LUND UNIVERSITY

LDPC Convolutional Codes Based on Braided Convolutional Codes

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

Link to publication

LDPC Convolutional Codes
Based on Braided Convolutional Codes

Marcos B.S. Tavares†, Michael Lentmaier†, Kamil Sh. Zigangirov‡ and Gerhard P. Fettweis†

† Vodafone Chair Mobile Communications Systems, Technische Universität Dresden, 01062 Dresden, Germany
Emails: {tavares, michael.lentmaier, fettweis}@ifn.et.tu-dresden.de
‡ Department of Electrical Engineering, Unviversity of Notre Dame, Notre Dame, IN 46556, USA
Email: kzigangi@nd.edu

Abstract— We introduce and analyze new constructions of LDPC convolutional codes and their tail-biting versions which are obtained from braided convolutional codes. The basic ideas behind the encoding and decoding architectures for these codes are presented. Additionally, asymptotic results concerning the iterative thresholds are shown for different ensembles. Finally, we evaluate the bit error rate performances of several codes by means of computer simulations.

I. INTRODUCTION

At the present time, the *low-density parity-check* (LDPC) codes invented by Gallager [1] and the *turbo codes* (TCs) proposed by Berrou *et al.* [2] are attracting very much attention of coding theorists and practitioners. The interest in these codes can be justified by their capacity approaching performance and low-complexity decoding algorithms.

In spite of their relatively good iterative thresholds, fast convergence and low encoding/decoding complexity [3], the conventional TCs are known for having poor distance properties. Motivated by the results shown in [4] – in which it can be observed that it is possible to construct turbo-like codes with good distance properties – other schemes are being investigated.

In this paper, we propose a new code construction method that uses conventional *convolutional codes* (CCs) as building elements for the construction of LDPC codes. This method is similar to the construction of LDPC codes based on protographs [5] and represents a new attempt to obtain code ensembles with good distance properties, good iterative thresholds and low-complexity encoding/decoding from simple codes. As we will see in the next sections, the codes obtained through this construction technique belong to the class of *LDPC convolutional codes* (LDPC-CCs) [6] and we call them *braided LDPC-CCs*. In addition to inheriting the properties belonging to LDPC-CCs (i.e., variable block lengths, linear time encoding based on shift-registers and pipeline decoding), the braided LDPC-CCs can be decoded with Gallager's *belief propagation* (BP) algorithm [1] or with the BCJR algorithm [7] depending on the constraints imposed on their syndrome former matrices. Similarly to [8] and [9], tail-biting versions of the braided LDPC-CCs can be derived to avoid the rate-loss problem due to the termination overhead.

This paper is organized as follows. In section II, we show the construction method and definition of the braided LDPC-

CCs and their tail-biting versions. In section III, we discuss the encoding architectures for these codes. Section IV describes the decoding architectures. In section V, iterative thresholds are computed for two ensembles of braided LDPC-CCs. The section VI presents simulation results on the error correcting capability of our codes. Finally, section VII concludes the paper.

II. CONSTRUCTION AND DEFINITION

To explain our construction method, we consider an example. For instance, we take the rate $R = 2/3$ CC with polynomial parity-check matrix given by $\mathbf{H}(D) = [1, 1+D^2, 1+D+D^2]$. Then, we connect two of such codes as shown by Fig. 1(a) and obtain a *tightly braided convolutional code* (TBCC) [4] with repeated parity-bits (i.e., $v_t^{(1)} = v_t^{(2)}$) that has its *syndrome former* $\bar{\mathbf{H}}_{[0,\infty]}^{\mathrm{T}}$ depicted in Fig. 1(b). The syndrome former of the associated braided LDPC-CC is obtained by substituting all 1's and 0's in $\bar{\mathbf{H}}_{[0,\infty]}^{\mathrm{T}}$ by *permutation matrices* and *all-zero matrices*, respectively. Observe that the blank cells in $\bar{\mathbf{H}}_{[0,\infty]}^{\mathrm{T}}$ also represent zeros and are originated due to the causality of the encoder.

