Shaping EXIT Functions of Turbo Codes

Land, Ingmar; Rasmussen, Lars; Grant, Alex

Published in:
Australian Communications Theory Workshop, 2008. AusCTW 2008

DOI (link to publication from Publisher):
10.1109/AUSCTW.2008.4460832

Publication date:
2008

Document Version
Publisher's PDF, also known as Version of record

Shaping EXIT Functions of Turbo Codes

Ingmar Land*‡, Lars Rasmussen*, Alex Grant*

* Institute for Telecommunications Research, University of South Australia, Adelaide, Australia.
‡ Department of Electronic Systems, Aalborg University, Aalborg, Denmark.
Email: {ingmar.land,lars.rasmussen,alex.grant}@unisa.edu.au.

***Abstract*—This paper investigates the shaping of the extrinsic information transfer (EXIT) functions of parallel turbo codes, when the turbo code is used as an outer code in a serially concatenated scheme. As opposed to other work, we consider the overall EXIT function of the turbo decoder and not the EXIT chart for the component codes. Three methods for shaping are investigated: (i) individual puncturing of the systematic part and the two parity parts of the turbo codes, (ii) puncturing of the overall turbo code, and (iii) repetition of the overall turbo code. For each case, the EXIT function of the resulting turbo code is computed from the EXIT function of its component codes. Exact results are obtained for the binary erasure channel.**

I. Introduction

In digital communication systems, the receiver typically consists of a receiver front-end followed by a decoder. The receiver front-end may for example be a demodulator, an equalizer, or a multi-user detector. To allow for an efficient iterative receiver, the code should be designed for the particular receiver front-end.

The extrinsic information transfer (EXIT) chart method has shown to be a very efficient tool for the design of codes [1]–[3]. In the EXIT chart method, the EXIT function of the receiver front-end is measured or determined analytically, and then the code is designed such that its EXIT function matches the one for the receiver front-end.

Efficient design procedures for low-density parity-check (LDPC) codes and for irregular repeat-accumulate (IRA) codes are presented in [4]–[6]. LDPC codes and IRA codes have many degrees of freedom for code design and thus allow for very good curve matching. Turbo codes [7], on the other hand, have already found their way into many applications, and thus they are of significant practical relevance. At a first view, it seems to be more difficult to match turbo codes to a particular receiver front-end. If, however, it is possible to improve systems that employ turbo codes by simple modifications, this would be very attractive from a practical point of view.

Punctured turbo codes were investigated in [8], [9] with respect to their decoding threshold on the AWGN channel and their distance properties; the adaptation to a receiver front-end was not considered. On the other hand, repeated convolutional codes (i.e. a serial concatenation of a convolutional code with repetition codes) were proposed for coded CDMA systems in [10]–[12] for iterative multi-user decoding. Therein a particular shape of the EXIT function due to the repetition of code symbols was observed. These concepts of constructing new codes by puncturing and repeating codes motivated the ideas in the present paper.

In this paper we consider the overall EXIT function of the parallel turbo code used as the outer code in a serially concatenated scheme. The decoding model is: a-priori L-values for the code symbols are given to the turbo decoder, the turbo decoder iterates until convergence is achieved, and then the turbo decoder delivers extrinsic L-values for the code symbols. The considered EXIT function is the mapping from a-priori mutual information about the code symbols to extrinsic mutual information about the code symbols. (This is different from the EXIT functions for the component codes depicted in the usual EXIT charts for turbo codes.)

In this paper we investigate several ways to shape the EXIT function of turbo codes by simple techniques, namely by puncturing and by repeating code symbols. In particular, the EXIT functions of the resulting turbo codes are analytically computed from the EXIT functions of the component codes. This provides an extension of the investigations on punctured turbo codes [8], [9], and analytical results for the observations for repeated convolutional codes [10]–[12].

In order to get exact results, we restrict ourselves in this paper to binary erasure channels (BEC). This approach is motivated by analytical results for randomly punctured convolutional codes on the BEC [13], and by other analytical results for EXIT functions for the BEC [14]. The methods presented, however, may be applied to AWGN channels in a similar way to obtain approximations of the exact EXIT functions.

The paper is structured as follows. In Section II, the definition of EXIT functions and the concept of information combining is revised. In Section III, the EXIT functions of punctured and repeated turbo codes are derived from the EXIT functions of the component codes. To illustrate these results, some examples are given in Section IV. In Section V, the main results are summarized, and possible extensions of the work presented are outlined.

Throughout the paper, random variables are written in uppercase letters and their realizations in the corresponding lowercase letters.

II. Theoretical Background

This section revises background information used in this paper. First EXIT functions for the BEC are defined. Then the equations for information combining for the BEC are given.

