EURASIP Journal on Applied Signal Processing 2005:6, 762–774
© 2005 Phillip A. Regalia

Iterative Decoding of Concatenated Codes: A Tutorial

Phillip A. Regalia

Département Communications, Images et Traitement de l' Information, Institut National des Télécommunications,
91011 Evry Cedex, France

Department of Electrical Engineering and Computer Science, Catholic University of America, Washington, DC 20064, USA
Email: phillip.regalia@int-evry.fr

Received 29 September 2003; Revised 1 June 2004

The turbo decoding algorithm of a decade ago constituted a milestone in error-correction coding for digital communications, and has inspired extensions to generalized receiver topologies, including turbo equalization, turbo synchronization, and turbo CDMA, among others. Despite an accrued understanding of iterative decoding over the years, the "turbo principle" remains elusive to master analytically, thereby inciting interest from researchers outside the communications domain. In this spirit, we develop a tutorial presentation of iterative decoding for parallel and serial concatenated codes, in terms hopefully accessible to a broader audience. We motivate iterative decoding as a computationally tractable attempt to approach maximum-likelihood decoding, and characterize fixed points in terms of a "consensus" property between constituent decoders. We review how the decoding algorithm for both parallel and serial concatenated codes coincides with an alternating projection algorithm, which allows one to identify conditions under which the algorithm indeed converges to a maximum-likelihood solution, in terms of particular likelihood functions factoring into the product of their marginals. The presentation emphasizes a common framework applicable to both parallel and serial concatenated codes.

Keywords and phrases: iterative decoding, maximum-likelihood decoding, information geometry, belief propagation.

1. INTRODUCTION

The advent of the turbo decoding algorithm for parallel concatenated codes a decade ago [1] ranks among the most significant breakthroughs in modern communications in the past half century: a coding and decoding procedure of reasonable computational complexity was finally at hand offering performance approaching the previously elusive Shannon limit, which predicts reliable communications for all channel capacity rates slightly in excess of the source entropy rate. The practical success of the iterative turbo decoding algorithm has inspired its adaptation to other code classes, notably serially concatenated codes [2, 3], and has rekindled interest [4, 5] in low-density parity-check codes [6], which give the definitive historical precedent in iterative decoding.

The serial concatenated configuration holds particular interest for communication systems, since the "inner encoder" of such a configuration can be given more general interpretations, such as a "parasitic" encoder induced by a convolutional channel or by the spreading codes used in CDMA. The corresponding iterative decoding algorithm can then be extended into new arenas, giving rise to turbo equalization [7, 8, 9] or turbo CDMA [10, 11], among doubtless other possibilities. Such applications demonstrate the power of iterative techniques which aim to jointly optimize receiver components, compared to the traditional approach of adapting such components independently of one another.

The turbo decoding algorithm for error-correction codes is known not to converge, in general, to a maximum-likelihood solution, although in practice it is usually observed to give comparable performance [12, 13, 14]. The quest to understand the convergence behavior has spawned numerous inroads, including extrinsic information transfer (or EXIT) charts [15], density evolution of intermediate quantities [16, 17], phase trajectory techniques [18], Gaussian approximations which simplify the analysis [19], and cross-entropy minimization [20], to name a few. Some of these analysis techniques have been applied with success to other configurations, such as turbo equalization [21, 22]. Connections to the belief propagation algorithm [23] have also been identified [24], which approach in turn is closely linked to earlier work on graph theoretic methods [25, 26, 27, 28]. In this context, the turbo decoding algorithm gives rise to a directed graph having cycles; the belief propagation algorithm is known to converge provided no cycles appear in the directed graph, although less can be said in general once cycles appear.

Interest in turbo decoding and related topics now extends beyond the communications community, and has been met with useful insights from other fields; some references in this direction include [29] which draws on nonlinear system analysis, [30] which draws on computer science, in addition to [31] (predating turbo codes) and [32] (more recent) which inject ideas from statistical physics, which in turn can be rephrased in terms of information geometry [33, 34]. Despite this impressive pedigree of analysis techniques, the "turbo principle" remains difficult to master analytically and, given its fair share of specialized terminology if not a certain degree of mystique, is often perceived as difficult to grasp to the nonspecialist. In this spirit, the aim of this paper is to provide a reasonably self-contained and tutorial development of iterative decoding for parallel and serial concatenated codes, in terms hopefully accessible to a broader audience. The paper does not aim at a comprehensive survey of available analysis techniques and implementation tricks surrounding iterative decoding (for which the texts [12, 13, 14] would be more appropriate), but rather chooses a particular vantage point which steers clear of unnecessary sophistication and avoids approximations.

We begin in Section 2 by reviewing optimum (maximum a posteriori and maximum-likelihood) decoding of parallel concatenated codes. We motivate the turbo decoding algorithm as a computationally tractable attempt to approach maximum-likelihood decoding. A characterization of fixed points is obtained in terms of a "consensus" property between the two constituent decoders, and a simple proof of the existence of fixed points is obtained as an application of the Brouwer fixed point theorem.

Section 3 then reexamines the calculation of marginal distributions in terms of a projection operator, leading to a compact formulation of the turbo decoding algorithm as an alternating projection algorithm. The material of the section aims at a concrete transcription of ideas originally developed by Richardson [29]; we include in addition a minimum-distance property of the projector in terms of the Kullback-Leibler divergence, and review how the turbo decoding algorithm indeed converges to a maximum-likelihood solution whenever specific likelihood functions factor into the product of its marginals. The factorization is known [18] to hold in extreme signal-to-noise ratios.

Section 4 shows that the iterative decoding algorithm for serial concatenated codes also admits an alternating projection interpretation, allowing us to transcribe all results for parallel concatenated codes to their serial concatenated counterparts. This should also facilitate unified studies of both code classes. Concluding remarks are summarized in Section 5.

2. TURBO DECODING OF PARALLEL CONCATENATED CODES

We begin by reviewing the classical turbo decoding algorithm for parallel concatenated codes. For simplicity, we restrict our development to the binary signaling case; the m-ary case can be handled by direct extension (see, e.g., [24] for a particularly clear treatment) or by mapping the m-ary constellation back to its binary origins.

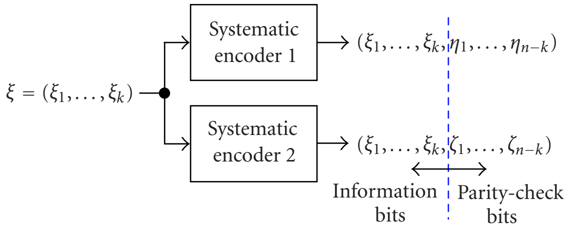
$\xi = (\xi_1, \dots, \xi_k)$
Systematic encoder 1
$(\xi_1, \dots, \xi_k, \eta_1, \dots, \eta_{n-k})$
Systematic encoder 2
$(\xi_1, \dots, \xi_k, \zeta_1, \dots, \zeta_{n-k})$
Information bits
Parity-check bits

Figure 1: Parallel concatenated encoder structure.

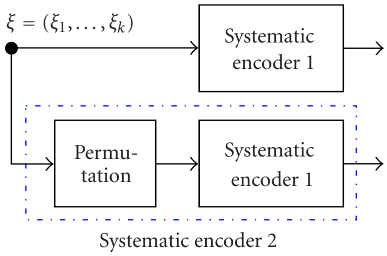

Figure 2: Particular realization of the second encoder by using the first encoder with an interleaver.

To begin, a binary (0 or 1) information block $\xi = (\xi_1, \dots, \xi_k)$ is passed through two constituent encoders, as in Figure 1, to create two codewords:

$$(\xi_1, \dots, \xi_k, \eta_1, \dots, \eta_{n-k}), \qquad (\xi_1, \dots, \xi_k, \zeta_1, \dots, \zeta_{n-k}). \quad (1)$$

Both encoders are systematic and of rate k/n, so that the information bits $\xi_1, \dots, \xi_k$ are directly available in either codeword. Note also that the two encoders need not share a common rate, although we will adhere to this case for ease of notation.

In practice, an expedient method of realizing the second systematic encoder is to permute (or interleave) the information bits ξ_i and duplicate the first encoder, as in Figure 2. Since this is a particular instance of Figure 1, we will simply consider two separate encodings of $\xi = (\xi_1, \dots, \xi_k)$ in what follows and avoid explicit reference to the interleaving operation, despite its importance in the study of the distance properties of concatenated codes [35].

The encoder outputs are converted to antipodal signaling (±1) and transmitted over a channel containing additive noise, giving the received signals x_i, y_i, and z_i:

$$\begin{aligned} x_i &= (2\xi_i - 1) + b_{x,i}, \quad i = 1, 2, \dots, k; \\ y_i &= (2\eta_i - 1) + b_{y,i}, \quad i = 1, 2, \dots, n-k; \\ z_i &= (2\zeta_i - 1) + b_{z,i}, \quad i = 1, 2, \dots, n-k. \end{aligned} \quad (2)$$

We assume that the noise samples $b_{x,i}$, $b_{y,i}$, and $b_{z,i}$ are Gaussian and mutually independent, sharing a common variance σ^2. For notational convenience, we arrange the received signals into the vectors

$$\mathbf{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_k \end{bmatrix}, \qquad \mathbf{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_{n-k} \end{bmatrix}, \qquad \mathbf{z} = \begin{bmatrix} z_1 \\ \vdots \\ z_{n-k} \end{bmatrix}. \tag{3}$$