In order to simplify notations, we give below the formal definition for braided LDPC-CCs and their tail-biting versions with rates $R = 1/3$, which are constructed from CCs of rate

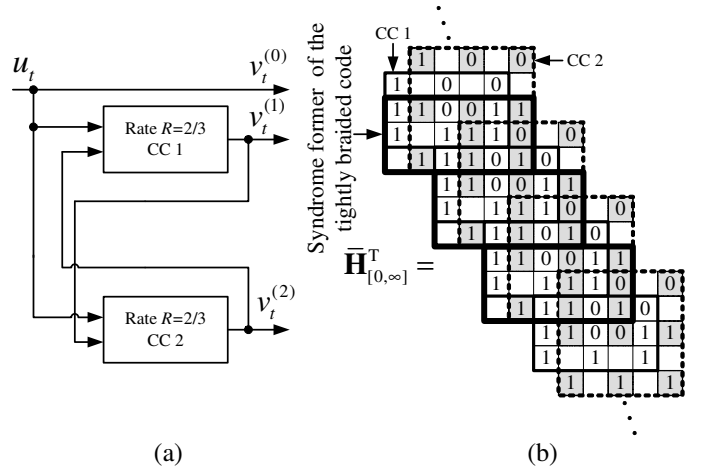Fig. 1. Tightly braided convolutional code (TBCC) [4] with repeated parity-bit used in the construction of braided LDPC-CCs. **(a)** Encoder for the TBCC. **(b)** Composition of the syndrome former $\bar{\mathbf{H}}_{[0,\infty]}^{\mathrm{T}}$ of the TBCC based on the syndrome formers of the component convolutional codes CC 1 (continuous rectangles, white cells) and CC 2 (dashed rectangles, gray cells).

$$\mathbf{H}^{\mathrm{T}}_{[0,\infty]} = \begin{pmatrix} \mathbf{P}^0_{\mathbf{v}_0}(0) & \mathbf{P}^1_{\mathbf{v}_0}(0) & \mathbf{0} & \mathbf{0} & \mathbf{P}^4_{\mathbf{v}_0}(0) & \mathbf{P}^5_{\mathbf{v}_0}(0) & \mathbf{0} & \mathbf{0} \\ \mathbf{P}^0_{\mathbf{v}_1}(0) & \mathbf{0} & \mathbf{P}^2_{\mathbf{v}_1}(0) & \mathbf{P}^3_{\mathbf{v}_1}(0) & \mathbf{P}^4_{\mathbf{v}_1}(0) & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{P}^1_{\mathbf{v}_2}(0) & \mathbf{P}^2_{\mathbf{v}_2}(0) & \mathbf{P}^3_{\mathbf{v}_2}(0) & \mathbf{0} & \mathbf{P}^5_{\mathbf{v}_2}(0) & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{P}^0_{\mathbf{v}_0}(1) & \mathbf{P}^1_{\mathbf{v}_0}(1) & \mathbf{0} & \mathbf{0} & \mathbf{P}^4_{\mathbf{v}_0}(1) & \mathbf{P}^5_{\mathbf{v}_0}(1) \\ \mathbf{0} & \mathbf{0} & \mathbf{P}^0_{\mathbf{v}_1}(1) & \mathbf{0} & \mathbf{P}^2_{\mathbf{v}_1}(1) & \mathbf{P}^3_{\mathbf{v}_1}(1) & \mathbf{P}^4_{\mathbf{v}_1}(1) & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{P}^1_{\mathbf{v}_2}(1) & \mathbf{P}^2_{\mathbf{v}_2}(1) & \mathbf{P}^3_{\mathbf{v}_2}(1) & \mathbf{0} & \mathbf{P}^5_{\mathbf{v}_2}(1) \\ & & & & \ddots & & \ddots & & \ddots \end{pmatrix} \tag{1}$$