A. *Definition of EXIT Functions*

The definition of EXIT functions follows the one in [14]. Consider a binary symmetric source and a binary linear encoder of rate R with information word length K and code word length N. As required in the EXIT chart method, we assume infinite length codes, i.e., $N \to \infty$. The binary information symbols are denoted by U_k, $k = 1, \ldots, K$, and the binary code symbols are denoted by X_n, $n = 1, \ldots, N$.

The information symbols are transmitted over a BEC, called the information-symbol a-priori channel, with erasure probability $\delta_{a,u}$. For the information symbol U_k, the channel output is denoted by $Y_{u,k}$, and the corresponding a-priori L-value by $a_{u,k} := L(U_k | y_{u,k})$. Thus the a-priori mutual information for the information symbols is given by

$$I_{a,u} := I(U_k; A_{u,k}) = 1 - \delta_{a,u},$$

which is independent of k.

Similarly, the code symbols are transmitted over a BEC, called the code-symbol a-priori channel, with erasure probability $\delta_{a,x}$. For the code symbol X_n, the channel output is denoted by $Y_{x,n}$ and the corresponding a-priori L-value by $a_{x,n} := L(X_n | y_{x,n})$. The a-priori mutual information for the code symbols is given by

$$I_{a,x} := I(X_n; A_{x,n}) = 1 - \delta_{a,x},$$

which is independent of n. Notice that the two a-priori channels are actually test channels and not the communication channels [14].

All a-priori L-values are given to an APP decoder. For each information symbol U_k, the decoder computes the extrinsic L-value $e_{u,k} := L(U_k | \boldsymbol{a}_{u,\setminus k}, \boldsymbol{a}_x)$. For each code symbol X_n, the decoder computes the extrinsic L-value $e_{x,n} := L(X_n | \boldsymbol{a}_u, \boldsymbol{a}_{x,\setminus n})$. The vector $\boldsymbol{a}_u$ comprises all a-priori L-values $a_{u,k}$ for information symbols U_k; the vector $\boldsymbol{a}_{u,\setminus k}$ comprises all a-priori L-values $a_{u,i}$ without element $a_{u,k}$. The vectors $\boldsymbol{a}_x$ and $\boldsymbol{a}_{x,\setminus n}$ have similar meanings for the code symbols X_n. Since the information-symbol a-priori channel and the code-symbol a-priori channel are BECs, also the channel from U_k to $E_{u,k}$ and the channel from X_n to $E_{x,n}$ are BECs.

Based on these extrinsic values, we define the extrinsic mutual information for information symbols

$$I_{e,u} := \frac{1}{K} \sum_{k=1}^{K} I(U_k; E_{u,k}),$$

where $K = RN$ and $N \to \infty$. In a similar way, we define the extrinsic mutual information for code symbols,

$$I_{e,u} := \frac{1}{N} \sum_{n=1}^{N} I(X_n; E_{x,n}),$$

where $N \to \infty$. Using these values of mutual information, we define the EXIT function T_u for information symbols by

$$I_{e,u} = \mathrm{T}_u(I_{a,u}, I_{a,x}). \tag{1}$$

Similarly, we define the EXIT function T_x for code symbols by

$$I_{e,x} = \mathrm{T}_x(I_{a,u}, I_{a,x}). \tag{2}$$

The following discussions and derivations are based on these two functions. For further details on EXIT functions, we refer the reader to [1], [2], [14].

B. Information Combining

Assume a binary symbol X is transmitted over two independent BECs yielding the outputs Y_1 and Y_2. Using the mutual information values of the two individual channels, $I_1 := I(X; Y_1)$ and $I_2 := I(X; Y_2)$, the mutual information between X and $[Y_1, Y_2]$, $I := I(X; Y_1, Y_2)$, can be expressed as

$$I = I_1 \boxplus I_2 := 1 - (1 - I_1)(1 - I_2). \tag{3}$$

This is referred to as the combined mutual information. The notation with the operator $\boxplus$ is introduced for convenience. Generalizing this leads to

$$I_1 \boxplus \cdots \boxplus I_n := 1 - (1 - I_1) \cdots (1 - I_n). \tag{4}$$

For details we refer the reader to [15]–[18].

III. Modification of EXIT Functions

In this section, we show how to obtain the EXIT function of a turbo code from the EXIT functions of the component codes. In particular, we emphasize how various ways of puncturing the systematic part and the two parity parts of the parent turbo code affect the EXIT function of the punctured turbo code. Then we discuss how overall puncturing and repeating changes the EXIT function.

