus, 0 is a fixed point of the map. Moreover, this fixed point is stable if

$$\lim_{x \to 0} \frac{\partial f}{\partial x}(x,\sigma) < 1.$$

It is shown in Appendix I that

$$\lim_{x \to 0} \frac{\partial f}{\partial x}(x,\sigma) = e^{-\frac{1}{2\sigma^2}} \lambda'(0)\rho'(1)$$

which completes the proof. □

2) Rayleigh-Fading Channel: On the Rayleigh-fading channel, the density of the channel log-likelihood ratios is [17]

$$p_{u_0}(u_0) = \frac{\sigma^2}{2\sqrt{1+2\sigma^2}} \exp\left(\frac{u_0 - \sqrt{1+2\sigma^2}|u_0|}{2}\right) \tag{8}$$

while the channel crossover probability is given by

$$p = \int_{-\infty}^{0} p_{u_0}(u_0)du_0 = \frac{1}{2}\left(1 - \frac{1}{\sqrt{1+2\sigma^2}}\right). \tag{9}$$

The check-node messages have a density which can be approximated as a symmetric Gaussian with mean m_u^{l+1} at iteration $(l+1)$. It follows that, at iteration $(l+1)$, the density of a variable node v of degree i is the convolution of $p_{u_0}(u_0)$ with a symmetric Gaussian of mean $(i-1)m_u^{l+1}$. The probability that $v < 0$ is known as [17]

$$\begin{aligned} &Q\left(\sqrt{\frac{(i-1)m_u^{l+1}}{2}}\right) - \frac{1}{\sqrt{1+2\sigma^2}} \\ &\quad \times Q\left(\sqrt{(1+2\sigma^2)\frac{(i-1)}{2}m_u^{l+1}}\right) \\ &\quad \times \exp\left(\frac{\sigma^2(i-1)m_u^{l+1}}{2}\right). \end{aligned} \tag{10}$$

By averaging this expression over all the variable node degrees

$$\begin{aligned} P_e^{l+1}(\sigma) = \sum_{i=2}^{d_v} \lambda_i \Bigg\{ & Q\left(\sqrt{\frac{(i-1)m_u^{l+1}}{2}}\right) \\ & - \frac{1}{\sqrt{1+2\sigma^2}} Q\left(\sqrt{(1+2\sigma^2)\frac{(i-1)}{2}m_u^{l+1}}\right) \\ & \times \exp\left(\frac{\sigma^2(i-1)m_u^{l+1}}{2}\right)\Bigg\}. \end{aligned}$$

Recalling that $\frac{m_u^{l+1}}{2} = \left[Q^{-1}\left(s(P_e^l(\sigma))\right)\right]^2$, we can define the map in (11) (see the bottom of the page). The iterates of the error probability of the variable node messages are then given by

$$\begin{cases} P_e^{l+1}(\sigma) & = & f\left(P_e^l(\sigma), \sigma\right) \\ P_e^0(\sigma) & = & \frac{1}{2}\left(1 - \frac{1}{\sqrt{1+2\sigma^2}}\right). \end{cases}$$

Similarly to the AWGN case, 0 is a fixed point of f, whose stability condition (see Appendix II) is given by

$$\lambda'(0)\rho'(0) < 1 + \frac{1}{2\sigma^2} \tag{12}$$

which is consistent with [21].

III. Model of the Iterative Decoding of Product Codes

A. Preliminaries on Product Codes

A product code $C_p = C_1 \bigotimes C_2$ is defined by the serial concatenation of two block codes $C_1(n_1, k_1, d_1)$ and $C_2(n_2, k_2, d_2)$. We assume here that binary codes are used. The information bits are placed in an array of k_1 lines and k_2 columns. The columns (resp., rows) are encoded using C_1 (resp., C_2), as described in Fig. 1.

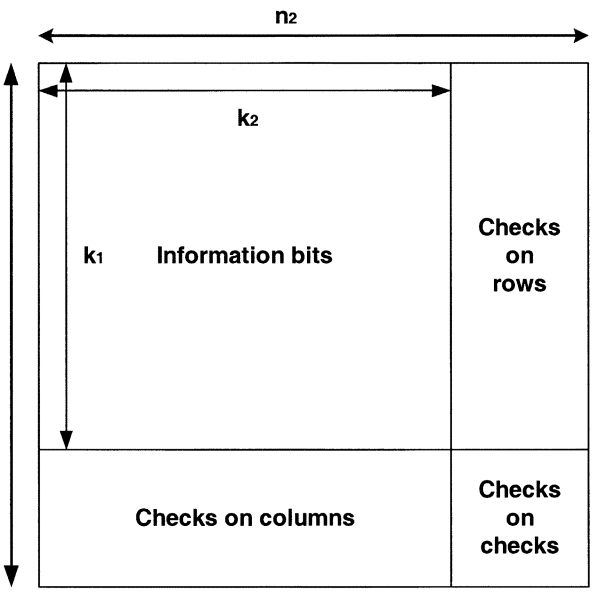

Fig. 1. Product code $C_p = C_1 \bigotimes C_2$.