$$\tilde{\mathbf{H}}^{\mathrm{T}}_{[0,L-1]} = \begin{pmatrix} \mathbf{P}^0_{\mathbf{v}_0}(0) & \mathbf{P}^1_{\mathbf{v}_0}(0) & \mathbf{0} & \mathbf{0} & \mathbf{P}^4_{\mathbf{v}_0}(0) & \mathbf{P}^5_{\mathbf{v}_0}(0) & \cdots & \cdots & \cdots & \mathbf{0} \\ \mathbf{P}^0_{\mathbf{v}_1}(0) & \mathbf{0} & \mathbf{P}^2_{\mathbf{v}_1}(0) & \mathbf{P}^3_{\mathbf{v}_1}(0) & \mathbf{P}^4_{\mathbf{v}_1}(0) & \mathbf{0} & \cdots & \cdots & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{P}^1_{\mathbf{v}_2}(0) & \mathbf{P}^2_{\mathbf{v}_2}(0) & \mathbf{P}^3_{\mathbf{v}_2}(0) & \mathbf{0} & \mathbf{P}^5_{\mathbf{v}_2}(0) & \cdots & \cdots & \cdots & \mathbf{0} \\ \ddots & & & & \ddots & & & & & \ddots \\ \mathbf{P}^4_{\mathbf{v}_0}(L-2) & \mathbf{P}^5_{\mathbf{v}_0}(L-2) & \cdots & \cdots & \cdots & \cdots & \mathbf{P}^0_{\mathbf{v}_0}(L-2) & \mathbf{P}^1_{\mathbf{v}_0}(L-2) & \mathbf{0} & \mathbf{0} \\ \mathbf{P}^4_{\mathbf{v}_1}(L-2) & \mathbf{0} & \cdots & \cdots & \cdots & \cdots & \mathbf{P}^0_{\mathbf{v}_1}(L-2) & \mathbf{0} & \mathbf{P}^2_{\mathbf{v}_1}(L-2) & \mathbf{P}^3_{\mathbf{v}_1}(L-2) \\ \mathbf{0} & \mathbf{P}^5_{\mathbf{v}_2}(L-2) & \cdots & \cdots & \cdots & \cdots & \mathbf{0} & \mathbf{P}^1_{\mathbf{v}_2}(L-2) & \mathbf{P}^2_{\mathbf{v}_2}(L-2) & \mathbf{P}^3_{\mathbf{v}_2}(L-2) \\ \mathbf{0} & \mathbf{0} & \mathbf{P}^4_{\mathbf{v}_0}(L-1) & \mathbf{P}^5_{\mathbf{v}_0}(L-1) & \cdots & \cdots & \cdots & \cdots & \mathbf{P}^0_{\mathbf{v}_0}(L-1) & \mathbf{P}^1_{\mathbf{v}_0}(L-1) \\ \mathbf{P}^2_{\mathbf{v}_1}(L-1) & \mathbf{P}^3_{\mathbf{v}_1}(L-1) & \mathbf{P}^4_{\mathbf{v}_1}(L-1) & \mathbf{0} & \cdots & \cdots & \cdots & \cdots & \mathbf{P}^0_{\mathbf{v}_1}(L-1) & \mathbf{0} \\ \mathbf{P}^2_{\mathbf{v}_2}(L-1) & \mathbf{P}^3_{\mathbf{v}_2}(L-1) & \mathbf{0} & \mathbf{P}^5_{\mathbf{v}_2}(L-1) & \cdots & \cdots & \cdots & \cdots & \mathbf{0} & \mathbf{P}^1_{\mathbf{v}_2}(L-1) \end{pmatrix} \tag{2}$$

$R = 2/3$. Generalization to constructions of different rates are straightforward and should follow the same principle.