A. Punctured Turbo Codes

The EXIT function of a punctured turbo code when used as a outer code in a serial concatenated scheme is addressed in the following. For decoding, the iterative decoder for the parent turbo code is employed. We assume that this decoder iterates until convergence is achieved. As the a-priori mutual information for information symbols is zero in this case, $I_{a,u} = 0$, we are looking for the EXIT function T_x with

$$I_{e,x} = \mathrm{T}_x^{\mathrm{pTC}}\Big(0, I_{a,x}\Big). \tag{5}$$

First the encoder and the decoder are described. This is then translated into the computation of the EXIT function based on the EXIT functions of the component codes. We refer to the unpunctured code as the parent turbo code (TC) and to the punctured code as the punctured turbo code (pTC) for the sake of clarity. To simplify notation, we restrict ourselves in the following to the case where a rate $1/3$ code is punctured to a rate $1/2$ code. The generalization is straight forward.

Encoder and Decoder: The encoder for the TC consists of a classical turbo-code encoder [7] of rate $1/3$ with a systematic branch and two parity branches. The two component encoders are referred to as Encoder 1 and Encoder 2 and they are assumed to be recursive convolutional encoders of rate 1.

The information symbols are denoted by U_i, $i = 1, \ldots, K$; they also form the systematic part of the TC. They are fed to Encoder 1 to generate the code symbols $X_i^{(1)}$, $i = 1, \ldots, K$, which form the first parity part of the TC. Interleaving the information symbols and feeding them to Encoder 1 yields the code symbols $X_i^{(2)}$, $i = 1, \ldots, K$, which form the second parity part of the TC. Thus the TC has rate $K/(3K) = 1/3$. As always in the EXIT chart method, we assume $K \to \infty$.

Each of the three parts of the TC is then randomly punctured to obtain the pTC. The puncturing is characterized by the permeability [8], [9]

$$\boldsymbol{\rho} = [\rho^{(s)} \rho^{(1)} \rho^{(2)}].$$

The value $\rho^{(s)}$ specifies the relative amount of systematic symbols that are not punctured; i.e., out of the K systematic symbols before puncturing, $\rho^{(s)}K$ systematic symbols are left after puncturing. Similarly, $\rho^{(1)}$ and $\rho^{(1)}$ specify the amount of code symbols from Encoder 1 and Encoder 2, respectively, left after puncturing. The rate of the pTC is thus

$$\frac{K}{\rho^{(s)}K + \rho^{(1)}K + \rho^{(1)}K} = \frac{1}{\rho^{(s)} + \rho^{(1)} + \rho^{(1)}}.$$

To obtain the desired rate 1/2 pTC, we require $\rho^{(s)} + \rho^{(1)} + \rho^{(1)} = 2$. For $\rho^{(s)} < 1$, we obtain partially systematic turbo codes, introduced and studied in [8].

The pTC is transmitted over a BEC with erasure probability δ, i.e., the a-priori mutual information for code symbols is $I_{a,x} = 1-\delta$. (The systematic symbols are also code symbols.) The observations are de-punctured, and then the a-priori L-values for the code symbols of the parent TC are computed, i.e., $L^{(s)}_{a,u,k}$ for the systematic code symbols U_k, $L^{(1)}_{a,x,k}$ for the code symbols $X^{(1)}_k$, and $L^{(2)}_{a,x,k}$ for the code symbols $X^{(2)}_k$.

In each iteration, Decoder 1 obtains two L-values for each information symbol U_k, namely the extrinsic L-value $L^{(2)}_{e,u,k}$ from Decoder 2 and the a-priori L-value $L^{(s)}_{a,u,k}$ based on the observation from the channel. The actual input to Decoder 1 is thus

$$L^{(1)}_{a,u,k} = L^{(s)}_{a,u,k} + L^{(2)}_{e,u,k}. \tag{6}$$

Based on these a-priori L-values, Decoder 1 computes the new extrinsic L-values $L^{(1)}_{e,u,k}$. (To simplify notation, the iteration numbers are omitted.) This holds in a similar way for Decoder 2. This iterative process is repeated until convergence is achieved. Let $L^{(1)*}_{e,u,k}$ and $L^{(2)*}_{e,u,k}$ denote the extrinsic L-values at the point of convergence.

After convergence, the extrinsic L-values for all code symbols are computed. The extrinsic L-value for the systematic code symbol U_k is the sum of the two extrinsic L-values of Decoder 1 and Decoder 2,

$$L^{(s)}_{e,u,k} = L^{(1)*}_{e,u,k} + L^{(2)*}_{e,u,k}. \tag{7}$$

The extrinsic L-value for the code symbol $X^{(1)}_k$ is determined by Decoder 1 using the values $L^{(1)*}_{e,u,k}$ for the information symbols and the values $L^{(1)}_{a,x,k}$ for the code symbols; similarly the extrinsic L-value for the code symbol $X^{(2)}_k$ is determined by Decoder 2 using the values $L^{(2)*}_{e,u,k}$ for the information symbols and the values $L^{(2)}_{a,x,k}$ for the code symbols.

The resulting extrinsic L-values are then punctured according to the permeability $\boldsymbol{\rho}$ to obtain the vectors of L-values for the actually transmitted code symbols of the pTC.