2.1. *Optimum decoding*

The maximum a posteriori decoding rule aims to calculate the a posteriori probability ratios

$$\frac{\Pr(\xi_i = 1|\mathbf{x},\mathbf{y},\mathbf{z})}{\Pr(\xi_i = 0|\mathbf{x},\mathbf{y},\mathbf{z})}, \quad i = 1,2,\ldots,k, \tag{4}$$

with the decision rule favoring a 1 for the ith bit if this ratio is greater than one, and 0 if the ratio is less than one. By using Bayes's rule, each ratio can be developed as

$$\begin{aligned}\frac{\Pr(\xi_i = 1|\mathbf{x},\mathbf{y},\mathbf{z})}{\Pr(\xi_i = 0|\mathbf{x},\mathbf{y},\mathbf{z})} &= \frac{\sum_{\xi:\xi_i=1}\Pr(\xi|\mathbf{x},\mathbf{y},\mathbf{z})}{\sum_{\xi:\xi_i=0}\Pr(\xi|\mathbf{x},\mathbf{y},\mathbf{z})} \\ &= \frac{\sum_{\xi:\xi_i=1} p(\mathbf{x},\mathbf{y},\mathbf{z}|\xi)\Pr(\xi)}{\sum_{\xi:\xi_i=0} p(\mathbf{x},\mathbf{y},\mathbf{z}|\xi)\Pr(\xi)},\end{aligned} \tag{5}$$

involving the a priori probability mass function $\Pr(\xi)$ and the likelihood function $p(\mathbf{x},\mathbf{y},\mathbf{z}|\xi)$, which is evaluated for the received $\mathbf{x}$, $\mathbf{y}$, and $\mathbf{z}$ as a function of the candidate information bits $\xi = (\xi_1,\ldots,\xi_k)$; the sum in the numerator (resp., denominator) is over all the configurations of the vector ξ for which the ith bit is a "1" (resp., "0"). Since the noise samples are assumed independent, the likelihood function naturally factors as

$$p(\mathbf{x},\mathbf{y},\mathbf{z}|\xi) = p(\mathbf{x}|\xi)p(\mathbf{y}|\xi)p(\mathbf{z}|\xi). \tag{6}$$

For the Gaussian noise case considered here, the three likelihood evaluations appear as

$$\begin{aligned} p(\mathbf{x}|\xi) &\sim \exp\left(-\frac{\|\mathbf{x}-\mathbf{c}_x(\xi)\|^2}{2\sigma^2}\right), \\ p(\mathbf{y}|\xi) &\sim \exp\left(-\frac{\|\mathbf{y}-\mathbf{c}_y(\xi)\|^2}{2\sigma^2}\right), \\ p(\mathbf{z}|\xi) &\sim \exp\left(-\frac{\|\mathbf{z}-\mathbf{c}_z(\xi)\|^2}{2\sigma^2}\right), \end{aligned} \tag{7}$$

where $\mathbf{c}_x(\xi)$, $\mathbf{c}_y(\xi)$, and $\mathbf{c}_z(\xi)$ contain the antipodal symbols ± 1 which would be received as a function of the candidate information bits ξ, in the absence of noise. For non-Gaussian noise, the likelihood functions would, of course, assume different forms.

The a posteriori probability ratios may therefore be written as

$$\frac{\Pr(\xi_i = 1|\mathbf{x},\mathbf{y},\mathbf{z})}{\Pr(\xi_i = 0|\mathbf{x},\mathbf{y},\mathbf{z})} = \frac{\sum_{\xi:\xi_i=1} p(\mathbf{x}|\xi)p(\mathbf{y}|\xi)p(\mathbf{z}|\xi)\Pr(\xi)}{\sum_{\xi:\xi_i=0} p(\mathbf{x}|\xi)p(\mathbf{y}|\xi)p(\mathbf{z}|\xi)\Pr(\xi)}, \quad i = 1,2,\ldots,k. \tag{8}$$

If the a priori probability mass function $\Pr(\xi)$ is uniform (i.e., $\Pr(\xi) = 1/2^k$ for all ξ), then this reduces to the maximum-likelihood decision metric:

$$\begin{aligned}&\frac{\Pr(\xi_i = 1|\mathbf{x},\mathbf{y},\mathbf{z})}{\Pr(\xi_i = 0|\mathbf{x},\mathbf{y},\mathbf{z})} \\ &\quad = \frac{\sum_{\xi:\xi_i=1} p(\mathbf{x}|\xi)p(\mathbf{y}|\xi)p(\mathbf{z}|\xi)}{\sum_{\xi:\xi_i=0} p(\mathbf{x}|\xi)p(\mathbf{y}|\xi)p(\mathbf{z}|\xi)} \quad \text{if } \Pr(\xi) \text{ is uniform.}\end{aligned} \tag{9}$$

If this expression were evaluated as written, the complexity of an optimum decision rule would be $\mathcal{O}(2^k)$, since there are 2^k configurations of the k information bits comprising ξ, leading to as many likelihood function evaluations. This clearly becomes impractical for sizable k.

Observe now that if we instead consider an optimum decoding rule using only one of the constituent encoders, we may write, by a development parallel to that above,

$$\frac{\Pr(\xi_i = 1|\mathbf{x},\mathbf{y})}{\Pr(\xi_i = 0|\mathbf{x},\mathbf{y})} = \frac{\sum_{\xi:\xi_i=1} p(\mathbf{x}|\xi)p(\mathbf{y}|\xi)\Pr(\xi)}{\sum_{\xi:\xi_i=0} p(\mathbf{x}|\xi)p(\mathbf{y}|\xi)\Pr(\xi)}, \tag{10}$$

$$\frac{\Pr(\xi_i = 1|\mathbf{x},\mathbf{z})}{\Pr(\xi_i = 0|\mathbf{x},\mathbf{z})} = \frac{\sum_{\xi:\xi_i=1} p(\mathbf{x}|\xi)p(\mathbf{z}|\xi)\Pr(\xi)}{\sum_{\xi:\xi_i=0} p(\mathbf{x}|\xi)p(\mathbf{z}|\xi)\Pr(\xi)}. \tag{11}$$

If each constituent encoder implements a trellis code, then $\mathbf{x}$ and $\mathbf{y}$ form a Markov chain, as do $\mathbf{x}$ and $\mathbf{z}$; the complexity of either decoding expression can then be reduced to $\mathcal{O}(k)$ by using the forward-backward algorithm from [36] (which, in turn, is a particular case of the sum-product algorithm [27]).

If the a priori probability function $\Pr(\xi)$ is indeed uniform, then it weighs all terms in the numerator and denominator equally and, as such, is effectively relegated to an unused variable in either decoding expression (10) or (11). Rather than accepting this status, one can imagine replacing the a priori probability function $\Pr(\xi)$, or "usurping" its position, by some other function in an attempt to "bias" either decoding rule (10) or (11) towards the maximum-likelihood decoding rule in (9). In particular, if $\Pr(\xi)$ were replaced by $p(\mathbf{z}|\xi)$ in (10), or by $p(\mathbf{y}|\xi)$ in (11), then either expression would agree formally with (9).

In order to retain the $\mathcal{O}(k)$ complexity of the forward-backward algorithm from [36], however, the a priori probability function $\Pr(\xi)$ is assumed to factor into the product of its bitwise marginals:

$$\Pr(\xi) = \Pr(\xi_1)\Pr(\xi_2)\cdots\Pr(\xi_k). \tag{12}$$

The likelihood function $p(\mathbf{y}|\xi)$ or $p(\mathbf{z}|\xi)$ does not, on the other hand, generally factor into its bitwise marginals, that is,

$$p(\mathbf{y}|\xi) \neq p(\mathbf{y}|\xi_1)p(\mathbf{y}|\xi_2)\cdots p(\mathbf{y}|\xi_k). \tag{13}$$

As such, a direct usurpation of the a priori probability by the likelihood function of the parity-check bits of the other constituent coder is not feasible. Rather, one must approximate the likelihood function $p(\mathbf{y}|\xi)$ or $p(\mathbf{z}|\xi)$ by a function that does factor into the product of its marginals. Many candidate approximations may be envisaged; that which has proved the most useful relies on extrinsic information values, which are reviewed next.

2.2. Extrinsic information values

We reexamine the likelihood function for the systematic bits:

$$\begin{aligned} p(\mathbf{x}|\xi) &= \frac{1}{(\sqrt{2\pi}\sigma)^k}\exp\left(-\sum_{i=1}^{k}\frac{[x_i-(2\xi_i-1)]^2}{2\sigma^2}\right) \\ &= \prod_{i=1}^{k}\frac{\exp\left(-[x_i-(2\xi_i-1)]^2/2\sigma^2\right)}{\sqrt{2\pi}\sigma} \\ &= p(x_1|\xi_1)p(x_2|\xi_2)\cdots p(x_k|\xi_k). \end{aligned} \tag{14}$$

This shows that the likelihood function $p(\mathbf{x}|\xi)$ for the systematic bits factors into the product of its marginals,[1] just like the a priori probability mass function:

$$\Pr(\xi) = \Pr(\xi_1)\Pr(\xi_2)\cdots\Pr(\xi_k). \tag{15}$$

Owing to these factorizations, each term from the numerator of (10) contains a factor $p(x_i|\xi_i = 1)\Pr(\xi_i = 1)$, and each term from the denominator contains a factor $p(x_i|\xi_i = 0)\Pr(\xi_i = 0)$. By isolating these common factors, we may rewrite the ratio from (10) as

$$\begin{aligned} &\frac{\Pr(\xi_i = 1|\mathbf{x},\mathbf{y})}{\Pr(\xi_i = 0|\mathbf{x},\mathbf{y})} \\ &\quad = \underbrace{\frac{p(x_i|\xi_i = 1)}{p(x_i|\xi_i = 0)}}_{\substack{\text{intrinsic}\\ \text{information}}}\underbrace{\frac{\Pr(\xi_i = 1)}{\Pr(\xi_i = 0)}}_{\substack{\text{a priori}\\ \text{information}}} \\ &\quad\quad \times \underbrace{\frac{\sum_{\xi:\xi_i=1}p(\mathbf{y}|\xi)\prod_{j\neq i}p(x_j|\xi_j)\Pr(\xi_j)}{\sum_{\xi:\xi_i=0}p(\mathbf{y}|\xi)\prod_{j\neq i}p(x_j|\xi_j)\Pr(\xi_j)}}_{\text{extrinsic information}}. \end{aligned} \tag{16}$$

The three terms on the right-hand side may be interpreted as follows:

(i) the first term indicates what the ith received bit x_i contributes to the determination of the ith transmitted bit ξ_i; hence the name "intrinsic information." It coincides with the maximum-likelihood metric for determining the ith bit when no coding is used,

(ii) the second term expresses the a priori probability ratio for the ith bit, and will be usurped shortly,

(iii) the third term expresses what the remaining bits in the packet (i.e., of index $j \neq i$) contribute to the determination of the ith bit; hence the name "extrinsic information."