The iterative decoding process is described in Fig. 2. The decoding is performed iteratively column-wise then row-wise using the modified Chase SISO algorithm [8]. The column decoder uses channel observations Z and *a priori* information A_c in the form of log-likelihood ratios to generate an *a posteriori* log-likelihood ratio L_c for each bit. The *extrinsic* information is then defined as $E_c = L_c - Z - A_c$. After block interleaving, E_c is used as *a priori* information A_r in conjunction with Z by the row decoder to generate an *a posteriori* log-likelihood ratio L_r for each bit. The *extrinsic* information is then defined as $E_r = L_r - Z - A_r$ and is used as *a priori* information for the columns after block interleaving.

B. Analysis of the Iterative Decoder

Let us assume that Chase Algorithm 2 is used as the row/column decoder with the modification proposed in [8] to obtain soft outputs. Let $C(n,k,d)$ be one of the constituent codes; the error-correcting capability of the code is $t = \lfloor (d-1)/2 \rfloor$ and the number of least reliable positions used to generate the list of candidate codewords is $l = \lfloor d/2 \rfloor$. Assume also that BPSK signaling is used (binary $0 \rightarrow +1$, binary $1 \rightarrow -1$) and that the all-zero codeword is transmitted. We modify the method proposed in [22] to obtain a good approximation of the BER of the constituent decoder. Assume the decoder admits as its input a log-likelihood ratio vector $\boldsymbol{r} = (r_1, \ldots, r_n)$ instead of the channel outputs. Let $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n)$ be the corresponding reliability vector with $\alpha_i = |r_i|$, $i = 1, \ldots, n$. If i transmission errors occur, the reliability values corresponding to the i hard decision errors (resp., $n-i$ correct hard decisions) are reordered in decreasing order: $\beta_1(i) \geq \beta_2(i) \geq \cdots \geq \beta_i(i)$ (resp., $\gamma_1(n-i) \geq \gamma_2(n-i) \geq \cdots \geq$

$$\begin{aligned} f(x;\sigma) = \sum_{i=2}^{d_v} \lambda_i \Bigg\{ & Q\left(\sqrt{(i-1)\left[Q^{-1}\left(s(x)\right)\right]^2}\right) \\ & - \frac{1}{\sqrt{1+2\sigma^2}} Q\left(\sqrt{(1+2\sigma^2)(i-1)\left[Q^{-1}\left(s(x)\right)\right]^2}\right) \exp\left(\sigma^2(i-1)\left[Q^{-1}\left(s(x)\right)\right]^2\right)\Bigg\}. \end{aligned} \tag{11}$$

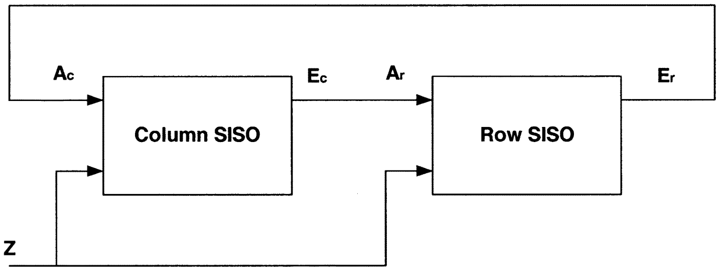

Fig. 2. Block diagram of the iterative decoder of a product code.

$\gamma_{n-i}(n-i))$. A good approximation of the word error rate of Chase Algorithm 2 is given by [22]

$$P_e = \sum_{i=t+1}^{l+t} \binom{n}{i} p^i (1-p)^{n-i} P\left(\beta_{t+1}(i) \geq \gamma_{n-l-t}(n-i)\right) + \sum_{i=l+t+1}^{n} \binom{n}{i} p^i (1-p)^{n-i} \quad (13)$$

where p represents the channel crossover probability. With a slight modification, we obtain an approximation of the BER as

$$P_b = \sum_{i=t+1}^{l+t} \frac{i}{n} \binom{n}{i} p^i (1-p)^{n-i} P\left(\beta_{t+1}(i) \geq \gamma_{n-l-t}(n-i)\right) + \sum_{i=l+t+1}^{n} \frac{i}{n} \binom{n}{i} p^i (1-p)^{n-i}. \quad (14)$$

Assume that the elements of $\boldsymbol{r}$ are i.i.d. with probability density function $f(x)$ and let $f_\alpha^c(x)$ (resp., $f_\alpha^e(x)$) be the density associated with a reliability corresponding to a correct (resp., erroneous) hard decision, then

$$p = \int_{-\infty}^{0} f(x)dx \quad (15)$$

$$f_\alpha^c(x) = \frac{f(x)}{1-p} u(x) \quad (16)$$

$$f_\alpha^e(x) = \frac{f(-x)}{p} u(x), \quad (17)$$

where $u(x)$ is the unit step function. The method to calculate the term $P\left(\beta_{t+1}(i) \geq \gamma_{n-l-t}(n-i)\right)$ in (14) using (16) and (17) can be found in [22].

C. Gaussian Density Approximation

A widely used model for the density of the *extrinsic* information is the *symmetric* Gaussian [13], [15], [18] which is described solely by its mean m_{E_r} for the rows and m_{E_c} for the columns, the variance being twice the mean. We will use this model in the rest of this analysis. Furthermore, we assume that the *a priori* information is i.i.d., even if this is true in practice only for very large interleavers.