EXIT Function: The decoding operations described above are now translated into processing of mutual information. Following (1) and (2), the EXIT functions of Decoder 1 are denoted by $\mathrm{T}^{(1)}_u$ and $\mathrm{T}^{(1)}_x$, and the EXIT functions of Decoder 2 are denoted by $\mathrm{T}^{(2)}_u$ and $\mathrm{T}^{(2)}_x$. The a-priori mutual information for code symbols of the pTC is given by $I_{a,x} = 1 - \delta$. The goal is to determine the average extrinsic mutual information $I_{e,x}$ for the code symbols of the pTC, which is a weighted average of the mutual information values for U_k, $X^{(1)}_k$ and $X^{(2)}_k$.

Puncturing symbols prior to transmission over a BEC and de-puncturing them prior to decoding is equivalent to having a BEC with larger erasure probability [13]. Thus the a-priori mutual information for the systematic code symbols U_k, the code symbols $X^{(1)}_k$ and the code symbols $X^{(2)}_k$ becomes

$$I^{(s)}_{a,u} = \rho^{(s)} I_{a,x}, \quad I^{(1)}_{a,x} = \rho^{(1)} I_{a,x}, \quad I^{(2)}_{a,x} = \rho^{(2)} I_{a,x}. \tag{8}$$

These are the mutual information values seen by the iterative decoder for the parent TC.

The extrinsic mutual information for information symbols produced by Decoder 1 is denoted by $I^{(1)}_{e,u}$, and the one produced by Decoder 2 is denoted by $I^{(2)}_{e,u}$.

Consider first Decoder 1. Similarly to (6), the combination of the two mutual information values $I^{(s)}_{a,u}$ and $I^{(2)}_{e,u}$ forms the a-priori mutual information about the information symbols. The value $I^{(1)}_{a,x}$ represents the a-priori mutual information about the code symbols. Using the EXIT function $\mathrm{T}^{(1)}_u$ of Decoder 1, the extrinsic mutual information about information symbols is thus

$$I^{(1)}_{e,u} = \mathrm{T}^{(1)}_u\Big(I^{(s)}_{a,u} \boxplus I^{(2)}_{e,u}, I^{(1)}_{a,x}\Big). \tag{9}$$

In a similar way, we obtain for Decoder 2

$$I^{(2)}_{e,u} = \mathrm{T}^{(2)}_u\Big(I^{(s)}_{a,u} \boxplus I^{(1)}_{e,u}, I^{(2)}_{a,x}\Big). \tag{10}$$

(Notice that the EXIT charts in [1] plot $I^{(1)}_{e,u}$ versus $I^{(2)}_{e,u}$ for Decoder 1, and $I^{(2)}_{e,u}$ versus $I^{(1)}_{e,u}$ for Decoder 1.) At the point of convergence, we have the two extrinsic mutual information values fulfilling (9) and (10) simultaneously. We denote them by $I^{(1)}_{e,u} = I^{(1)*}_{e,u}$ and $I^{(2)}_{e,u} = I^{(2)*}_{e,u}$.

Following (7), the extrinsic mutual information for the systematic code symbols U_k is then computed by information combining,

$$I^{(s)}_{e,u} = I^{(1)*}_{e,u} \boxplus I^{(2)*}_{e,u}. \tag{11}$$

Using similar arguments as above, the extrinsic mutual information for the code symbols $X^{(1)}_i$ are determined using the EXIT function $\mathrm{T}^{(1)}_x$,

$$I^{(1)}_{e,x} = \mathrm{T}^{(1)}_x\Big(I^{(s)}_{a,u} \boxplus I^{(2)*}_{e,u}, I^{(1)}_{a,x}\Big). \tag{12}$$

Similarly we obtain for Decoder 2

$$I^{(2)}_{e,x} = \mathrm{T}^{(2)}_x\Big(I^{(s)}_{a,u} \boxplus I^{(1)*}_{e,u}, I^{(2)}_{a,x}\Big). \tag{13}$$

These are mutual information values for the code symbols of the parent TC.

To obtain the average extrinsic mutual information $I_{e,x}$ for the code symbols of the pTC, these mutual information values have to be weighted according to the relative number of symbols in the pTC. The length of the rate $1/2$ code is $2K$. Out of those, there are $\rho^{(s)}K$ systematic code symbols U_k, $\rho^{(1)}K$ code symbols $X_k^{(1)}$, and $\rho^{(2)}K$ code symbols $X_k^{(2)}$. Therefore, we have

$$I_{e,x} = \frac{\rho^{(s)}}{2} \cdot I_{e,u}^{(s)} + \frac{\rho^{(1)}}{2} \cdot I_{e,x}^{(1)} + \frac{\rho^{(2)}}{2} \cdot I_{e,x}^{(2)} \qquad (14)$$

for the average extrinsic information.