Let $T(\xi) = T_1(\xi_1)T_2(\xi_2)\cdots T_k(\xi_k)$ be a factorable probability mass function whose bitwise ratios are chosen to match the extrinsic information values above:

$$\begin{aligned} &\frac{T_i(\xi_i = 1)}{T_i(\xi_i = 0)} \\ &\quad = \frac{\sum_{\xi:\xi_i=1}p(\mathbf{y}|\xi)\prod_{j\neq i}p(x_j|\xi_j)\Pr(\xi_j)}{\sum_{\xi:\xi_i=0}p(\mathbf{y}|\xi)\prod_{j\neq i}p(x_j|\xi_j)\Pr(\xi_j)}, \quad i = 1,2,\ldots,k. \end{aligned} \tag{17}$$

Since these values depend on the likelihood function $p(\mathbf{y}|\xi)$ (in addition to the systematic bits save for x_i), we may consider $T(\xi)$ a factorable function which approximates, in some sense, the likelihood function $p(\mathbf{y}|\xi)$. (We will see in Theorem 2 a condition under which this approximation becomes exact). We now let $T(\xi)$ usurp the place reserved for the a priori probability function $\Pr(\xi)$ (denoted $\Pr(\xi) \leftarrow T(\xi)$) in the evaluation of the second decoder (11); since both $p(\mathbf{x}|\xi)$ and $T(\xi)$ factor into the product of their respective marginals, we have

$$\begin{aligned} &\frac{\sum_{\xi:\xi_i=1}p(\mathbf{x}|\xi)p(\mathbf{z}|\xi)\Pr(\xi)}{\sum_{\xi:\xi_i=0}p(\mathbf{x}|\xi)p(\mathbf{z}|\xi)\Pr(\xi)} \\ &\quad \longleftarrow \frac{\sum_{\xi:\xi_i=1}p(\mathbf{x}|\xi)p(\mathbf{z}|\xi)T(\xi)}{\sum_{\xi:\xi_i=0}p(\mathbf{x}|\xi)p(\mathbf{z}|\xi)T(\xi)} \\ &\quad = \underbrace{\frac{p(x_i|\xi_i = 1)}{p(x_i|\xi_i = 0)}}_{\substack{\text{intrinsic}\\ \text{information}}}\underbrace{\frac{T_i(\xi_i = 1)}{T_i(\xi_i = 0)}}_{\text{pseudoprior}} \\ &\quad\quad \times \underbrace{\frac{\sum_{\xi:\xi_i=1}p(\mathbf{z}|\xi)\prod_{j\neq i}p(x_j|\xi_j)T_j(\xi_j)}{\sum_{\xi:\xi_i=0}p(\mathbf{z}|\xi)\prod_{j\neq i}p(x_j|\xi_j)T_j(\xi_j)}}_{\text{extrinsic information}}. \end{aligned} \tag{18}$$

Here we adopt the term "pseudoprior" for $T(\xi)$ since it usurps the a priori probability function; similarly, the result of this substitution may be termed a "pseudoposterior" which usurps the true a posteriori probability ratio.

Let now $U(\xi) = U_1(\xi_1)U_2(\xi_2)\cdots U_k(\xi_k)$ denote another factorable probability function whose bitwise ratios match the extrinsic information values furnished by this second decoder:

$$\begin{aligned} &\frac{U_i(\xi_i = 1)}{U_i(\xi_i = 0)} \\ &\quad = \frac{\sum_{\xi:\xi_i=1}p(\mathbf{z}|\xi)\prod_{j\neq i}p(x_j|\xi_j)T_j(\xi_j)}{\sum_{\xi:\xi_i=0}p(\mathbf{z}|\xi)\prod_{j\neq i}p(x_j|\xi_j)T_j(\xi_j)}, \quad i = 1,2,\ldots,k. \end{aligned} \tag{19}$$

This function may then usurp the a priori probability values used in the first decoder, and the process iterates. If we let a superscript (m) denote an iteration index, the coupling of

[1] Although we show this factorization here for a Gaussian channel, the factorization holds, of course, for any memoryless channel model.

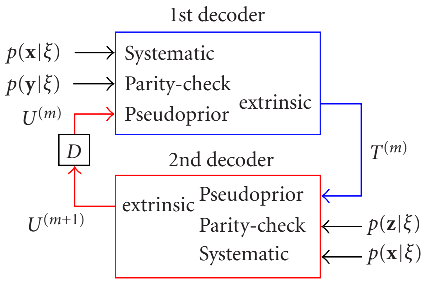

Figure 3: Flow graph of the turbo decoding algorithm.

the two decoders admits an external description as

$$\begin{aligned}\frac{p(x_i|\xi_i=1)}{p(x_i|\xi_i=0)}\,\frac{U_i^{(m)}(1)}{U_i^{(m)}(0)}\,\frac{T_i^{(m)}(1)}{T_i^{(m)}(0)} \\ = \frac{\sum_{\xi:\xi_i=1} p(\mathbf{y}|\xi)p(\mathbf{x}|\xi)U^{(m)}(\xi)}{\sum_{\xi:\xi_i=0} p(\mathbf{y}|\xi)p(\mathbf{x}|\xi)U^{(m)}(\xi)},\end{aligned} \tag{20}$$

$$\begin{aligned}\frac{p(x_i|\xi_i=1)}{p(x_i|\xi_i=0)}\,\frac{T_i^{(m)}(1)}{T_i^{(m)}(0)}\,\frac{U_i^{(m+1)}(1)}{U_i^{(m+1)}(0)} \\ = \frac{\sum_{\xi:\xi_i=1} p(\mathbf{z}|\xi)p(\mathbf{x}|\xi)T^{(m)}(\xi)}{\sum_{\xi:\xi_i=0} p(\mathbf{z}|\xi)p(\mathbf{x}|\xi)T^{(m)}(\xi)},\end{aligned} \tag{21}$$

in which (20) furnishes $T^{(m)}(\xi)$ and (21) furnishes $U^{(m+1)}(\xi)$. This is depicted in Figure 3. A fixed point corresponds to $U^{(m+1)}(\xi) = U^{(m)}(\xi)$ which, by inspection of the pseudoposteriors above, yields the following property.

Property 1. A fixed point is attained if and only if the two decoders yields the same pseudoposteriors (the left-hand sides of (20) and (21)) for $i = 1, 2, \ldots, k$.

A fixed point is therefore reflected by a state of "consensus" between the two decoders [15, 29, 37].

2.3. Existence of fixed points

A necessary (but not sufficient) condition for the algorithm to converge is that a fixed point exist, reflected by a state of consensus according to Property 1. A convenient tool in this direction is the Brouwer fixed point theorem [38], which asserts that any continuous map from a closed, bounded, and convex set into itself admits a fixed point; its application in the present context gives the following result [18, 29].

Theorem 1. *The turbo decoding algorithm from* (20) *and* (21) *always admits a fixed point.*

To verify, consider the pseudopriors $U^{(m)}(\xi_i)$ evaluated for $\xi_i = 1$, which, at any iteration m, are (pseudo-) probabilities lying between 0 and 1:

$$0 \le U^{(m)}(1) \le 1, \qquad i = 1, 2, \ldots, k. \tag{22}$$

This clearly gives a closed, bounded, and convex set. Since the updated pseudopriors $U^{(m+1)}$ also lie in this set, and since the map from $U^{(m)}(\xi)$ to $U^{(m+1)}(\xi)$ is continuous [18, 29], the conditions of the Brouwer theorem are satisfied, to show existence of a fixed point.

3. PROJECTIONS AND PRODUCT DISTRIBUTIONS

A key element of the development thus far concerns the calculation of bitwise marginal ratios which, according to [20], provide the troublesome element which accounts for the difference between a provably convergent algorithm [20] which is not practically implementable, and the implementable—but difficult to grasp—turbo decoding algorithm. We develop here an alternate viewpoint of the calculation of bitwise marginals in terms of a certain projection operator, adapted from the seminal work of Richardson [29].

Let $q(\xi)$ be a distribution, for example, a probability mass function, or a likelihood function, which assigns a nonnegative number to each of the 2^k evaluations of $\xi = (\xi_1, \ldots, \xi_k)$. We let $\mathbf{q}$ be the vector built from these 2^k evaluations:

$$\mathbf{q} = \left.\begin{bmatrix} q[\xi = (0,\ldots,0,0)] \\ q[\xi = (0,\ldots,0,1)] \\ \vdots \\ q[\xi = (1,\ldots,1,1)] \end{bmatrix}\right\} 2^k \text{ evaluations.} \tag{23}$$

We assume that $\mathbf{q}$ is scaled such that its entries sum to one. The k marginal distributions determined from $q(\xi)$, each having two evaluations at $\xi_i = 0$ and $\xi_i = 1$ $(1 \le i \le k)$, are given by

$$\begin{aligned} q_1(\xi_1 = 0) &= \sum_{\xi:\xi_1=0} q(\xi), & q_1(\xi_1 = 1) &= \sum_{\xi:\xi_1=1} q(\xi), \\ q_2(\xi_2 = 0) &= \sum_{\xi:\xi_2=0} q(\xi), & q_2(\xi_2 = 1) &= \sum_{\xi:\xi_2=1} q(\xi), \\ &\vdots & &\vdots \\ q_k(\xi_k = 0) &= \sum_{\xi:\xi_k=0} q(\xi), & q_k(\xi_k = 1) &= \sum_{\xi:\xi_k=1} q(\xi). \end{aligned} \tag{24}$$

Definition 1. The distribution $q(\xi)$ is a *product distribution* if it coincides with the product of its marginals:

$$q(\xi) = q_1(\xi_1)q_2(\xi_2)\cdots q_k(\xi_k). \tag{25}$$

The set of all product distributions is denoted by $\mathcal{P}$.

