Extrinsic Information Transfer Functions: Model and Erasure Channel Properties

Alexei Ashikhmin, *Member, IEEE*, Gerhard Kramer, *Member, IEEE*, and Stephan ten Brink, *Member, IEEE*

***Abstract*—Extrinsic information transfer (EXIT) charts are a tool for predicting the convergence behavior of iterative processors for a variety of communication problems. A model is introduced that applies to decoding problems, including the iterative decoding of parallel concatenated (turbo) codes, serially concatenated codes, low-density parity-check (LDPC) codes, and repeat–accumulate (RA) codes. EXIT functions are defined using the model, and several properties of such functions are proved for erasure channels. One property expresses the area under an EXIT function in terms of a conditional entropy. A useful consequence of this result is that the design of capacity-approaching codes reduces to a curve-fitting problem for all the aforementioned codes. A second property relates the EXIT function of a code to its Helleseth–Kløve–Levenshtein information functions, and thereby to the support weights of its subcodes. The relation is via a refinement of information functions called split information functions, and via a refinement of support weights called split support weights. Split information functions are used to prove a third property that relates the EXIT function of a linear code to the EXIT function of its dual.**

***Index Terms*—Concatenated codes, duality, error-correction coding, iterative decoding, mutual information.**

I. Introduction

THE seminal paper of Gallager [1, p. 48] suggested to evaluate the convergence behavior of iterative decoders for low-density parity-check (LDPC) codes by tracking the probability distributions of extrinsic log-likelihood ratios (L-values). The procedure is particularly simple for erasure channels (ECs) because one must compute only the fraction of erasures being passed from one component decoder to another. For example, this is done in [2], [3] for irregular LDPC codes. However, for other channels, one must track entire probability density functions. A detailed analysis for such cases is described in [4], [5], where the procedure is called *density evolution*.

Density evolution can be simplified in several ways. First, empirical evidence shows that good EC codes are also good for many practical channels. This motivates designing codes for ECs, and then adapting the design for the actual channel [6, Ch. 6]. A second approach is to track only one number per iteration rather than density functions. For instance, one might track a statistic of the extrinsic L-values based on their mean, variance, an error probability, a fidelity or a mutual information [7]–[17]. We refer to [13, Sec. IV] and [18] for a comparison of some of these tools. We consider tracking a per-letter average mutual information, i.e., we use extrinsic information transfer (EXIT) charts. Several reasons for choosing EXIT charts are as follows.

- Mutual information seems to be the most accurate statistic [13, Sec. IV], [18].
- Mutual information is the most robust statistic, in the sense that it applies without change to the widest range of channels, modulations, and detectors. For instance, EXIT functions apply to ECs without change. They further apply to symbol-based decoders [19] and to suboptimal decoders such as hard-decision decoders.
- EXIT functions have analytic properties that have useful implications for designing codes and iterative processors.

One aim of this paper is to justify the last claim. For example, we prove that if the decoder's *a priori* L-values come from a binary EC (or BEC) then the *area* under an EXIT function is one minus a conditional entropy. This property is used to show that code design for BECs reduces to a curve-fitting problem for several classes of codes including parallel concatenated (PC or turbo) [20], serially concatenated (SC) [21], LDPC, and repeat–accumulate (RA) codes [22], [23]. This fact gives theoretical support for the curve-fitting techniques already being applied in the communications literature, see, e.g., [24]–[30]. The success of these techniques relies on the robustness of EXIT charts: the transfer functions change little when BEC *a priori* L-values are replaced by, e.g., *a priori* L-values generated by transmitting binary phase-shift keying (BPSK) symbols over an additive white Gaussian noise (AWGN) channel. Moreover, the resulting transfer functions continue to predict the convergence behavior of iterative decoders rather accurately.

For the special case of LDPC codes, the area property is related to the *flatness condition* of [31] and has similar implications. For both LDPC and RA codes on a BEC, the curve-fitting technique is known through a polynomial equation [2], [23]. However, the area property applies to many communication problems beyond LDPC or RA decoding. For instance, it applies to problems with PC codes, SC codes, modulators, detectors, and channels with memory.

A second property we prove is that EXIT functions for BECs can be expressed in terms of what we call split information functions and split support weights. The former are refinements of the information functions of a code introduced in [32], while

Manuscript received March 25, 2003; revised March 15, 2004. The material in this paper was presented in part at the Conference on Information Sciences and Systems, Princeton University, Princeton, NJ, March 2002; the IEEE International Symposium on Information Theory, Lausanne, Switzerland, June/July 2002; and the 3rd International Symposium on Turbo Codes, Brest, France, September 2003.

A. Ashikhmin and G. Kramer are with Bell Laboratories, Lucent Technologies, Murray Hill, NJ 07974 USA (e-mail: aea@bell-labs.com; gkr@bell-labs.com).

S. ten Brink was with Bell Laboratories, Lucent Technologies, Crawford Hill, NJ. He is now with Realtek, Irvine, CA 92618 USA (e-mail: stenbrink@realtek-us.com).

Communicated by S. Litsyn, Associate Editor for Coding Theory.

Digital Object Identifier 10.1109/TIT.2004.836693

the latter are refinements of the *weight enumerators* of a code [33]–[35]. The split information functions are used to prove a third property that relates the EXIT function of a linear code to the EXIT function of its dual. As far as we know, these are the first applications of information functions and support weights to *a posteriori* probability (APP) decoding.

This paper is organized as follows. The first part of the paper, comprising Sections II and III, deals with general channels. In Section II, we describe a decoding model that can be used for a wide variety of communication problems. In Section III, we use this model to define EXIT functions and derive general properties of such functions. The second part of the paper deals with BECs. In Section IV, we derive several properties of EXIT functions when the *a priori* L-values are modeled as coming from a BEC. Sections V–VIII show how the area property guides code design. Section IX summarizes our results.

II. Preliminaries

A. EXIT Chart Example

Consider the EXIT chart for an LDPC code. Such a code is often represented by a bipartite graph whose left vertices are called *variable* nodes and whose right vertices are called *check* nodes (see [4]). Suppose the variable and check nodes have degrees 2 and 4, respectively, so that we have a $(2,4)$-regular LDPC code. The code has a design rate of $1/2$ and, as is often done, we assume the code is long and its interleaver has large girth.

Suppose we transmit over a BEC with erasure probability q. A belief-propagation decoder then passes only one of three probabilities: $0, 1$, and $1/2$ (erasure). As we will show, the EXIT functions turn out to be one minus the fraction of erasures being passed from one side of the graph to the other, i.e., the analysis is equivalent to that of [2]. Fig. 1 shows the EXIT functions when $q = 0.3$ and $q = 0.5$. The curve for the check nodes is the one starting at 0 on the I_{Ac} axis, and its functional form is $I_{Ec} = (I_{Ac})^3$. The curve for the variable nodes depends on q and is given by $I_{Ev} = 1 - q(1 - I_{Av})$.

The decoding trajectories are depicted in Fig. 1 by the dashed lines marked with arrows. For instance, when $q = 0.3$, we begin on the I_{Ac} axis at $1-q = 0.7$ and move right to the check-node curve. We then move up to the variable-node curve marked $q = 0.3$, back to the check-node curve, and so forth. The $q = 0.3$ curve does *not* intersect the check-node curve, which means the decoder's per-edge erasure probability can be made to approach zero. We say that there is an open *convergence tunnel* between the curves. In contrast, the $q = 0.5$ curve intersects the check-node curve, which means the decoder gets "stuck." Convergence is, in fact, guaranteed if $q < 1/3$, and $q = 1/3$ is therefore called a *threshold* for this decoder.

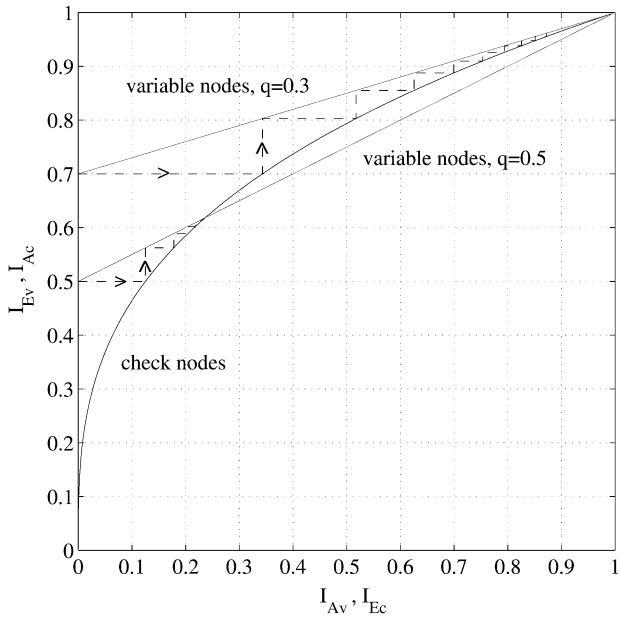

Fig. 1. EXIT chart for a $(2,4)$-regular LDPC code on the BEC.

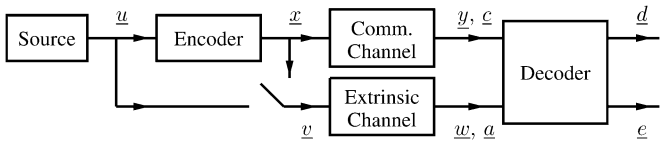

Fig. 2. A decoding model for PC and SC codes.

B. Decoding Model

Consider the decoding model shown in Fig. 2. A binary-symmetric source produces a vector $\underline{u}$ of k independent information bits each taking on the values 0 and 1 with probability $1/2$. An encoder maps $\underline{u}$ to a binary length n codeword $\underline{x}$. We write random variables using upper case letters and their realizations by the corresponding lower case letters. For example, we consider $\underline{u}$ to be a realization of $\underline{U}$. The decoder receives two vectors: a noisy version $\underline{y}$ of $\underline{x}$ and a noisy version $\underline{w}$ of $\underline{v}$, where $\underline{v}$ is either $\underline{u}$ or $\underline{x}$. We call the $\underline{x}$ to $\underline{y}$ channel $P(\underline{y}|\underline{x})$ the *communication channel*, and the $\underline{v}$ to $\underline{w}$ channel $P(\underline{w}|\underline{v})$ the *extrinsic channel*. One might alternatively choose to call the extrinsic channel the *a priori* channel because we use its outputs as if they were *a priori* information. However, we will consider *iterative* decoding where this channel models extrinsic information [36] coming from another decoder rather than true *a priori* information. Either way, the terminology "extrinsic" reminds us that $\underline{w}$ originates from outside the communication channel.

Fig. 2 depicts how we will model the information $(\underline{y}, \underline{w})$ that the component decoders of a PC or SC code receive. For example, suppose we perform iterative decoding for an SC code. The inner decoder receives extrinsic information about the *input* bits of the inner encoder, so we set $\underline{v} = \underline{u}$. The outer encoder, in contrast, receives extrinsic information about the *output* bits of the outer encoder, so we set $\underline{v} = \underline{x}$. In both cases, the extrinsic channel is an artificial device that does not exist. We introduce it only to help us analyze the decoder's operation.

Often both the communication and extrinsic channels are memoryless, but we remark that the area property derived below remains valid when the communication channel has memory. For example, suppose we parse the bits $\underline{x}$ into 4-bit blocks, and map each of these blocks to a 16 quadrature amplitude modulation (16-QAM) symbol. We send these symbols through an AWGN channel and they arrive at the receiver as $\underline{y}$. We view the communication channel as including the parser,

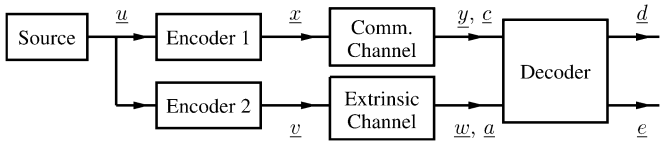

Fig. 3. A decoding model with two encoders.

the 16-QAM mapper, and the AWGN channel. This channel has a 4-bit "block memory" but the area property applies. (Of course, for such channels the property's applicability to iterative decoding is hampered by the fact that the extrinsic channels are not accurately modeled as BECs.) As a second example, suppose we map $\underline{x}$ onto BPSK symbols that are sent over an intersymbol interference (ISI) channel. The communication channel now consists of the BPSK mapper and the ISI channel. We will write C for the capacity of the communication channel.

Fig. 3 depicts another decoding model with *two* encoders. In fact, Fig. 3 includes Fig. 2: let Encoder 1 be the Encoder in Fig. 2, and if $\underline{v} = \underline{u}$ choose Encoder 2 to be the identity mapping, and if $\underline{v} = \underline{x}$ choose Encoder 2 to be Encoder 1. The reason for introducing the second encoder is that, when dealing with LDPC or RA codes, we need to make Encoder 1 the identity mapping and Encoder 2 a repetition code or single parity-check code. This situation is not included in Fig. 2.

An even more general approach is to replace $P(\underline{y}|\underline{x})$ and $P(\underline{w}|\underline{v})$ with a combined channel $P(\underline{w}, \underline{y}|\underline{v}, \underline{x})$. Such a model could be useful for analyzing the effect of dependencies between the channel and *a priori* L-values. For other problems, the vector $\underline{u}$ might have complex entries and Encoder 1 might be a discrete-time linear filter. We will, however, consider only the model of Fig. 3.

Let m be the length of $\underline{v}$, $\underline{w}$, $\underline{a}$, and $\underline{e}$. The decoder uses $\underline{y}$ and $\underline{w}$ to compute two estimates of $\underline{v}$: the *a posteriori* L-values $\underline{d}$ and the *extrinsic* L-values $\underline{e}$. The symbol w_i, $i = 1, 2, \ldots, m$, gives *a priori* information about the random variable V_i with L-value

$$a_i = \log \frac{P(w_i \mid V_i = 0)}{P(w_i \mid V_i = 1)} \tag{1}$$

where $P(w_i \mid V_i = 0)$ is the probability that $W_i = w_i$ conditioned on the event $V_i = 0$. Similarly, for memoryless communication channels, the symbol y_i gives information about the random variable X_i with L-value

$$c_i = \log \frac{P(y_i \mid X_i = 0)}{P(y_i \mid X_i = 1)}. \tag{2}$$

We will use (2) when dealing with PC codes in Section VIII. For simplicity, we assume that all random variables are discrete. Continuous random variables can be treated by replacing certain probabilities by probability density functions.