Thus we have the desired relation between the a-priori mutual information $I_{a,x}$ and the extrinsic mutual information $I_{e,x}$ for the pTC, given by (8)-(14). To compute those, only the EXIT functions of the component codes and the permeability characterizing the puncturing are required. Some examples are provided in Section IV.

B. Punctured and Repeated Code

This section again deals with the EXIT function of a code used as an outer code in a serially concatenated scheme. This code may be a turbo code (examples for this case are provided in Section IV) but it may also be any other code. We assume that the EXIT function

$$I_{e,x}^{\mathrm{C}} = \mathrm{T}_x^{\mathrm{C}}\Big(0, I_{a,x}^{\mathrm{C}}\Big) \qquad (15)$$

of the parent code is given. (As before, the a-priori mutual information for information symbols, $I_{a,u}^{\mathrm{C}}$, is zero in this case.)

From this EXIT function we can compute the EXIT function of two other codes: the punctured code and the repeated code. The punctured code (pC) is equivalent to the one in the previous section when we use the TC as the parent code and when all parts of the TC are punctured in the same way, i.e., $\rho^{(s)} = \rho^{(1)} = \rho^{(2)} = \rho$. The repeated code (rC) results from from repeating the code symbols of the parent code. Repeated convolutional codes were considered in multiuser systems with iterative decoding in [10]–[12], which motives this investigation. The EXIT functions of punctured codes and repeated codes are discussed in the following. Notice that in Section III-A the starting point are the EXIT functions of the component codes, whereas here we start with the EXIT function of the parent code.

Punctured Code: Consider first the code with puncturing. The parent code is punctured prior to transmission, and the decoder for the unpunctured code is used. We assume random puncturing and characterize the puncturing by the permeability ρ. As above, if the length of the original code is N, the length of the punctured code is ρN. Furthermore, we assume that the punctured code is transmitted over a BEC with erasure probability δ. Thus the a-priori mutual information about code symbols of the punctured code is $I_{a,x} = 1 - \delta$. The goal is to determine the extrinsic mutual information $I_{e,x}$ about the code symbols of the punctured code.

Puncturing symbols prior to transmission over a BEC and de-puncturing them prior to decoding is equivalent to having a BEC with larger erasure probability [13]. Thus the a-priori mutual information for the code symbols available to the decoder for the parent code is

$$I_{a,x}^{\mathrm{C}} = \rho\, I_{a,x}. \qquad (16)$$

The extrinsic mutual information for the code symbols of the parent code becomes thus

$$I_{e,x}^{\mathrm{C}} = \mathrm{T}_x^{\mathrm{C}}\Big(0, \rho\, I_{a,x}^{\mathrm{C}}\Big). \qquad (17)$$

The average extrinsic mutual information for the code symbols of the punctured code is equal to that for the parent code, i.e., $I_{e,x} = I_{e,x}^{\mathrm{C}}$. Thus we have the desired relation between the a-priori mutual information and the extrinsic mutual information for the punctured code:

$$I_{e,x} = \mathrm{T}_x^{\mathrm{C}}\Big(0, \rho\, I_{a,x}\Big). \qquad (18)$$

This EXIT function depends only on the EXIT function of the parent code and the permeability, characterizing the puncturing.

Repeated Code: Consider now the case where each code symbol is repeated n times. As before, we assume transmission over a BEC with erasure probability δ. Thus the a-priori mutual information about code symbols of the repeated code is $I_{a,x} = 1 - \delta$. We assume optimal decoding of the repeated code. The goal is to determine the extrinsic mutual information $I_{e,x}$ for the code symbols of the repeated code.

Optimal decoding can be separated into three steps. Similarly the EXIT function is computed in three steps.

Step 1: For each code symbol of the parent code, the a-priori L-values of the repeated symbols are added. This corresponds to information combining,

$$I_{a,x}^{\mathrm{C}} = \underbrace{I_{a,x} \boxplus \ldots \boxplus I_{a,x}}_{n \text{ terms}}, \qquad (19)$$

leading to the a-priori mutual information of the code symbols of the parent code.

Step 2: For each code symbol of the parent code, the extrinsic L-value is computed. The corresponding extrinsic mutual information is obtained from the EXIT function of the parent code,

$$I_{e,x}^{\mathrm{C}} = \mathrm{T}_x^{\mathrm{C}}\Big(0, I_{a,x}^{\mathrm{C}}\Big), \qquad (20)$$

where the a-priori mutual information for information symbols is zero, of course.