It is straightforward to check that $q(\xi) \in \mathcal{P}$ if and only if its vector representation is Kronecker decomposable as

$$\mathbf{q} = \mathbf{q}_1 \otimes \mathbf{q}_2 \otimes \cdots \otimes \mathbf{q}_k \tag{26}$$

with

$$\mathbf{q}_i = \begin{bmatrix} q_i(\xi_i = 0) \\ q_i(\xi_i = 1) \end{bmatrix}, \qquad i = 1, 2, \ldots, k. \tag{27}$$

We note also that $\mathcal{P}$ is closed under multiplication: if $q(\xi)$ and $r(\xi)$ belong to $\mathcal{P}$, so does their product:

$$s(\xi) = \alpha q(\xi) r(\xi) \in \mathcal{P}, \tag{28}$$

where the scalar α is chosen so that the evaluations of $s(\xi)$ sum to one. This operation can be expressed in vector notation using the Hadamard (or term-by-term) product:

$$\mathbf{s} = \alpha \mathbf{q} \odot \mathbf{r}. \tag{29}$$

To simplify notations, the scalar α will not be explicitly indicated, with the tacit understanding that the elements of the vector must be scaled to sum to one; we will henceforth write $\mathbf{s} = \mathbf{q} \odot \mathbf{r}$, omitting explicit mention of the scale factor α.

Suppose now $r(\xi)$ is not a product distribution. If $r_1(\xi_1), \dots, r_k(\xi_k)$ denote its marginal distributions, then we can set

$$q(\xi) = r_1(\xi_1) r_2(\xi_2) \cdots r_k(\xi_k), \tag{30}$$

to create a product distribution $q(\xi) \in \mathcal{P}$ which, by construction, generates the same marginals as $r(\xi)$:

$$q_i(\xi_i) = r_i(\xi_i), \quad i = 1, 2, \dots, k. \tag{31}$$

This operation will be denoted by

$$\mathbf{q} = \pi(\mathbf{r}). \tag{32}$$

We can observe that $\mathbf{q}$ is a product distribution ($\mathbf{q} \in \mathcal{P}$) if and only if $\pi(\mathbf{q}) = \mathbf{q}$, and since $\pi(\mathbf{r}) \in \mathcal{P}$ for any distribution $\mathbf{r}$, we must have $\pi(\pi(\mathbf{r})) = \pi(\mathbf{r})$, so that $\pi(\cdot)$ is a projection operator.

Definition 2. The distribution $\mathbf{q}$ is the projection of $\mathbf{r}$ into $\mathcal{P}$ if (i) $\mathbf{q} \in \mathcal{P}$ and (ii) $q_i(\xi_i) = r_i(\xi_i)$.

The following section details some simple information-theoretic properties which reinforce the interpretation as a projection.

3.1. Information-theoretic properties of the projector

The results summarized in this section may be understood as concrete transcriptions of ultimately deeper results from the field of information geometry [33, 34]. To begin, we recall that the entropy of a distribution $r(\xi)$ is defined as [39]

$$H(r) = -\sum_{\xi} r(\xi) \log_2 r(\xi), \tag{33}$$

involving the sum over all 2^k configurations of the vector $\xi = (\xi_1, \dots, \xi_k)$. A basic result of information theory asserts that the entropy of any joint distribution is upper bounded by the sum of the entropies of its marginal distributions [39], that is,

$$H(r) \le \sum_{i=1}^{k} H(r_i) = \sum_{i=1}^{k} \left(-\sum_{\xi_i=0}^{1} r_i(\xi_i) \log_2 \left(r_i(\xi_i) \right) \right), \tag{34}$$

with equality if and only if $r(\xi)$ factors into the product of its marginals [$r(\xi) \in \mathcal{P}$]. Therefore, if $\mathbf{r} \notin \mathcal{P}$, then by setting $\mathbf{q} = \pi(\mathbf{r})$, we have

$$H(r) \le \sum_{i=1}^{k} H(r_i) = \sum_{i=1}^{k} H(q_i) = H(q), \tag{35}$$

because $q_i(\xi_i) = r_i(\xi_i)$ and $q(\xi) \in \mathcal{P}$. This shows that the projection $\mathbf{q} = \pi(\mathbf{r})$ maximizes the entropy over all distributions that generate the same marginals as $r(\xi)$.

We recall next that the Kullback-Leibler distance (or relative entropy) between two distributions $r(\xi)$ and $s(\xi)$ is given by [20, 39]

$$D(r \| s) = \sum_{\xi} r(\xi) \log_2 \frac{r(\xi)}{s(\xi)} \ge 0, \tag{36}$$

with $D(r\|s) = 0$ if and only if $r(\xi) = s(\xi)$ for all ξ. If $s(\xi) \in \mathcal{P}$ and $\mathbf{q} = \pi(\mathbf{r})$, then we may verify (see the appendix) that

$$D(r\|s) = D(r\|q) + D(q\|s) \ge D(r\|q), \tag{37}$$

since $D(q\|s) \ge 0$, with equality if and only if $s(\xi) = q(\xi)$. This shows that the projection $q(\xi)$ is the closest product distribution to $r(\xi)$ using the Kullback-Leibler distance.

3.2. Application to turbo decoding

The added complication of accounting for the calculation of bitwise marginals noted in [20] can be offset by appealing to the previous section, which interprets bitwise marginals as resulting from a projection. Accordingly, we show in this section how the turbo decoding algorithm of (20) and (21) falls out as an alternating projection algorithm [29].

Let $\mathbf{p}_x$, $\mathbf{p}_y$, and $\mathbf{p}_z$ denote the vectors which collect the 2^k evaluations of the likelihood functions $p(\mathbf{x}|\xi)$, $p(\mathbf{y}|\xi)$, and $p(\mathbf{z}|\xi)$, respectively, that is,

$$\mathbf{p}_x = \left.\begin{bmatrix} p(\mathbf{x}|\xi = [0,\dots,0,0]) \\ p(\mathbf{x}|\xi = [0,\dots,0,1]) \\ \vdots \\ p(\mathbf{x}|\xi = [1,\dots,1,1]) \end{bmatrix}\right\} 2^k \text{ evaluations}, \tag{38}$$

and similarly for $\mathbf{p}_y$ and $\mathbf{p}_z$. Likewise, let the vectors $\mathbf{t}^{(m)}$ and $\mathbf{u}^{(m)}$ collect the 2^k evaluations of $T^{(m)}(\xi)$ and $U^{(m)}(\xi)$, respectively, at a given iteration m.

We can observe that the right-hand side of (20) calculates the bitwise marginal ratios of the distribution $p(\mathbf{y}|\xi)p(\mathbf{x}|\xi)U^{(m)}(\xi)$; this distribution admits a vector representation of the form $\mathbf{p}_y \odot \mathbf{p}_x \odot \mathbf{u}^{(m)}$. The left-hand side of (20) displays the bitwise marginal ratios of the product distribution $\mathbf{p}_x \odot \mathbf{u}^{(m)} \odot \mathbf{t}^{(m)}$ which generates, by construction, the same bitwise marginals as $\mathbf{p}_y \odot \mathbf{p}_x \odot \mathbf{u}^{(m)}$. This confirms that $\mathbf{p}_x \odot \mathbf{u}^{(m)} \odot \mathbf{t}^{(m)}$ is the projection of $\mathbf{p}_y \odot \mathbf{p}_x \odot \mathbf{u}^{(m)}$ in $\mathcal{P}$. By applying the same reasoning to (21), we establish the following [29].

Proposition 1. *The turbo decoding algorithm of* (20) *and* (21) *admits an exact description as the alternating projection algorithm*

$$\mathbf{p}_x \odot \mathbf{u}^{(m)} \odot \mathbf{t}^{(m)} = \pi\big(\mathbf{p}_y \odot \mathbf{p}_x \odot \mathbf{u}^{(m)}\big), \tag{39}$$

$$\mathbf{p}_x \odot \mathbf{u}^{(m+1)} \odot \mathbf{t}^{(m)} = \pi\big(\mathbf{p}_z \odot \mathbf{p}_x \odot \mathbf{t}^{(m)}\big). \tag{40}$$

From this, a connection with maximum-likelihood decoding follows readily [18].

Theorem 2. *If* $\mathbf{p}_x \odot \mathbf{p}_y$ *and/or* $\mathbf{p}_x \odot \mathbf{p}_z$ *is a product distribution, then*

(1) *the turbo decoding algorithm* ((39) *and* (40)) *converges in a single iteration;*

(2) *the pseudoposteriors so obtained agree with the maximum-likelihood decision rule for the code.*

For the proof, assume that $\mathbf{p}_x \odot \mathbf{p}_y \in \mathcal{P}$. We already have $\mathbf{u}^{(m)} \in \mathcal{P}$, and since $\mathcal{P}$ is closed under multiplication, we see that $\mathbf{p}_y \odot \mathbf{p}_x \odot \mathbf{u}^{(m)} \in \mathcal{P}$. Since the projector behaves as the identity operation for distributions in $\mathcal{P}$, the first decoder step of the turbo decoding algorithm from (39) becomes

$$\mathbf{p}_x \odot \mathbf{u}^{(m)} \odot \mathbf{t}^{(m)} = \pi\big(\mathbf{p}_y \odot \mathbf{p}_x \odot \mathbf{u}^{(m)}\big) = \mathbf{p}_y \odot \mathbf{p}_x \odot \mathbf{u}^{(m)}. \tag{41}$$

From this, we identify $\mathbf{p}_x \odot \mathbf{t}^{(m)} = \mathbf{p}_x \odot \mathbf{p}_y$ for all iterations m to show that a fixed point is attained. The second decoder from (40) then gives

$$\pi\big(\mathbf{p}_z \odot \mathbf{p}_x \odot \mathbf{t}^{(m)}\big) = \pi\big(\mathbf{p}_z \odot \mathbf{p}_x \odot \mathbf{p}_y\big), \tag{42}$$

which furnishes the bitwise marginal ratios of $p(\mathbf{x}|\xi)p(\mathbf{y}|\xi)p(\mathbf{z}|\xi)$. This agrees with the maximum-likelihood decision rule seen previously in (9). The proof when instead $\mathbf{p}_x \odot \mathbf{p}_z \in \mathcal{P}$ follows by exchanging the role of the two decoders.