1) AWGN Channel: On the AWGN channel, the channel log-likelihood ratio Z is distributed as a *symmetric* Gaussian with mean $m_Z = 2/\sigma^2$, where σ is the standard deviation of the channel noise. Therefore, the input log-likelihood ratio of the column decoder $Z + A_c$ is also distributed as a *symmetric* Gaussian with mean $m_Z + m_{E_r}$. We obtain the post-decoding column BER P_b^c by applying the method described in Section III-B with

$$\boxed{f(x) = \frac{q\left(\dfrac{x-(m_Z+m_{E_r})}{\sqrt{2(m_Z+m_{E_r})}}\right)}{\sqrt{2(m_Z+m_{E_r})}}} \quad (18)$$

where $q(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2}$. Since $L_c = Z + A_c + E_c$, the density $p_{L_c}(x)$ of L_c is again Gaussian and *symmetric*. Let m_{L_c} denote its mean, it follows that

$$P_b^c = \int_{-\infty}^{0} p_{L_c}(x)dx = Q\left(\sqrt{\frac{m_{L_c}}{2}}\right) \quad (19)$$

where $Q(x) = \int_x^{+\infty} q(t)dt$. As a result

$$m_{E_c} = 2\left[Q^{-1}\left(P_b^c\right)\right]^2 - m_Z - m_{E_r}. \quad (20)$$

An analog method is used to describe the row decoder. The input log-likelihood ratio of the row decoder is $Z + A_r$ with mean $m_Z + m_{E_c}$. We obtain the post-decoding row BER P_b^r by applying the method described in Section III-B with

$$\boxed{f(x) = \frac{q\left(\dfrac{x-(m_Z+m_{E_c})}{\sqrt{2(m_Z+m_{E_c})}}\right)}{\sqrt{2(m_Z+m_{E_c})}}.} \quad (21)$$

Noting that $L_r = Z + A_r + E_r$ has a *symmetric* Gaussian distribution with mean of m_{L_r}, it follows that

$$P_b^r = Q\left(\sqrt{\frac{m_{L_r}}{2}}\right) \quad (22)$$

and finally

$$m_{E_r} = 2\left[Q^{-1}\left(P_b^r\right)\right]^2 - m_Z - m_{E_c}. \quad (23)$$

By iterating this process with the initialization $m_{E_r} = 0$, we obtain a description of the iterative decoding of product codes on the AWGN channel.

2) Rayleigh-Fading Channel: On the Rayleigh-fading channel, the distribution of the channel log-likelihood ratio Z is [17]

$$p_Z(z) = \frac{\sigma^2}{2\sqrt{1+2\sigma^2}} \exp\left(\frac{z - \sqrt{1+2\sigma^2}|z|}{2}\right). \quad (24)$$

Therefore, the distribution of the input log-likelihood ratio of the column decoder $Z + A_c$ is the convolution of $p_Z(z)$ with a *symmetric* Gaussian with mean m_{E_r}[17] shown in (25) at the top of the next page. Using $f(x)$, we obtain the column BER P_b^c by applying the method described in Section III-B. Since $L_c = Z + A_c + E_c$, the density $p_{L_c}(x)$ of L_c is also the convolution of $p_Z(z)$ with a *symmetric* Gaussian with mean $m = m_{E_r} + m_{E_c}$. It follows that [17]

$$P_b^c = \int_{-\infty}^{0} p_{L_c}(x)dx = T_\sigma(m) \quad (26)$$

where

$$T_\sigma(m) = Q\left(\sqrt{\frac{m}{2}}\right) - Q\left(\sqrt{\frac{(1+2\sigma^2)m}{2}}\right) \frac{1}{\sqrt{1+2\sigma^2}} \exp\left(\frac{\sigma^2 m}{2}\right). \quad (27)$$

As a result

$$m_{E_c} = T_\sigma^{-1}\left(P_b^c\right) - m_{E_r}. \quad (28)$$

$$\boxed{\begin{aligned} f(x) = \frac{\sigma^2}{4\sqrt{1+2\sigma^2}} \quad \exp\left(\frac{\sigma^2 m_{E_r}}{2}\right) \Bigg[& \exp\left(\frac{1+\sqrt{1+2\sigma^2}}{2}x\right) \operatorname{erfc}\left(\frac{x/\sqrt{m_{E_r}} + \sqrt{m_{E_r}(1+2\sigma^2)}}{2}\right) \\ & + \exp\left(\frac{1-\sqrt{1+2\sigma^2}}{2}x\right) \operatorname{erfc}\left(\frac{-x/\sqrt{m_{E_r}} + \sqrt{m_{E_r}(1+2\sigma^2)}}{2}\right)\Bigg]. \end{aligned}} \tag{25}$$

The row decoder is described with the same equations, but replacing P_b^c with P_b^r and inverting the roles of m_{E_r} and m_{E_c}. By iterating this process with the initialization $m_{E_r} = 0$, we obtain a description of the iterative decoding of product codes on the Rayleigh-fading channel.[1]

IV. Application: Threshold Computation

This part of the correspondence is concerned with the computation of a decoding threshold in the iterative decoding algorithm. As will be shown, the existence of the threshold can be explained in terms of the nonlinear dynamics of the 1-D model describing the iterative decoding system.