Step 3: For each code symbol of the repeated code, the extrinsic L-value from Step 2 is added with $n-1$ a-priori L-values of the repeated symbols. Thus the extrinsic mutual information $I_{e,x}$ for code symbols of the repeated code can be determined by combining the extrinsic mutual information $I_{e,x}^{\mathrm{C}}$ from Step 2 with $n-1$ a-priori mutual information values $I_{a,x}$ for the code symbols of the repeated code,

$$I_{e,x} = \underbrace{I_{a,x} \boxplus \ldots \boxplus I_{a,x}}_{n-1 \text{ terms}} \boxplus I_{e,x}^{\mathrm{C}}. \qquad (21)$$

The EXIT function of the repeated code can thus be computed from the EXIT function of the parent code using (19), (20) and (21). Examples for EXIT functions of punctured and repeated turbo codes are provided in Section IV.

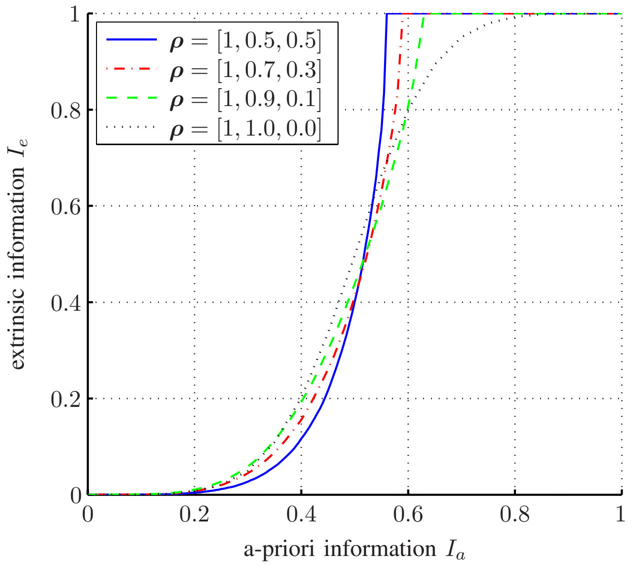

Fig. 1. EXIT functions of rate $1/2$ turbo codes with non-symmetric puncturing. The labels of the curves give the permeability $\boldsymbol{\rho} = [\rho^{(s)}\rho^{(1)}\rho^{(2)}]$. The classical (symmetric punctured) turbo code is obtained for $\boldsymbol{\rho} = [1, 0.5, 0.5]$ (solid line). The rate $1/2$ RSC code is obtained for $\boldsymbol{\rho} = [1, 1, 0]$ (dotted line).

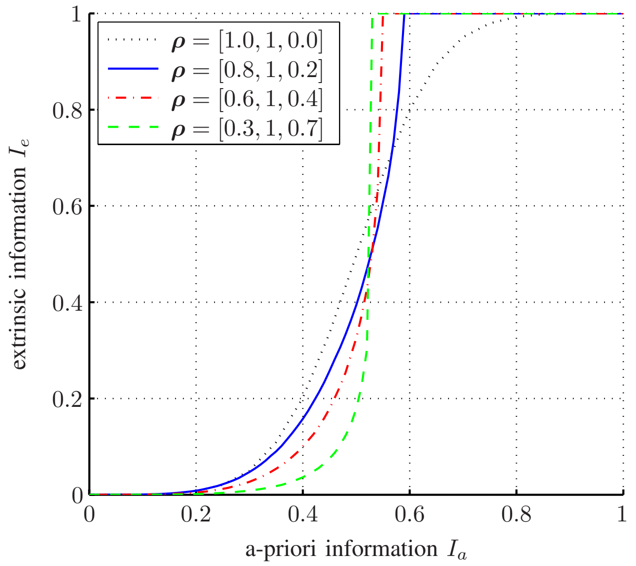

Fig. 2. EXIT functions of partially systematic rate $1/2$ turbo codes with non-symmetric puncturing. The labels of the curves give the permeability $\boldsymbol{\rho} = [\rho^{(s)}\rho^{(1)}\rho^{(2)}]$. The rate $1/2$ RSC code is obtained for $\boldsymbol{\rho} = [1, 1, 0]$ (dotted line).

Extension to the AWGN Channel: We restricted ourselves to the BEC, as above methods provide exact results for this channel. These methods, however, may also be applied to the EXIT functions for AWGN channels, similar to [9]. Then the resulting EXIT functions represent approximations of the true ones.

To do so, the following modifications are required: (i) In Section III-A, $\mathrm{T}_u^{(1)}$, $\mathrm{T}_x^{(1)}$, $\mathrm{T}_u^{(2)}$, and $\mathrm{T}_x^{(2)}$ have to be replaced by the EXIT functions for AWGN channels; similarly in Section III-B, $\mathrm{T}_x^{\mathrm{C}}$ has to be replaced by the EXIT function for AWGN channels. (ii) In both Section III-A and Section III-B, the operator $\boxplus$ for information combining for the BEC has to be replaced by the corresponding operator for the AWGN channel (see references in Section II-B).