Note that since $\mathbf{p}_x$ is already a product distribution (i.e., $\mathbf{p}_x \in \mathcal{P}$), it is sufficient (but not necessary) that $\mathbf{p}_y \in \mathcal{P}$ to have $\mathbf{p}_x \odot \mathbf{p}_y \in \mathcal{P}$. One may anticipate from this result that if $\mathbf{p}_x \odot \mathbf{p}_y$ and/or $\mathbf{p}_x \odot \mathbf{p}_z$ is "close" to a product distribution, then the algorithm should converge "rapidly;" formal steps confirming this notion are developed in [18]. Such proximity to a product distribution can be verified, in particular, in extreme signal-to-noise ratios [18].

Example 1 (high signal-to-noise ratios). Let ξ_* denote the vector of true information bits. The joint likelihood evaluation for $\mathbf{x}$ and $\mathbf{y}$ becomes

$$p(\mathbf{x},\mathbf{y}|\xi) \sim \exp\left(- \frac{\|\mathbf{c}_x(\xi_*) - \mathbf{c}_x(\xi) + \mathbf{b}_x\|^2}{2\sigma^2} - \frac{\|\mathbf{c}_y(\xi_*) - \mathbf{c}_y(\xi) + \mathbf{b}_y\|^2}{2\sigma^2}\right), \tag{43}$$

where $\mathbf{c}_x(\xi)$ and $\mathbf{c}_y(\xi)$ denote the antipodal (± 1) representation of the coded information bits ξ, and where $\mathbf{b}_x$ and $\mathbf{b}_y$ are the vectors of channel noise samples. As the noise variance σ^2 tends to zero, we have $\mathbf{b}_x, \mathbf{b}_y \to \mathbf{0}$, and

$$p(\mathbf{x},\mathbf{y}|\xi) \xrightarrow{\sigma^2 \to 0} \delta(\xi - \xi_*) = \begin{cases} 1, & \xi = \xi_*, \\ 0, & \xi \neq \xi_*. \end{cases} \tag{44}$$

We note that the delta function can always be written as the product of its marginals (which are themselves delta functions of the individual bits of ξ_*). Experimental evidence confirms that, in high signal-to-noise ratios, the algorithm converges rapidly to decoded symbols of high reliability.

Example 2 (poor signal-to-noise ratios). As the noise variance σ^2 increases, the likelihood evaluations are dominated by the presence of the noise terms; ratios of candidate likelihood evaluations then tend to 1, which is to say that $p(\mathbf{x},\mathbf{y}|\xi)$ approaches a uniform distribution:

$$p(\mathbf{x},\mathbf{y}|\xi) \xrightarrow{\sigma^2 \to \infty} \frac{1}{2^k} \quad \forall \xi. \tag{45}$$

We note that a uniform distribution can always be written as the product of its marginals (which are themselves uniform distributions). Experimental evidence again confirms (e.g., [15, 18]) that, in poor signal-to-noise ratios, the algorithm converges rapidly to a fixed point, but offers low confidence in the decoded symbols.

Although the above examples assume a Gaussian channel for simplicity, the basic reasoning can be extended to other memoryless channel models. More interesting, of course, is the convergence behavior for intermediate signal-to-noise ratios, which still presents a challenging problem. A natural question at this stage, however, is whether there exist constituent encoders which would give $\mathbf{p}_x \odot \mathbf{p}_y$ or $\mathbf{p}_x \odot \mathbf{p}_z$ as a product distribution irrespective of the signal-to-noise ratio. The answer is in the affirmative by considering, for example, a repetition code for the second constituent encoder. The arguments showing that $\mathbf{p}_x \in \mathcal{P}$ can then be copied to show that $\mathbf{p}_z \in \mathcal{P}$ as well (and therefore that $\mathbf{p}_x \odot \mathbf{p}_y \in \mathcal{P}$). But the distance properties of the resulting concatenated code are not very impressive, being basically the same as for the first constituent encoder. This concurs with an observation from [24], namely that "easily decodable" codes do not tend to be good codes.

4. SERIAL CONCATENATED CODES

We turn our attention now to serial concatenated codes, which have been studied extensively by Benedetto and his coworkers [2, 3, 35], and which encompass an ultimately richer structure. Our aim in this section is to show that the alternating projection interpretation again carries through, affording thus a unified study of serial and parallel concatenated codes.

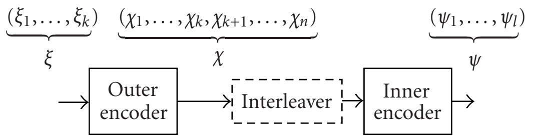

Figure 4: Flow graph for a serial concatenated code, with optional interleaver.

The basic flow graph for serial concatenated codes is depicted in Figure 4 in which the information bits $\xi = (\xi_1,\ldots,\xi_k)$ are first processed by an outer encoder, which here is systematic, so that the first k bits of its output $\chi = (\chi_1,\ldots,\chi_n)$ are the information bits:

$$\chi_i = \xi_i, \quad i = 1,2,\ldots,k. \tag{46}$$

The remaining bits $\chi_{k+1},\ldots,\chi_n$ furnish the $n-k$ parity-check bits. The cascaded inner encoder may admit different interpretations.

(i) The inner encoder may be a second (block or convolutional) encoder, perhaps endowed with an interleaver to offer protection against burst errors, consistent with conventional serial concatenated codes [2, 3]. Each input configuration χ is mapped to an output configuration ψ. With reference to Figure 4, the rate of the inner encoder is n/l.

(ii) The inner encoder may be a differential encoder, in order to endow the receiver with robustness against phase ambiguity in the received signal. Since a differential encoder is a particular case of a rate 1 convolutional encoder (with $l = n$ or perhaps $l = n+1$), this case is accommodated by the previous case.

(iii) The inner encoder may represent the convolutional effect induced by a channel whose memory is longer than the symbol period. In this case, taking into account that the symbols $\{\chi_i\}$ will have been converted to antipodal signaling (± 1), the baseband channel output appears as

$$\mathbf{v}_i = \underbrace{\sum_m \mathbf{h}_m\left(2\chi_{m-i}-1\right)}_{\psi_i} + \mathbf{b}_i, \tag{47}$$

where $\{\mathbf{h}_m\}$ denotes the equivalent impulse response of the baseband model, $\mathbf{b}_i$ is the additive channel noise, and where $\mathbf{v}_i$ may be scalar-valued (for a single-input single-output channel) or vector-valued (for a single-input multiple-output channel).

Certainly other interpretations may be developed as well; the above list may nonetheless be considered representative of some common configurations.

4.1. Optimum decoding

With $\mathbf{v}$ denoting the noisy received signal ψ (after conversion to antipodal form, possibly corrupted by intersymbol interference), the optimum decoding metric is again based on the a posteriori marginal probability ratios

$$\begin{aligned}\frac{\Pr\left(\xi_i = 1|\mathbf{v}\right)}{\Pr\left(\xi_i = 0|\mathbf{v}\right)} &= \frac{\sum_{\xi:\xi_i=1}\Pr(\xi|\mathbf{v})}{\sum_{\xi:\xi_i=0}\Pr(\xi|\mathbf{v})} \\ &= \frac{\sum_{\xi:\xi_i=1} p(\mathbf{v}|\xi)\Pr(\xi)}{\sum_{\xi:\xi_i=0} p(\mathbf{v}|\xi)\Pr(\xi)}, \quad i = 1,2,\ldots,k.\end{aligned} \tag{48}$$

If all input configurations are equally probable, we have $\Pr(\xi) = 1/2^k$ and we recover the maximum-likelihood decoding rule.

If no interleaver is used between the two coders, then the mapping from ξ to $\mathbf{v}$ is a noisy convolution, allowing a trellis structure to perform optimum decoding at a reasonable computational cost. In the presence of an interleaver, on the other hand, the convolutional structure between ξ and $\mathbf{v}$ is compromised, such that a direct evaluation of (48) leads to a computational complexity that grows exponentially with the block length. Iterative decoding, to be reviewed next, represents an attempt to reduce the decoding complexity to a reasonable value.

4.2. Iterative decoding for serial concatenated codes

Iterative serial decoding [2] amounts to implementing locally optimum decoders which infer χ from $\mathbf{v}$, and then ξ from χ, and subsequently exchanging information until consensus is reached. Our development emphasizes the external descriptions of the local decoding operations in order to better identify the form of consensus that is reached, as well as to justify the seemingly heuristic coupling between the coders by way of connections with maximum-likelihood decoding.

Consider first the inner decoding rule, which seeks to determine the inner encoder's input $\chi = (\chi_1,\ldots,\chi_n)$ from the noisy received signal $\mathbf{v}$:

$$\begin{aligned}\frac{\Pr\left(\chi_i = 1|\mathbf{v}\right)}{\Pr\left(\chi_i = 0|\mathbf{v}\right)} &= \frac{\sum_{\chi:\chi_i=1}\Pr(\chi|\mathbf{v})}{\sum_{\chi:\chi_i=0}\Pr(\chi|\mathbf{v})} \\ &= \frac{\sum_{\chi:\chi_i=1} p(\mathbf{v}|\chi)\Pr(\chi)}{\sum_{\chi:\chi_i=0} p(\mathbf{v}|\chi)\Pr(\chi)}, \quad i = 1,2,\ldots,n.\end{aligned} \tag{49}$$

The inner decoder assumes that the a priori probability mass function $\Pr(\chi)$ factors into the product of its marginals as

$$\Pr(\chi) = \Pr\left(\chi_1\right)\Pr\left(\chi_2\right)\cdots\Pr\left(\chi_n\right). \tag{50}$$