A. Definition of Braided LDPC-CCs

A braided LDPC-CC $\mathcal{C}$ associated with a TBCC $\mathcal{C}_0$ is the set of all sequences $\mathbf{v}_{[0,\infty]} = (\mathbf{v}_0^{(0)}, \mathbf{v}_0^{(1)}, \mathbf{v}_0^{(2)}, \cdots, \mathbf{v}_t^{(0)}, \mathbf{v}_t^{(1)}, \mathbf{v}_t^{(2)}, \cdots)$ satisfying the equation $\mathbf{v}_{[0,\infty]}\mathbf{H}^{\mathrm{T}}_{[0,\infty]} = \mathbf{0}$, where $\mathbf{v}_t^{(0)} = \mathbf{u}_t = (u_{t,0}, \cdots, u_{t,M-1})$, $\forall t \in \mathbb{Z}^+$ with $u_{t,m} \in GF(2)$ for $0 \le m \le M-1$, are the information bits and $\mathbf{v}_t^{(i)} = (v_{t,0}^{(i)}, \cdots, v_{t,M-1}^{(i)})$, $i = 1, 2$, $\forall t \in \mathbb{Z}^+$ with $v_{t,m}^{(i)} \in GF(2)$ for $0 \le m \le M-1$, are the parity bits. The syndrome former $\mathbf{H}^{\mathrm{T}}_{[0,\infty]}$ with elements in $GF(2)$ is obtained by substituting the 1's in the syndrome former $\bar{\mathbf{H}}^{\mathrm{T}}_{[0,\infty]}$ of the TBCC $\mathcal{C}_0$ by permutation matrices of dimension $M \times M$ and the 0's by $M \times M$ all-zero matrices.

For instance, the braided LDPC-CC associated with the TBCC from Fig. 1 has the syndrome former given by (1), where $\mathbf{0}$ represents an $M \times M$ all-zero matrix and the other elements are $M \times M$ permutation matrices. Due to the convolutional character of the braided LDPC-CCs, it is also possible to define the *syndrome former memory* m_s as the number of time instants being checked by the parity equations. For the example in (1), m_s is equal to two. Moreover, if we consider practical applications, periodic syndrome former matrices with period T are used, i.e., $\mathbf{P}^i_{\mathbf{v}_0}(t) = \mathbf{P}^i_{\mathbf{v}_0}(t+T)$, $\mathbf{P}^i_{\mathbf{v}_1}(t) = \mathbf{P}^i_{\mathbf{v}_1}(t+T)$ and $\mathbf{P}^i_{\mathbf{v}_2}(t) = \mathbf{P}^i_{\mathbf{v}_2}(t+T)$.

We will speak of *constrained codes*, if the non-zero permutation matrices of $\mathbf{H}^{\mathrm{T}}_{[0,\infty]}$ are subject to the constraints $\mathbf{P}^{2n}_{\mathbf{v}_0}(t) = \mathbf{P}^{\mathrm{even}}_{\mathbf{v}_0}(t)$, $\mathbf{P}^{2n+1}_{\mathbf{v}_0}(t) = \mathbf{P}^{\mathrm{odd}}_{\mathbf{v}_0}(t)$, $\mathbf{P}^{2n}_{\mathbf{v}_1}(t) = \mathbf{P}^{\mathrm{even}}_{\mathbf{v}_1}(t)$, $\mathbf{P}^{2n+1}_{\mathbf{v}_1}(t) = \mathbf{P}^{\mathrm{odd}}_{\mathbf{v}_1}(t)$, $\mathbf{P}^{2n}_{\mathbf{v}_2}(t) = \mathbf{P}^{\mathrm{even}}_{\mathbf{v}_2}(t)$ and $\mathbf{P}^{2n+1}_{\mathbf{v}_2}(t) = \mathbf{P}^{\mathrm{odd}}_{\mathbf{v}_2}(t)$; where $t, n \in \mathbb{Z}^+$, and $\mathbf{P}^{\mathrm{even}}_{(\cdot)}(t)$ and $\mathbf{P}^{\mathrm{odd}}_{(\cdot)}(t)$ are arbitrary non-zero permutation matrices. Furthermore, we will consider two ensembles of braided LDPC-CCs throughout this paper. Ψ_1 is an ensemble of regular codes with degrees $(J = 4, K = 6)$, which is defined by the TBCC of Fig. 1. The second ensemble Ψ_2, which is constructed in the same
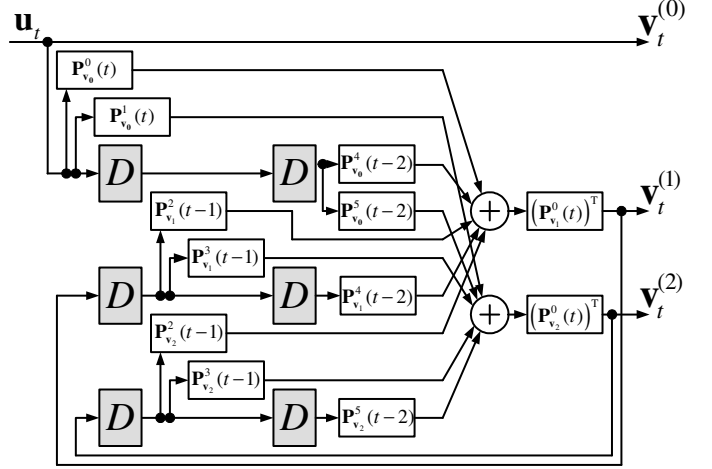