IV. Examples

In this section, some EXIT functions are computed using the methods from the previous section to illustrate the effects of puncturing and repeating. For all examples, we use turbo codes with the $(5/7)$ recursive convolutional encoder (feedforward polynomial 5, feedback polynomial 7) of rate 1 as component encoders.

EXIT functions of several rate $1/2$ turbo codes with various permeabilities $\boldsymbol{\rho} = [\rho^{(s)}\rho^{(1)}\rho^{(2)}]$ are depicted in Figure 1 and Figure 2. They are computed using the method from Section III-A.

Figure 1 shows systematic turbo codes, i.e., none of the systematic code symbols are punctured, and thus $\rho^{(s)} = 1$. The puncturing of the two parity parts, however, is varied between two extreme cases: (a) both parity parts are punctured in the same way, i.e., $\rho^{(1)} = 1/2$ and $\rho^{(2)} = 1/2$; (b) the first parity part is not punctured at all and the second parity part is completely punctured, i.e., $\rho^{(1)} = 1$ and $\rho^{(2)} = 0$. Case (a) corresponds to the classical, symmetric punctured turbo code [19]. Case (b) corresponds to the $(1, 5/7)$ recursive systematic convolutional encoder of rate $1/2$. By selecting the corresponding permeabilities, various shapes of EXIT functions in between those two extreme cases can be obtained.

Figure 2 shows partially systematic turbo codes, i.e., also systematic code symbols are punctured. This time, the first parity part is not punctured, i.e., $\rho^{(1)} = 1$, and the puncturing of the systematic part and the second parity part is varied between two extreme cases: (a) all of the systematic code symbols are punctured and no symbols of the second parity part are punctured, i.e., $\rho^{(s)} = 0$ and $\rho^{(2)} = 1$; (b) no systematic symbols are punctured and the second parity part is completely punctured, i.e., $\rho^{(s)} = 1$ and $\rho^{(2)} = 0$. As above, case (b) corresponds to the $(1, 5/7)$ recursive systematic convolutional encoder of rate $1/2$. Case (a) itself is not depicted, since convergence is impossible for this setting (see [20]).

Apparently, puncturing systematic code symbols improves the decoding threshold (provided that not too many are punctured, see [20]); at the same time, the EXIT function becomes more square-edged. This, however, is the opposite of what is desired when adapting turbo-code EXIT functions to EXIT functions of receiver front-ends.

The effects of puncturing and repeating according to Section III-B are illustrated in Figure 3. The parent code is the classical rate $1/2$ turbo code from above. Puncturing this code with permeability $\rho = 3/4$ leads to a rate $2/3$ turbo code, and puncturing this code with permeability $\rho = 2/3$ leads to a rate $3/4$ turbo code. The figure shows that puncturing the code stretches its EXIT function in direction of the x-axis.

The figure shows also the EXIT function of the turbo code resulting from repeating the code symbols ($n = 2$). The

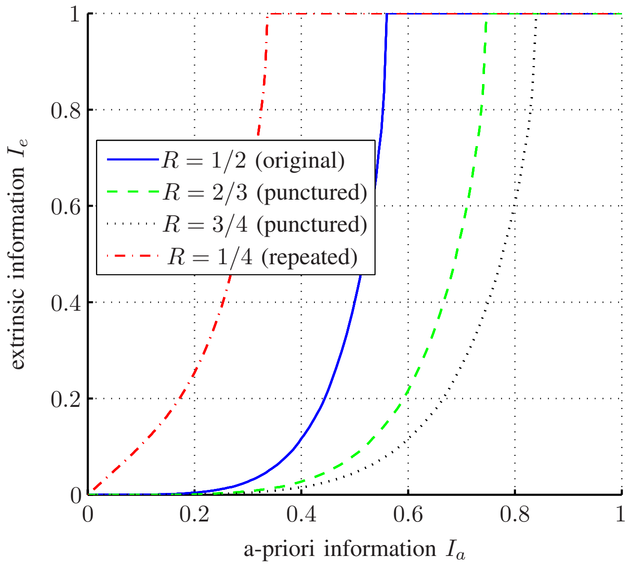

Fig. 3. EXIT functions of punctured and repeated turbo codes. The parent turbo code (original) has rate $1/2$ (solid line). The punctured turbo codes have rate $2/3$ ($\rho = 3/4$) (dashed line) and rate $3/4$ ($\rho = 2/3$) (dotted line). The repeated code has rate $1/4$ (rate $1/2$ repetition) (dash-dotted line).

resulting EXIT function looks compressed compared to the one of the parent turbo code (original code). In addition to that, for low a-priori mutual information it has slope 1 and thus the shape of the EXIT function of the rate $1/2$ repetition code (see e.g. [15]). This follows directly from the analytical expression of the EXIT function in (21), and it explains the observations in [10]–[12]: for low a-priori information, the EXIT function of the repeated convolutional code is very similar to the EXIT function of the repetition code.