This assumption, strictly speaking, is incorrect, because the bits $\{\chi_i\}$ are produced by the outer encoder, which imposes dependencies between the bits for error control purposes. The forward-backward algorithm from [36], however, cannot exploit these dependencies without incurring a significant increase in computational complexity. By turning a "blind eye" to this fact, and therefore admitting the factorization of $\Pr(\chi)$ into the product of its marginals, each term from the numerator (resp., denominator) of (49) will contain a factor $\Pr(\chi_i = 1)$ (resp., $\Pr(\chi_i = 0)$), which gives

$$\frac{\Pr(\chi_i=1|\mathbf{v})}{\Pr(\chi_i=0|\mathbf{v})} = \frac{\Pr(\chi_i=1)}{\Pr(\chi_i=0)} \underbrace{\frac{\sum_{\chi:\chi_i=1} p(\mathbf{v}|\chi)\prod_{j\neq i}\Pr(\chi_j)}{\sum_{\chi:\chi_i=0} p(\mathbf{v}|\chi)\prod_{j\neq i}\Pr(\chi_j)}}_{\text{extrinsic information}}, \quad i=1,2,\dots,n. \tag{51}$$

We now let $T(\chi) = T_1(\chi_1)\cdots T_n(\chi_n)$ denote a factorable probability mass function whose marginal ratios match the extrinsic information values above:

$$\frac{T_i(\chi_i = 1)}{T_i(\chi_i = 0)} = \frac{\sum_{\chi:\chi_i=1} p(\mathbf{v}|\chi)\prod_{j\neq i}\Pr(\chi_j)}{\sum_{\chi:\chi_i=0} p(\mathbf{v}|\chi)\prod_{j\neq i}\Pr(\chi_j)}. \tag{52}$$

The outer decoder would normally aim to determine the information bits ξ based on an estimate (denoted by $\hat{\chi}$) of the outer encoder's output, according to the a posteriori probability ratios

$$\frac{\Pr(\xi_i = 1|\hat{\chi})}{\Pr(\xi_i = 0|\hat{\chi})} = \frac{\sum_{\xi:\xi_i=1}\Pr(\xi|\hat{\chi})}{\sum_{\xi:\xi_i=0}\Pr(\xi|\hat{\chi})} = \frac{\sum_{\xi:\xi_i=1} p(\hat{\chi}|\xi)\Pr(\xi)}{\sum_{\xi:\xi_i=0} p(\hat{\chi}|\xi)\Pr(\xi)}. \tag{53}$$

The estimate $\hat{\chi}$, however, is not immediately available. If it were, then each likelihood function evaluation would appear as

$$p(\hat{\chi}|\xi) \sim \exp\left(-\sum_{j=1}^{n}\frac{\left[\hat{\chi}_j - (2\chi_j(\xi)-1)\right]^2}{2\sigma^2}\right), \tag{54}$$

assuming a Gaussian channel, in which $\chi_j(\xi)$ is either 0 or 1, depending on $\xi = (\xi_1,\dots,\xi_k)$. To each hypothetical bit $\hat{\chi}_j$, therefore, we associate two evaluations as $\exp[-(\hat{\chi}_j \pm 1)^2/(2\sigma^2)]$ (corresponding to $\chi_j(\xi) = 0$ or 1), which are usurped by the two evaluations of $T_j(\chi_j)$ from (52):

$$\frac{\exp\left[-(\hat{\chi}_j - 1)^2/(2\sigma^2)\right]}{\exp\left[-(\hat{\chi}_j + 1)^2/(2\sigma^2)\right]} \longleftarrow \frac{T_j(1)}{T_j(0)}. \tag{55}$$

The forward-backward algorithm [36] may then run, following this systematic substitution.

To develop an external description of the decoding algorithm which results, we note that this substitution amounts to usurping the likelihood function $p(\hat{\chi}|\xi)$ by

$$p(\hat{\chi}|\xi) \longleftarrow \prod_{j=1}^{n} T_j(\chi_j(\xi)), \tag{56}$$

in which the right-hand side notationally emphasizes that only those bit combinations $\chi_1,\dots,\chi_n$ that lie in the outer codebook make sense.

To arrive at a more convenient form, let $\phi(\chi)$ denote the indicator function for the outer codebook:

$$\phi(\chi) = \begin{cases} 1 & \text{if } \chi \text{ lies in the outer codebook,} \\ 0 & \text{otherwise.} \end{cases} \tag{57}$$

The 2^n configurations of $(\chi_1,\dots,\chi_n)$ generate 2^n evaluations of $\prod_{j=1}^{n} T_j(\chi_j)$, but only 2^k of these evaluations survive in the product $\phi(\chi)\prod_j T_j(\chi_j)$, namely, the 2^k evaluations from the right-hand side of (56) which are generated as ξ varies over its 2^k configurations. We may then establish a one-to-one correspondence between the 2^k "surviving" evaluations in $\phi(\chi)\prod_j T_j(\chi_j)$ and the 2^k evaluations of the likelihood function $p(\hat{\chi}|\xi)$ which are usurped in (56). Assuming that $\Pr(\xi)$ is a uniform distribution, the usurped pseudoposteriors from (53) become

$$\begin{aligned} \frac{\sum_{\xi:\xi_i=1} p(\hat{\chi}|\xi)}{\sum_{\xi:\xi_i=0} p(\hat{\chi}|\xi)} &\longleftarrow \frac{\sum_{\chi:\chi_i=1}\phi(\chi)\prod_{j=1}^{n} T_j(\chi_j)}{\sum_{\chi:\chi_i=0}\phi(\chi)\prod_{j=1}^{n} T_j(\chi_j)} \\ &= \frac{T_i(1)}{T_i(0)} \underbrace{\frac{\sum_{\chi:\chi_i=1}\phi(\chi)\prod_{j\neq i} T_j(\chi_j)}{\sum_{\chi:\chi_i=0}\phi(\chi)\prod_{j\neq i} T_j(\chi_j)}}_{\text{extrinsic information}}, \end{aligned} \tag{58}$$

in which we note the following:

(i) since the outer code is systematic, the first k bits $\chi_1,\dots,\chi_k$ coincide with the information bits $\xi_1,\dots,\xi_k$, allowing therefore a direct substitution for the variables of summation. In addition, the formula above may be evaluated as written for the parity-check bits $\chi_{k+1},\dots,\chi_n$;

(ii) each term in the numerator (resp., denominator) contains a factor $T_i(\chi_i = 1)$ (resp., $T_i(\chi_i = 0)$), so that the ratio $T_i(1)/T_i(0)$ naturally factors out. Let

$$U(\chi) = U_1(\chi_1)\cdots U_n(\chi_n) \tag{59}$$

be a factorable probability function whose marginal ratios match the extrinsic information values:

$$\frac{U_i(1)}{U_i(0)} = \frac{\sum_{\chi:\chi_i=1}\phi(\chi)\prod_{j\neq i} T_j(\chi_j)}{\sum_{\chi:\chi_i=0}\phi(\chi)\prod_{j\neq i} T_j(\chi_j)}, \quad i = 1,2,\dots,n. \tag{60}$$

These values may then usurp the a priori probability function $\Pr(\chi)$ of the inner decoder: $\Pr(\chi) \leftarrow U(\chi)$.

If we let a superscript (m) denote an iteration number, then the coupling of the two decoders admits an external description of the form

$$\frac{T_i^{(m)}(1)}{T_i^{(m)}(0)}\frac{U_i^{(m)}(1)}{U_i^{(m)}(0)} = \frac{\sum_{\chi:\chi_i=1} p(\mathbf{v}|\chi)\prod_{j=1}^{n} U_j^{(m)}(\chi_j)}{\sum_{\chi:\chi_i=0} p(\mathbf{v}|\chi)\prod_{j=1}^{n} U_j^{(m)}(\chi_j)}, \quad i = 1,2,\dots,n, \tag{61}$$

$$\frac{T_i^{(m)}(1)}{T_i^{(m)}(0)}\frac{U_i^{(m+1)}(1)}{U_i^{(m+1)}(0)} = \frac{\sum_{\chi:\chi_i=1}\phi(\chi)\prod_{j=1}^{n} T_j^{(m)}(\chi_j)}{\sum_{\chi:\chi_i=0}\phi(\chi)\prod_{j=1}^{n} T_j^{(m)}(\chi_j)}, \quad i = 1,2,\dots,n, \tag{62}$$

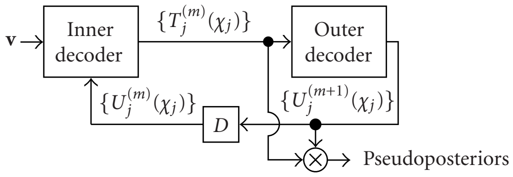

FIGURE 5: Flow graph for iterative decoding of serial concatenated codes.

as depicted in Figure 5. A fixed point corresponds to $U^{(m+1)}(\chi) = U^{(m)}(\chi)$ which, in analogy with the parallel concatenated code case, can be characterized as the following "consensus" property.

Property 2. A fixed point in the serial decoding algorithm occurs if and only if the two decoders yield the same pseudoposteriors (left-hand sides of (61) and (62)) for $i = 1, 2, \ldots, n$.

Note that the consensus here covers the information bits plus the parity-check bits furnished by the outer decoder. As with the parallel concatenated code case, the existence of fixed points follows by applying the Brouwer fixed point theorem (cf. Section 2.3).

4.3. Projection interpretation

The iterative decoding algorithm for serial concatenated codes can also be rephrased as an alternating projection algorithm, analogously to the parallel concatenated code case of Section 3, as we develop presently.

We continue to denote by $\mathcal{P}$ the set of distributions $q(\xi)$ which factor into the product of their marginals:

$$q(\chi) = q_1(\chi_1)q_2(\chi_2)\cdots q_n(\chi_n). \tag{63}$$

The only modification here is that we now have n marginal distributions to consider, to account for the k information bits plus the $n-k$ parity-check bits which intervene in the consensus of Property 2. If $r(\chi)$ is an arbitrary distribution, then $\mathbf{q} = \pi(\mathbf{r})$ yields a distribution $q(\chi) \in \mathcal{P}$ which generates the same n marginal distributions as $r(\chi)$.