Fig. 2. Encoder for the braided LDPC-CC defined by (1).

way as Ψ_1 but uses CCs with parity-check matrix given by $\mathbf{H}(D) = [1 + D, 1, 1 + D + D^2]$ as component codes, consists of irregular codes and has degree distribution given by $J(x) = (1/3)x + (2/3)x^4$ and $K = 6$.

B. Definition of Tail-Biting Braided LDPC-CCs

A rate $R = 1/3$ tail-biting braided LDPC-CC $\tilde{\mathcal{C}}$ is the set of all code sequences $\tilde{\mathbf{v}}_{[0,L-1]}$ satisfying the equation $\tilde{\mathbf{v}}_{[0,L-1]}\tilde{\mathbf{H}}^{\mathrm{T}}_{[0,L-1]} = \mathbf{0}$, where the transposed parity-check matrix $\tilde{\mathbf{H}}^{\mathrm{T}}_{[0,L-1]}$ with block length $N = 3LM$ is obtained by wrapping the last $2m_s$ columns of permutation matrices of the syndrome former $\mathbf{H}^{\mathrm{T}}_{[0,\infty]}$ after L time instants. For the example of Fig. 1, the parity-check matrix of the tail-biting code is given by (2).

III. ENCODING

The encoding of braided LDPC-CCs can be easily defined based on their syndrome formers $\mathbf{H}^{\mathrm{T}}_{[0,\infty]}$. Considering a rate $R = 1/3$ systematic braided LDPC-CC, the recursive encoding equations for the parity-bits are given by (3).
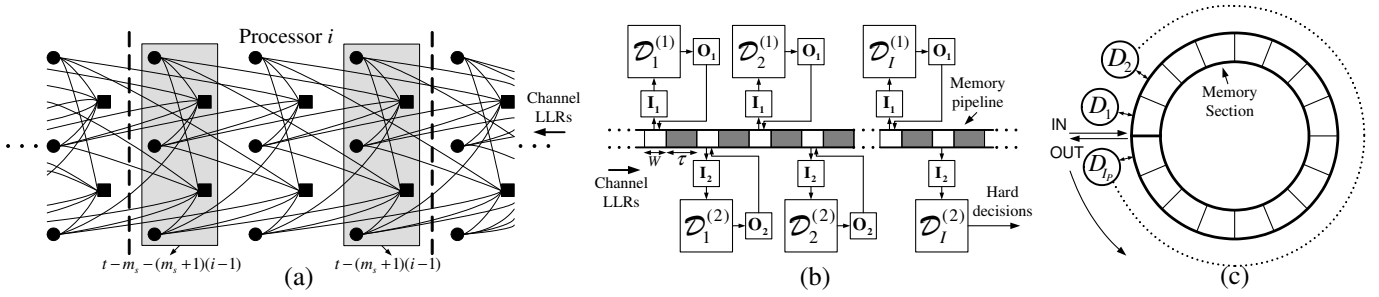
Fig. 3. Decoders for braided LDPC-CCs and tail-biting braided LDPC-CCs. **(a)** Graph representation of the code defined by (1) with $m_s = 2$. **(b)** Architecture for decoding braided LDPC-CCs with constrained syndrome former in a turbo-like fashion. **(c)** Circular pipeline architecture for decoding tail-biting braided LDPC-CCs.