A. LDPC Codes

We study the convergence properties of the message passing algorithm by means of (6) and the 1-D map derived in Section II-B for the AWGN channel.

Theorem 4.1: Define the threshold

$$\sigma^* = \sup\left\{\sigma > 0 : \lim_{l \to +\infty} P_e^l(\sigma) = 0\right\}.$$

If $\sigma \leq \sigma^*$, $P_e^l(\sigma)$ converges to 0, otherwise $P_e^l(\sigma)$ converges to a value strictly larger than 0.

Proof: From (7) it is clear that $f(x;\sigma) \geq 0$ and $\frac{\partial f}{\partial x}(x,\sigma) > 0$, $\forall \sigma > 0$ and $\forall x \in [0, P_e^0(\sigma)]$; therefore, $P_e^l(\sigma)$ is decreasing and converges to a fixed point. It follows that if there is no fixed point in $[0, P_e^0(\sigma)]$ other than 0 hence, $P_e^l(\sigma)$ converges to 0. Conversely, assume there is a fixed point $x > 0$ in $[0, P_e^0(\sigma)]$ then $P_e^l(\sigma) \geq x, \forall l > 0$ since $f(x;\sigma)$ is increasing on $[x, P_e^0(\sigma)]$. Therefore, $P_e^l(\sigma)$ converges to a fixed point strictly greater than 0. Now, from (30) in Appendix I it is easily seen that

$$\frac{\partial f}{\partial \sigma}(x,\sigma) > 0, \qquad \forall \sigma > 0 \quad \text{and} \quad \forall x \in [0, P_e^0(\sigma)]$$

therefore, $\sigma > \sigma^*$ implies $P_e^l(\sigma) > P_e^l(\sigma^*)$, which completes the proof. □

Example 4.2: The following example illustrates the threshold effect for a $(d_v = 3, d_c = 27)$ rate $\frac{8}{9}$ regular LDPC code on the AWGN channel. The threshold found with the analysis presented in Section II-B is $\sigma^* = 0.496$. Figs. 3–5 show the map derived by (7) and the successive iterates $P_e^l(\sigma)$, along with the bisectrix line, for values of σ smaller, equal, and larger than σ^*, respectively. At $\sigma = \sigma^*$, a *tangent* (or *saddle-node*) bifurcation occurs [20]: two fixed points, one stable (*S*) the other unstable (*U*), appear (see Fig. 5).

A similar behavior can be observed for the map derived by (11) on the Rayleigh-fading channel.

TABLE I
Thresholds of Regular LDPC Codes Obtained With Density Evolution (σ_{DE}^*) and Gaussian Analysis (σ_{GA}^*) With Their Corresponding $\frac{E_b}{N_0}$

d_v	d_c	rate	σ_{DE}^*	$\left(\frac{E_b}{N_0}\right)_{DE}^*$	σ_{GA}^*	$\left(\frac{E_b}{N_0}\right)_{GA}^*$
3	6	1/2	0.880	1.11 dB	0.848	1.43 dB
3	9	2/3	0.708	1.75 dB	0.690	1.97 dB
3	12	3/4	0.632	2.22 dB	0.619	2.41 dB
3	15	4/5	0.587	2.59 dB	0.577	2.74 dB
3	18	5/6	0.557	2.86 dB	0.548	3.01 dB
3	21	6/7	0.534	3.11 dB	0.527	3.22 dB
3	24	7/8	0.517	3.30 dB	0.510	3.42 dB
3	27	8/9	0.503	3.47 dB	0.496	3.59 dB

Before proceeding to the numerical evaluation of thresholds we add the following remark on code optimization [11], [13].

Remark 4.3: The expression of $\frac{\partial f}{\partial \sigma}(x,\sigma)$ in (30) is minimal for all $x \in [0, P_e^0(\sigma)]$ when $s(x)$ is minimal for all x: this is the case when the polynomial $\rho(x)$ is concentrated on the lowest possible right degree, for a given $\lambda(x)$. Noting that $\lim_{\sigma \to 0} f(x;\sigma) = 0$ and reminding that $f(x;\sigma)$ is increasing on all $\sigma > 0$ and $\forall x \in [0, P_e^0(\sigma)]$, this means intuitively that the threshold σ^* is maximum for concentrated check degree distribution polynomials. When performing code optimization of irregular codes, once $\rho(x)$ is fixed, it is easy to find a suitable $\lambda(x)$ so as to maximize the threshold. This is consistent with the fact mentioned in Section II-B that for good LDPC codes, the polynomial $\rho(x)$ is concentrated and with the fact that the performance of a single parity-check codes is increased if the number of information bits is reduced. Similar results for the binary-symmetric channel (BSC) and the binary erasure channel (BEC) have been presented in [23] and [24], respectively.

1) Numerical Results: We conclude this subsection by comparing the threshold values σ^* and their corresponding ratios $\left(\frac{E_b}{N_0}\right)^*$ obtained with our analysis, versus density evolution. In Table I, we give the thresholds for rate $R = \frac{m}{m+1}$ regular LDPC codes with $d_v = 3$, for $m = 1, \ldots, 8$. It can be seen that the model proposed in Section II-B estimates the E_b/N_0 threshold with accuracy between 0.1 and 0.3 dB. The accuracy obtained by the authors in [13] is better by approximately one order of magnitude; however, our closed-form analytical model provides more insight into the decoder dynamics.

