Asymptotic Distance and Convergence Analysis of Braided Protograph Convolutional Codes

Tavares, Marcos B.S.; Lentmaier, Michael; Fettweis, Gerhard; Zigangirov, Kamil

Published in:
[Host publication title missing]

DOI:
10.1109/ALLERTON.2008.4797678

2008

Asymptotic Distance and Convergence Analysis of Braided Protograph Convolutional Codes

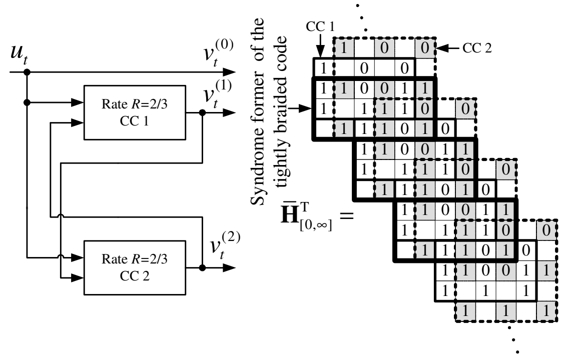
Marcos B.S. Tavares, Michael Lentmaier,
and Gerhard P. Fettweis
Vodafone Chair Mobile Communications Systems
Dresden University of Technology
01062 Dresden, Germany
{tavares, michael.lentmaier, fettweis}@ifn.et.tu-dresden.de

Kamil Sh. Zigangirov
Department of Electrical Engineering
University of Notre Dame
Notre Dame, IN 46556, USA
Email: kzigangi@nd.edu

***Abstract*— We analyze a class of LDPC convolutional codes that are constructed from tightly braided convolutional base codes by a lifting procedure. For these braided protograph convolutional codes, we show that the distances grow linearly with the constraint length, and we present lower bounds on their asymptotic segment distance and free distance as well as on the asymptotic minimum distance of their tail-biting versions. With some constraints imposed on the lifting permutations, braided protograph convolutional codes can also be decoded as turbo-like codes by iterative application of the BCJR algorithm. For this case, we derive an explicit upper-bound on the asymptotic decoding error probability as a function of the number of iterations.**

I. Introduction

Braided convolutional codes form a class of iteratively decodable convolutional codes that are constructed from component convolutional codes [1]. Like turbo codes, they are decoded by iterative application of the BCJR algorithm [2], which operates in the component decoding trellises. A special property of braided convolutional codes is that both information and parity bits are connected to each of the component decoders in a symmetric manner. This feature makes them more similar to (generalized) low-density parity-check ((G)LDPC) codes [3] [4] than other turbo-like constructions.


u_t
$v_t^{(0)}$
$v_t^{(1)}$
$v_t^{(2)}$
Rate R=2/3
CC 1
Rate R=2/3
CC 2
CC 1
CC 2
Syndrome former of the tightly braided code
$\bar{\mathbf{H}}^{\mathrm{T}}_{[0,\infty]}$ =
1
0
0
1
1

(a) (b)

Fig. 1. **(a)** Encoder for tightly braided convolutional codes (TBCC) [1]. **(b)** Composition of the base code syndrome former $\bar{\mathbf{H}}^{\mathrm{T}}_{[0,\infty]}$ by the (here identical) syndrome formers of the component convolutional codes CC 1 and CC 2.

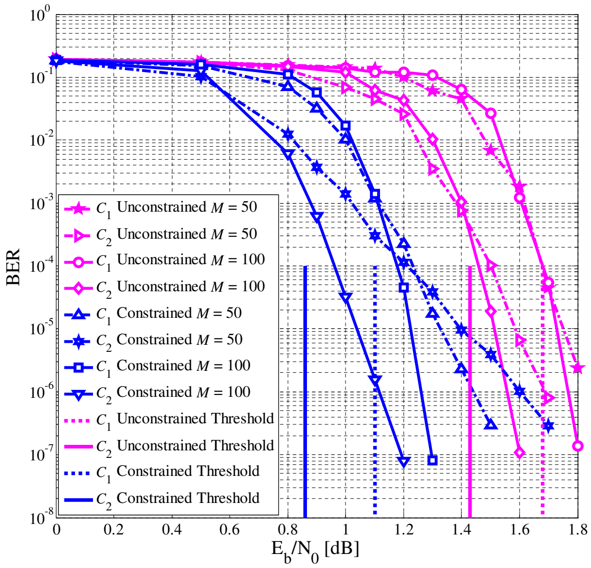


Fig. 2. Simulation results for different ensembles of braided protograph convolutional codes using continuous pipeline decoding. M denotes the sizes of the lifting matrices.

In this paper, we consider another variant of braided convolutional codes in which very simple tighly braided convolutional codes are used as a base code. A lifting procedure replaces in the syndrome former of this base code all ones by a permutation matrix and all zeros by an all-zero matrix [5]. An example of such a base code and its syndrome former, as well as simulation results are shown in Fig. 1 and Fig. 2, respectively. Codes from the unconstrained ensembles are decoded like conventional LDPC convolutional codes using belief propagation, while for the constrained ensembles a turbo-like decoder is used. Already for relatively small lifting matrices the codes achieve low error probabilities close to the respective convergence thresholds. A more detailed description of braided protograph convolutional codes, including efficient encoder and decoder implementations, is given in [5]. In the following two sections we perform an analysis of their asymptotic distance and convergence properties.

$$\mathbf{H}^{\mathrm{T}}_{[0,3]} = \begin{array}{cccccccccccc|c}
\mathbf{C}_0^{(0)} & \mathbf{C}_0^{(1)} & \mathbf{C}_1^{(0)} & \mathbf{C}_1^{(1)} & \mathbf{C}_2^{(0)} & \mathbf{C}_2^{(1)} & \mathbf{C}_3^{(0)} & \mathbf{C}_3^{(1)} & \mathbf{C}_4^{(0)} & \mathbf{C}_4^{(1)} & \mathbf{C}_5^{(0)} & \mathbf{C}_5^{(1)} & \\
\hline
\lambda_{0,0} & \lambda_{0,1} & 0 & 0 & \lambda_{0,4} & \lambda_{0,5} & & & & & & & \rho_0^{(0)} \\
\lambda_{1,0} & 0 & \lambda_{1,2} & \lambda_{1,3} & \lambda_{1,4} & 0 & M \times M & & & \text{permutation} & & & \rho_0^{(1)} \\
0 & \lambda_{2,1} & \lambda_{2,2} & \lambda_{2,3} & 0 & \lambda_{2,5} & \downarrow & & & \text{matrix} & & & \rho_0^{(2)} \\
 & & \lambda_{3,2} & \lambda_{3,3} & 0 & 0 & \lambda_{3,6} & \lambda_{3,7} & & \mathbf{0} & & & \rho_1^{(0)} \\
 & & \lambda_{4,2} & 0 & \lambda_{4,4} & \lambda_{4,5} & \lambda_{4,6} & 0 & & & & & \rho_1^{(1)} \\
 & & 0 & \lambda_{5,3} & \lambda_{5,4} & \lambda_{5,5} & 0 & \lambda_{5,7} & & & & & \rho_1^{(2)} \\
 & & & & \lambda_{6,4} & \lambda_{6,5} & 0 & 0 & \lambda_{6,8} & \lambda_{6,9} & & & \rho_2^{(0)} \\
 & & & & \lambda_{7,4} & 0 & \lambda_{7,6} & \lambda_{7,7} & \lambda_{7,8} & 0 & & & \rho_2^{(1)} \\
 & \mathbf{0} & & & 0 & \lambda_{8,5} & \lambda_{8,6} & \lambda_{8,7} & 0 & \lambda_{8,9} & & & \rho_2^{(2)} \\
 & & & & & & \lambda_{9,6} & \lambda_{9,7} & 0 & 0 & \lambda_{9,10} & \lambda_{9,11} & \rho_3^{(0)} \\
 & & & & & & \lambda_{10,6} & 0 & \lambda_{10,8} & \lambda_{10,9} & \lambda_{10,10} & 0 & \rho_3^{(1)} \\
 & & & & & & 0 & \lambda_{11,7} & \lambda_{11,8} & \lambda_{11,9} & 0 & \lambda_{11,11} & \rho_3^{(2)} \\
\hline
\psi_0^{(0)} & \psi_0^{(1)} & \psi_1^{(0)} & \psi_1^{(1)} & \psi_2^{(0)} & \psi_2^{(1)} & \psi_3^{(0)} & \psi_3^{(1)} & \psi_4^{(0)} & \psi_4^{(1)} & \psi_5^{(0)} & \psi_5^{(1)} &
\end{array}$$