$$\mathbf{v}_t^{(1)} = \left[\mathbf{u}_t \mathbf{P}_{\mathbf{v}_0}^0(t) + \right.$$
$$\left. \sum_{k=1}^{m_s} (\mathbf{u}_{t-k} \mathbf{P}_{\mathbf{v}_0}^{2k}(t-k) + \mathbf{v}_{t-k}^{(1)} \mathbf{P}_{\mathbf{v}_1}^{2k}(t-k) + \mathbf{v}_{t-k}^{(2)} \mathbf{P}_{\mathbf{v}_2}^{2k}(t-k)) \right] (\mathbf{P}_{\mathbf{v}_1}^0(t))^{\mathrm{T}}$$

$$\mathbf{v}_t^{(2)} = \left[\mathbf{u}_t \mathbf{P}_{\mathbf{v}_0}^1(t) + \right.$$
$$\left. \sum_{k=1}^{m_s} (\mathbf{u}_{t-k} \mathbf{P}_{\mathbf{v}_0}^{2k+1}(t-k) + \mathbf{v}_{t-k}^{(1)} \mathbf{P}_{\mathbf{v}_1}^{2k+1}(t-k) + \mathbf{v}_{t-k}^{(2)} \mathbf{P}_{\mathbf{v}_2}^{2k+1}(t-k)) \right] (\mathbf{P}_{\mathbf{v}_2}^1(t))^{\mathrm{T}}$$
$$(3)$$

Fig. 2 shows exemplarily the architecture of the encoder for the code defined by (1). In this picture, each D represents M delay elements.

Periodic tail-biting braided LDPC-CCs are quasi-cyclic codes. Therefore, they can be encoded in linear time, using circuits based on shift-registers [10]. Alternatively, the techniques presented in [11] can be applied and very efficient encoding structures can be obtained.

IV. DECODING

In general, LDPC-CCs have the fundamental property that the separation between connected nodes of their Tanner graphs is not bigger than $m_s + 1$ time instants. One direct consequence of this property is the decoding by means of pipeline architectures [6]. For the codes that we are considering in this paper, pipeline decoders can also be derived. Below, we discuss the main aspects of the decoding architectures.

A. Decoding of the Braided LDPC-CCs

In the most general case, where no constraints are imposed and short cycles are avoided in the construction of $\mathbf{H}_{[0,\infty]}^{\mathrm{T}}$, the pipelined belief propagation algorithm shall be used. This algorithm works as follows. Let the transmitted sequence be $\mathbf{v}_t = (\mathbf{v}_t^{(0)}, \mathbf{v}_t^{(1)}, \mathbf{v}_t^{(2)})$, $t = 0, 1, 2, \cdots$, and the received sequence be $\mathbf{r}_t = (\mathbf{r}_t^{(0)}, \mathbf{r}_t^{(1)}, \mathbf{r}_t^{(2)})$, $t = 0, 1, 2, \cdots$. At time instant t, the *log-likelihood ratios* (LLRs) corresponding to $\mathbf{r}_t = (\mathbf{r}_t^{(0)}, \mathbf{r}_t^{(1)}, \mathbf{r}_t^{(2)})$ arrive at the decoder, which consists of a delay line and the concatenation of I identical parallel operating processors [6]. Figure 3(a) shows one of such processors and its connection to the graph representation of the code. In this graph, each node and each edge correspond to M permuted[1] nodes and M permuted edges of the same type, respectively. The next step of the decoding procedure is to shift all stored values in the delay line in order to

---

[1] The permutations are defined by the permutation matrices in $\mathbf{H}_{[0,\infty]}^{\mathrm{T}}$.

accommodate the new LLRs corresponding to $\mathbf{v}_t$. The values dropped from the delay line due to this shift action are already the final decoding results of $\mathbf{v}_{t-I(m_s+1)-1}$ after I iterations. In a further step, the check nodes corresponding to the time instants $\mathbb{T}_{\mathrm{CN}} = \{t - (m_s + 1)(i - 1)\}$, with $i = 1, \cdots, I$, are updated. Finally, the variable nodes corresponding to the time instants $\mathbb{T}_{\mathrm{VN}} = \{t - m_s - (m_s + 1)(i - 1)\}$, with $i = 1, \cdots, I$, are updated and the whole process is repeated for $\mathbf{r}_{t+1}$ and so on.