The decoder we are mainly interested in is the APP decoder [1] that computes the L-values

$$d_i = \log \frac{\Pr(V_i = 0 \mid \underline{y}, \underline{w})}{\Pr(V_i = 1 \mid \underline{y}, \underline{w})} \tag{3}$$

where $\Pr(V_i = 0 \mid \underline{y}, \underline{w})$ is the probability of the event $V_i = 0$ conditioned on $\underline{Y} = \underline{y}$ and $\underline{W} = \underline{w}$. For example, suppose we perform a maximum *a posteriori* probability (MAP) decoding of the bits in $\underline{u}$. The appropriate model is then Fig. 3 (or Fig. 2) with $\underline{v} = \underline{u}$ and an extrinsic channel that is absent or completely noisy.

For further analysis, we write $\underline{v}_{[i]}$ for the vector $\underline{v}$ with the ith entry removed, i.e., $\underline{v}_{[i]} = [v_1, \ldots, v_{i-1}, v_{i+1}, \ldots, v_m]$. We expand the numerator in (3) as

$$\begin{aligned}
&\Pr(V_i = 0 \mid \underline{y}, \underline{w}) \\
&\quad = \sum_{\underline{u}: v_i(\underline{u})=0} P(\underline{u} \mid \underline{y}, \underline{w}) \\
&\quad = \sum_{\underline{u}: v_i(\underline{u})=0} \frac{P(\underline{u}) P(\underline{w} \mid \underline{u}) P(\underline{y} \mid \underline{u}, \underline{w})}{P(\underline{y}, \underline{w})} \\
&\quad = \sum_{\underline{u}: v_i(\underline{u})=0} \frac{P(\underline{u}) P(\underline{w} \mid \underline{v}(\underline{u})) P(\underline{y} \mid \underline{x}(\underline{u}))}{P(\underline{y}, \underline{w})} \\
&\quad = \frac{P(w_i | V_i = 0)}{P(\underline{y}, \underline{w})} \\
&\qquad \cdot \sum_{\underline{u}: v_i(\underline{u})=0} P(\underline{u}) P(\underline{w}_{[i]} | \underline{v}_{[i]}(\underline{u})) P(\underline{y} | \underline{x}(\underline{u}))
\end{aligned} \tag{4}$$

where $\underline{v}(\underline{u})$ and $\underline{x}(\underline{u})$ are vectors corresponding to $\underline{u}$, and where the last step follows if the extrinsic channel is memoryless. Expanding the denominator of (3) in the same way and inserting the result into (3), we have

$$d_i = a_i + e_i \tag{5}$$

where

$$\begin{aligned}
e_i &= \log \frac{\Pr(V_i = 0 \mid \underline{y}, \underline{w}_{[i]})}{\Pr(V_i = 1 \mid \underline{y}, \underline{w}_{[i]})} \\
&= \log \frac{\sum_{\underline{u}: v_i(\underline{u})=0} P(\underline{w}_{[i]} | \underline{v}_{[i]}(\underline{u})) P(\underline{y} \mid \underline{x}(\underline{u}))}{\sum_{\underline{u}: v_i(\underline{u})=1} P(\underline{w}_{[i]} \mid \underline{v}_{[i]}(\underline{u})) P(\underline{y} \mid \underline{x}(\underline{u}))}.
\end{aligned} \tag{6}$$

The value e_i is called the *extrinsic* L-*value* about v_i.

III. EXIT Functions

A. Average Extrinsic Information

An iterative decoder has two or more component decoders that exchange extrinsic L-values. Alternatively, the decoders could exchange extrinsic *probabilities*. The ensuing analysis does not depend on how the reliabilities are represented because we use mutual information.

Continuing, the e_i from one decoder pass through an interleaver and are fed to another decoder as *a priori* L-values $\underline{a}$. We model $\underline{a}$ as being output from a channel as in Fig. 3. We define two quantities (see also [13]–[16]), namely

$$I_A = \frac{1}{m} \sum_{i=1}^{m} I(V_i; A_i) \tag{7}$$

$$I_E = \frac{1}{m} \sum_{i=1}^{m} I(V_i; E_i). \tag{8}$$

As done here, we adopt the notation of [37, Ch. 2] for mutual information and entropies. The value I_A is called the *average a priori information* going into the decoder, and I_E is called

the *average extrinsic information* coming out of the decoder. *An EXIT chart plots I_E as a function of I_A.*

Consider first I_A, and suppose the V_i all have the same distribution, and that the extrinsic channel is memoryless and time invariant. We then have

$$I_A = I(V_1; A_1). \tag{9}$$

We further have $0 \leq I_A \leq 1$ because V_i is binary. We will usually consider codes for which the V_i are uniform and identically distributed, and codes and extrinsic channels for which I_A can take on all values between 0 and 1.

Consider next I_E. Observe from (6) that E_i is a function of $\underline{Y}$ and $\underline{W}_{[i]}$, and that $\underline{W}_{[i]}$ and $\underline{A}_{[i]}$ are interchangeable since one defines the other. This implies (see [38, Sec. 2.10])

$$I(V_i; E_i) \leq I(V_i; \underline{Y}\,\underline{W}_{[i]}) = I(V_i; \underline{Y}\,\underline{A}_{[i]}). \tag{10}$$

We will use $\underline{A}_{[i]}$ rather than $\underline{W}_{[i]}$ simply because $\underline{A}_{[i]}$ better reminds us of the word *a priori*. Some authors prefer to add commas and write $I(V_i; \underline{Y}, \underline{A}_{[i]})$, and similarly for entropies with multiple random variables. However, we will adhere to the notation of [37]. The following proposition shows that the inequality in (10) is in fact an equality for APP decoders with extrinsic message passing.

Proposition 1:

$$I(V_i; E_i) = I(V_i; \underline{Y}\,\underline{A}_{[i]}). \tag{11}$$

Proof: See Appendix A. □

We remark that for non-APP decoders, the average extrinsic information put out by the decoder will usually not satisfy (10) with equality. The import of Proposition 1 is that we need to consider only random variables in *front* of the decoder, i.e., we have

$$I_E = \frac{1}{m}\sum_{i=1}^{m} I(V_i; \underline{Y}\,\underline{A}_{[i]}). \tag{12}$$

B. Examples

Example 1: (Repetition Codes on a BEC) Consider Fig. 2 with a length n repetition code. Suppose the communication and extrinsic channels are BECs with erasure probabilities q and p, respectively. We have $I_A = I(V_1; A_1) = 1 - p$. Furthermore, for the case $\underline{v} = \underline{x}$ we have

$$\begin{aligned} I_E &= \frac{1}{n}\sum_{i=1}^{n} I(X_i; \underline{Y}\,\underline{A}_{[i]}) \\ &= H(X_1) - H(X_1 \mid \underline{Y}\,\underline{A}_{[1]}) \\ &= 1 - q^n p^{n-1} \end{aligned} \tag{13}$$

where the second step follows by the symmetry of the code. We plot I_E versus I_A in Fig. 4 where we have chosen $q = 0.9$ and $n = 2, 3, 4$.

Example 2: (Repetition Code on a BSC) Consider the repetition codes of Example 1, but where the communication and extrinsic channels are binary-symmetric channels (BSCs) with crossover probabilities ϵ and δ, respectively. We now have

$$I_A = I(V_1; A_1) = 1 - h(\delta)$$

where

$$h(x) = -x\log_2(x) - (1-x)\log_2(1-x)$$

is the binary entropy function. For the case $\underline{v} = \underline{x}$ and $n = 2$ we use (12) to compute

$$\begin{aligned} I_E = 1 - \Bigg\{ & [(1-\epsilon)^2(1-\delta) + \epsilon^2\delta]h\left(\frac{\epsilon^2\delta}{(1-\epsilon)^2(1-\delta)+\epsilon^2\delta}\right) \\ & + [(1-\epsilon)^2\delta + \epsilon^2(1-\delta)]h \\ & \cdot \left(\frac{\epsilon^2(1-\delta)}{(1-\epsilon)^2\delta + \epsilon^2(1-\delta)}\right) + 2\epsilon(1-\epsilon)h(\delta)\Bigg\}. \end{aligned} \tag{14}$$

Similar curves can be computed for $n > 2$. We plot I_E in Fig. 4 for $n = 2$ with $\epsilon = 0.316$ (it is the lowest solid curve). We thus have $C = 1 - h(0.316) \approx 0.1$, i.e., C is approximately the same as in Example 2. Observe that the BSC curve is close to but below the BEC curve. Similar observations concerning thresholds were made in [6, Chs. 6 and 7].

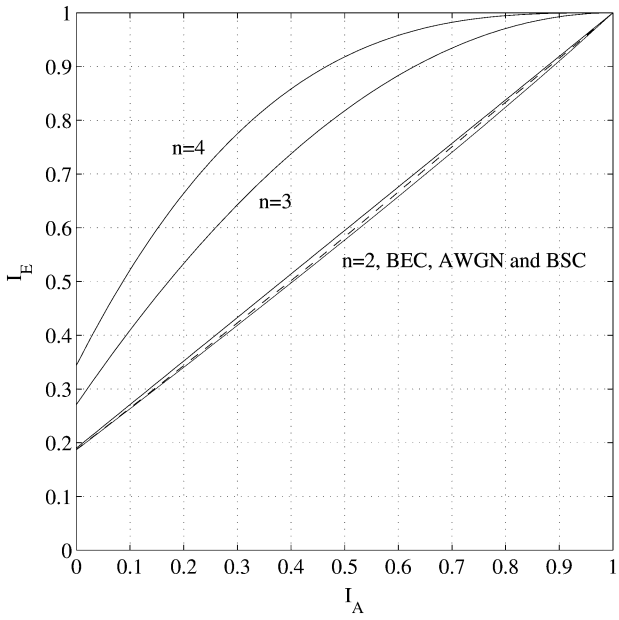

Fig. 4. EXIT chart for repetition codes on BECs (upper three solid lines), a BSC (lower solid line), and BPSK on an AWGN channel (dashed line).

Example 3: (Repetition Code on an AWGN Channel) Consider the repetition codes of Example 1, but where the communication and extrinsic channels are BPSK-input AWGN channels with noise variances σ_{ch}^2 and σ_{ex}^2, respectively. We convert these variances to the variances of their L-values [36], namely, $\widetilde{\sigma}_{ch}^2 = 4/\sigma_{ch}^2$ and $\widetilde{\sigma}_{ex}^2 = 4/\sigma_{ex}^2$, respectively. One can compute $I_A = J(\widetilde{\sigma}_{ex})$ and $I_E = J\left(\sqrt{(n-1)\widetilde{\sigma}_{ex}^2 + n\widetilde{\sigma}_{ch}^2}\right)$, where

$$J(\sigma) = 1 - \int_{-\infty}^{\infty} \frac{e^{-\frac{\left(\xi - \sigma^2/2\right)^2}{2\sigma^2}}}{\sqrt{2\pi\sigma^2}} \cdot \log_2[1 + e^{-\xi}]d\xi. \tag{15}$$

We plot I_E in Fig. 4 for $n = 2$ as the dashed curve, where $\widetilde{\sigma}_{ch} = 0.77$ so that $C = J(\widetilde{\sigma}_{ch}) \approx 0.1$. Observe that the AWGN curve lies between the BEC and BSC curves. Again, a similar result concerning thresholds was found in [6, Chs. 6 and 7]. In fact, recent work has shown that the BEC and BSC curves give upper and lower bounds, respectively, on the EXIT curves for repetition codes on binary-input, symmetric channels [39], [40].

Example 4: Consider LDPC variable nodes of degree d_v. We use the model of Fig. 3 where $\underline{u}$ is one bit, $\underline{x} = \underline{u}$, and Encoder 2 is a length d_v repetition code. We again make the communication and extrinsic channels BECs with erasure probabilities q and p, respectively. We compute

$$\begin{aligned} I_E &= 1 - qp^{d_v - 1} \\ &= 1 - q(1 - I_A)^{d_v - 1}. \end{aligned} \tag{16}$$

Fig. 1 shows two examples of such curves when $d_v = 2$ and $q = 0.3, 0.5$.

Example 5: Consider LDPC check nodes of degree d_c. We use the model of Fig. 3 with $\underline{y} = \underline{0}$ and with Encoder 2 a length d_c single parity-check code. Let the extrinsic channel be a BEC with erasure probability p so that $I_A = 1 - p$ and (12) simplifies to

$$I_E = (1 - p)^{d_c - 1} = (I_A)^{d_c - 1}. \tag{17}$$

An example of such a curve with $d_c = 4$ is plotted in Fig. 1.

Example 6: Consider Example 4 but where $\underline{u} = \underline{x}$ has k bits and Encoder 2 is a length $d_v = k + 1$ single parity-check code. The code with codewords $(\underline{x}, \underline{v})$ is thus a systematic code. We compute

$$\begin{aligned} I_E = \frac{d_v - 1}{d_v} \left[(1 - q) + q(1 - p)(1 - qp)^{d_v - 2} \right] \\ + \frac{1}{d_v} (1 - qp)^{d_v - 1}. \end{aligned} \tag{18}$$

We will use (18) for generalized LDPC codes in Section VII-A.

C. Mixtures of Codes

Suppose we split $\underline{u}$ into several vectors $\underline{u}_j$, $j = 1, 2, \ldots, n_u$, and encode each $\underline{u}_j$ separately. Let $\underline{v}_j$ and $\underline{e}_j$ be those portions of the respective $\underline{v}$ and $\underline{e}$ corresponding to $\underline{u}_j$, and denote the length of $\underline{v}_j$ by ℓ_j. Equation (8) simplifies to

$$I_E = \sum_{j=1}^{n_u} \frac{\ell_j}{m} \left[\frac{1}{\ell_j} \sum_{i=1}^{\ell_j} I(V_{ji}; E_{ji}) \right] = \sum_{j=1}^{n_u} \gamma_j \, I_{Ej} \tag{19}$$

where V_{ji} and E_{ji} are the ith entries of $\underline{V}_j$ and $\underline{E}_j$, respectively, $\gamma_j = \ell_j / m$, and I_{Ej} is the expression in square brackets in (19). Observe that I_{Ej} is simply the average extrinsic information for component code j. Thus, the EXIT function I_E is the average of the component EXIT functions I_{Ej}. This *mixing* property is known and was used in [24]–[28] to improve codes.