(a)

$$\tilde{\mathbf{H}}^{\mathrm{T}}_{[0,3]} = \begin{array}{cccccccc|c}
\tilde{\mathbf{C}}_0^{(0)} & \tilde{\mathbf{C}}_0^{(1)} & \tilde{\mathbf{C}}_1^{(0)} & \tilde{\mathbf{C}}_1^{(1)} & \tilde{\mathbf{C}}_2^{(0)} & \tilde{\mathbf{C}}_2^{(1)} & \tilde{\mathbf{C}}_3^{(0)} & \tilde{\mathbf{C}}_3^{(1)} & \\
\hline
\lambda_{0,0} & \lambda_{0,1} & 0 & 0 & \lambda_{0,4} & \lambda_{0,5} & & & \tilde{\rho}_0^{(0)} \\
\lambda_{1,0} & 0 & \lambda_{1,2} & \lambda_{1,3} & \lambda_{1,4} & 0 & \mathbf{0} & & \tilde{\rho}_0^{(1)} \\
0 & \lambda_{2,1} & \lambda_{2,2} & \lambda_{2,3} & 0 & \lambda_{2,5} & & & \tilde{\rho}_0^{(2)} \\
 & & \lambda_{3,2} & \lambda_{3,3} & 0 & 0 & \lambda_{3,6} & \lambda_{3,7} & \tilde{\rho}_1^{(0)} \\
\mathbf{0} & & \lambda_{4,2} & 0 & \lambda_{4,4} & \lambda_{4,5} & \lambda_{4,6} & 0 & \tilde{\rho}_1^{(1)} \\
 & & 0 & \lambda_{5,3} & \lambda_{5,4} & \lambda_{5,5} & 0 & \lambda_{5,7} & \tilde{\rho}_1^{(2)} \\
\lambda_{6,0} & \lambda_{6,1} & & & \lambda_{6,4} & \lambda_{6,5} & 0 & 0 & \tilde{\rho}_2^{(0)} \\
\lambda_{7,0} & 0 & \mathbf{0} & & \lambda_{7,4} & 0 & \lambda_{7,6} & \lambda_{7,7} & \tilde{\rho}_2^{(1)} \\
0 & \lambda_{8,1} & & & 0 & \lambda_{8,5} & \lambda_{8,6} & \lambda_{8,7} & \tilde{\rho}_2^{(2)} \\
0 & 0 & \lambda_{9,2} & \lambda_{9,3} & & & \lambda_{9,6} & \lambda_{9,7} & \tilde{\rho}_3^{(0)} \\
\lambda_{10,0} & \lambda_{10,1} & \lambda_{10,2} & 0 & \mathbf{0} & & \lambda_{10,6} & 0 & \tilde{\rho}_3^{(1)} \\
\lambda_{11,0} & \lambda_{11,1} & 0 & \lambda_{11,3} & & & 0 & \lambda_{11,7} & \tilde{\rho}_3^{(2)} \\
\hline
\tilde{\psi}_0^{(0)} & \tilde{\psi}_0^{(1)} & \tilde{\psi}_1^{(0)} & \tilde{\psi}_1^{(1)} & \tilde{\psi}_2^{(0)} & \tilde{\psi}_2^{(1)} & \tilde{\psi}_3^{(0)} & \tilde{\psi}_3^{(1)} &
\end{array}$$

(b)

Fig. 3. Different variables used to compute the distances bounds. **(a)** Convolutional ensemble $\mathcal{C}_1$. **(b)** Tail-biting ensemble $\tilde{\mathcal{C}}_1$.

II. Asymptotic Distance Analysis

Classically, an important parameter that is used to characterize the error correction performance of a block code is its *minimum distance* d_{min}. The *Gilbert-Varshamov (GV) bound* [6] states that for a sufficiently large block length N there exists a linear binary block code with rate $0 < R < 1$ which has a minimum distance lower-bounded by $d_{\text{min}} \geq \alpha_{\text{GV}}(R)N$. The rate-dependent term $\alpha_{\text{GV}}(R)$ is called the *Gilbert-Varshamov parameter*. Analogously, the *Costello bound* [7] states that for a sufficiently large constraint length ν there exists a linear binary polynomially encoded convolutional code with rate $0 < R < 1$ which has a *free distance* d_{free} lower-bounded by $d_{\text{free}} \geq \alpha_{\text{C}}(R)\nu$. In this sense, ensembles of linear block/convolutional codes for which the minimum/free distances $d_{\text{min}}/d_{\text{free}}$ increase linearly with the block/constraint length N/ν are said to be *asymptotically good*.

Considering the LDPC block codes originally proposed by Gallager in [3] much work has been done on the study of the asymptotic behavior of minimum distances. For instance, Gallager himself analyzed regular ensembles of LDPC block codes composed by permutation matrices and showed that they are asymptotically good [3]. Further works for more general ensembles include, e.g., [8]–[10] and [11], [12], where the last two references are dedicated to protograph-based LDPC block code ensembles [13].

In this context, the free distance bounds of the LDPC convolutional codes [14] have also received attention. In [15], [16], combinatorial-probabilistic methods were developed for the analysis of regular LDPC convolutional codes composed by independent permutation matrices and their tail-biting versions [17]. Furthermore, the weight enumeration techniques presented in [11], [12] were applied in [18] for the analysis of protograph-based LDPC convolutional codes, which are derived from block codes using a parity-check matrix unwrapping procedure.

In this paper we extend the bounding methods of [15], [16] to the braided protograph convolutional code ensembles and their tail-biting versions. However, it is worth to mention that the technique that we will present in this section enables an asymptotic distance analysis of any protograph-based LDPC code construction (i.e., regular or irregular, block or convolutional) composed by independent permutation matrices.