B. Product Codes

In order to analyze a 1-D system, we choose to study the iterates of the BER at the output of the row decoder $P_b^r(l)$ as a function of the iteration index l and the noise parameter σ. $P_b^r(0)$ is arbitrarily set to the channel crossover probability. It can be verified that 0 is a fixed point of the iterative decoding model described in Section III-B. As for LDPC codes, a threshold

$$\sigma^* = \sup\left\{\sigma > 0 : \lim_{l \to +\infty} P_b^r(l) = 0\right\}$$

exists.

[1] Note that at the first iteration the distribution of the input log-likelihood ratio of the row decoder is the convolution of $p_Z(z)$ with a Dirac function, as initially $m_{E_r} = 0$.

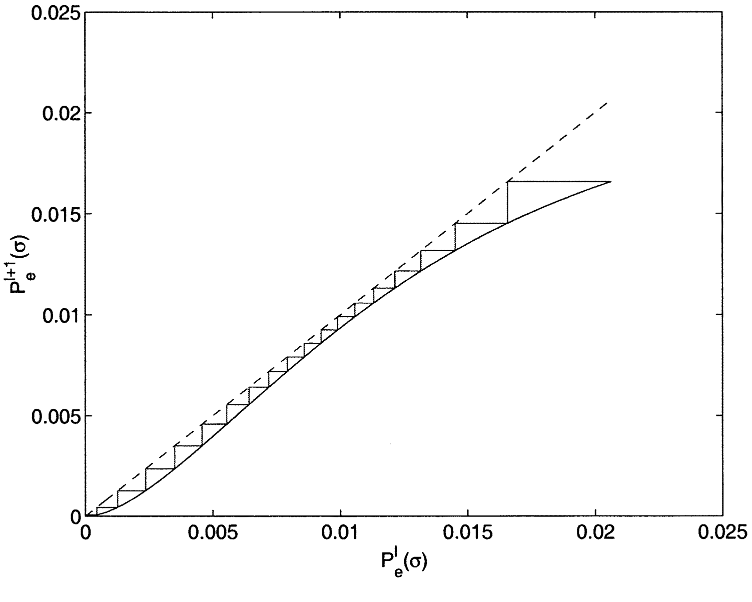

Fig. 3. $P_e^{l+1}(\sigma)$ as a function of $P_e^l(\sigma)$ at $\sigma = 0.49$ for the $(d_v = 3, d_c = 27)$ regular LDPC code.

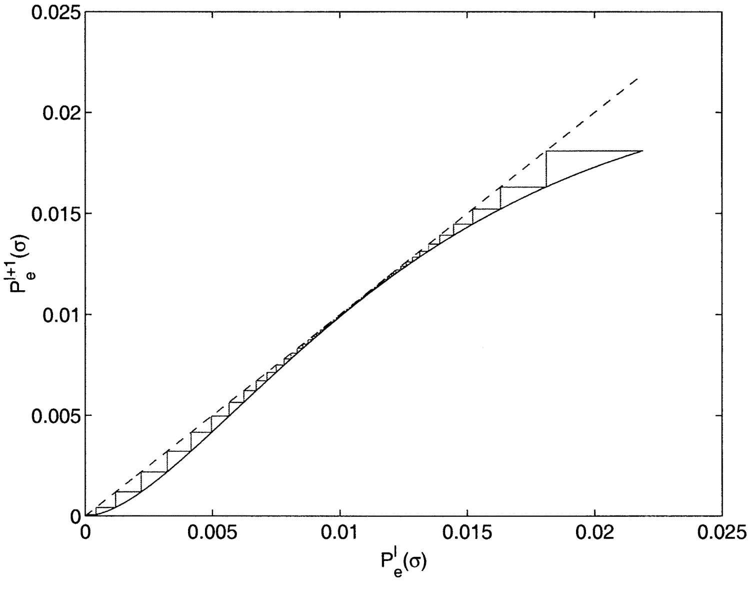

Fig. 4. $P_e^{l+1}(\sigma)$ as a function of $P_e^l(\sigma)$ at the threshold value $\sigma^* = 0.496$ for the $(d_v = 3, d_c = 27)$ regular LDPC code.

Example 4.4: The following example illustrates the threshold effect for the BCH$(64, 51, 6)^2$ product code. The threshold found with the analysis presented in Section III-B is $\sigma^* = 0.745$. Fig. 6 shows the successive iterates of P_b^r, along with the bisectrix line, for $\sigma = \sigma^*$. The decoding trajectory starts in the upper right corner and ends at the origin. At $\sigma = \sigma^*$, the decoding trajectory enters a *bottleneck or tunnel region* [15], [17] near the bisectrix line with a characteristic slowing down of the convergence rate.

In general, we observed that for $C(n, k, d)^2$ product codes where n and k are fixed, the threshold σ^* is an increasing function of d. This seems to indicate that well-known constituent codes with high minimum distance such as the Bose-Chaudhuri–Hocquenghem (BCH) codes are suitable to obtain product codes with a good threshold under iterative decoding.