Example 7: (Irregular LDPC Codes) An *irregular* LDPC code [2] can be viewed as follows: Encoder 2 in Fig. 3 is a mixture of either repetition codes or single parity-check codes. For example, suppose that 40% and 60% of the edges are connected to degree-2 and degree-3 variables nodes, respectively. Inserting (16) into (19) with $\gamma_1 = 0.4$ and $\gamma_2 = 0.6$, we have

$$I_E = 1 - q(0.4p + 0.6p^2). \tag{20}$$

The γ_j are here the same as the left degree distribution coefficients λ_{j+1} of [2].

IV. Erasure Channel Properties

The rest of this paper is concerned with the special case where the *a priori* symbols $\underline{w}$ are modeled as coming from a BEC with erasure probability p. We derive three results for EXIT functions for such situations. The first, an area property, is valid for any codes and communication channels. The second, an equation showing how to compute EXIT functions via the Helleseth–Kløve–Levenshtein information functions of a code [32], is valid when both the communication and extrinsic channels are BECs. The third, a duality property, relates the EXIT functions of a linear code and its dual.

A. Area Property

We have

$$I_A = I_{A,\max} \cdot (1 - p) \tag{21}$$

where

$$I_{A,\max} = \frac{1}{m} \sum_{i=1}^{m} H(V_i). \tag{22}$$

Let $\mathcal{A} = \int_0^{I_{A,\max}} I_E(I_A) dI_A$ be the area under the EXIT function. We have the following result.

Theorem 1: If the extrinsic channel (i.e., *a priori* channel) is a BEC, then for any codes (linear or not) and any communication channel (memoryless or not) we have

$$\mathcal{A} = I_{A,\max}^2 \left[1 - \frac{H(\underline{V}|\underline{Y})}{\sum_{i=1}^{m} H(V_i)} \right]. \tag{23}$$

Proof: See Appendix B. □

Note that

$$0 \le H(\underline{V}|\underline{Y}) \le H(\underline{V}) \le \sum_{i=1}^{m} H(V_i) \le m$$

which implies $0 \le \mathcal{A} \le I_{A,\max}^2 \le 1$. We will consider primarily cases where $H(V_i) = 1$ for all i so that $I_{A,\max} = 1$. For instance, suppose that Encoder 2 is linear and has no idle components, i.e., Encoder 2's generator matrix has no all-zeros columns. This implies $H(V_i) = 1$ for all i so that (23) becomes

$$\mathcal{A} = 1 - \frac{1}{m} H(\underline{V}|\underline{Y}). \tag{24}$$

TABLE I
EXIT FUNCTIONS FOR SYSTEMATIC LINEAR BLOCK CODES

Code with generator matrix $G = [I\ P]$	EXIT Function	$\mathcal{A} = 1 - R$
n=6, k=2, $P = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \end{bmatrix}$	$I_{E1}(p) = 1 - 2p^3 + p^5$	2/3
n=4, k=2, $P = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}$	$I_{E2}(p) = 1 - \frac{1}{2}p - \frac{3}{2}p^2 + p^3$	1/2
n=8, k=4, $P = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix}$	$I_{E3}(p) = 1 - 7p^3 + 21p^5 - 21p^6 + 6p^7$	1/2
n=3, k=2, $P = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$	$I_{E4}(p) = (1-p)^2$	1/3

Furthermore, if both Encoders 1 and 2 are one-to-one (invertible) mappings then we can interchange $\underline{U}$, $\underline{V}$, and $\underline{X}$. For example, we can write (24) as

$$\mathcal{A} = 1 - \frac{1}{m} H(\underline{X}|\underline{Y}). \tag{25}$$

An important consequence of Theorem 1 is that it restricts the form of the EXIT function. Moreover, one can sometimes relate $\mathcal{A}$ to the code *rate*, as shown in several examples below.

We remark that we will use two definitions of rate interchangeably: $R = k/n$ and $R = H(\underline{V})/n$. In most cases, these two rates are identical, but if Encoder 2 is not a one-to-one mapping then the second rate is smaller than the first. We will ignore this subtlety and assume that $H(\underline{V}) = k$.

Example 8: (Scramblers) Suppose Encoder 1 in Fig. 3 is a *filter* or *scrambler*, i.e., it is a rate *one* code. Suppose further that Encoder 2 is the identity mapping and the communication channel is *memoryless*. We then have $H(\underline{X}|\underline{Y}) = k \cdot H(X_1|Y_1)$ and $m = k$ so that (25) gives

$$\mathcal{A} = I(X_1; Y_1). \tag{26}$$

The area is therefore C if independent and uniformly distributed binary inputs maximize (26). This result was discovered in [41] and motivated Theorem 1.

Example 9: (No Communication Channel) Suppose there is no communication channel. Such a situation occurs for the *outer* decoder of an SC code for which we also have $m = n$. The area (24) is thus

$$\mathcal{A} = 1 - H(\underline{V})/n = 1 - R. \tag{27}$$

Several EXIT functions for this situation are given in Table I, where we write $I_E(p)$ to mean the average extrinsic information when $I_A = 1 - p$. The functions are plotted in Fig. 5. We will show that (27) has important consequences for code design.

Example 10: Consider Fig. 2 with $\underline{v} = \underline{u}$. This situation fits the decoding of the *inner* code of an SC code (Section V) and the component codes of a PC code (Section VIII). We have $m = k$, $H(\underline{X}) = k$, and

$$\mathcal{A} = 1 - H(\underline{X}|\underline{Y})/k = \frac{I(\underline{X};\underline{Y})/n}{R}. \tag{28}$$

Observe that, by definition, we have $I(\underline{X};\underline{Y})/n \le C$.

Example 11: Consider Fig. 2 with $\underline{v} = \underline{x}$. We have $m = n$ and (25) becomes

$$\mathcal{A} = 1 - H(\underline{X}|\underline{Y})/n = I(\underline{X};\underline{Y})/n + (1 - R). \tag{29}$$

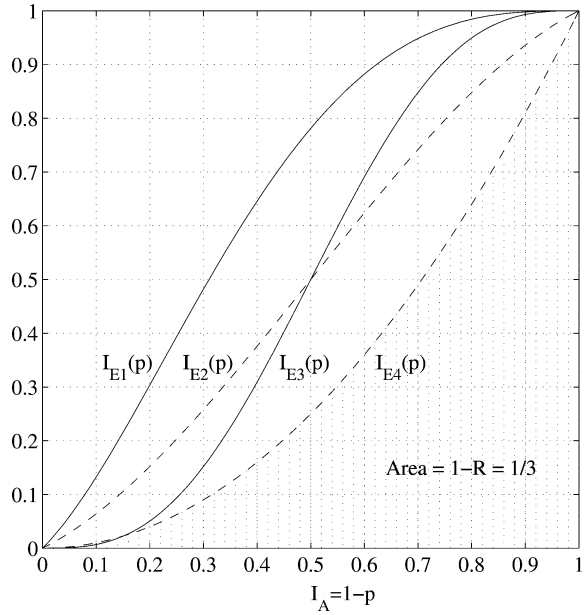

Fig. 5. EXIT chart for the systematic linear block codes of Table I.

Example 12: Consider the LDPC variable nodes of Example 4. We have $m = d_v$, so (25) becomes

$$\mathcal{A} = 1 - \frac{1 - I(X_1; Y_1)}{d_v}. \tag{30}$$

Example 13: Consider the LDPC check nodes of Example 5. There is no communication channel, so we apply (27) with $m = d_c$ and $H(\underline{V}) = d_c - 1$ to obtain

$$\mathcal{A} = 1/d_c. \tag{31}$$

B. EXIT and Information Functions of a Code

The information function in h positions of a code $\mathcal{C}$ was defined in [32] as the average amount of information in h positions of $\mathcal{C}$. More precisely, let n be the code length and $\mathcal{S}_h$ be the set of all subsets of $\{1, 2, \ldots, n\}$ of size h. Let $\mathcal{S} \in \mathcal{S}_h$ with $\mathcal{S} = \{i_1, i_2, \ldots, i_h\}$. We write

$$\underline{x}_{\mathcal{S}} = [x_{i_1}, x_{i_2}, \ldots, x_{i_h}]$$
$$\mathcal{C}_{\mathcal{S}} = \{\underline{x}_{\mathcal{S}} : \underline{x} \in \mathcal{C}\}.$$

Let $\mathcal{C}$ be a linear code and $k_{\mathcal{S}}$ the dimension of $\mathcal{C}_{\mathcal{S}}$. The information function in h positions of $\mathcal{C}$ is

$$e_h = \frac{1}{\binom{n}{h}} \sum_{\mathcal{S} \in \mathcal{S}_h} k_{\mathcal{S}}. \tag{32}$$

We write the unnormalized version of e_h as

$$\tilde{e}_h = \sum_{\mathcal{S}\in\mathcal{S}_h} k_{\mathcal{S}}. \tag{33}$$

We remark that the above definitions and the following theory can be extended to nonlinear codes (cf. [32]).

Consider the following simple generalization of e_h. Let $\mathcal{C}$ be the code formed by all pairs $(\underline{v},\underline{x})$ in Fig. 3. Suppose Encoders 1 and 2 are linear, and that $\mathcal{C}$ is a linear $[m+n,k]$ code (which means that $\mathcal{C}$ has dimension k). Let $\mathcal{S}_{g,h}$ be the set of all subsets of $\{1,2,\ldots,m+n\}$ of the form $\{i_1,i_2,\ldots,i_g,j_1,j_2,\ldots,j_h\}$ where

$$1\le i_1<i_2<\cdots i_g\le m$$
$$m+1\le j_1<j_2<\cdots j_h\le m+n.$$

In other words, $\mathcal{S}_{g,h}$ is the set of subsets of g positions from the first m positions of $\mathcal{C}$, and h positions from the last n positions of $\mathcal{C}$. We define the split information function in (g,h) positions of $\mathcal{C}$ as

$$e_{g,h} = \frac{1}{\binom{m}{g}}\frac{1}{\binom{n}{h}}\sum_{\mathcal{S}\in\mathcal{S}_{g,h}} k_{\mathcal{S}}. \tag{34}$$

We write the unnormalized version of $e_{g,h}$ as

$$\tilde{e}_{g,h} = \sum_{\mathcal{S}\in\mathcal{S}_{g,h}} k_{\mathcal{S}}. \tag{35}$$

We remark that $\tilde{e}_{0,h}$ is the information function of Encoder 1, and $\tilde{e}_{g,0}$ is the information function of Encoder 2.

The following theorem shows that EXIT functions on a BEC can be computed from the split information functions. We write $I_E(p)$ for the average extrinsic information when $I_A=1-p$ and there is no communication channel, and $I_E(p,q)$ when $I_A=1-p$ and $C=1-q$.

Theorem 2: If the extrinsic and communication channels are BECs with respective erasure probabilities p and q, and if Encoders 1 and 2 are linear with no idle components, then we have

$$I_E(p,q) = 1-\frac{1}{m}\sum_{h=0}^{n}(1-q)^h q^{n-h}\sum_{g=1}^{m}(1-p)^{g-1}p^{m-g} \cdot[g\cdot\tilde{e}_{g,h}-(m-g+1)\cdot\tilde{e}_{g-1,h}]. \tag{36}$$

Proof: See Appendix C. □

Theorem 2 can be used to prove Theorem 1 for BEC communication channels: integrating (36) with (108), we have

$$\begin{aligned}\int_0^1 I_E(p,q)dp &= 1-\frac{1}{m}\sum_{h=0}^{n}(1-q)^h q^{n-h}[\tilde{e}_{m,h}-\tilde{e}_{0,h}]\\ &= 1-\frac{1}{m}\left[H(\underline{V}\,\underline{Y})-H(\underline{Y})\right]\\ &= 1-\frac{1}{m}H(\underline{V}|\underline{Y}).\end{aligned} \tag{37}$$

Example 14: (Repetition Codes) Consider the repetition codes of Example 1. We compute

$$\tilde{e}_{g,h} = \begin{cases} 0, & \text{if } g=h=0\\ \binom{m}{g}\binom{n}{h}, & \text{else}\end{cases} \tag{38}$$

so that in (36) we have

$$g\cdot\tilde{e}_{g,h}-(m-g+1)\cdot\tilde{e}_{g-1,h} = \begin{cases} n, & \text{if } g=1,h=0\\ 0, & \text{else.}\end{cases} \tag{39}$$

The resulting EXIT curve coomputed from (36) is (13).

Example 15: (No Communication Channel) Suppose $n=0$ (see Example 9), in which case we have

$$I_E(p) = 1-\frac{1}{m}\sum_{g=1}^{m}(1-p)^{g-1}p^{m-g} \cdot[g\cdot\tilde{e}_g-(m-g+1)\cdot\tilde{e}_{g-1}]. \tag{40}$$

Example 16: (MAP Decoding) Recall from Section II-B that MAP decoding of the bits in $\underline{u}$ corresponds to Fig. 2 with $\underline{v}=\underline{u}$ and $p=1$, i.e., there is no *a priori* information. We have

$$I_E(1,q) = 1-\frac{1}{m}\sum_{h=0}^{n}(1-q)^h q^{n-h}[\tilde{e}_{1,h}-k\cdot\tilde{e}_h]. \tag{41}$$

Note that the decoder's average erasure probability $\overline{P}_e$ is simply $1-I_E(1,q)$. For instance, suppose we transmit using an $[n,k]$ maximum distance separable (MDS) code [42, Ch. 11]. Any h positions of such a code have rank h for $h\le k$, and rank k for $h>k$. This implies

$$\tilde{e}_h = \begin{cases}\binom{n}{h}h, & \text{if } h<k\\ \binom{n}{h}k, & \text{if } h\ge k\end{cases} \tag{42}$$

and

$$\tilde{e}_{1,h} = \begin{cases}\binom{n}{h}k\left[\frac{h}{n}h+\left(1-\frac{h}{n}\right)(h+1)\right], & \text{if } h<k\\ \binom{n}{h}k^2, & \text{if } h\ge k.\end{cases} \tag{43}$$

Inserting (42) and (43) into (41), we obtain

$$\overline{P}_e = \sum_{h=0}^{k-1}(1-q)^h q^{n-h}\binom{n}{h}\left(1-\frac{h}{n}\right). \tag{44}$$

We point out that only few binary MDS codes exist. However, most of the theory developed above can be extended to b-ary sources and codes. For example, suppose $\underline{u}$ is a b-ary vector, Encoder 1 is an $[n,k]$ Reed–Solomon code over GF(b), and $\underline{v}=\underline{u}$. Reed–Solomon codes are MDS and the average symbol erasure probability turns out to be precisely (44).