A. Distance Bounding Technique

Without loss of generality, we explain the distance bounding technique using the code ensemble $\mathcal{C}_1$ resulting from the base code depicted in Fig. 1.

Let us consider the length $N = 3ML$ codeword

$$\mathbf{v}_{[0,L-1]} = (\mathbf{v}_0^{(0)}, \mathbf{v}_0^{(1)}, \mathbf{v}_0^{(2)}, \cdots, \mathbf{v}_{L-1}^{(0)}, \mathbf{v}_{L-1}^{(1)}, \mathbf{v}_{L-1}^{(2)})$$

where $\mathbf{v}_t^{(p)} = (v_{t,0}^{(p)}, \cdots, v_{t,M-1}^{(p)})$, with $v_{t,i}^{(p)} \in GF(2)$ for $t = 0, \cdots, L-1$, $i = 0, \cdots, M-1$ and $p = 0, 1, 2$. If $\mathbf{v}_{[0,L-1]}$ is a codeword of $\mathcal{C}_1$, the following equation must be satisfied:

$$\mathbf{v}_{[0,L-1]}\mathbf{H}^{\mathrm{T}}_{[0,L-1]} = \mathbf{0}, \tag{1}$$

where

$$\mathbf{H}^{\mathrm{T}}_{[0,L-1]} = [\mathbf{C}_0^{(0)}, \mathbf{C}_0^{(1)}, \cdots, \mathbf{C}_{L+1}^{(0)}, \mathbf{C}_{L+1}^{(1)}]$$

is the syndrome-former and each sub-matrix $\mathbf{C}_t^{(p)}$, $t = 0, \cdots, L+1$, $p = 0, 1$, is a column of independent $M \times M$ permutation matrices and all-zero matrices, as shown by Fig. 3(a) for $L = 4$.

The starting point of our bounding technique is the calculation of the probability that the codeword $\mathbf{v}_{[0,L-1]}$ satisfies the constraint equation expressed by (1). Because of the statistical independence of the permutation matrices in Fig. 3(a), the total constraint satisfaction probability can be expressed as the product of the probabilities that the codeword $\mathbf{v}_{[0,L-1]}$ satisfies the subset of constraints represented by each of the columns $\mathbf{C}_t^{(p)}$. In this context, the expression below shows the composition of the total constraint satisfaction probability for the ensemble $\mathcal{C}_1$:

$$\psi_{[0,L-1]} = \prod_{i=0}^{t=L+1} \prod_{p=0}^{1} \psi_t^{(p)}, \tag{2}$$

where $\psi_t^{(p)}$ are the probabilities that the subset of constraints represented by $\mathbf{C}_t^{(p)}$ are satisfied. The correspondences between $\psi_t^{(p)}$ and $\mathbf{C}_t^{(p)}$ are also shown in Fig. 3(a). The probabilities $\psi_t^{(p)}$ can be determined as a function of the Hamming weight composition

$$\mathbf{d}_{[0,L-1]} = (d_0^{(0)}, d_0^{(1)}, d_0^{(2)}, \cdots d_{L-1}^{(0)}, d_{L-1}^{(1)}, d_{L-1}^{(2)})$$

using the combinatorial approach presented in [15]. In this case, we have:

$$\psi_0^{(0)} = \psi_0^{(1)} = \binom{M}{M\rho_0^{(0)}}^{-1}, \tag{3}$$

$$\psi_{L+1}^{(0)} = \psi_{L+1}^{(1)} = \binom{M}{M\rho_{L-1}^{(0)}}^{-1}, \tag{4}$$

$$\psi_t^{(p)} \leq \left[\prod_{\{i,j:\rho_i^{(j)} \in \boldsymbol{\rho}_{4,t}^{(p)}\}} \binom{M}{M\rho_i^{(j)}} \right]^{-1} \exp\left(MG_4(\boldsymbol{\lambda}_{4,t}^{(p)}, \boldsymbol{\rho}_{4,t}^{(p)}) \right),$$
$$\text{for } t = 1, L \text{ and } p = 0, 1, \tag{5}$$

$$\psi_t^{(p)} \leq \left[\prod_{\{i,j:\rho_i^{(j)} \in \boldsymbol{\rho}_{6,t}^{(p)}\}} \binom{M}{M\rho_i^{(j)}} \right]^{-1} \exp\left(MG_6(\boldsymbol{\lambda}_{6,t}^{(p)}, \boldsymbol{\rho}_{6,t}^{(p)}) \right),$$
$$\text{for } 2 \leq t \leq L-1 \text{ and } p = 0, 1, \tag{6}$$

where $\boldsymbol{\lambda}_{4,t}^{(p)}$ and $\boldsymbol{\lambda}_{6,t}^{(p)}$ are the vectors containing the formal variables $\lambda_{i,j}$ corresponding to the columns $\mathbf{C}_t^{(p)}$. E.g., in Fig. 3(a) we have

$$\boldsymbol{\lambda}_{4,1}^{(0)} = (\lambda_{1,2}, \lambda_{2,2}, \lambda_{3,2}, \lambda_{4,2}) \quad \text{and}$$
$$\boldsymbol{\lambda}_{6,2}^{(0)} = (\lambda_{0,4}, \lambda_{1,4}, \lambda_{4,4}, \lambda_{5,4}, \lambda_{6,2}, \lambda_{7,4}).$$

The vectors $\boldsymbol{\rho}_{4,t}^{(p)}$ and $\boldsymbol{\rho}_{6,t}^{(p)}$ contain the normalized Hamming weight compositions (i.e., $\rho_t^{(p)} = d_t^{(p)}/M$ for $t = 0, \cdots, L-1$ and $p = 0, 1, 2$) corresponding to the columns $\mathbf{C}_t^{(p)}$. Taking the example from Fig. 3(a), we have $\boldsymbol{\rho}_{4,1}^{(0)} = (\rho_0^{(1)}, \rho_0^{(2)}, \rho_1^{(0)}, \rho_1^{(1)})$ and $\boldsymbol{\rho}_{6,2}^{(0)} = (\rho_0^{(0)}, \rho_0^{(1)}, \rho_1^{(1)}, \rho_1^{(2)}, \rho_2^{(0)}, \rho_2^{(1)})$. Finally, the function $G_K(\boldsymbol{\lambda}_{K,t}^{(p)}, \boldsymbol{\rho}_{K,t}^{(p)})$ is defined as:

$$G_K(\boldsymbol{\lambda}_{K,t}^{(p)}, \boldsymbol{\rho}_{K,t}^{(p)}) = -\sum_{i=0}^{K-1} \boldsymbol{\lambda}_{K,t}^{(p)}(i) \boldsymbol{\rho}_{K,t}^{(p)}(i) \tag{7}$$
$$+ \ln \frac{\prod_{i=0}^{K-1}(1 + e^{\boldsymbol{\lambda}_{K,t}^{(p)}(i)}) + \prod_{i=0}^{K-1}(1 - e^{\boldsymbol{\lambda}_{K,t}^{(p)}(i)})}{2},$$

where $\boldsymbol{\lambda}_{K,t}^{(p)}(i)$ ($\boldsymbol{\rho}_{K,t}^{(p)}(i)$) corresponds to the i-th element of the vector $\boldsymbol{\lambda}_{K,t}^{(p)}$ ($\boldsymbol{\rho}_{K,t}^{(p)}$).