1) Numerical Results: Tables II and III give the thresholds calculated with the method presented in Section III-B for the product codes simulated in [8] on the AWGN and Rayleigh channels, respectively. We emphasize that although it is not possible to define rigorously a threshold for codes with finite block lengths, the threshold existing in

TABLE II
THRESHOLDS OF PRODUCT BCH CODES σ^* WITH THE CORRESPONDING $\frac{E_b}{N_0}$ FOR THE AWGN CHANNEL

Code	Rate	σ^*	$\left(\frac{E_b}{N_0}\right)^*_{dB}$
$(32, 21, 6)^2$	0.431	1.106	-0.2
$(32, 26, 4)^2$	0.660	0.803	0.7
$(64, 51, 6)^2$	0.635	0.745	1.5
$(64, 57, 4)^2$	0.793	0.615	2.2
$(128, 113, 6)^2$	0.779	0.588	2.7
$(128, 120, 4)^2$	0.879	0.513	3.3
$(256, 247, 4)^2$	0.931	0.447	4.3
$(512, 502, 4)^2$	0.961	0.401	5.1

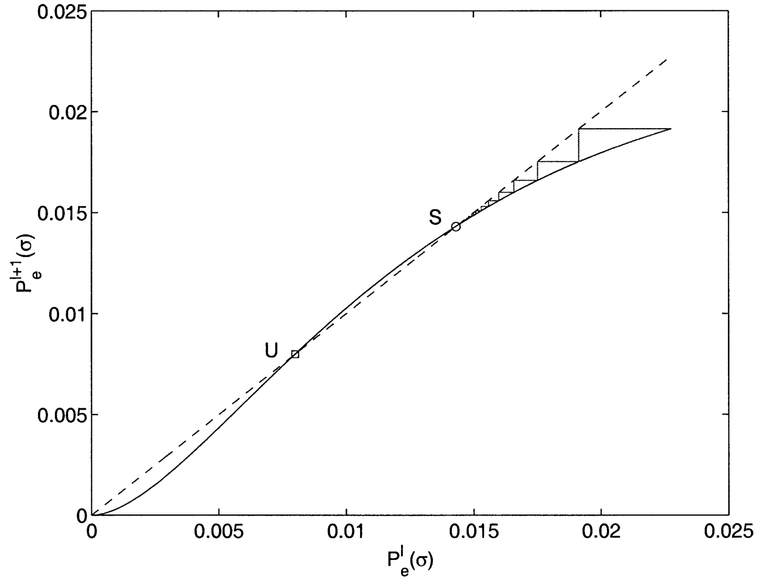

Fig. 5. $P_e^{l+1}(\sigma)$ as a function of $P_e^l(\sigma)$ at $\sigma = 0.50$ for the $(d_v = 3, d_c = 27)$ regular LDPC code. Note the appearance of a stable (*S*) and unstable (*U*) fixed points due to the occurrence of a tangent bifurcation.

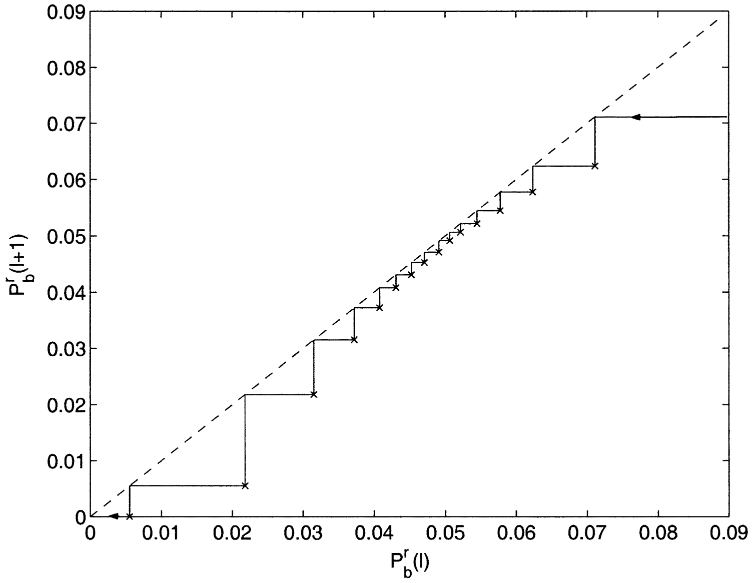

Fig. 6. $P_b^r(l+1)$ as a function of $P_b^r(l)$ at $\sigma = 0.745$ for the BCH$(64, 51, 6)^2$ product code on the AWGN channel.

TABLE III
THRESHOLDS OF PRODUCT BCH CODES σ^* WITH THE CORRESPONDING $\frac{E_b}{N_0}$ FOR THE RAYLEIGH-FADING CHANNEL

Code	Rate	σ^*	$\left(\frac{E_b}{N_0}\right)^*_{dB}$
$(32, 21, 6)^2$	0.431	0.896	1.6
$(32, 26, 4)^2$	0.660	0.580	3.5
$(64, 51, 6)^2$	0.635	0.508	4.8
$(64, 57, 4)^2$	0.793	0.366	6.7
$(128, 113, 6)^2$	0.779	0.330	7.7
$(128, 120, 4)^2$	0.879	0.246	9.7
$(256, 247, 4)^2$	0.931	0.170	12.7
$(512, 502, 4)^2$	0.961	0.118	15.7

our model is a good indicator of the beginning of the *waterfall region* of the BER curves, except when the block length of the constituent codes is 32. In this particular case, the calculated thresholds are even below the capacity. This confirms that the hypothesis of i.i.d. extrinsic information is approximately justified only for large block lengths.