C. EXIT Functions and Support Weights

The information functions of a code are known to be related to the *support weights* of its subcodes [32]. The support weight $w(\mathcal{C})$ of a code $\mathcal{C}$ is the number of positions where *not* all codewords of $\mathcal{C}$ are zero. For example, the code

$$\mathcal{C} = \{[0\,0\,0],[1\,0\,0],[0\,1\,0],[1\,1\,0]\} \tag{45}$$

has $w(\mathcal{C})=2$. The *rth support weight* A_i^r of $\mathcal{C}$ is the number of unique subspaces of $\mathcal{C}$ of dimension r and support weight i. For example, the code (45) has

$$\begin{aligned}&A_0^0=1\\ &A_0^1=0,\quad A_1^1=2,\quad A_2^1=1,\quad A_3^1=0\\ &A_0^2=0,\quad A_1^2=0,\quad A_2^2=1,\quad A_3^2=0.\end{aligned} \tag{46}$$

The sequence $A_0^0, A_1^1, A_2^1, A_3^1, \ldots, A_n^1$ is the usual weight distribution of a code. We also have $A_i^r = 0$ for $i < r$, and we write $A_i^r = 0$ for $r < 0$.

The numbers A_i^r have been investigated in [34], [35], and more recently in [43]–[48]. For instance, it is known that $\tilde{e}_h$ can be written in terms of the A_i^r as follows (see [32, Theorem 5]):

$$\tilde{e}_h = \sum_{r=1}^{h} r \sum_{s=0}^{r} (-1)^s 2^{\binom{s}{2}} \begin{bmatrix} k-r+s \\ s \end{bmatrix} \cdot \sum_{i=0}^{n} \binom{n-i}{h} A_i^{k-r+s} \quad (47)$$

where for all i we define $\binom{i}{0} = 1$ and $\begin{bmatrix} i \\ 0 \end{bmatrix} = 1$, and for $j > 0$ we write

$$\binom{i}{j} = \prod_{\ell=0}^{j-1} \frac{i-\ell}{j-\ell}, \quad \begin{bmatrix} i \\ j \end{bmatrix} = \prod_{\ell=0}^{j-1} \frac{2^i - 2^\ell}{2^j - 2^\ell}.$$

The numbers $\begin{bmatrix} i \\ j \end{bmatrix}$ are known as Gaussian binomial coefficients [42, p. 443]. The A_i^r have been determined for some codes. For instance, the $[2^k - 1, k]$ simplex code has (see [32, Sec. IV])

$$A_i^r = \begin{cases} \begin{bmatrix} k \\ r \end{bmatrix}, & \text{if } i = 2^k - 2^{k-r} \text{ for } 0 \le r \le k \\ 0, & \text{else.} \end{cases} \quad (48)$$

Inserting (48) into (47), and performing manipulations, we have (see [32, Sec. IV])

$$\tilde{e}_h = \sum_{r=1}^{h} r \sum_{s=0}^{r-1} (-1)^s 2^{\binom{s}{2}} \begin{bmatrix} k \\ r \end{bmatrix} \begin{bmatrix} r \\ s \end{bmatrix} \binom{2^{r-s}-1}{h}. \quad (49)$$

Example 17: (Simplex Code with $k = 2$) The $[3, 2]$ simplex code is a single parity-check code. Equation (49) yields

$$[\tilde{e}_1, \tilde{e}_2, \tilde{e}_3] = [3, 6, 2]$$
$$[e_1, e_2, e_3] = [1, 2, 2]. \quad (50)$$

Inserting (50) into (40), we have $I_E(p) = (1-p)^2$.

Example 18: (Simplex Code with $k = 3$) The $[7, 3]$ simplex code has

$$[\tilde{e}_1, \tilde{e}_2, \ldots, \tilde{e}_7] = [7, 42, 98, 105, 63, 21, 3]$$
$$[e_1, e_2, \ldots, e_7] = [1, 2, 2.8, 3, 3, 3, 3]. \quad (51)$$

Inserting (51) into (40), we have

$$I_E(p) = 1 - \frac{1}{7}\left[7p^6 + 42\overline{p}p^5 + 84\overline{p}^2p^4 + 28\overline{p}^3p^3\right] \quad (52)$$

where $\overline{p} = 1 - p$. This curve is plotted in Fig. 6 as the solid line, where $I_A = 1 - p$.

Example 19: (Uncoded Transmission) Consider the uncoded transmission of k bits. We use [35, eq. (7)] to compute

$$A_i^r = \binom{k}{i} \sum_{s=0}^{i} (-1)^s \begin{bmatrix} i-s \\ r \end{bmatrix} \binom{i}{s}. \quad (53)$$

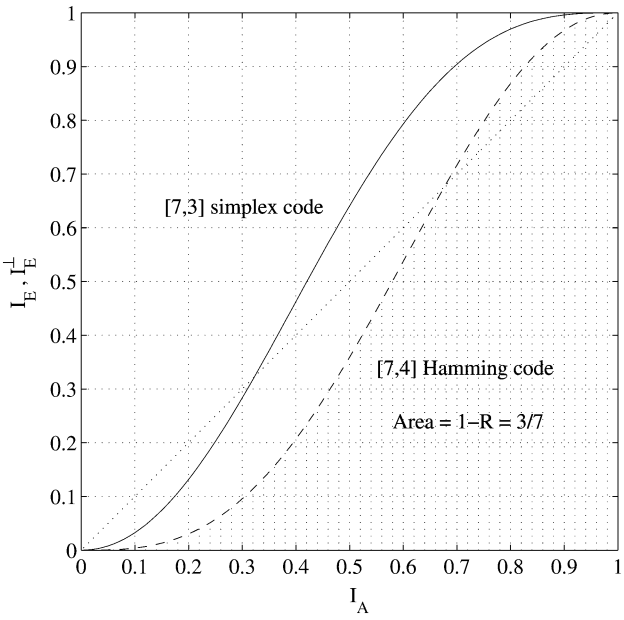

Fig. 6. EXIT chart for the $[7, 3]$ simplex code and its dual.

For example, for $k = 3$ we have

$$\begin{aligned} &A_0^0 = 1 \\ &A_0^1 = 0, \quad A_1^1 = 3, \quad A_2^1 = 3, \quad A_3^1 = 1 \\ &A_0^2 = 0, \quad A_1^2 = 0, \quad A_2^2 = 3, \quad A_3^2 = 4 \\ &A_0^3 = 0, \quad A_1^3 = 0, \quad A_2^3 = 0, \quad A_3^3 = 1 \end{aligned} \quad (54)$$

and $\tilde{e}_0 = 0$, $\tilde{e}_1 = 3$, $\tilde{e}_2 = 6$, $\tilde{e}_3 = 3$. In fact, we easily compute

$$\tilde{e}_h = \binom{k}{h} h \quad (55)$$

for general k. This shows that it can be simpler to compute $\tilde{e}_h$ directly rather than through (47).

D. Split Support Weights

The fact that $\tilde{e}_h$ can be expressed in terms of the A_i^r motivates the question whether $\tilde{e}_{g,h}$ can be written in terms of appropriate generalizations of the A_i^r. This is indeed possible, as we proceed to show.

Consider again the linear code $\mathcal{C}$ formed by all pairs $(\underline{v}, \underline{x})$ in Fig. 3. We define the *split support weights* $A_{i,j}^r$ of $\mathcal{C}$ as the number of unique subspaces of $\mathcal{C}$ that have dimension r, support weight i in the first m positions of $\mathcal{C}$, and support weight j in the last n positions of $\mathcal{C}$. We have the following generalization of (47).

Theorem 3:

$$\tilde{e}_{g,h} = \sum_{r=1}^{g+h} r \sum_{s=0}^{r} (-1)^s 2^{\binom{s}{2}} \begin{bmatrix} k-r+s \\ s \end{bmatrix} \cdot \sum_{i=0}^{m} \sum_{j=0}^{n} \binom{m-i}{g} \binom{n-j}{h} A_{i,j}^{k-r+s}. \quad (56)$$

Proof: See Appendix D for a sketch of the proof. □

We remark that k in (56) could be larger than the dimensions of the code for Encoders 1 and 2. Thus, it is not immediately clear how to relate $A^r_{i,j}$ and the support weights for Encoders 1 and 2.

Example 20: (Input–Output Weight Enumerators) Suppose Encoder 2 is the identity mapping. The $A^1_{i,j}$ are then the $A_{i,j}$ used in [21], [49] to determine the input–output weight enumerator function for the code generated by Encoder 1.

Example 21: (Simplex Code) Consider a systematic generator matrix for the $[2^k-1,k]$ simplex code, and suppose that Encoder 1 transmits the k systematic bits while Encoder 2 transmits the 2^k-k-1 redundant bits. We have $m=2^k-k-1$ and $n=k$, and compute

$$A^r_{i,j}=\begin{cases} A^r_j \text{ of (53)}, & \text{if } i=2^k-2^{k-r}-j \\ 0, & \text{else.}\end{cases} \tag{57}$$

For instance, for $k=3$ we use (57) in (56) to obtain

$$\begin{array}{llll} \tilde{e}_{0,0}=0, & \tilde{e}_{0,1}=3, & \tilde{e}_{0,2}=6, & \tilde{e}_{0,3}=3 \\ \tilde{e}_{1,0}=4, & \tilde{e}_{1,1}=24, & \tilde{e}_{1,2}=33, & \tilde{e}_{1,3}=12 \\ \tilde{e}_{2,0}=12, & \tilde{e}_{2,1}=51, & \tilde{e}_{2,2}=54, & \tilde{e}_{2,3}=18 \\ \tilde{e}_{3,0}=11, & \tilde{e}_{3,1}=36, & \tilde{e}_{3,2}=36, & \tilde{e}_{3,3}=12 \\ \tilde{e}_{4,0}=3, & \tilde{e}_{4,1}=9, & \tilde{e}_{4,2}=9, & \tilde{e}_{4,3}=3. \end{array} \tag{58}$$

Example 22: (Identity and Simplex Code) Suppose Encoder 1 transmits the k information bits directly, and Encoder 2 generates the $[2^k-1,k]$ simplex code. We have $m=2^k-1$ and $n=k$, and compute

$$A^r_{i,j}=\begin{cases} A^r_j \text{ of (53)}, & \text{if } i=2^k-2^{k-r} \\ 0, & \text{else.}\end{cases} \tag{59}$$

For example, for $k=3$ we use (59) in (56) to obtain

$$\begin{array}{llll} \tilde{e}_{0,0}=0, & \tilde{e}_{0,1}=3, & \tilde{e}_{0,2}=6, & \tilde{e}_{0,3}=3 \\ \tilde{e}_{1,0}=7, & \tilde{e}_{1,1}=39, & \tilde{e}_{1,2}=54, & \tilde{e}_{1,3}=21 \\ \tilde{e}_{2,0}=42, & \tilde{e}_{2,1}=162, & \tilde{e}_{2,2}=180, & \tilde{e}_{2,3}=63 \\ \tilde{e}_{3,0}=98, & \tilde{e}_{3,1}=306, & \tilde{e}_{3,2}=312, & \tilde{e}_{3,3}=105 \\ \tilde{e}_{4,0}=105, & \tilde{e}_{4,1}=315, & \tilde{e}_{4,2}=315, & \tilde{e}_{4,3}=105 \\ \tilde{e}_{5,0}=63, & \tilde{e}_{5,1}=189, & \tilde{e}_{5,2}=189, & \tilde{e}_{5,3}=63 \\ \tilde{e}_{6,0}=21, & \tilde{e}_{6,1}=63, & \tilde{e}_{6,2}=63, & \tilde{e}_{6,3}=21 \\ \tilde{e}_{7,0}=3, & \tilde{e}_{7,1}=9, & \tilde{e}_{7,2}=9, & \tilde{e}_{7,3}=3. \end{array} \tag{60}$$

Note that $\tilde{e}_{g,0}$ is the $\tilde{e}_g$ of (51). We use (60) in (36) to compute the EXIT curve to be

$$\begin{aligned} I_E(p,q)=1-\frac{1}{7} & \Big\{ q^3\left[\,7p^6+42\overline{p}p^5+84\overline{p}^2p^4+28\overline{p}^3p^3\,\right] \\ & +\overline{q}q^2\left[\,18p^6+90\overline{p}p^5+108\overline{p}^2p^4+36\overline{p}^3p^3\,\right] \\ & +\overline{q}^2q\left[\,12p^6+36\overline{p}p^5+36\overline{p}^2p^4+12\overline{p}^3p^3\,\right]\Big\} \end{aligned} \tag{61}$$

where $\overline{p}=1-p$ and $\overline{q}=1-q$. This curve is plotted for various q in Fig. 7 as the solid lines. We recover (52) by using $q=1$.

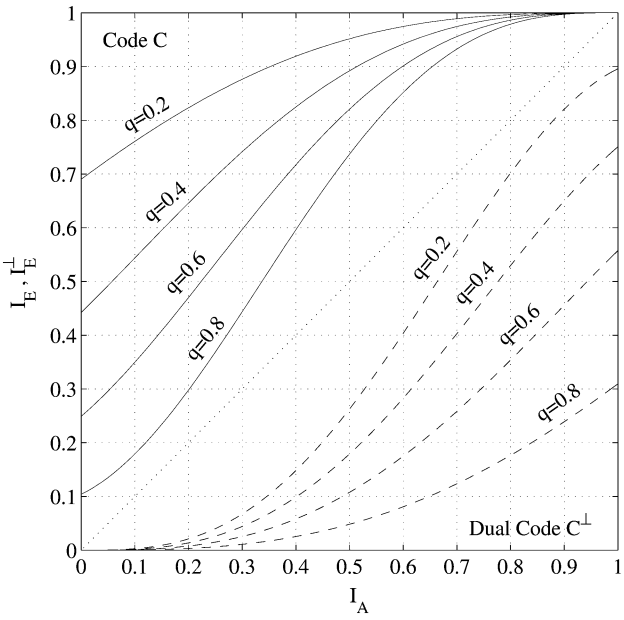

Fig. 7. EXIT chart for Example 22 (solid lines) and Example 25 (dashed lines).