Here it is worth to mention that, in contrast to [15], a more general arrangement of the syndrome former is assumed in Fig. 3(a). In this case, some all-zero matrices are placed between the permutation matrices. Furthermore, we extended this analysis to irregular code ensembles, as shown by the numerical results of Section II-C. Once that the probability $\psi_{[0,L-1]}$ has been determined, the expectation of the normalized Hamming weight composition $\boldsymbol{\rho}_{[0,L-1]} = \mathbf{d}_{[0,L-1]}/M$ is defined as follows:

$$E(\boldsymbol{\rho}_{[0,L-1]}) = \psi_{[0,L-1]} \prod_{t=0}^{L-1} \prod_{p=0}^{2} \binom{M}{M\rho_t^{(p)}}. \tag{8}$$

After substituting the expressions in (3)-(6) and applying the Lemma 1 of [15], the expectation above is upper-bounded by

$$E(\boldsymbol{\rho}_{[0,L-1]}) \leq \left(\prod_{t=0}^{L-1} \prod_{p=0}^{2} \sigma_t^{(p)} \right)^3 \exp\left(MF(\boldsymbol{\lambda}, \boldsymbol{\rho}_{[0,L-1]}) \right), \tag{9}$$

where $\sigma_t^{(p)} = \sqrt{2\pi e M \rho_t^{(p)}}$, the function $F(\boldsymbol{\lambda}, \boldsymbol{\rho}_{[0,L-1]})$ is given by

$$F(\boldsymbol{\lambda}, \boldsymbol{\rho}_{[0,L-1]}) = \sum_{p=0}^{1} G_4(\boldsymbol{\lambda}_{4,0}^{(p)}, \boldsymbol{\rho}_{4,0}^{(p)})$$
$$+ \sum_{t=1}^{L-2} \sum_{p=0}^{1} G_6(\boldsymbol{\lambda}_{6,t}^{(p)}, \boldsymbol{\rho}_{6,t}^{(p)}) + \sum_{p=0}^{1} G_4(\boldsymbol{\lambda}_{4,L-1}^{(p)}, \boldsymbol{\rho}_{4,L-1}^{(p)})$$
$$- 3 \sum_{t=0}^{L-1} \sum_{p=0}^{2} H(\rho_t^{(p)}) + 2H(\rho_0^{(0)}) + 2H(\rho_{L-1}^{(0)}), \tag{10}$$

and $H(\cdot)$ is the binary entropy function. The function $F(\boldsymbol{\lambda}, \boldsymbol{\rho}_{[0,L-1]})$ is the component that governs the upper bound given by (9). For instance, if $F(\boldsymbol{\lambda}, \boldsymbol{\rho}_{[0,L-1]}) < 0$ for some $\boldsymbol{\lambda}$, then the expected number of codewords with normalized weight composition $\boldsymbol{\rho}_{[0,L-1]}$ goes to zero exponentially with M as M tends to infinity.

Defining ρ_L^* as solution to the max-min optimization problem

$$\rho_L^* = \max\left\{ \rho_{\text{sum}} : \max_{\boldsymbol{\rho}_{[0,L-1]}} \left\{ \min_{\boldsymbol{\lambda}} \left\{ F(\boldsymbol{\lambda}, \boldsymbol{\rho}_{[0,L-1]}) < 0 \right\} \right\} \right\},$$

where $\rho_{\text{sum}} = \sum_{t=0}^{L-1} \sum_{p=0}^{2} \rho_t^{(p)}$, allows us to state the following theorem.

Theorem 1: In the ensemble $\mathcal{C}_1$, there exists a convolutional code with segment distance lower-bounded by

$$d_L \geq \rho_L^* M = (\rho_L^*/9)\nu_s, \tag{11}$$

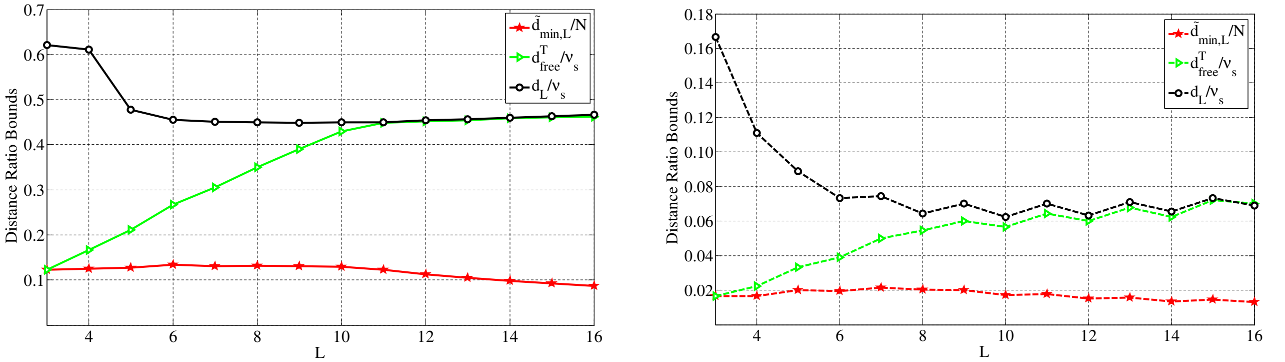
where $\nu_s = 9M$ is the constraint length of the convolutional code. □

Similar to [15], we have the following lower bound on the free distance of the ensemble $\mathcal{C}_1$.

Theorem 2: In the ensemble $\mathcal{C}_1$, there exists a convolutional code with free distance lower-bounded by

$$d_{\text{free}} \geq (\rho^*/9)\nu = \alpha_{\mathcal{C}_1}\nu, \tag{12}$$

where ρ_L^* converges to ρ^* as L tends to infinity. □



(a)

(b)

Fig. 4. Distance to constraint length (block length) ratio bounds. **(a)** Distance ratio bounds for the regular ensembles $\mathcal{C}_1$ and $\tilde{\mathcal{C}}_1$. **(b)** Distance ratio bounds for the irregular ensembles $\mathcal{C}_2$ and $\tilde{\mathcal{C}}_2$. d_L/ν_s is the segment distance to constraint length ratio of the convolutional code. $\tilde{d}_{\text{min},L}/N$ is the minimum distance to block length ratio of the tail-biting code. $d^{\text{T}}_{\text{free}}/\nu_s$ is the free distance to constraint length ratio of the convolutional code obtained from the tail-biting code.

B. Distance Bounds for Tail-Biting Ensembles

The techniques shown above can be applied to lower-bound the minimum distances of tail-biting codes in an analogous way. In order to simplify the explanation, we use the tail-biting code ensemble $\tilde{\mathcal{C}}_1$, which is derived from the convolutional code ensemble $\mathcal{C}_1$.