We let $\mathbf{p}_v$ denote the vector containing the 2^n likelihood evaluations of $p(\mathbf{v}|\chi)$:

$$\mathbf{p}_v = \left.\begin{bmatrix} p[\mathbf{v}|\chi = (0,\ldots,0,0)] \\ p[\mathbf{v}|\chi = (0,\ldots,0,1)] \\ \vdots \\ p[\mathbf{v}|\chi = (1,\ldots,1,1)] \end{bmatrix}\right\} 2^n \text{ evaluations.} \tag{64}$$

Similarly, let the vectors $\mathbf{t}^{(m)}$, $\mathbf{u}^{(m)}$, and ϕ collect their respective 2^n evaluations:

$$\mathbf{t}^{(m)} = \begin{bmatrix} T_1^{(m)}(0)\cdots T_n^{(m)}(0) \\ T_1^{(m)}(0)\cdots T_n^{(m)}(1) \\ \vdots \\ T_1^{(m)}(1)\cdots T_n^{(m)}(1) \end{bmatrix},$$
$$\mathbf{u}^{(m)} = \begin{bmatrix} U_1^{(m)}(0)\cdots U_n^{(m)}(0) \\ U_1^{(m)}(0)\cdots U_n^{(m)}(1) \\ \vdots \\ U_1^{(m)}(1)\cdots U_n^{(m)}(1) \end{bmatrix}, \tag{65}$$
$$\phi = \begin{bmatrix} \phi[\chi = (0,\ldots,0,0)] \\ \phi[\chi = (0,\ldots,0,1)] \\ \vdots \\ \phi[\chi = (1,\ldots,1,1)] \end{bmatrix}.$$

With respect to the inner decoder, we see that the right-hand side of (61) calculates the marginal ratios of the distribution $p(\mathbf{v}|\chi)U^{(m)}(\chi)$, which distribution admits a vector representation as $\mathbf{p}_v \odot \mathbf{u}^{(m)}$. The left-hand side of (61) contains the marginal ratios of $\mathbf{t}^{(m)} \odot \mathbf{u}^{(m)} \in \mathcal{P}$, which agree with those of $\mathbf{p}_v \odot \mathbf{u}^{(m)}$, consistent with our projection operation. By applying the same reasoning to (62), we obtain a natural counterpart to Proposition 1.

Proposition 2. *The iterative serial decoding algorithm of* (61) *and* (62) *coincides with the alternating projection algorithm*

$$\begin{aligned} \mathbf{t}^{(m)} \odot \mathbf{u}^{(m)} &= \pi(\mathbf{p}_v \odot \mathbf{u}^{(m)}), \\ \mathbf{t}^{(m)} \odot \mathbf{u}^{(m+1)} &= \pi(\phi \odot \mathbf{t}^{(m)}). \end{aligned} \tag{66}$$

From this follows a natural analogue to Theorem 2 establishing a key link with maximum-likelihood decoding.

Theorem 3. *If $p(\mathbf{v}|\chi)$ factors into the product of its marginals, then*

(1) *the iterative algorithm* (61) *and* (62) *converges in a single iteration;*
(2) *the pseudoposteriors so obtained agree with the maximum-likelihood decision metric for the code.*

The proof parallels that of Theorem 2, but displays its own particularities which merit its inclusion here. If $p(\mathbf{v}|\chi)$ factors into the product of its marginals, then $\mathbf{p}_v \in \mathcal{P}$, giving $\mathbf{p}_v \odot \mathbf{u}^{(m)} \in \mathcal{P}$ as well. Since the projector behaves as the identity when applied to elements of $\mathcal{P}$, the first displayed equation of Proposition 2 becomes

$$\mathbf{t}^{(m)} \odot \mathbf{u}^{(m)} = \pi(\mathbf{p}_v \odot \mathbf{u}^{(m)}) = \mathbf{p}_v \odot \mathbf{u}^{(m)}. \tag{67}$$

From this we identify $\mathbf{t}^{(m)} = \mathbf{p}_v$ for all iterations m, giving a fixed point. Substituting $\mathbf{t}^{(m)} = \mathbf{p}_v$ into the projector of the second displayed equation of Proposition 2 reveals

$$\mathbf{t}^{(m)} \odot \mathbf{u}^{(m+1)} = \pi(\phi \odot \mathbf{t}^{(m)}) = \pi(\phi \odot \mathbf{p}_v). \tag{68}$$

This calculates the marginal functions of $\phi(\chi)p(\mathbf{v}|\chi)$, whose surviving evaluations are the restriction of the likelihood function $p(\mathbf{v}|\chi)$ to the outer codebook:

$$\phi(\chi)p(\mathbf{v}|\chi) = \begin{cases} p(\mathbf{v}|\chi(\xi)) = p(\mathbf{v}|\xi) & \text{if } \phi(\chi) = 1, \\ 0 & \text{otherwise.} \end{cases} \tag{69}$$

Since the outer code is systematic, we have $\chi_i = \xi_i$ for $i = 1, \dots, k$. Therefore, the first k marginal ratios from $\phi(\chi)p(\mathbf{v}|\chi)$ coincide with those from $p(\mathbf{v}|\xi)$; these in turn agree with the maximum-likelihood decoding rule which results from (48) when the a priori probability function $\Pr(\xi)$ is uniform.

As with the case of parallel concatenated codes, the likelihood function $p(\mathbf{v}|\chi)$ will be "close" to a factorable distribution when the signal-to-noise ratio is sufficiently high or sufficiently low. The conclusions from [18, Section 3, Examples 1 and 2] therefore apply to serial concatenated codes as well.

5. CONCLUDING REMARKS

We have developed a tutorial overview of iterative decoding for parallel and serial concatenated codes, in the hopes of rendering this material accessible to a wider audience. Our development has emphasized descriptions and properties which are valid irrespective of the block length, which may facilitate the analysis of such algorithms for short block lengths. At the same time, the presentation emphasizes how decoding algorithms for parallel and serial concatenated codes may be addressed in a unified manner.

Although different properties have been exposed, the critical question of convergence domains versus code choice and signal-to-noise ratio remains less immediate to develop. The natural extension of the projection viewpoint favored here involves studying the stability properties of the dynamic system which results. This is pursued in [18, 29] (among others) in which explicit expressions for the Jacobian of the system feedback matrix are obtained; once a fixed point is isolated, local stability properties can then be studied [18, 29], but they depend in a complicated manner on the specific code and channel properties (distance, block length, signal-to-noise ratio, etc.).

One may observe that a fixed point occurs whenever the pseudoposteriors assume uniform distributions, and that this gives a convergent point in pessimistic signal-to-noise ratios [18]. With some further code constraints [40], fixed points are also shown to occur at codeword configurations (i.e., where $T_i(1) = \xi_i$), consistent with the observed convergence behavior for signal-to-noise ratios beyond the waterfall region, and corresponding to an unequivocal fixed point in the terminology of [18]. Interestingly, the convergence of pseudoprobabilities to 0 or 1 was observed for low-density parity-check codes as far back as [6]. Deducing the stability properties of different fixed points versus the signal-to-noise ratio and block length, however, remains a challenging problem.

By allowing the block length to become arbitrarily long, large sample approximations may be invoked, which typically take the form of log-pseudoprobability ratios approaching independent Gaussian random variables. Many insightful analyses may then be developed (e.g., [15, 16, 17, 19], among others). Such approximations, however, are known to be less than faithful for shorter block lengths, of greater interest in two-way communication systems, and analyses exploiting large sample approximations do not adequately predict the behavior of iterative decoding algorithms for shorter block lengths.

Graphical methods (including [25, 26, 27, 28]) provide another powerful analysis technique in this direction. Present trends include studying how code design impacts the cycle length of the decoding algorithm, based on the plausible conjecture that longer cycles should have a greater "stability margin" in an ultimately closed-loop system. Further study, however, is required to better understand the stability properties of iterative decoding algorithms in the general case.

APPENDIX

VERIFICATION OF IDENTITY (37)

Let $r(\xi)$ be an arbitrary distribution, and let $q(\xi)$ be its projection in $\mathcal{P}$, giving a product distribution $q(\xi) = q_1(\xi_1) \cdots q_k(\xi_k)$ whose marginals match those of $r(\xi)$: $q_i(\xi_i) = r_i(\xi_i)$. Consider first

$$\begin{aligned} D(r\|q) &= \sum_{\xi_1=0}^{1} \cdots \sum_{\xi_k=0}^{1} r(\xi) \log_2 \frac{r(\xi)}{q_1(\xi_1) \cdots q_k(\xi_k)} \\ &= \underbrace{\sum_{\xi_1=0}^{1} \cdots \sum_{\xi_k=0}^{1} r(\xi) \log_2 r(\xi)}_{-H(r)} - \sum_{\xi_1=0}^{1} \cdots \sum_{\xi_k=0}^{1} r(\xi) \log_2 \left(q_1(\xi_1) \cdots q_k(\xi_k)\right) \\ &= -H(r) + \underbrace{\left(\sum_{\xi_1=0}^{1} \cdots \sum_{\xi_k=0}^{1} r(\xi) \log_2 q_1(\xi_1) + \cdots + \sum_{\xi_1=0}^{1} \cdots \sum_{\xi_k=0}^{1} r(\xi) \log_2 q_k(\xi_k) \right)}_{(a)}. \end{aligned} \tag{A.1}$$