E. Duality Property

Consider the linear code $\mathcal{C}$ formed by all pairs $(\underline{v},\underline{x})$ in Fig. 3. Let $I^{\perp}_E(\cdot)$ be the EXIT function of the dual code $\mathcal{C}^{\perp}$ of $\mathcal{C}$, i.e., $\mathcal{C}^{\perp}$ is an $[m+n,m+n-k]$ code. We have the following result.

Theorem 4: If the extrinsic and communication channels are BECs with respective erasure probabilities p and q, then we have

$$I^{\perp}_E(p,q)=1-I_E(1-p,1-q). \tag{62}$$

Proof: See Appendix E. □

If there is no communication channel $(n=0)$, then we have

$$I^{\perp}_E(p)=1-I_E(1-p). \tag{63}$$

Example 23: (LDPC Check Nodes) Consider the LDPC check nodes of Example 5 for which there is no communication channel. The duals of single parity-check codes are repetition codes, and Example 4 with $q=1$ tells us that the repetition code curve is $I_E(p)=1-p^{n-1}$. We apply (63) and obtain

$$I^{\perp}_E(p)=(1-p)^{n-1}. \tag{64}$$

This duality result was described in [6, Lemma 7.7]. We remark that (63) with p replaced by I_A is rather accurate for channels such as AWGN channels (see [60]). The use of (63) in this way is called a "reciprocal channel approximation" in [6, Lemma 7.7]. This approximation was used to design good codes in [27], [30]. The more general (62) was used in the same way for RA codes in [28].

Example 24: (Hamming Codes) The dual of a simplex code is a Hamming code [42, p. 30]. Consider the $[7,4]$ Hamming code for which (52) and Theorem 4 give

$$I_E(p)=\frac{1}{7}\left[7\overline{p}^6+42p\overline{p}^5+84p^2\overline{p}^4+28p^3\overline{p}^3\right]. \tag{65}$$

The curve (65) is depicted in Fig. 6 as the dashed line.

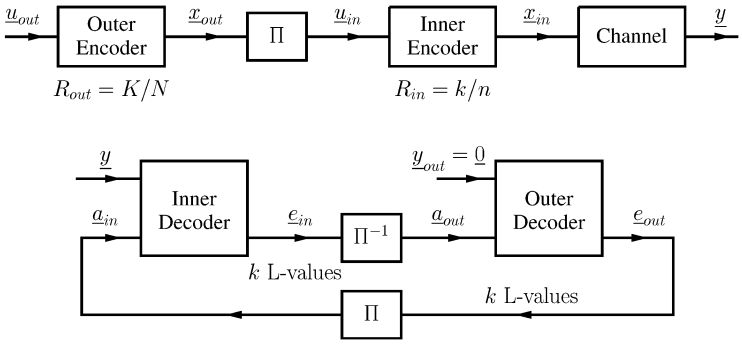

Fig. 8. Encoder and iterative decoder for an SC code.

Example 25: Consider Example 22 and the code $\mathcal{C}$ formed by all pairs $(\underline{v}, \underline{x})$. A generator matrix for $\mathcal{C}$ is

$$G = [I_k \mid P \parallel I_k] \tag{66}$$

where I_k is the $k \times k$ identity matrix, and P is the $k \times (2^k - k - 1)$ parity-check matrix of the simplex code. A generator matrix for $\mathcal{C}^{\perp}$ is

$$H = \left[\begin{array}{c|c||c} P^T & I_{2^k-k-1} & 0_{(2^k-k-1)} \\ I_k & 0_{k\times(2^k-k-1)} & I_k \end{array}\right] \tag{67}$$

where P^T is the transpose of P, and $0_{k\times\ell}$ is the $k \times \ell$ all-zeros matrix. Thus, the situation for the dual code $\mathcal{C}^{\perp}$ is that Encoder 1 transmits k out of the $n = 2^k - 1$ information bits, while the generator matrix of Encoder 2 is the first of $2^k - 1$ columns of (67). The corresponding EXIT function can be computed using Example 22 and (62). For instance, for $k = 3$, we use (61) and (62) to plot the EXIT functions shown in Fig. 7 as dashed lines.

V. EXIT for SC Codes

An SC code has an $[N,\ K]$ *outer code* and an $[n,\ k]$ *inner code* with $k = N$ (see Fig. 8). The outer encoder maps K information bits $\underline{u}_{\rm out}$ to N coded bits $\underline{x}_{\rm out}$. An interleaver permutes the bits in $\underline{x}_{\rm out}$ to $\underline{u}_{\rm in}$, and the inner encoder maps $\underline{u}_{\rm in}$ to the length n vector $\underline{x}_{\rm in}$. The overall rate of the code is $R = R_{\rm out}R_{\rm in} = K/n$ where $R_{\rm out} = K/N = K/k$ and $R_{\rm in} = k/n$. We consider only the case where both codes are linear.

An iterative decoder for the SC code is shown in Fig. 8. Consider first the outer decoder for which we use the model of Fig. 2 with $\underline{v} = \underline{x} = \underline{x}_{\rm out}$ and $\underline{y} = \underline{0}$. Example 9 tells us that for a BEC we have

$$\mathcal{A}_{\rm out} = 1 - R_{\rm out}. \tag{68}$$

Next, for the inner decoder we use Fig. 2 with $\underline{v} = \underline{u}_{\rm in}$. Example 10 tells us that

$$\mathcal{A}_{\rm in} = \frac{I(\underline{X};\underline{Y})/n}{R_{\rm in}}. \tag{69}$$

We will use these area results to interpret how to design SC codes, and later RA, LDPC, and PC codes. Before proceeding, however, we caution that the interpretations rely on the accuracy of the model in Fig. 3. In particular, we *assume* that $\underline{u}$ has independent and uniformly distributed bits, and that the extrinsic channel is a BEC, i.e., the extrinsic channel is memoryless *and* time invariant. These assumptions are not necessarily valid in practice even if the communication channel is a BEC, e.g., for finite-length codes and many decoder iterations. A common approach to enable analysis is to study *ensembles* of codes rather than specific codes [1]–[4], [50]. We will not use this approach, and instead consider only sufficiently long codes and interleavers for which the EXIT analysis becomes exact.

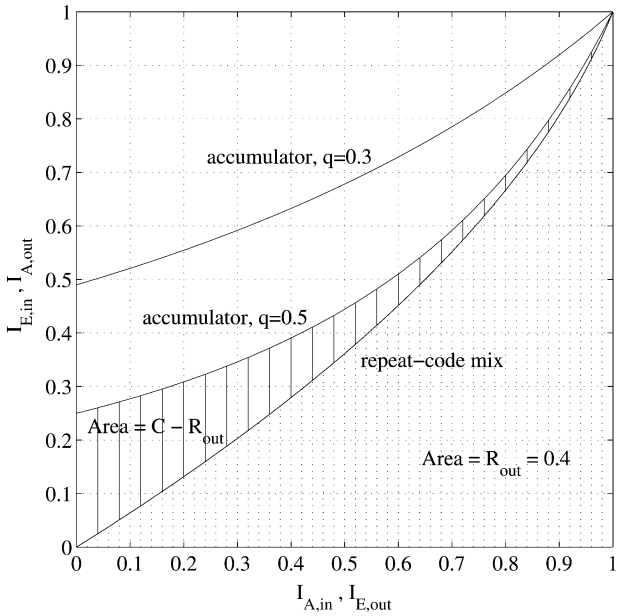

Fig. 9. EXIT chart for an RA code on the BEC.

Equations (68) and (69) have the following implications. For successful decoding, the outer code EXIT curve must lie above the inner code EXIT curve. For SC codes this implies $1 - \mathcal{A}_{\rm out} < \mathcal{A}_{\rm in}$ or, using (68) and (69)

$$R_{\rm out}R_{\rm in} < I(\underline{X};\underline{Y})/n \le C. \tag{70}$$

Thus, we get the satisfying result that the overall rate must be less than capacity for successful decoding [51]. However, (68) and (69) say more.

First, suppose that $1 - \mathcal{A}_{\rm out} = \gamma\mathcal{A}_{\rm in}$ for some γ satisfying $0 \le \gamma < 1$. We have

$$R_{\rm out}R_{\rm in} = \gamma I(\underline{X};\underline{Y})/n \le \gamma C \tag{71}$$

i.e., any area gap between the outer and inner code EXIT curves implies a rate loss compared to C. In fact, if the inner code is a scrambler (i.e., a rate one code as in Example 8) and $I(\underline{X};\underline{Y})/n = C$, then $\mathcal{A}_{\rm in} = C$ and the area between the curves is *exactly* the rate loss $C - R$ (cf. Example 26 and Fig. 9).

The bound (71) means that the design of capacity-approaching codes reduces to a *curve-fitting* problem, i.e., one must match the outer code curve exactly to the inner code curve. Moreover, the smaller the area between the curves, the larger the number of iterations needed to achieve a desired per-edge erasure probability. The EXIT chart thus shows graphically how the decoding complexity (in terms of the number of iterations) increases as one approaches capacity.

Second, if the inner code has rate less than one then we have $I(\underline{X};\underline{Y})/n < C$. That is, any inner code with $R_{\text{in}} < 1$ has an inherent capacity loss that the outer code cannot recover. Of course, *strong* codes with $R_{\text{in}} < 1$ do have $I(\underline{X};\underline{Y})/n \approx C$. However, *iterative* decoding is attractive just because powerful codes can be built from easily decodable component codes. Such component codes with $R_{\text{in}} < 1$ will inevitably leave significant gaps between $I(\underline{X};\underline{Y})/n$ and C.

For example, suppose that the inner code is a length n repetition code and that the communication channel is a BEC with erasure probability q, $0 \le q < 1$. We have

$$I(\underline{X};\underline{Y})/n = (1-q^n)/n \le C \tag{72}$$

with equality if and only if $n = 1$. Thus, repetition codes are a particularly poor choice as inner codes when iteratively decoding. Similar large gaps to capacity for certain convolutional codes are computed in Section VIII. This discussion suggests that it is a good idea to use a rate one inner code when iteratively decoding. In fact, an inner code with rate larger than one will also do, as shown in Example 26.

Example 26: (RA Codes on a BEC) An RA code [22] has an accumulator (a differential encoder) as the inner code and a mixture of repetition codes as the outer code. One can further *puncture* the accumulator output by placing a layer of check nodes between the interleaver and accumulator, as was done in [23]. Suppose the check nodes all have a edges going into the interleaver, and that the communication channel is a BEC with erasure probability q. The EXIT function of the combined check-node–layer/accumulator inner decoder is (see [23, eq. (17)])

$$I_{E,acc}(p,q) = \left[\frac{1-q}{1-q(1-p)^a}\right]^2 (1-p)^{a-1}. \tag{73}$$

The inner code rate is a, which can be *larger* than one. One can check that the area under (73) is precisely $\mathcal{A}_{\text{in}} = C/a = (1-q)/a$, as required by (69).

As an example of a code design, suppose we choose $a = 1$ and connect 40% of the edges to degree-2 variable nodes and 60% of the edges to degree-3 variable nodes. We then have $R_{\text{out}} = 0.4$ and, using (13) and (19)

$$I_{E,out} = 1 - 0.4p - 0.6p^2. \tag{74}$$

The area under the curve is precisely $1 - R_{\text{out}} = 0.6$. The EXIT curve is plotted in Fig. 9 for erasure probabilities $q = 0.3$ and $q = 0.5$. In this figure, we have $I_{A,in} = 1-p$ and $I_{A,out} = 1-p$. Observe that the decoder's per-edge erasure probability can be made to approach zero for both channels. The threshold for this code is, in fact, $q = 5/9$ so that $C = 4/9 \approx 0.444$. Thus, these repeat–accumulate codes cannot approach capacity.

As a second example, suppose we choose $a > 1$. We then have the problem that both the inner and outer code EXIT curves start from $I_E = 0$, and decoding cannot even begin to converge. This problem can be overcome in several ways: make the code systematic [23], let a small fraction of the check nodes have degree 1 [28], or use inner or outer *code doping* [52]. The first of these approaches makes the code fall outside the class of serially concatenated codes, so we consider it next.

VI. EXIT for Systematic RA Codes

A systematic RA code of rate $R = k/n$ has k variable nodes and an accumulator of length $n-k$. The design of [23] further specifies a check-node layer with $n-k$ check nodes each having a edges going into the interleaver. Let $\overline{d}_v$ be the *average* number of edges going into the interleaver from the variable nodes. The number of edges is both $k\overline{d}_v$ and $(n-k)a$, giving

$$R = \frac{1}{1-\overline{d}_v/a}. \tag{75}$$

Next, suppose the communication channel is memoryless with $I(X;Y) = C$ if X is uniformly distributed. The areas $\mathcal{A}_v$ and $\mathcal{A}_{\text{acc}}$ under the respective variable and check-node/accumulator curves are

$$\mathcal{A}_v = 1 - (1-C)/\overline{d}_v \tag{76}$$

$$\mathcal{A}_{\text{acc}} = C/a. \tag{77}$$

For successful decoding, the variable-node curve must lie above the check-node/accumulator curve, which requires $1 - \mathcal{A}_v < \mathcal{A}_{\text{acc}}$. So suppose we have $1-\mathcal{A}_v = \gamma\mathcal{A}_{\text{acc}}$ for some γ satisfying $0 \le \gamma < 1$. From (76) and (77) we have

$$R = \frac{\gamma C}{1-(1-\gamma)C} < C. \tag{78}$$

Equation (78) has the same implications as (71), namely, that any area gap between the two curves corresponds to a rate loss compared to C. We thus again have the result that to approach capacity one must match the variable-node curve exactly to the check-node/accumulator curve. This is done in [23] by using a Taylor series expansion of an approximation to the check-node curve. The paper [23] thereby followed the approach of [53] which we discuss later in Example 27.