In this case, the total constraint satisfaction probability for ensemble $\tilde{\mathcal{C}}_1$ is given by

$$\tilde{\psi}_{[0,L-1]} = \prod_{i=0}^{t=L-1} \prod_{p=0}^{1} \tilde{\psi}_t^{(p)}, \tag{13}$$

and the probabilities $\tilde{\psi}_t^{(p)}$ are given by

$$\tilde{\psi}_t^{(p)} \le \left[\prod_{\{i,j:\tilde{\rho}_i^{(j)} \in \tilde{\boldsymbol{\rho}}_{6,t}^{(p)}\}} \binom{M}{M\tilde{\rho}_i^{(j)}} \right]^{-1} \exp\left(MG_6(\tilde{\boldsymbol{\lambda}}_{6,t}^{(p)}, \tilde{\boldsymbol{\rho}}_{6,t}^{(p)}) \right),$$
$$\text{for } 0 \le t \le L-1, \text{ and } p = 0, 1, \tag{14}$$

where the correspondences between the variables from above and the parity-check matrix are shown in Fig. 3(b). The upper bound on the expectation of the normalized Hamming weight composition is similar to (9). However, the function multiplied by M within the exponent is given by

$$\tilde{F}(\boldsymbol{\lambda}, \tilde{\boldsymbol{\rho}}_{[0,L-1]}) = \sum_{t=0}^{L-1} \sum_{p=0}^{1} G_6(\boldsymbol{\lambda}_{6,t}^{(p)}, \tilde{\boldsymbol{\rho}}_{6,t}^{(p)}) - 3 \sum_{t=0}^{L-1} \sum_{p=0}^{2} H(\tilde{\rho}_t^{(p)}). \tag{15}$$

Furthermore, if we define $\tilde{\rho}_L^*$ as solution to the max-min optimization problem

$$\tilde{\rho}_L^* = \max \left\{ \tilde{\rho}_{\text{sum}} : \max_{\tilde{\boldsymbol{\rho}}_{[0,L-1]}} \left\{ \min_{\boldsymbol{\lambda}} \left\{ \tilde{F}(\boldsymbol{\lambda}, \tilde{\boldsymbol{\rho}}_{[0,L-1]}) < 0 \right\} \right\} \right\},$$

where $\tilde{\rho}_{\text{sum}} = \frac{1}{3L} \sum_{t=0}^{L-1} \sum_{p=0}^{2} \tilde{\rho}_t^{(p)}$, the theorem below can be stated.

Theorem 3: In the ensemble $\tilde{\mathcal{C}}_1$, there exists a tail-biting convolutional code with minimum distance lower-bounded by

$$\tilde{d}_{\text{min},L} \ge \tilde{\rho}_L^* 3ML = \alpha_{\tilde{\mathcal{C}}_1} N, \tag{16}$$

where $N = 3ML$ is the block length. □

C. Numerical Results

Using the bounding techniques presented above, we calculated lower bounds on the distances for the ensembles $\mathcal{C}_1$ and $\tilde{\mathcal{C}}_1$ that have been used as examples in the previous section, and also for the ensembles $\mathcal{C}_2$ and $\tilde{\mathcal{C}}_2$, which are constructed using component codes with polynomial parity-check matrix given by $\mathbf{H}(D) = [1 + D, 1, 1 + D + D^2]$. $\mathcal{C}_1$ and $\tilde{\mathcal{C}}_1$ are regular ensembles with degree distribution $(4, 6)$, while $\mathcal{C}_2$ and $\tilde{\mathcal{C}}_2$ are irregular and have degree distribution given by $(\frac{1}{3}x + \frac{2}{3}x^4, 6)$.

Fig. 4(a) and Fig. 4(b) show the asymptotic ratios d_L/ν_s for the ensembles $\mathcal{C}_1$ and $\mathcal{C}_2$. We also calculated lower bounds on the minimum distance $\tilde{d}_{\text{min},L}$ for the tail-biting ensembles $\tilde{\mathcal{C}}_1$ and $\tilde{\mathcal{C}}_2$ with block length $N = 3ML$ and period $T = L$. It has been shown in [16] that the free distance d^T_{free} of the corresponding periodically time-varying convolutional code of period T and constraint length $\nu_s = 9M$ is lower-bounded by $\tilde{d}_{\text{min},L}$ for each L. The distance ratios $\tilde{d}_{\text{min},L}/N$ and d^T_{free}/ν_s, $T = L$, for the tail-biting ensembles $\tilde{\mathcal{C}}_1$ and $\tilde{\mathcal{C}}_2$ and their mother convolutional ensembles $\mathcal{C}_1$ and $\mathcal{C}_2$, respectively, are also shown in Fig. 4(a) and Fig. 4(b). It can be observed that for large L the bound on d^T_{free}/ν_s of the periodically time-varying codes approaches the same value as the bound on d_L/ν_s of the non-periodic codes, which approaches d_{free}/ν_s as $L \to \infty$. On the other hand, for the shortest tail-biting code block length $N = 9M$ the bound on $\tilde{d}_{\text{min},3}/N$ of the regular ensemble becomes equal to the distance ratio of Gallager's regular $(4, 6)$ block codes. Interestingly, we can also observe that for the irregular ensembles in Fig. 4(b), the curves of the distance ratios show a zig-zag behavior as if there were different ensembles for odd L and for even L. The results also show that if M

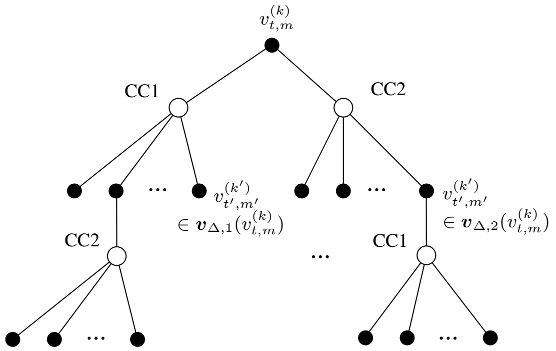

$v^{(k)}_{t,m}$
CC1
CC2
$v^{(k')}_{t',m'}$
$\in \boldsymbol{v}_{\Delta,1}(v^{(k)}_{t,m})$
$v^{(k')}_{t',m'}$
$\in \boldsymbol{v}_{\Delta,2}(v^{(k)}_{t,m})$
CC2
CC1

Fig. 5. Computation tree for iterative decoding of a symbol $v^{(k)}_{t,m}$. Solid circles denote symbol nodes, empty circles constraint nodes.

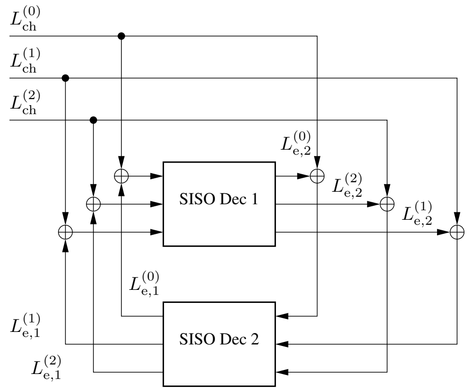


Fig. 6. Illustration of the LLRs exchanged in the turbo-like decoder.

is fixed but L is increased up to $L = 10$, the minimum distance still grows linearly with N even if the constraint length of the associated convolutional code is kept constant. This means that tail-biting codes get stronger with larger L (i.e., $3 < L \leq 10$ in this example) although the structure of the graph is maintained and the parity-check matrix gets sparser.