V. CONCLUSION

In this correspondence, we have presented 1-D models for the iterative decoding of LDPC and product codes, based on Gaussian densities. These simple 1-D maps describe the evolution of the error probabilities, as a function of the number of iterations, on both the AWGN and the Rayleigh-fading channel. For LDPC codes, our analysis leads to a stability condition which is consistent with the density evolution method.

Our approach allows a qualitative analysis of the nonlinear dynamics of the decoding algorithm near the threshold. Also, we have verified that the thresholds obtained with our approximate models are in good agreement with the values computed through density evolution or Monte Carlo simulations.

APPENDIX I

The partial derivatives of the map given by (7) admit a closed-form expression. Namely, it is easily seen that the partial derivative with respect to x can be expressed as

$$\begin{aligned}\frac{\partial f}{\partial x}(x,\sigma) &= e^{-\frac{1}{2\sigma^2}}\sum_{j=2}^{d_c}\rho_j(j-1)(1-2x)^{(j-2)}\\ &\quad\times\sum_{i=2}^{d_v}\lambda_i\frac{(i-1)Q^{-1}(s(x))}{\sqrt{\frac{1}{\sigma^2}+(i-1)\left\{Q^{-1}(s(x))\right\}^2}}\\ &\quad\times\exp\left[\left(1-\frac{i}{2}\right)\left\{Q^{-1}(s(x))\right\}^2\right] \qquad (29)\end{aligned}$$

and, if we take the limit at $x=0$, it can be verified that

$$\lim_{x\to 0}\frac{\partial f}{\partial x}(x,\sigma) = e^{-\frac{1}{2\sigma^2}}\lambda_2\sum_{j=2}^{d_c}\rho_j(j-1) = e^{-\frac{1}{2\sigma^2}}\lambda'(0)\rho'(1).$$

Similarly, the partial derivative with respect to σ can be expressed as

$$\begin{aligned}\frac{\partial f}{\partial \sigma}(x,\sigma) &= \frac{e^{-\frac{1}{2\sigma^2}}}{\sigma^3\sqrt{2\pi}}\sum_{i=2}^{d_v}\lambda_i\frac{1}{\sqrt{\frac{1}{\sigma^2}+(i-1)\left\{Q^{-1}(s(x))\right\}^2}}\\ &\quad\times\exp\left[\frac{1}{2}(1-i)\left\{Q^{-1}(s(x))\right\}^2\right]. \qquad (30)\end{aligned}$$

APPENDIX II

The partial derivative of the map given by (11) is given by

$$\begin{aligned}\frac{\partial f(x,\sigma)}{\partial x} &= \frac{2\sigma^2}{\sqrt{1+2\sigma^2}}\sum_{j=2}^{d_c}\rho_j(j-1)(1-2x)^{j-2}\\ &\quad\times\sum_{i=2}^{d_v}\lambda_i\Bigg\{\sqrt{2\pi}\exp\left(\frac{1}{2}(1+2\sigma^2)(i-1)[Q^{-1}(s(x))]^2\right)\\ &\quad\times Q\left(\sqrt{(1+2\sigma^2)(i-1)[Q^{-1}(s(x))]^2}\right)\\ &\quad\times(i-1)Q^{-1}(s(x))\exp\left(\left(1-\frac{i}{2}\right)[Q^{-1}(s(x))]^2\right)\Bigg\}. \qquad (31)\end{aligned}$$

Noting that $Q^{-1}(s(x))\to+\infty$ for $x\to 0$ and using the fact that

$$Q(\sqrt{y})\to\frac{1}{\sqrt{2\pi}}\frac{e^{-\frac{y}{2}}}{\sqrt{y}},\qquad \text{for } y\to+\infty$$

it follows that for $x\to 0$

$$\begin{aligned}\frac{\partial f(x,\sigma)}{\partial x} &\to \left(\sum_{j=2}^{d_c}\rho_j(j-1)\right)\frac{2\sigma^2}{\sqrt{1+2\sigma^2}}\\ &\quad\times\sum_{i=2}^{d_v}\lambda_i\frac{(i-1)Q^{-1}(s(x))}{\sqrt{(1+2\sigma^2)(i-1)[Q^{-1}(s(x))]^2}}\\ &\quad\times\exp\left(\left(1-\frac{i}{2}\right)[Q^{-1}(s(x))]^2\right)\end{aligned}$$

and

$$\begin{aligned}\frac{\partial f(x,\sigma)}{\partial x} &\to \left(\sum_{j=2}^{d_c}\rho_j(j-1)\right)\frac{1}{1+\frac{1}{2\sigma^2}}\\ &\quad\times\sum_{i=2}^{d_v}\lambda_i\sqrt{i-1}\exp\left(\left(1-\frac{i}{2}\right)[Q^{-1}(s(x))]^2\right).\end{aligned}$$

Finally

$$\begin{aligned}\lim_{x\to 0}\frac{\partial f}{\partial x}(x,\sigma) &= \frac{1}{1+\frac{1}{2\sigma^2}}\lambda_2\sum_{j=2}^{d_c}\rho_j(j-1)\\ &= \frac{1}{1+\frac{1}{2\sigma^2}}\lambda'(0)\rho'(1). \qquad (32)\end{aligned}$$

ACKNOWLEDGMENT

The authors would like to thank Dr. Ljupco Kocarev (UCSD), Prof. Ruediger Urbanke (EPFL), and Henry Pfister (UCSD) for the stimulating discussions.