The ith sum from the term (a) appears as

$$\begin{aligned}
&\sum_{\xi_1=0}^{1}\cdots\sum_{\xi_i=0}^{1}\cdots\sum_{\xi_k=0}^{1} r(\xi)\log_2 q_i(\xi_i)\\
&\quad= \sum_{\xi_i=0}^{1}\log_2 q_i(\xi_i)\underbrace{\sum_{j\neq i}\left(\sum_{\xi_j=0}^{1} r(\xi_1,\ldots,\xi_k)\right)}_{r_i(\xi_i)=q_i(\xi_i)}\\
&\quad= \sum_{\xi_i=0}^{1} r_i(\xi_i)\log_2 q_i(\xi_i) = -H(r_i) = -H(q_i),
\end{aligned}\tag{A.2}$$

since the sums over bits other than i extract the ith marginal function $r_i(\xi_i)$, which coincides with $q_i(\xi_i)$. Combining with the previous expression, we see that

$$D(r\|q) = \sum_{i=1}^{k} H(r_i) - H(r).\tag{A.3}$$

Now let $s(\xi) = s_1(\xi_i)\cdots s_k(\xi_k)$ be an arbitrary product distribution. The same steps illustrated above give

$$\begin{aligned}
D(r\|s) &= \sum_{\xi_1=0}^{1}\cdots\sum_{\xi_k=0}^{1} r(\xi)\log_2\frac{r(\xi)}{s_1(\xi_1)\cdots s_k(\xi_k)}\\
&= -H(r) - \Bigg(\sum_{\xi_1=0}^{1}\cdots\sum_{\xi_k=0}^{1} r(\xi)\log_2 s_1(\xi_1) + \cdots\\
&\qquad + \sum_{\xi_1=0}^{1}\cdots\sum_{\xi_k=0}^{1} r(\xi)\log_2 s_k(\xi_k)\Bigg)\\
&= -H(r) - \Bigg(\sum_{\xi_1=0}^{1} r_1(\xi_1)\log_2 s_1(\xi_1) + \cdots\\
&\qquad + \sum_{\xi_k=0}^{1} r_k(\xi_k)\log_2 s_k(\xi_k)\Bigg)\\
&= -H(r) - \Bigg(\sum_{\xi_1=0}^{1} q_1(\xi_1)\log_2 s_1(\xi_1) + \cdots\\
&\qquad + \sum_{\xi_n=0}^{1} q_k(\xi_k)\log_2 s_k(\xi_k)\Bigg).
\end{aligned}\tag{A.4}$$

Adding and subtracting the sums

$$\begin{aligned}
&\sum_{\xi_1=0}^{1} q_1(\xi_1)\log_2 q_1(\xi_1) + \cdots + \sum_{\xi_n=0}^{1} q_k(\xi_k)\log_2 q_k(\xi_k)\\
&\quad= -\sum_{i=1}^{k} H(q_i),
\end{aligned}\tag{A.5}$$

and regrouping gives

$$\begin{aligned}
&D(r\|s)\\
&\quad= \underbrace{-H(r) + \sum_{i=1}^{k} H(q_i)}_{D(r\|q)}\\
&\qquad+ \underbrace{\sum_{\xi_1=0}^{1} q_1(\xi_1)\log_2\frac{q_1(\xi_1)}{s_1(\xi_1)} + \cdots + \sum_{\xi_k=0}^{1} q_k(\xi_k)\log_2\frac{q_k(\xi_n)}{s_k(\xi_k)}}_{\sum_{i=1}^{k} D(q_i\|s_i)=D(q\|s)},
\end{aligned}\tag{A.6}$$

which is the identity (37).

ACKNOWLEDGMENT

This work was supported by the Scientific Services Program of the US Army, Contract no. DAAD19-02-D-0001.

REFERENCES

[1] C. Berrou and A. Glavieux, "Near optimum error correcting coding and decoding: turbo-codes," *IEEE Trans. Commun.*, vol. 44, no. 10, pp. 1261–1271, 1996.

[2] S. Benedetto and G. Montorsi, "Iterative decoding of serially concatenated convolutional codes," *Electronics Letters*, vol. 32, no. 13, pp. 1186–1188, 1996.

[3] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara, "Analysis, design, and iterative decoding of double serially concatenated codes with interleavers," *IEEE J. Select. Areas Commun.*, vol. 16, no. 2, pp. 231–244, 1998.

[4] D. J. C. MacKay, "Good error-correcting codes based on very sparse matrices," *IEEE Trans. Inform. Theory*, vol. 45, no. 2, pp. 399–431, 1999.

[5] Y. Kou, S. Lin, and M. P. C. Fossorier, "Low-density parity-check codes based on finite geometries: a rediscovery and new results," *IEEE Trans. Inform. Theory*, vol. 47, no. 7, pp. 2711–2736, 2001.

[6] R. G. Gallager, "Low-density parity-check codes," *IRE Trans. Inform. Theory*, vol. 8, no. 1, pp. 21–28, 1962.

[7] C. Douillard, M. Jezequel, C. Berrou, P. Picart, P. Didier, and A. Glavieux, "Iterative correction of intersymbol interference: turbo equalization," *European Transactions on Telecommunications*, vol. 6, no. 5, pp. 507–511, 1995.

[8] C. Laot, A. Glavieux, and J. Labat, "Turbo equalization: adaptive equalization and channel decoding jointly optimized," *IEEE J. Select. Areas Commun.*, vol. 19, no. 9, pp. 1744–1752, 2001.

[9] M. Tüchler, R. Kotter, and A. Singer, "Turbo equalization: principles and new results," *IEEE Trans. Commun.*, vol. 50, no. 5, pp. 754–767, 2002.

[10] X. Wang and H. V. Poor, "Iterative (turbo) soft interference cancellation and decoding for coded CDMA," *IEEE Trans. Commun.*, vol. 47, no. 7, pp. 1046–1061, 1999.

[11] X. Wang and H. V. Poor, "Blind joint equalization and multiuser detection for DS-CDMA in unknown correlated noise," *IEEE Trans. Circuits Syst. II*, vol. 46, no. 7, pp. 886–895, 1999.

[12] C. Heegard and S. B. Wicker, *Turbo Coding*, Kluwer Academic Publishers, Boston, Mass, USA, 1999.

[13] B. Vucetic and J. Yuan, *Turbo Codes: Principles and Applications*, Kluwer Academic Publishers, Boston, Mass, USA, 2000.

[14] L. Hanzo, T. H. Liew, and B. L. Yeap, *Turbo Coding, Turbo Equalisation and Space-Time Coding*, John Wiley & Sons, Chichester, UK, 2002.

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

[16] T. Richardson and R. Urbanke, "An introduction to the analysis of iterative coding systems," in *Codes, Systems, and Graphical Models*, IMA Volume in Mathematics and Its Applications, pp. 1–37, New York, NY, USA, 2001.

[17] D. Divsalar, S. Dolinar, and F. Pollara, "Iterative turbo decoder analysis based on density evolution," *IEEE J. Select. Areas Commun.*, vol. 19, no. 5, pp. 891–907, 2001.

[18] D. Agrawal and A. Vardy, "The turbo decoding algorithm and its phase trajectories," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 699–722, 2001.

[19] H. El Gamal and A. R. Hammons Jr., "Analyzing the turbo decoder using the Gaussian approximation," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 671–686, 2001.

[20] M. Moher and T. A. Gulliver, "Cross-entropy and iterative decoding," *IEEE Trans. Inform. Theory*, vol. 44, no. 7, pp. 3097–3104, 1998.

[21] R. Le Bidan, C. Laot, D. LeRoux, and A. Glavieux, "Analyse de la convergence en turbo-détection," in *Proc. Colloque GRETSI sur le Traitement du Signal et des Images (GRETSI '01)*, Toulouse, France, September 2001.

[22] A. Roumy, A. J. Grant, I. Fijalkow, P. D. Alexander, and D. Pirez, "Turbo-equalization: convergence analysis," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '01)*, vol. 4, pp. 2645–2648, Salt Lake City, Utah, USA, May 2001.

[23] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann Publishers, San Mateo, Calif, USA, 1988.

[24] 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, no. 2, pp. 140–152, 1998.

[25] R. M. Tanner, "A recursive approach to low complexity codes," *IEEE Trans. Inform. Theory*, vol. 27, no. 5, pp. 533–547, 1981.

[26] N. Wiberg, *Codes and decoding on general graphs*, Ph.D. thesis, Linköping University, Linköping, Sweden, April 1996.

[27] F. R. Kschischang and B. J. Frey, "Iterative decoding of compound codes by probability propagation in graphical models," *IEEE J. Select. Areas Commun.*, vol. 16, no. 2, pp. 219–230, 1998.

[28] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 498–519, 2001.

[29] T. Richardson, "The geometry of turbo-decoding dynamics," *IEEE Trans. Inform. Theory*, vol. 46, no. 1, pp. 9–23, 2000.

[30] 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 Symposium on Theory of Computing*, pp. 249–258, Dallas, Tex, USA, 1998.

[31] N. Sourlas, "Spin-glass models as error-correcting codes," *Nature*, vol. 339, no. 6227, pp. 693–695, 1989.

[32] A. Montanari and N. Sourlas, "Statistical mechanics and turbo codes," in *Proc. 2nd International Symposium on Turbo Codes and Related Topics*, pp. 63–66, Brest, France, September 2000.

[33] S. Ikeda, T. Tanaka, and S. Amari, "Information geometry of turbo and low-density parity-check codes," *IEEE Trans. Inform. Theory*, vol. 50, no. 6, pp. 1097–1114, 2004.

[34] S. Amari and H. Nagaoka, *Methods of information geometry*, American Mathematical Society, Providence, RI, USA; Oxford University Press, New York, NY, USA, 2000.

[35] S. Benedetto and G. Montorsi, "Unveiling turbo codes: some results on parallel concatenated coding schemes," *IEEE Trans. Inform. Theory*, vol. 42, no. 2, pp. 409–428, 1996.

[36] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate (corresp.)," *IEEE Trans. Inform. Theory*, vol. 20, no. 2, pp. 284–287, 1974.

[37] J. Hagenauer, E. Offer, and L. Papke, "Iterative decoding of binary block and convolutional codes," *IEEE Trans. Inform. Theory*, vol. 42, no. 2, pp. 429–445, 1996.

[38] T. L. Saaty and J. Bram, *Nonlinear Mathematics*, McGraw-Hill, New York, NY, USA, 1964.

[39] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, John Wiley & Sons, New York, NY, USA, 1991.

[40] P. A. Regalia, "Contractivity in turbo iterations," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '04)*, vol. 4, pp. 637–640, Montreal, Canada, May 2004.

Phillip A. Regalia was born in Walnut Creek, California, in 1962. He received the B.S. (with honors), M.S., and Ph.D. degrees in electrical and computer engineering in 1985, 1987, and 1988, respectively, from the University of California at Santa Barbara, and the Habilitation à Diriger des Recherches (necessary to advance to the level of Professor in the French academic system) from the University of Paris-Orsay in 1994. He is presently with the Department of Electrical Engineering and Computer Science, the Catholic University of America, Washington, DC, and Editor-in-Chief of the EURASIP Journal of Wireless Communications and Networking.