The presented distance bounds correspond to the unconstrained codes where all permutation matrices are independent from each other. For the constrained codes some of the permutations matrices are equal, what complicates the application of combinatorial approach we presented above. We are currently working on this problem, however, we conjecture that the constrained ensembles will also have asymptotic distances with linear growth behavior.

III. Asymptotic Analysis of Turbo-Like Decoding

A. Convergence Behavior of (G)LDPC Codes and Turbo Codes

A measure for the asymptotically achievable performance of iterative decoding is the convergence threshold. It is an upper bound on the smallest signal-to-noise ratio (SNR) for which the decoding error probability converges to zero with the number of iterations for a specific class of codes and decoding algorithms. To evaluate such thresholds, the probability density functions (PDFs) of the messages that are exchanged within the decoder can be tracked as function of the iterations (density evolution). Estimates of the threshold values for braided protograph convolutional codes have been presented in [5].

The evolution of the Bhattacharyya parameter during density evolution has been investigated in [19], resulting in an upper-bound on the rate at which the decoding error probability converges to zero for SNRs above the threshold. For (G)LDPC codes the bit error probability can be upper-bounded by

$$P_{\mathrm{b}} < \exp\Big(-a\big((d_{\min}-1)(J-1)\big)^{I}\Big) \ , \qquad (17)$$

where a is some positive constant, $d_{\min}$ is the smallest minimum distance among the component codes, J is the number of component codes per symbol (i.e., the variable node degree), and I is the number of decoding iterations. Note that a doubly exponential decrease requires either $d_{\min} > 2$ or $J > 2$, which indicates that the number of component code per symbol can be traided against their strength. For (multiple) turbo codes, on the other hand, the bit error probability is upper-bounded by

$$P_{\mathrm{b}} < \exp\left(-a\Delta\right) + \exp\left(-bI\right) \ , \qquad J = 2 \ , \qquad (18)$$

$$P_{\mathrm{b}} < \exp\left(-a'\Delta\right) + \exp\left(-b'\Delta I\right) \ , \qquad J > 2 \ , \qquad (19)$$

where a, a', b, b' are some positive constants and Δ is the size of the decoding window, specifying how many neighboring sections in the component decoder trellis are taken into account. The weaker convergence rate of turbo codes is related to their property that parity-check symbol estimates are not improved during the decoding iterations. From this also follows, that the free distance d_{free} of the component codes does not appear in (18) and (19).

In this paper, we derive corresponding results for turbo-like iterative decoding of braided protograph convolutional codes.

B. Convergence Analysis of Braided Protograph Convolutional Codes

To analyze iterative decoding of a particular symbol $v^{(k)}_{t,m}$, it is convenient to consider a computation tree around that symbol, as illustrated in Fig. 5. The computation tree shows, how different code symbols contribute to the decoding of the symbol at its root during the iterations $i = 1, \ldots, I$. The decoder proceeds iteration by iteration upwards through this tree, starting from symbol node level $\ell_0 = I$. Let $\boldsymbol{v}_{\Delta,j}(v^{(k)}_{t,m})$ denote the set of $3(2\Delta+1)$ code symbols in the Δ-truncated trellis of component decoder j around $v^{(k)}_{t,m}$. The constraint nodes in the tree represent the rule that the symbols $\boldsymbol{v}_{\Delta,j}(v^{(k)}_{t,m})$ have to correspond to valid paths in that trellis.

The messages from constraint nodes to variable nodes are defined as the extrinsic LLRs at the output of the soft-input component APP decoders. As illustrated in Fig. 6, at the output of each component decoder $j = 1, 2$ there exist three types of messages $L^{(k)}_{\mathrm{e},j}$ corresponding to the different symbol types $v_{t,m}^{(0)}$, $v_{t,m}^{(1)}$ and $v_{t,m}^{(2)}$. The density evolution approach relies on the assumption that all messages exchanged during the iterations are statistically independent. This is the case if within every computation tree all symbol nodes up to level ℓ_0 correspond to different code symbols. Under this assumption the message distributions are independent of time t and position m. To simplify notation, when the context allows we will omit the indices t and m. In order to make such an independence possible, following the approach in [19], we consider in our analysis a windowed BCJR decoder that operates on a truncated trellis of size $2\Delta+1$, centered at the symbol to be decoded.

The messages $L_{i,j}^{(k)}$ that are passed to the input of component decoder j after iteration i can be written as

$$L_{i,j}^{(k)}(v^{(k)}) = L_{\mathrm{ch}}^{(k)} + \log \frac{\sum_{\boldsymbol{v}_{\Delta,j}:v^{(k)}=0} \Pr(\boldsymbol{v}_{\Delta,\bar{j}}|\boldsymbol{L}_{i-1,\bar{j}})}{\sum_{\boldsymbol{v}_{\Delta,j}:v^{(k)}=1} \Pr(\boldsymbol{v}_{\Delta,\bar{j}}|\boldsymbol{L}_{i-1,\bar{j}})} \ , \tag{20}$$

where $L_{\mathrm{ch}}^{(k)}$ is the intrinsic LLR of $v^{(k)}$, $\boldsymbol{L}_{i-1,\bar{j}}$ is the set of LLRs corresponding to $\boldsymbol{v}_{\Delta,\bar{j}}$, and $\bar{j}=\{1,2\}\setminus j$ denotes the complement of j. The second term in (20) is the extrinsic output LLR computed by the other component APP decoder.

In order to estimate the decoding error probability, we observe the evolution of the Bhattacharyya parameters of the LLRs $L_{i,j}^{(k)}$ during the decoding iterations, defined as

$$B_{i,j}^{(k)} = E\left[\exp\left(-L_{i,j}^{(k)}/2\right)\right] \ , \tag{21}$$

where the expectation is over the distribution of $L_{i,j}^{(k)}$ conditioned on the all-zero transmitted sequence. To derive explicit bounds on $B_{i,j}^{(k)}$, we introduce the *windowed path enumerator* functions for the Δ-truncated component decoder trellises around a symbol $v_{t,m}^{(k)}$

$$F_{\Delta,j}^{(k)}\left(B_0,B_1,B_2\right) = \sum_{d_0,d_1,d_2} \alpha_\Delta\left(d_0,d_1,d_2\right) {B_0}^{d_0} {B_1}^{d_1} {B_2}^{d_2} \ ,$$

where B_k and d_k are the Bhattacharyya parameters and trellis path weights corresponding to symbols $v^{(k)}$, respectively, and α_Δ denotes the number of paths in the set $\hat{\mathcal{S}}(v_{t,m}^{(k)}=1)$ with weights d_k (the symbol $v_{t,m}^{(k)}=1$ itself is excluded in the weight d_k). A state segment (corresponding to a trellis path) is called *convex* if all states between an arbitray pair of nonzero states are nonzero as well, and $\hat{\mathcal{S}}(v_{t,m}^{(k)}=1)$ is defined as the set of all convex state segments in the trellis with $v_{t,m}^{(k)}=1$ [19]. The windowed path enumerator functions for a fixed Δ and for the asymptotic case $\Delta\to\infty$ can be obtained by transfer function methods from a state transition description of the encoder.