REFERENCES

[1] C. E. Shannon, “A mathematical theory of communication,” *Bell Syst. Tech. J.*, vol. 27, pp. 379–423, 1948.

[2] R. G. Gallager, “Low density parity check codes,” *IRE Trans. Inform. Theory*, vol. IT-8, pp. 21–28, Jan. 1962.

[3] D. J. C. MacKay and R. M. Neal, “Near shannon limit performance of low density parity check codes,” *Electron. Lett.*, vol. 32, pp. 1645–1646, 1996.

[4] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, “Factor graphs and the sum-product algorithm,” *IEEE Trans. Inform. Theory*, vol. 47, pp. 498–519, Feb. 2001.

[5] P. Elias, “Error-free coding,” *IRE Trans. Inform. Theory*, vol. PGIT-4, pp. 29–37, Sept. 1954.

[6] C. Berrou, A. Glavieux, and P. Thitimajshima, “Near shannon limit error-correcting coding and decoding: Turbo-codes,” in *Proc. IEEE Int. Conf. Communications*, 1993, pp. 1064–70.

[7] R. Pyndiah, A. Glavieux, A. Picart, and S. Jacq, “Near optimum decoding of product codes,” in *Proc. GLOBECOM’94*, vol. 1, San Francisco, CA, Nov. 1994, pp. 339–343.

[8] R. M. Pyndiah, “Near optimum decoding of product codes: Block turbo codes,” *IEEE Trans. Commun.*, vol. 46, pp. 1003–1010, Aug. 1998.

[9] D. Chase, “A class of algorithms for decoding block codes with channel measurement information,” *IEEE Trans. Inform. Theory*, vol. IT-18, pp. 170–182, Jan. 1972.

[10] T. J. Richardson and R. L. 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] T. J. Richardson, M. A. Shokrollahi, and R. L. Urbanke, “Design of capacity-approaching irregular low-density parity-check codes,” *IEEE Trans. Inform. Theory*, vol. 47, pp. 619–637, Feb. 2001.

[12] N. Wiberg, “Codes and decoding on general graphs,” Ph.D. dissertation, Linköping Univ., Linköping, Sweden, 1996.

[13] S.-Y. Chung, T. J. Richardson, and R. L. Urbanke, “Analysis of sum-product decoding of low-density parity-check codes using a Gaussian approximation,” *IEEE Trans. Inform. Theory*, vol. 47, pp. 657–670, Feb. 2001.

[14] H. E. Gammal and A. R. Hammons, Jr., “Analysis the turbo decoder using the Gaussian approximation,” *IEEE Trans. Inform. Theory*, vol. 47, pp. 671–686, Feb. 2001.

[15] D. Divsalar, S. Dolinar, and F. Pollara, “Iterative turbo decoder analysis based on density evolution,” *J. Select. Areas Commun.*, vol. 19, pp. 891–907, May 2001.

[16] S. ten Brink, “Convergence of iterative decoding,” *Electron. Lett.*, vol. 35, no. 10, pp. 806–808, May 1999.

[17] ——, “Convergence behavior of iteratively decoded parallel concatenated codes,” *IEEE Trans. Commun.*, vol. 49, pp. 1727–1737, Oct. 2001.

[18] F. Lehmann and G. M. Maggio, “An approximate analytical model of the message passing decoder of LDPC codes,” in *Proc. 2002 IEEE Int. Symp. Information Theory*, Lausanne, Switzerland, June/July 2002, p. 31.

[19] P. Rusmevichientong and B. Van Roy, “An analysis of belief propagation on the turbo decoding graph with Gaussian densities,” *IEEE Trans. Inform. Theory*, vol. 47, pp. 745–765, Feb. 2001.

[20] S. H. Strogatz, *Nonlinear Dynamics and Chaos*. Cambridge, MA: Perseus, 1994.

[21] J. Hou, P. H. Siegel, and L. B. Milstein, “Performance analysis and code optimization of low density parity-check codes on Rayleigh fading channels,” *IEEE J. Select. Areas Commun.*, vol. 19, pp. 924–934, May 2001.

[22] M. P. C. Fossorier and S. Lin, “Error performance analysis for reliability-based decoding algorithms,” *IEEE Trans. Inform. Theory*, vol. 48, pp. 287–293, Jan. 2002.

[23] L. Bazzi, T. Richardson, and R. Urbanke, “Exact thresholds and optimal codes for the binary symmetric channel and Gallager’s decoding algorithm A,” in *Proc 2000 IEEE Int. Symp. Information Theory*, Sorrento, Italy, June 2000.

[24] M. A. Shokrollahi, “New sequences of linear time erasure codes approaching the channel capacity,” in *Proc. AAECC’99: 13th AAECC Symp. Applied Algebra, Algebraic Algorithms, and Error-Correcting Codes*, Honolulu, HI, Nov 1999, pp. 65–76.