VII. EXIT for LDPC Codes

The design rate R of an LDPC code is determined by the number of variable nodes n_v and the number of check nodes n_c via (see [4, Sec. II-A])

$$R = (n_v - n_c)/n_v = 1 - n_c/n_v. \tag{79}$$

The true rate could be larger than the design rate because some of the checks could be redundant. However, as is usually done, we will ignore this subtlety. Note that n_v is the code length n. Let $\overline{d}_v$ and $\overline{d}_c$ be the *average* degrees of the variable and check nodes, respectively. The number of interleaver edges is both $n_v\overline{d}_v$ and $n_c\overline{d}_c$, giving

$$R = 1 - \overline{d}_v/\overline{d}_c. \tag{80}$$

Suppose again that the communication channel is memoryless with $I(X;Y) = C$ if X is uniformly distributed. We use (19), (30), and (31), and find that the areas under the respective variable and check-node curves are

$$\mathcal{A}_v = 1 - (1 - C)/\overline{d}_v \tag{81}$$

$$\mathcal{A}_c = 1/\overline{d}_c. \tag{82}$$

For successful decoding the variable-node curve must lie above the check-node curve. This implies $1 - \mathcal{A}_v < \mathcal{A}_c$, so suppose that $1 - \mathcal{A}_v = \gamma \mathcal{A}_c$ for some γ satisfying $0 \leq \gamma < 1$. From (81) and (82) we have

$$R = \frac{C - (1 - \gamma)}{\gamma} < C. \tag{83}$$

Equation (83) has the same implications as (71) and (78): an area gap between the two curves translates into a rate loss compared to C. We can again approach capacity only by matching the variable-node curve exactly to the check-node curve. We point out that this result is related to the flatness condition of [31]. The curve fitting is accomplished in [53] and [54] via Taylor series expansions of the check-node EXIT curve.

Example 27: (Curve Fit Via Taylor Series) We follow the approach of [53] and choose a right-regular (or check-regular) LDPC code with $d_c = 4$ for all check nodes. The inverse EXIT curve is $I_{Ec}^{-1}(p) = (1 - p)^{1/3}$, and its Taylor series expansion about $p = 0$ is

$$I_{Ec}^{-1}(p) = 1 - \frac{1}{3}p - \frac{1}{9}p^2 - \frac{5}{81}p^3 - \cdots. \tag{84}$$

It is easy to check that all the remaining terms in the expansion are negative, so any truncation of (84) will lie above $I_{Ec}^{-1}(p)$. Suppose we truncate after four terms to obtain

$$I_{Ec}^{-1}(p) \approx 1 - \frac{41}{81}\left[\frac{27}{41}p + \frac{9}{41}p^2 + \frac{5}{41}p^3\right]. \tag{85}$$

We now choose $27/41, 9/41$, and $5/41$ of the edges to be incident to degree-2, -3, and -4 variable nodes, respectively. As a result, the EXIT curve of the variable-node decoder is the right-hand side of (85) if $q = 41/81$. We further have $\overline{d}_v = 101/41$ and $R = 63/164 \approx 0.384$, while $C = 40/81 \approx 0.494$. The resulting curve is shown in Fig. 10 with $I_{Av} = 1 - p$ and $I_{Ac} = 1 - p$. The curve fit is rather tight near $(I_A, I_E) = (1, 1)$, which means the decoder needs many iterations to correct all erasures. Of course, as q decreases the convergence tunnel widens.

Example 28: (Unequal Error Protection) Consider the code of Example 27 and suppose we choose only so many iterations so that $I_E \geq 0.99$. This means that the coded bits associated with variable nodes of degree-2, -3, and -4 have erasure probabilities less than $q(1 - 0.99)^2$, $q(1 - 0.99)^3$, and $q(1 - 0.99)^4$, respectively. Thus, if we have a systematic code, it makes sense to design the encoder to associate the information bits with the high-degree variable nodes [54].

A. Generalized LDPC Codes

Generalized LDPC codes are codes where the repetition codes of Example 4 and/or the single parity-check codes of Example 5 are replaced by other codes. For example, the paper [55] replaces the single parity-check codes by Hamming codes. The motivation for doing this is to reduce the number of decoder iterations and/or to lower the error floor. A disadvantage is that one must implement more complex APP decoders. However, we point out that low-complexity APP decoders exist for trellis codes, and also for first-order Reed–Muller and Hamming codes [56].

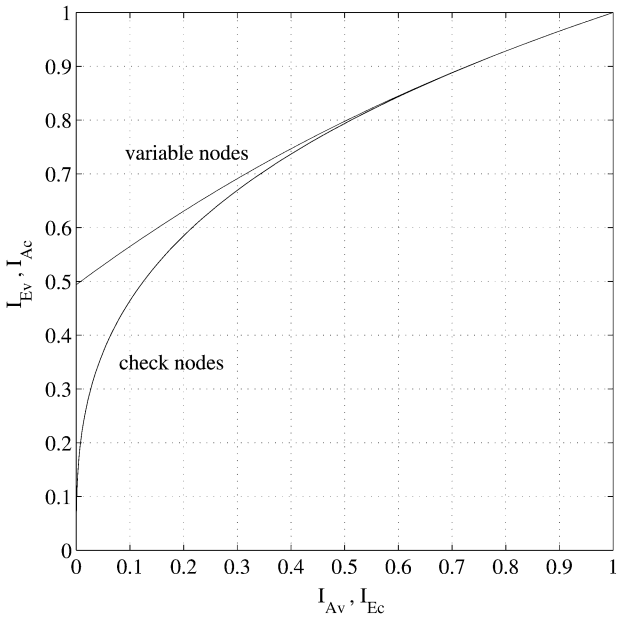

Fig. 10. EXIT chart for a right-regular LDPC code on the BEC.

We next derive the design rate of generalized LDPC codes. Suppose there are n_v variable nodes and n_c check nodes. Suppose further that variable node j represents a $[d_{v,j}, k_{v,j}]$ linear code that has the $k_{v,j}$ information bits going through the communication channel and the $d_{v,j}$ coded bits going through the extrinsic channel. For instance, the variable nodes of Examples 4 and 12 represent $[d_{v,j} = d_v, k_{v,j} = 1]$ repetition codes. The average number of coded bits and degrees per variable node is

$$\overline{k}_v = \frac{1}{n_v}\sum_{j=1}^{n_v} k_{v,j}, \quad \overline{d}_v = \frac{1}{n_v}\sum_{j=1}^{n_v} d_{v,j}. \tag{86}$$

Similarly, suppose check node j represents a $[d_{c,j}, k_{c,j}]$ linear code with no communication channel and with the $d_{v,j}$ coded bits going through the extrinsic channel. We write

$$\overline{k}_c = \frac{1}{n_c}\sum_{j=1}^{n_c} k_{c,j}, \quad \overline{d}_c = \frac{1}{n_c}\sum_{j=1}^{n_c} d_{c,j}. \tag{87}$$

The number of interleaver edges is both $n_v\overline{d}_v$ and $n_c\overline{d}_c$, and the number of constraints on the $n = n_v\overline{k}_v$ coded bits is $\sum_j (d_{c,j} - k_{c,j}) = n_c(\overline{d}_c - \overline{k}_c)$. The design rate is therefore

$$\begin{aligned} R &= [n - n_c(\overline{d}_c - \overline{k}_c)]/n \\ &= 1 - \frac{1 - R_c}{R_v} \end{aligned} \tag{88}$$

where $R_v = \overline{k}_v/\overline{d}_v$ and $R_c = \overline{k}_c/\overline{d}_c$. For example, for LDPC codes we have $R_v = 1/\overline{d}_v$ and $R_c = 1 - 1/\overline{d}_c$.

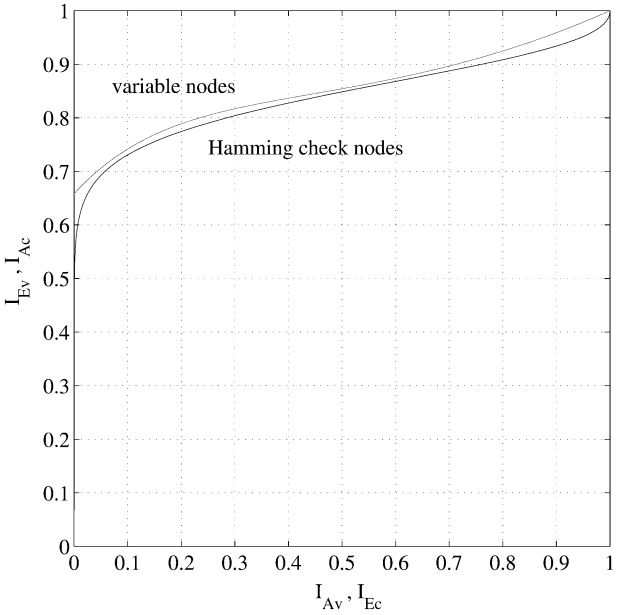

Fig. 11. EXIT chart for a generalized LDPC code.

The decoding model for variable node j is Fig. 3 with Encoder 1 passing $k_{v,j}$ independent bits directly to the communication channel, and with Encoder 2 putting out the $d_{v,j}$ bits of the linear code. Suppose the communication channel is memoryless with $I(X;Y) = C$ if X is uniformly distributed. We use (19) and (25) to compute

$$\mathcal{A}_v = 1 - (1 - C)R_v \tag{89}$$

$$\mathcal{A}_c = 1 - R_c. \tag{90}$$

We again require $1 - \mathcal{A}_v < \mathcal{A}_c$ for successful decoding, so suppose $1 - \mathcal{A}_v = \gamma \mathcal{A}_c$ for some γ satisfying $0 \leq \gamma < 1$. We then recover (83) and again find that one must match the variable-node curve exactly to the check-node curve.

Example 29: (Hamming Check-Regular Codes) Consider using $[31, 26]$ Hamming codes as the check nodes. We computethe check-node EXIT curve via (49), (40), and Theorem 4, and plot it in Fig. 11 as the lower curve. We mix the variable nodes of Examples 4 and 6 by connecting one half of the interleaver edges to $d_v = 8$ repetition codes and the other half to $d_v = 7$ single parity-check nodes. This means that 7/15 of the variable nodes are repetition codes, and 8/15 are single parity-check codes. We therefore have $\overline{k}_v = 55/15$, $\overline{d}_v = 112/15$, and $R_v = 55/112$. The variable-node curve is shown in Fig. 11 for $q = 0.3$, and it is reasonably well-matched to the Hamming code curve. The design rate is $R = 229/341 = 0.672$ which is close to $C = 0.7$.

VIII. EXIT for PC Codes

The most common PC code puts out three kinds of bits: k systematic bits $\underline{u} = \underline{x}_0$, n_1 parity-check bits $\underline{x}_1$, and n_2 parity-check bits $\underline{x}_2$. The overall rate is thus $R = k/(k + n_1 + n_2)$. An encoder and decoder structure for such a code is shown in Fig. 12. Suppose the communication channel is memoryless.

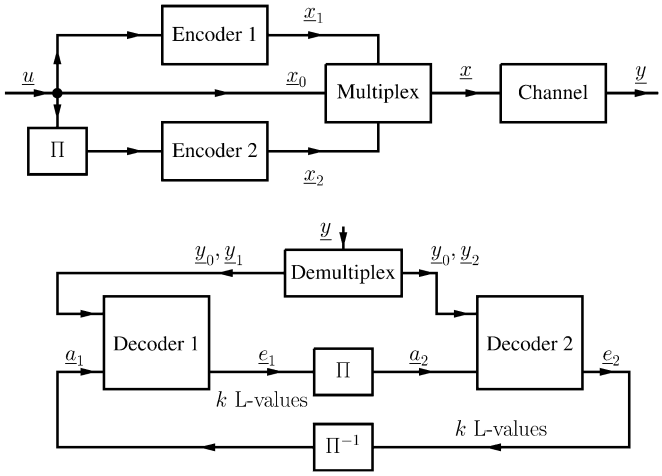

Fig. 12. Encoder and iterative decoder for a PC code.

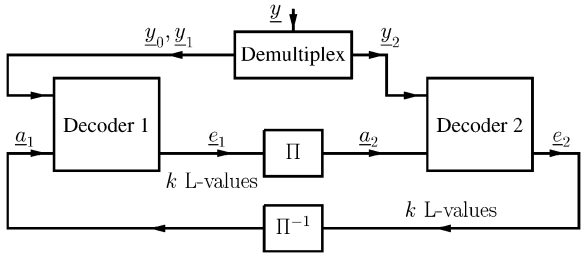

Fig. 13. Alternative iterative decoder for a PC code. The second decoder receives only $\underline{y}_2$.

The vector $\underline{y}_j$ then represents the communication channel outputs corresponding to $\underline{x}_j$ for $j = 0, 1, 2$.

Fig. 12 has $\underline{y}_0$ going to *both* decoders. This overlap means that one must subtract the channel L-value c_i from the e_i of (6) before passing it to the other component decoder. In other words, for memoryless communication channels, the new extrinsic L-value is

$$e_i = \log \frac{\sum_{\underline{u}: v_i(\underline{u})=0} P(\underline{w}_{[i]} | \underline{v}_{[i]}(\underline{u})) P(\underline{y}_{[i]} \mid \underline{x}_{[i]}(\underline{u}))}{\sum_{\underline{u}: v_i(\underline{u})=1} P(\underline{w}_{[i]} \mid \underline{v}_{[i]}(\underline{u})) P(\underline{y}_{[i]} \mid \underline{x}_{[i]}(\underline{u}))} \tag{91}$$

where $\underline{y}_{[i]}$ is the same as $[\underline{y}_0, \underline{y}_1, \underline{y}_2]$ without the ith component of $\underline{y}_0$. Furthermore, we now have

$$d_i = a_i + c_i + e_i. \tag{92}$$

Observe that e_i is a function of $\underline{y}_{[i]}$ and $\underline{a}_{[i]}$. One can show that $I(V_i; E_i) = I(V_i; \underline{Y}_{[i]}\, \underline{A}_{[i]})$, which means that (8) becomes

$$I_E = \frac{1}{m} \sum_{i=1}^{m} I(V_i; \underline{Y}_{[i]}\, \underline{A}_{[i]}). \tag{93}$$

Rather than dealing with (93), suppose we use the decoder shown in Fig. 13. That is, we consider new Decoders 1 and 2 where only Decoder 1 uses $\underline{y}_0$. The advantage of this asymmetric approach is that one can apply all the results of Section IV.