V. Summary and Outlook

This paper has presented some basic ideas about how to shape EXIT functions of turbo codes by random puncturing and by repeating, and how to compute the resulting EXIT functions from the EXIT functions of the component codes. The methods presented are (i) puncturing a parent turbo code with individual puncturing schemes for each the systematic part and the two parity parts; (ii) puncturing a parent turbo code with the same puncturing scheme for all three parts; (iii) repeating all code symbols. (This direct serial concatenation of the original code with a repetition code has been suggested for application in multi-user systems with iterative receivers.) The effects on the EXIT function of the turbo code have been illustrated by examples.

There are some obvious extensions to the methods presented. Instead of using a symmetric rate $1/3$ parent turbo code as a starting point, other multiple turbo codes with more than two encoders and non-symmetric turbo encoders may be used. In the present paper, only repeating all code symbols has been studied. Repeating only the systematic code symbols, only the parity symbols, or even the combination of repeating some parts of the code and puncturing other parts may lead to other useful shapes of the overall EXIT function.

To obtain an optimal code for a given receiver front-end, other code classes, like LDPC codes or IRA codes, may be more suitable. From a practical point of view, however, the use of a standard turbo code adapted by simply puncturing or repeating code symbols may be more attractive for some applications.

Acknowledgment

This work has been supported in parts by the Australian Research Council under ARC Discovery Grant DP0663567.

References

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

[2] ——, "Code characteristic matching for iterative decoding of serially concatenated codes," *Ann. Télécommun.*, vol. 56, no. 7-8, pp. 394–408, 2001.

[3] S. ten Brink, J. Speidel, and R.-H. Yan, "Iterative demapping for QPSK modulation," *Electron. Letters*, vol. 34, no. 15, pp. 1459–1460, Jul. 1998.

[4] G. Lechner, J. Sayir, and I. Land, "Optimization of LDPC codes for receiver frontends," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Seattle, Washington, USA, Jul. 2006.

[5] S. ten Brink, G. Kramer, and A. Ashikhmin, "Design of low-density parity-check codes for modulation and detection," *IEEE Trans. Inform. Theory*, vol. 52, no. 14, pp. 670–678, Apr. 2004.

[6] S. ten Brink and G. Kramer, "Design of repeat-accumulate codes for iterative detection and decoding," *IEEE Trans. Signal Processing*, vol. 51, no. 11, pp. 2764–2772, Nov. 2003.

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

[8] I. Land and P. Hoeher, "Partially systematic rate 1/2 turbo codes," in *Proc. Int. Symp. on Turbo Codes & Rel. Topics*, Brest, France, Sep. 2000, pp. 287–290.

[9] F. Brännström and L. Rasmussen, "Multiple parallel concatenated codes with optimal puncturing and energy distribution," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Seoul, Korea, May 2005.

[10] H. Yi and L. K. Rasmussen, "Iterative switched-decoding for interleave-division multiple-access systems," accepted for publication in IEEE Trans. Vehicul. Techn.

[11] K. Kusume and G. Bauch, "Some aspects of interleave division multiple access in ad hoc networks," in *Proc. Int. Symp. on Turbo Codes & Rel. Topics*, Munich, Germany, Apr. 2006.

[12] ——, "A simple complexity reduction strategy for interleave division multiple access," in *Proc. IEEE Vehicular Tech. Conf. (VTC)*, Montreal, Canada, Sep. 2006.

[13] R. Thobaben, "EXIT functions for randomly punctured systematic codes," in *Proc. IEEE Inf. Theory Workshop*, Lake Tahoe, CA, USA, Sep. 2007.

[14] A. Ashikhmin, G. Kramer, and S. ten Brink, "Extrinsic information transfer functions: model and erasure channel properties," *IEEE Trans. Inform. Theory*, vol. 50, no. 11, pp. 2657–2673, Nov. 2004.

[15] I. Land and J. Huber, "Information combining," *Foundations and Trends in Communications and Information Theory*, vol. 3, no. 3, 2006.

[16] I. Land, S. Huettinger, P. A. Hoeher, and J. Huber, "Bounds on information combining," *IEEE Trans. Inform. Theory*, vol. 51, no. 2, pp. 612–619, Feb. 2005.

[17] ——, "Bounds on mutual information for simple codes using information combining," *Ann. Télécommun.*, vol. 60, no. 1/2, pp. 184–214, Jan./Feb. 2005.

[18] I. Sutskover, S. Shamai (Shitz), and J. Ziv, "Extremes of information combining," *IEEE Trans. Inform. Theory*, vol. 51, no. 4, pp. 1313–1325, Apr. 2005.

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

[20] S. ten Brink, "Code doping for triggering iterative decoding convergence," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Washington, DC, USA, Jun. 2001, p. 235.