Example 1: For an encoder j with polynomial parity-check matrix $\mathbf{H}(D)=[1,1+D^2,1+D+D^2]$, implemented in observer canonical form, the transfer function matrix $\boldsymbol{T}$ is given by

$$\boldsymbol{T} = \begin{bmatrix} 1 & B_0B_1 & B_1B_2 & B_0B_2 \\ B_1B_2 & B_0B_2 & 1 & B_0B_1 \\ B_0 & B_1 & B_0B_1B_2 & B_2 \\ B_0B_1B_2 & B_2 & B_0 & B_1 \end{bmatrix}$$

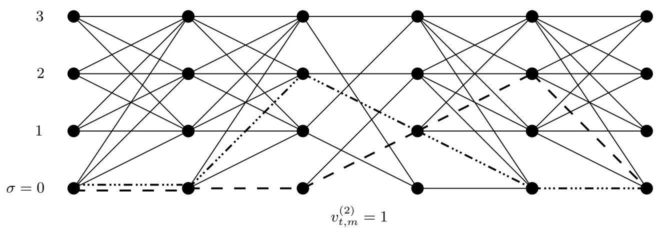


Fig. 7. Trellis paths of the convex state segments $\boldsymbol{\sigma}\in\hat{\mathcal{S}}(v_{t,m}^{(2)}=1)$ for a memory $m_{\mathrm{cc}}=2$ encoder and $\Delta=2$. The highlighted two paths are the paths of minimum weight $d_{\mathrm{free}}=3$ that remain when $\Delta\to\infty$.

The corresponding trellis paths contributing to $F_{\Delta,j}^{(k)}$ are shown in Fig. 7 for $k=2$ and $\Delta=2$. □

The following lemma connects the Bhattacharyya parameters of two consecutive decoding iterations.

Lemma 1: The Bhattacharyya parameters $B_{i,j}^{(k)}$, $k=0,1,2$, corresponding to the LLRs passed to component decoder j, $j=1,2$, after iteration i, $i=0,\ldots,I$, satisfy the following inequalities

$$\begin{aligned}
B_{i,1}^{(0)} &< A\,F_{\Delta,2}^{(0)}\left(B_{i-1,2}^{(0)},B_{i-1,2}^{(1)},B_{i-1,2}^{(2)}\right)\\
B_{i,2}^{(0)} &< A\,F_{\Delta,1}^{(0)}\left(B_{i-1,1}^{(0)},B_{i-1,1}^{(2)},B_{i-1,1}^{(1)}\right)\\
B_{i,1}^{(1)} &< A\,F_{\Delta,2}^{(1)}\left(B_{i-1,2}^{(0)},B_{i-1,2}^{(1)},B_{i-1,2}^{(2)}\right)\\
B_{i,2}^{(1)} &< A\,F_{\Delta,1}^{(2)}\left(B_{i-1,1}^{(0)},B_{i-1,1}^{(2)},B_{i-1,1}^{(1)}\right)\\
B_{i,1}^{(2)} &< A\,F_{\Delta,2}^{(2)}\left(B_{i-1,2}^{(0)},B_{i-1,2}^{(1)},B_{i-1,2}^{(2)}\right)\\
B_{i,2}^{(2)} &< A\,F_{\Delta,1}^{(1)}\left(B_{i-1,1}^{(0)},B_{i-1,1}^{(2)},B_{i-1,1}^{(1)}\right),
\end{aligned}$$

where A is the Bhattacharyya parameter of the intrinsic LLRs $L_{\mathrm{ch}}^{(k)}$, and $B_{0,j}^{(k)}=A$. □

Each of the inequalities given in Lemma 1 can be proved analogously to Lemma 3 in [19]. The essential difference is that several different types of symbols with different Bhattacharyya parameters and windowed path enumerators have to be considered. Their interrelations can be deduced from Fig. 6. Assuming equal component encoders, so that $F_{\Delta,1}^{(k)}=F_{\Delta,2}^{(k)}\overset{\mathrm{def}}{=}F_{\Delta}^{(k)}$, the outputs at the two decoders are identically distributed, i.e.,

$$\begin{aligned}
B_i^{(0)} &\overset{\mathrm{def}}{=} B_{i,1}^{(0)} = B_{i,2}^{(0)}\\
B_i^{(1)} &\overset{\mathrm{def}}{=} B_{i,1}^{(1)} = B_{i,2}^{(2)}\\
B_i^{(2)} &\overset{\mathrm{def}}{=} B_{i,1}^{(2)} = B_{i,2}^{(1)} \ .
\end{aligned}$$

Corollary 1: If the two component encoders are equal, then the Bhattacharyya parameters after iteration i, $i=1,\ldots,I$, satisfy the inequality

$$B_i^{(k)} < A\,F_{\Delta}^{(k)}\left(B_{i-1}^{(0)},B_{i-1}^{(2)},B_{i-1}^{(1)}\right) \quad , \quad k=0,1,2 \ , \tag{22}$$

from which it follows that

$$B_i^{(k)} \le B_i^{\max} < \max_k A\,F_{\Delta}^{(k)}\left(B_{i-1}^{\max}\right) \ , \tag{23}$$

where

$$B_i^{\max} = \max_k B_{i-1}^{(k)} \ ,$$

and

$$F_\Delta^{(k)}(B) \overset{\text{def}}{=} F_\Delta^{(k)}(B_0, B_1, B_2)\Big|_{B_0=B_1=B_2=B} \ .$$

□

Suppose now that after some iteration $I_1 < I$ all three Bhattacharyya parameters $B_{I_1}^{(k)}$ are smaller than the *breakout value* B_{br}, defined by the equation

$$B_{\text{br}} = \max_k A\, F_\Delta^{(k)}(B_{\text{br}}) \ . \qquad (24)$$

Then it follows from Corollary 1 that $B_i^{(k)}$ decreases with $i = I_1+1, \ldots, I$ and converges to zero as I tends to infinity. For two encoders equal to that in Example 1 with $\Delta = 4$, the windowed path enumerators and the breakout value B_{br} are illustrated in Fig. 8. The value A was selected for a channel with $E_{\text{b}}/N_0 = 1.1$ dB, which is the estimated convergence threshold for this code [5].

For analyzing the rate of convergence, we introduce a majorant function

$$\hat{f}_\Delta(B) \overset{\text{def}}{=} \hat{\beta} B^{d_{\text{free}}-1} \geq \max_k A\, F_\Delta^{(k)}(B) \ , \ 0 \leq B \leq B_{\text{br}} \ ,$$

where the value $\hat{\beta} = B_{\text{br}}^{d_{\text{free}}-2}$ is chosen such that equality holds for $B = B_{\text{br}}$ (see Fig. 8). Note that $d_{\text{free}} - 1$ is the lowest order in the windowed path enumerators $F_\Delta^{(k)}(B)$ if Δ is chosen sufficiently large. Consider now $B_{I_1}^{\max} < B_{\text{br}}$, which is the largest of the Bhattacharyya parameters after decoding iteration I_1. It follows from (23) that

$$B_i^{\max} < \hat{f}_\Delta\left(B_{i-1}^{\max}\right) \ , \qquad (25)$$

and a recursive application results in

$$B_I^{\max} < B_{\text{br}} \left(\left(B_{I_1}^{\max}/B_{\text{br}}\right)^{d_{\text{free}}-1}\right)^{I-I_1} \ . \qquad (26)$$

The corresponding trajectory is also illustrated in Fig. 8.