For example, consider Decoder 1 for which we use the model of Fig. 2 with $\underline{v} = \underline{u}$. Example 10 tells us that

$$\mathcal{A}_1 = \frac{I(\underline{X}_0\underline{X}_1;\underline{Y}_0\underline{Y}_1)/(k+n_1)}{k/(k+n_1)} \leq C/R_1 \qquad (94)$$

where $R_1 = k/(k+n_1)$. For Decoder 2 we similarly have

$$\mathcal{A}_2 = \frac{I(\underline{X}_2;\underline{Y}_2)/n_2}{k/n_2} \leq C/R_2 \qquad (95)$$

where $R_2 = k/n_2$. For successful decoding, Decoder 1's curve must lie above Decoder 2's curve. This implies $1-\mathcal{A}_1 < \mathcal{A}_2$, so suppose that $1-\mathcal{A}_1 = \gamma\mathcal{A}_2$ for some γ satisfying $0 \leq \gamma < 1$. From (94) and (95) we have

$$R = \frac{I(\underline{X}_0\underline{X}_1;\underline{Y}_0\underline{Y}_1) + \gamma I(\underline{X}_2;\underline{Y}_2)}{k+n_1+n_2} < C. \qquad (96)$$

We yet again find that an area gap between the two curves translates into a rate loss compared to C. The design of capacity-approaching PC codes thus reduces to a curve-fitting problem, as is the case for SC, RA, and LDPC codes.

The areas (94) and (95) have further implications. Just like the inner code of an SC code, if $R_1 < 1$ then we must have

$$I(\underline{X}_0\,\underline{X}_1;\underline{Y}_0\,\underline{Y}_1)/(k+n_1) < C.$$

Similarly, if $R_2 < 1$ then we must have

$$I(\underline{X}_2;\underline{Y}_2)/n_2 < C.$$

In either case, the component code has an inherent capacity loss that the other code cannot recover. For example, if $n_2 \approx n_1$ and Encoder 1 is a low-memory convolutional code with rate less than one, then one will have a significant gap between R and C in (96). We illustrate this point with the following example.

Example 30: (Turbo Code) Suppose Encoders 1 and 2 are accumulators for which $a-1$ of every a bits are punctured. We have $R = a/(a+2)$ and compute the respective first and second decoder EXIT curves to be

$$I_{E1}(p,q) = 1 - q\left(1 - \left[\frac{1-q}{1-q(1-pq)^a}\right]^2 (1-pq)^{a-1}\right) \qquad (97)$$

$$I_{E2}(p,q) = \left[\frac{1-q}{1-q(1-p)^a}\right]^2 (1-p)^{a-1}. \qquad (98)$$

The curve (97) follows by replacing p in (73) with pq, and then noting that the extrinsic L-value is a sum of an accumulator L-value and a channel L-value. The curve (98) is simply (73). We thus have $R_1 = a/(a+1)$, $R_2 = a$ and

$$\mathcal{A}_1 = \frac{1-q}{a}\left[a + \frac{1-(1-q)^a}{1-q(1-q)^a}\right] \leq C/R_1 \qquad (99)$$

$$\mathcal{A}_2 = (1-q)/a = C/R_2. \qquad (100)$$

Equality holds in (99) if and only if $q = 1$, so that $\mathcal{A}_1 < C/R_1$ for all interesting cases. We cannot, therefore, approach capacity *regardless* of which second component code we choose.

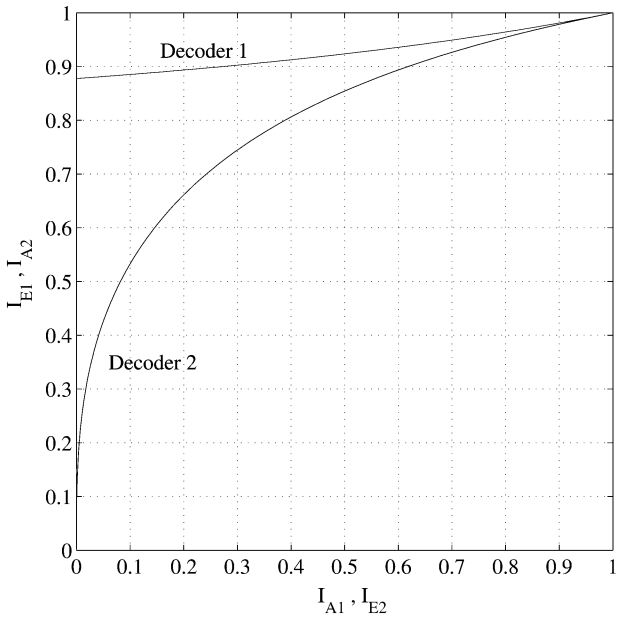

Fig. 14. EXIT chart for a PC code.

For instance, suppose we choose $a = 4$ so that $R = 2/3$. We plot the EXIT curves in Fig. 14 for $q = 0.2$, where $I_{A1} = 1-p$ and $I_{A2} = 1-p$. We have $\mathcal{A}_1 = 0.929$ which implies (see (94))

$$I(\underline{X}_0\,\underline{X}_1;\underline{Y}_0\,\underline{Y}_1)/(k+n_1) = 0.743.$$

This is less than $C = 0.8$, so that *any* choice of second component code with $n_2 \approx n_1$ cannot make the overall code approach capacity.

A. Nonsystematic PC Codes

Nonsystematic PC codes have the same encoder structure as in Fig. 12 except that no $\underline{x}_0$ is transmitted. The rate is therefore $R = k/(n_1+n_2)$ and (94) becomes

$$\mathcal{A}_1 = \frac{I(\underline{X}_1;\underline{Y}_1)/n_1}{k/n_1} \qquad (101)$$

which has the same form as (95). We again find that the design of capacity-approaching codes reduces to a curve-fitting problem. One can further make similar statements about the component codes as was done above for systematic PC codes.

As a final remark, RA codes can be converted to nonsystematic PC codes as described in [57]. The conversion involves splitting the inner code into two parts, and considering each part as a new component code. The paper [14] shows that this trick can improve decoding efficiency.

IX. Summary

A decoding model was introduced that applies to a wide variety of code structures and communication problems. EXIT functions were defined for this model by averaging per-letter mutual informations. Various properties of these functions were derived when transmitting information over a BEC. An

area property was used to show that the design of capacity-approaching codes reduces to a curve-fitting problem for SC, RA, LDPC, and PC codes. A duality property lets one compute the EXIT function of a linear code from the EXIT function of its dual.

We suspect that other interesting EXIT properties can yet be found. Potential extensions of our results could be as follows.

- The connection between EXIT functions and information functions motivates further investigations of support weights.
- Closed-form EXIT functions for convolutional codes on BECs would help explain how to design PC codes.
- Empirically, EXIT functions are robust to changes in the communication and extrinsic channels. For example, consider two modulators with different bit-to-symbol mappings but the same rate. The area property suggests that the area under both EXIT curves should be the same, and this is indeed approximately true for a wide range of mappings [9, Fig. 3], [41, eq. 12]. To understand the robustness better, bounds on EXIT functions for non-BEC channels would be helpful, and might improve the existing EXIT chart performance predictions. Some recent progress on this problem for LDPC codes has been reported in [39] and [40].

Appendix A
Proof of Proposition 1

We expand

$$\begin{aligned}\Pr(V_i=0 \mid e_i) &= \sum_{\underline{y},\underline{a}_{[i]}} P(\underline{y},\underline{a}_{[i]}|e_i)\Pr(V_i=0 \mid \underline{y},\underline{a}_{[i]},e_i)\\ &= \sum_{\underline{y},\underline{a}_{[i]}} P(\underline{y},\underline{a}_{[i]} \mid e_i)\frac{\exp(e_i)}{1+\exp(e_i)}\\ &= \frac{\exp(e_i)}{1+\exp(e_i)}. \end{aligned} \tag{102}$$

This last term is simply $\Pr(V_i=0 \mid \underline{y},\underline{a}_{[i]},e_i)$. Thus, because V_i is binary, we have $I(V_i;\underline{Y}\,\underline{A}_{[i]}|E_i)=0$. We further have

$$\begin{aligned} I(V_i;\underline{Y}\,\underline{A}_{[i]}) &= I(V_i;\underline{Y}\,\underline{A}_{[i]}E_i)\\ &= I(V_i;E_i)+I(V_i;\underline{Y}\,\underline{A}_{[i]} \mid E_i)\\ &= I(V_i;E_i). \end{aligned} \tag{103}$$

This proves (11).

Appendix B
Proof of the Area Property

The terms in the sum of (12) are

$$I(V_i;\underline{Y}\,\underline{A}_{[i]}) = H(V_i) - H(V_i|\underline{Y}\,\underline{A}_{[i]}). \tag{104}$$

We expand the conditional entropy as

$$H(V_i|\underline{Y}\,\underline{A}_{[i]}) = \sum_{\underline{a}_{[i]}} P(\underline{a}_{[i]})H(V_i|\underline{Y},\underline{A}_{[i]}=\underline{a}_{[i]}). \tag{105}$$

The extrinsic channel is a BEC, so let $\mathcal{S}$ be the set of positions of $\underline{a}$ that were *not* erased, not including position i. We have

$$H(V_i|\underline{Y},\underline{A}_{[i]}=\underline{a}_{[i]}) = H(V_i|\underline{Y},\underline{V}_{\mathcal{S}}=\underline{v}_{\mathcal{S}}) \tag{106}$$

where $\underline{V}_{\mathcal{S}}=\{V_j : j\in\mathcal{S}\}$. Inserting (106) into (105), we have

$$H(V_i|\underline{Y}\,\underline{A}_{[i]}) = \sum_{j=0}^{m-1}(1-p)^j p^{m-1-j}\sum_{|\mathcal{S}|=j,i\notin\mathcal{S}} H(V_i|\underline{Y}\,\underline{V}_{\mathcal{S}}) \tag{107}$$

where p is the extrinsic channel erasure probability, and where we have partitioned the set of $\underline{a}_{[i]}$ according to the number i of erasures in $\underline{a}_{[i]}$. We find that (see [58, p. 303])

$$\int_0^1 (1-p)^j p^{m-1-j}dp = \frac{1}{m\binom{m-1}{j}}. \tag{108}$$

The integral (108) is in fact a special case of Euler's integral of the first kind, or the beta function. Using (21), (104), (107), and (108), we have

$$\begin{aligned}\int_0^{I_{A,\max}} \frac{I_E(p)}{I_{A,\max}^2}dI_A &= 1-\int_0^1 \frac{\sum_{i=1}^m H(V_i|\underline{Y}\,\underline{A}_{[i]})}{m\cdot I_{A,\max}}dp\\ &= 1-\frac{1}{\sum_{i=1}^m H(V_i)}\sum_{i=1}^m\sum_{j=0}^{m-1}\frac{1}{m\binom{m-1}{j}}\\ &\quad\cdot\sum_{|\mathcal{S}|=j,i\notin\mathcal{S}} H(V_i|\underline{Y}\,\underline{V}_{\mathcal{S}}). \end{aligned} \tag{109}$$

It remains to show that the triple sum in (109) collapses to $H(\underline{V}|\underline{Y})$. Consider the vector $\underline{V}$ of length m, and observe that we can expand $H(\underline{V})$ in $m!$ different ways. For example, for $m=3$ we have the following six expansions:

$$\begin{aligned} H(\underline{V}) &= H(V_1)+H(V_2|V_1)+H(V_3|V_1V_2)\\ &= H(V_1)+H(V_3|V_1)+H(V_2|V_1V_3)\\ &= H(V_2)+H(V_1|V_2)+H(V_3|V_1V_2)\\ &= H(V_2)+H(V_3|V_2)+H(V_1|V_2V_3)\\ &= H(V_3)+H(V_1|V_3)+H(V_2|V_1V_3)\\ &= H(V_3)+H(V_2|V_3)+H(V_1|V_2V_3). \end{aligned}$$

We sum these expansions and divide by $m!$ to obtain

$$H(\underline{V}) = \sum_{i=1}^m\sum_{j=0}^{m-1}\frac{1}{m\binom{m-1}{j}}\sum_{|\mathcal{S}|=j,i\notin\mathcal{S}} H(V_i|\underline{V}_{\mathcal{S}}). \tag{110}$$

The entropy $H(\underline{V}|\underline{Y})$ has the same form as (110) except that one must add $\underline{Y}$ to the conditioning of all entropies. Inserting the result into (109) proves Theorem 1.

Appendix C
Proof of Theorem 2

Consider again (104), and note that $H(V_i)=1$ because Encoder 2 has no idle components. Let $\mathcal{S}$ and $\mathcal{T}$ be the sets of positions without erasures in the respective $\underline{a}$ and $\underline{y}$, not including position i for $\mathcal{S}$. For *linear* codes, we have

$$H(V_i|\underline{Y}=\underline{y},\underline{A}_{[i]}=\underline{a}_{[i]}) = k_{i\cup\mathcal{S}\cup\mathcal{T}}-k_{\mathcal{S}\cup\mathcal{T}}. \tag{111}$$

Averaging over all $\mathcal{S}$ and $\mathcal{T}$ and m positions, we obtain

$$\begin{aligned}\frac{1}{m}\sum_{i=1}^{m} H(V_i|\underline{Y}\,\underline{A}_{[i]}) &= \frac{1}{m}\sum_{i=1}^{m}\sum_{h=0}^{n}(1-q)^h q^{n-h}\sum_{|\mathcal{T}|=h}\sum_{g=1}^{m}(1-p)^{g-1}p^{m-g} \\ &\quad\cdot\left[\sum_{|\mathcal{S}|=g-1,i\notin\mathcal{S}} k_{i\cup\mathcal{S}\cup\mathcal{T}} - k_{\mathcal{S}\cup\mathcal{T}}\right]. \end{aligned} \tag{112}$$

We move the sum over i and $|\mathcal{T}| = h$ inside the square brackets and write

$$\begin{aligned}&\sum_{|\mathcal{T}|=h}\left(\sum_{i=1}^{m}\left\{\sum_{|\mathcal{S}|=g,i\in\mathcal{S}} k_{\mathcal{S}\cup\mathcal{T}} - \sum_{|\mathcal{S}|=g-1,i\notin\mathcal{S}} k_{\mathcal{S}\cup\mathcal{T}}\right\}\right) \\ &\quad= \sum_{|\mathcal{T}|=h}\left(\sum_{|\mathcal{S}|=g} g k_{\mathcal{S}\cup\mathcal{T}} - \sum_{|\mathcal{S}|=g-1}(m-g+1)k_{\mathcal{S}\cup\mathcal{T}}\right) \\ &\quad= g\cdot\tilde{e}_{g,h} - (m-g+1)\cdot\tilde{e}_{g-1,h}. \end{aligned} \tag{113}$$

Finally, we use (113) in (112), and subtract (112) from one. The result is (36).

Appendix D
Proof of Theorem 3

We give a sketch of the proof. For a given $\mathcal{S}\in\mathcal{S}_{g,h}$, let $\mathcal{C}'_{\mathcal{S}}$ be the linear code formed by those codewords of $\mathcal{C}$ which are all zero in those positions not in $\mathcal{S}$. Define

$$N^r_{g,h} = |\{\mathcal{S}\in\mathcal{S}_{g,h} : \dim\mathcal{C}'_{\mathcal{S}} = r\}| \tag{114}$$

and let $e_{g,h,r}$ be the number of sets $\mathcal{S}$ in $\mathcal{S}_{g,h}$ such that $k_{\mathcal{S}} = r$. One can check that

$$e_{g,h,r} = N^{k-r}_{m-g,n-h}. \tag{115}$$

Using arguments similar to those in [35, Lemma 1] (see also [42, Ch. 5.2]), we have

$$\sum_{i=0}^{g}\sum_{j=0}^{h}\binom{m-i}{g-i}\binom{n-j}{h-j}A^r_{i,j} = \sum_{a=r}^{k}\begin{bmatrix}a\\r\end{bmatrix}N^a_{g,h} \tag{116}$$

for $0\le g\le m$, $0\le h\le n$, and $0\le r\le k$. Solving this set of equations for $A^r_{i,j}$ and using (115) proves Theorem 3.

Appendix E
Proof of the Duality Property

We use [32, Corollary 3] (see also [59, Lemma 2.0]) to write

$$e_{g,h} = e^{\perp}_{m-g,n-h} - [(m-g)+(n-h)-k] \tag{117}$$

where $e^{\perp}_{g,h}$ is the split information function of the dual code. We can rewrite (117) as

$$\tilde{e}_{g,h} = \tilde{e}^{\perp}_{m-g,n-h} - \binom{m}{g}\binom{n}{h}[(m-g)+(n-h)-k] \tag{118}$$

where $\tilde{e}^{\perp}_{g,h}$ is the unnormalized split information function of the dual code. Inserting (118) into (36), we have

$$\begin{aligned}I_E(p,q) &= 1 - \frac{1}{m}\sum_{h=0}^{n}(1-q)^h q^{n-h}\sum_{g=1}^{m}(1-p)^{g-1}p^{m-g} \\ &\quad\cdot\Bigg[g\tilde{e}^{\perp}_{m-g,n-h} - g\binom{m}{g}\binom{n}{h} \\ &\qquad\cdot\{(m-g)+(n-h)-k\} \\ &\qquad-(m-g+1)\tilde{e}^{\perp}_{m-g+1,n-h} + (m-g+1) \\ &\qquad\cdot\binom{m}{g-1}\binom{n}{h}\{(m-g+1)+(n-h)-k\}\Bigg]. \end{aligned} \tag{119}$$

We make the change of variables $g = m-g'+1$ and $h = n-h'$, and collect terms to write

$$\begin{aligned}I_E(p,q) = &-I^{\perp}_E(1-p,1-q) \\ &+\sum_{h'=0}^{n}(1-q)^{n-h'}q^{h'}\binom{n}{h'} \\ &\cdot\sum_{g'=1}^{m}(1-p)^{m-g'}p^{g'-1}\binom{m-1}{g'-1}. \end{aligned} \tag{120}$$

The double sum is simply one. This proves (62).

References

[1] R. G. Gallager, *Low-Density Parity-Check Codes*. Cambridge, MA: MIT Press, 1963.

[2] M. G. Luby, M. Mitzenmacher, M. A. Shokrollahi, D. A. Spielman, and V. Stemann, "Practical loss-resilient codes," in *Proc. 29th Annu. ACM Symp. Theory of Computing*, 1997, pp. 150–159.

[3] M. G. Luby, M. Mitzenmacher, M. A. Shokrollahi, and D. A. Spielman, "Efficient erasure correcting codes," *IEEE Trans. Inform. Theory*, vol. 47, pp. 569–584, Feb. 2001.

[4] 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.

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

[6] S. Y. Chung, "On the construction of some capacity-approaching coding schemes," Ph.D. dissertation, MIT, Cambridge, MA, 2000.

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

[8] D. Divsalar, S. Dolinar, and F. Pollara, "Low complexity turbo-like codes," in *Proc. 2nd Int. Symp. Turbo Codes*, Sept. 2000, pp. 73–80.

[9] S. ten Brink, "Designing iterative decoding schemes with the extrinsic information transfer chart," *AEÜ Int. J. Electron. Commun.*, vol. 54, no. 6, pp. 389–398, Dec. 2000.

[10] S. Y. Chung, T. J. Richardson, and R. 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.

[11] H. El Gamal and A. R. Hammons Jr, "Analyzing the turbo decoder using the Gaussian approximation," *IEEE Trans. Inform. Theory*, vol. 47, pp. 671–686, Feb. 2001.

[12] K. Narayanan, "Effect of precoding on the convergence of turbo equalization for partial response channels," *IEEE J. Select. Areas Commun.*, vol. 19, pp. 686–698, Apr. 2001.

[13] S. ten Brink, "Convergence behavior of iteratively decoded parallel concatenated codes," *IEEE Trans. Commun.*, vol. 49, pp. 1727–1737, Oct. 2001.

[14] S. Huettinger, J. Huber, R. Johannesson, and R. Fischer, "Information processing in soft-output decoding," in *Proc. Allerton Conf. Communication, Control, and Computing*, Allerton, IL, Oct. 2001.

[15] J. Boutros and G. Caire, "Iterative multiuser joint decoding: Unified framework and asymptotic analysis," *IEEE Trans. Inform. Theory*, vol. 48, pp. 1772–1793, July 2002.

[16] J. Huber and S. Huettinger, "Information processing and combining in channel coding," in *Proc. 3rd Int. Symp. Turbo Codes*, Brest, France, Sept. 1–5, 2003, pp. 95–102.

[17] F. Lehmann and G. M. Maggio, "Analysis of the iterative decoding of LDPC and product codes using the Gaussian approximation," *IEEE Trans. Inform. Theory*, vol. 49, pp. 2993–3000, Nov. 2003.

[18] M. Tüchler, S. ten Brink, and J. Hagenauer, "Measures for tracing convergence of iterative decoding algorithms," in *Proc. 4th Int. ITG Conf. Source and Channel Coding*, Berlin, Germany, Jan. 2002.

[19] B. Scanavino, G. Montorsi, and S. Benedetto, "Convergence properties of iterative decoders working at bit and symbol level," in *Proc. 2001 IEEE Global Telecommunications Conf. (GLOBECOM '01)*, vol. 2, 2001, pp. 1037–1041.

[20] C. Berrou and A. Glavieux, "Near optimum error correcting coding and decoding: Turbo-codes," *IEEE Trans. Commun.*, vol. 44, pp. 1261–1271, Oct. 1996.

[21] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara, "Serial concatenation of interleaved codes: Performance analysis, design, and iterative decoding," *IEEE Trans. Inform. Theory*, vol. 44, pp. 909–926, May 1998.

[22] D. Divsalar, H. Jin, and R. J. McEliece, "Coding theorems for 'turbo-like' codes," in *Proc. Allerton Conf. Communication, Control, and Computing*, Allerton, IL, Sept. 1998, pp. 201–210.

[23] H. Jin, A. Khandekar, and R. McEliece, "Irregular repeat-accumulate codes," in *Proc. 2nd Int. Conf. Turbo Codes*, Brest, France, Sept. 2000.

[24] S. ten Brink, "Rate one-half code for approaching the Shannon limit by 0.1 dB," *Electron. Lett.*, vol. 36, no. 15, pp. 1293–1294, July 2000.

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

[26] M. Tüchler and J. Hagenauer, "EXIT charts of irregular codes," in *Proc. 36th Annu. Conf. Information Science and Systems*, Princeton, NJ, Mar. 2002.

[27] G. Caire, D. Burshtein, and S. Shamai (Shitz), "LDPC coding for interference mitigation at the transmitter," in *Proc. Allerton Conf. Communication, Control, and Computing*, Allerton, IL, Oct. 2002.

[28] S. ten Brink and G. Kramer, "Design of repeat-accumulate codes for iterative detection and decoding," *IEEE Trans. Signal Processing*, vol. 51, pp. 2764–2772, Nov. 2003.

[29] M. Tüchler, "Design of serially concatenated systems depending on the block length," *IEEE Trans. Commun.*, vol. 52, pp. 209–218, Feb. 2004.

[30] S. ten Brink, G. Kramer, and A. Ashikhmin, "Design of low-density parity-check codes for modulation and detection," *IEEE Trans. Commun.*, vol. 52, pp. 670–678, Apr. 2004.

[31] M. A. Shokrollahi, "Capacity-achieving sequences," in *Codes, Systems, and Graphical Models*, B. Marcus and J. Rosenthal, Eds. Minneapolis, MN: Inst. Math. and its Applic., Univ. Minnesota, 2000, vol. 123, IMA Volumes in Mathematics and its Applications, pp. 153–166.

[32] T. Helleseth, T. Kløve, and V. I. Levenshtein, "On the information function of an error-correcting code," *IEEE Trans. Inform. Theory*, vol. 43, pp. 549–557, Mar. 1997.

[33] V. K. Wei, "Generalized Hamming weights for linear codes," *IEEE Trans. Inform. Theory*, vol. 37, pp. 1412–1418, Sept. 1991.

[34] T. Kløve, "Support weight distribution for linear codes," *Discr. Math.*, vol. 106/107, pp. 311–316, 1992.

[35] J. Simonis, "The effective length of subcodes," *Applicable Algebra in Eng., Commun., and Comput.*, vol. 5, pp. 371–377, 1994.

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

[37] R. G. Gallager, *Information Theory and Reliable Communication*. New York: Wiley, 1968.

[38] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. New York: Wiley, 1991.

[39] I. Land, P. A. Hoeher, S. Huettinger, and J. Huber, "Bounds on information combining," in *Proc. 3rd Int. Symp. Turbo Codes*, Brest, France, Sept. 1–5, 2003, pp. 39–42.

[40] I. Sutskover, S. Shamai (Shitz), and J. Ziv, "Extremes of information combining," in *Proc. Allerton Conf. Communication, Control, and Computing*, Allerton, IL, Oct. 2003.

[41] S. ten Brink, "Exploiting the chain rule of mutual information for the design of iterative decoding schemes," in *Proc. Allerton Conf. Communication, Control, and Computing*, Allerton, IL, Oct. 2001.

[42] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error-Correcting Codes.*. Amsterdam, The Netherlands: North-Holland, 1977.

[43] T. Helleseth, T. Kløve, and J. Mykkeltveit, "The weight distribution of irreducible cyclic codes with block length $n_1((q^l-1)/N)$," *Discr. Math.*, vol. 18, pp. 179–211, 1977.

[44] C. Bachoc, "On harmonic weight enumerators of binary codes," *Des., Codes, Cryptogr.*, vol. 18, pp. 11–28, 1999.

[45] H. Chen and J. Coffey, "Trellis structure and higher weights of extremal self-dual codes," *Des., Codes, Cryptogr.*, vol. 24, pp. 15–36, 2001.

[46] S. Dougherty, A. Gulliver, and M. Oura, "Higher weights and graded rings for binary self-dual codes," unpublished paper, submitted for publication.

[47] O. Milenkovic, "Higher weight and coset weight enumerators of formally self-dual codes," *Des., Codes, Cryptogr.*, to be published.

[48] O. Milenkovic, S. T. Coffey, and K. J. Compton, "The third support weight enumerators of the doubly-even, self-dual $[32, 16, 8]$ codes," *IEEE Trans. Inform. Theory*, vol. 49, pp. 740–746, Mar. 2003.

[49] S. Benedetto and G. Montorsi, "Unveiling turbo codes: Some results on parallel concatenated coding schemes," *IEEE Trans. Inform. Theory*, vol. 42, pp. 409–428, Mar. 1996.

[50] C. Di, D. Proietti, İ. E. Telatar, T. J. Richardson, and R. L. Urbanke, "Finite-length analysis of low-density parity-check codes on the binary erasure channel," *IEEE Trans. Inform. Theory*, vol. 48, pp. 1570–1579, June 2002.

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

[52] S. ten Brink, "Code doping for triggering iterative decoding convergence," in *Proc. 2001 IEEE Int. Symp. Information Theory*, Washington, DC, June 2001, p. 235.

[53] M. A. Shokrollahi, "New sequences of linear time erasure codes approaching the channel capacity," in *Proc. 13th Conf. Applied Algebra, Error Correcting Codes, and Cryptography (Lecture Notes in Computer Science)*. Berlin, Germany: Springer-Verlag, 1999, pp. 65–76.

[54] P. Oswald and A. Shokrollahi, "Capacity-achieving sequences for the erasure channel," *IEEE Trans. Inform. Theory*, vol. 48, pp. 3017–3028, Dec. 2002.

[55] M. Lentmaier and K. S. Zigangirov, "On generalized low-density parity-check codes based on Hamming component codes," *IEEE Commun. Lett.*, vol. 3, pp. 248–250, Aug. 1999.

[56] A. Ashikhmin and S. Litsyn, "Fast MAP decoding of first order Reed-Muller and Hamming codes," *IEEE Trans. Inform.Theory*, submitted for publication.

[57] S. Huettinger, S. ten Brink, and J. Huber, "Turbo-code representation of RA-codes and DRS-codes for reduced complexity decoding," in *Proc. 2001 Conf. Information Science and Systems*, Mar. 21–23, 2001.

[58] I. N. Bronshtein and K. A. Semendyayev, *Handbook of Mathematics*, 3rd ed. Berlin, Germany: Springer-Verlag, 1997.

[59] F. J. MacWilliams, "A theorem on the distribution of weights in a systematic code," *Bell Syst. Tech. J.*, vol. 42, pp. 79–94, 1963.

[60] E. Sharon, A. Ashikhmin, and S. Litsyn, "EXIT functions for the Gaussian channel," in *Proc. Allerton Conf. Communication, Control, and Computing*, Monticello, IL, Oct. 2003, pp. 972–981.