Let $P_i^{\max}$ denote the maximal error probability P_i^k that results if hard decisions are made from the LLRs $L_{i,j}^{(k)}$. From the definition of $B_I^{\max}$ it follows that

$$P_{\text{b}} < B_I^{\max} < \sqrt{4P_i^{\max}} \ . \qquad (27)$$

The following Theorem formulates the main result of this section, which follows from (26) and (27).

Theorem 4: Consider turbo-like iterative decoding of braided protograph convolutional codes with two identical component encoders, where transmission takes place over a binary-input output-symmetric memoryless channel. Suppose that a total of I independent decoding iterations are performed and that after the first $I_1 < I$ iterations, the hard decision error probabilites of all symbols become less than $B_{\text{br}}^2/4$. Then the decoding bit error probability P_{b} is upper-bounded by

$$P_{\text{b}} < \exp\left(-a(d_{\text{free}}-1)^I\right) \ , \qquad (28)$$

where a is a positive constant and d_{free} is the free distance of the component codes. □

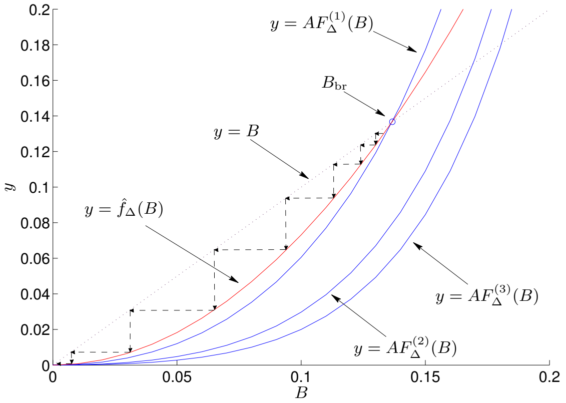


Fig. 8. Upper-bounding the evolution of $B_i^{\max}$ by the majorant function $\hat{f}_\Delta(B)$.

IV. Conclusions

The distance and convergence properties of braided protograph convolutional codes have been analyzed for the case when the size M of the lifting matrices tends to infinity. The presented distance bounding technique generalizes some earlier results on permutation-based regular LDPC convolutional code ensembles to the case of irregular base code graphs, and it can be applied to a wider range of code ensembles. The convergence analysis of braided protograph convolutional codes shows that for turbo-like decoding, like in the case of (generalized) LDPC codes, a doubly exponential convergence rate is achieved. This is a clear advantage compared to (multiple) turbo codes and can be prescribed to the particular way the component convolutional codes are concatenated, which is closely related to (G)LDPC codes.

V. Acknowledgments

The authors are grateful for the use of the high performance computing facilities of the ZIH at the Technische Universität Dresden.

References

[1] W. Zhang, M. Lentmaier, K.Sh. Zigangirov, and D.J. Costello, Jr., "Braided convolutional codes: a new class of turbo-like codes," *IEEE Trans. Inform. Theory*, May 2006, submitted for publication.

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

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

[4] M. Lentmaier and K. Sh. Zigangirov, "Iterative decoding of generalized low-density parity-check codes," in *Proc. IEEE International Symposium on Information Theory*, Boston, USA, August 1998, p. 149.

[5] M.B.S. Tavares, M. Lentmaier, K.Sh. Zigangirov, and G.P. Fettweis, "LDPC convolutional codes based on braided convolutional codes," in *Proc. of IEEE International Symposium of Information Theory (ISIT'08)*, Toronto, Canada, July 2008.

[6] W.W. Peterson and E.J. Weldon, Jr., *Error-Correcting Codes*. Cambridge, MA: MIT Press, 1972.

[7] D.J. Costello, Jr., "Free distance bounds for convolutional codes," *IEEE Trans. Inform. Theory*, vol. 20, no. 3, pp. 356–365, May 1974.

[8] S. Litsyn and V. Shevelev, “On ensembles of low-density parity-check codes: Asymptotic distance distributions,” *IEEE Trans. Inform. Theory*, vol. 48, no. 4, pp. 887–908, Apr. 2002.

[9] ——, “Distance distributions in ensembles of irregular low-density parity-check codes,” *IEEE Trans. Inform. Theory*, vol. 49, no. 12, pp. 3140–3159, Dec. 2003.

[10] D. Burshtein and G. Miller, “Asymptotic enumeration methods for analysing LDPC codes,” *IEEE Trans. Inform. Theory*, vol. 50, no. 6, pp. 1115–1131, June 2004.

[11] S.L. Fogal, R. McEliece, and J. Thorpe, “Enumerators for protograph ensambles of LDPC codes,” in *Proc. IEEE International Symposium on Information Theory*, Adelaide, Australia, Sept. 2005, pp. 2156–2160.

[12] D. Divsalar, “Ensemble weigth enumerators for protograph LDPC codes,” in *Proc. IEEE International Symposium on Information Theory*, Seattle, USA, July 2006, pp. 1554–1558.

[13] J. Thorpe, “Low-density parity-check (LDPC) codes constructed from protographs,” in *IPN Progress Report 42-154, JPL*, Aug. 2005.

[14] A. Jiménez Feltström and K.Sh. Zigangirov, “Periodic time-varying convolutional codes with low-density parity-check matrices,” *IEEE Trans. Inform. Theory*, vol. 45, no. 5, pp. 2181–2190, Sept. 1999.

[15] A. Sridharan, D.V. Truhachev, M. Lentmaier, D.J. Costello, Jr., and K.Sh. Zigangirov, “Distance bounds for an ensemble of LDPC convolutional codes,” *IEEE Trans. Inform. Theory*, vol. 53, no. 12, Dec. 2007.

[16] D.V. Truhachev, K.Sh. Zigangirov, and D.J. Costello, Jr., “Distance bounds for periodically time-varying and tail-biting LDPC convolutional codes,” Submitted to *IEEE Transactions on Information Theory*, Dec. 2007.

[17] M.B.S. Tavares, K.Sh. Zigangirov, and G.P. Fettweis, “Tail-biting LDPC convolutional codes,” in *Proc. of IEEE International Symposium of Information Theory (ISIT’07)*, Nice, France, June 2007.

[18] D.G.M. Mitchell, A.E. Pusane, K.Sh. Zigangirov, and D.J. Costello, Jr., “Asymptotically good LDPC convolutional codes based on protographs,” Accepted for publication in *Proc. of the IEEE International Symposium on Information Theory*, July 2008.

[19] M. Lentmaier, D.V. Truhachev, K.Sh. Zigangirov, and D.J. Costello, Jr., “An analysis of the block error probability performance of iterative decoding,” *IEEE Trans. Inform. Theory*, vol. 51, no. 11, pp. 3834–3855, Nov. 2005.