On the other hand, constrained braided LDPC-CCs can be decoded based on a turbo-like decoding scheme of $2M$ component CCs. Considering the braided LDPC-CC associated with Fig. 1, the component CCs have parity-check matrices given by $\mathbf{H}(D) = [1, 1 + D^2, 1 + D + D^2]$ and the code sequences are obtained using (3), where each vectorial equation in (3) represents a group of M CCs. In this sense, the M CCs from one group are made to exchange extrinsic information with the M CCs from the other group during the decoding and, thus, it can iterate in a turbo fashion. Fig. 3(b) shows the decoding architecture for these codes. As we can observe, this decoder consists of a delay line (memory pipeline), the concatenation of I pairs of parallel operating processors and write/read logics (i.e., $\mathbf{I}_1$, $\mathbf{I}_2$, $\mathbf{O}_1$ and $\mathbf{O}_2$) defined by the permutation matrices in $\mathbf{H}_{[0,\infty]}^{\mathrm{T}}$. Here, $\mathcal{D}_i^{(1)}$, with $i = 1, \cdots, I$, are the processors responsible for decoding the M CCs from the first group (first equation in (3)) and $\mathcal{D}_i^{(2)}$, with $i = 1, \cdots, I$, are the processors responsible for decoding the M CCs from the second group (second equation in (3)). Obviously, the communication between adjacent processors is done by means of extrinsic information. Furthermore, the decoding within each processor $\mathcal{D}_i^{(\cdot)}$ is performed using M parallel, windowed BCJR decoders [12] that operate on windows of length W. Additionally, adjacent processors are separated from each other by the separation delay τ in order to avoid memory conflicts. Within this context, the decoding of a certain $\mathbf{r}_t$ is accomplished when it is shifted through the whole memory pipeline and processed by the $2I$ processors, and hard decisions are taken at the output of processor $\mathcal{D}_I^{(2)}$.

B. Decoding of the Tail-Biting Braided LDPC-CCs

The decoding of tail-biting braided LDPC-CCs with unconstrained parity-check matrices can be performed using the circular pipeline architecture discussed in [8] and depicted in Fig. 3(c). This decoder will consist of $1 \le I_P \le N_{\max}/(m_s + 1)$

parallel processors $\mathcal{D}_i$, where $N_{\max}$ is a design parameter[2] and corresponds to the maximum length (in time instants) of the codewords that can be decoded. The operation of the circular decoder is as follows. After a word $\tilde{\mathbf{r}}_{[0,L-1]}$, corresponding to the transmitted codeword $\tilde{\mathbf{v}}_{[0,L-1]}$, is received and stored, the processors $\mathcal{D}_i$ begin to hop over the memory addresses in steps corresponding to the amount of data transmitted in one time instant of the tail-biting code while performing their decoding operations. When each processor has completed I_R rounds over the complete memory, a number $I = I_R \cdot I_P$ of decoding iterations has been performed on $\tilde{\mathbf{r}}_{[0,L-1]}$.

Similar to the braided LDPC-CCs with constrained syndrome formers, a turbo-like decoding scheme can be applied to decode the tail-biting codes with constrained parity-check matrices. In this case, the component CCs belonging to the two groups of M CCs are tail-biting CCs and each of them can be decoded using the techniques presented in [13]. A circular decoding architecture can be also derived for the constrained tail-biting braided LDPC-CCs. However, it is more convenient to perform the decoding iterations in a non-pipelined fashion in order to avoid the additional memory requirements due to the separation delay τ.

V. ITERATIVE THRESHOLD ANALYSIS

We were able to compute the iterative thresholds of the ensembles Ψ_1 and Ψ_2 using the *computation tree* methods presented in [3]. The results of our computations are summarized in Table I for the BPSK modulated *additive white Gaussian noise* (AWGN) channel.

As we can observe in Table I, there is more than 0.5 dB difference between the thresholds of unconstrained and constrained codes from the same ensemble. This is a particularly interesting phenomenon since the unconstrained and constrained codes from a particular ensemble still have the same degree distributions. In this case, the difference in the thresholds are due to the application of distinct decoding algorithms, i.e., unconstrained codes are decoded using Gallager's BP and constrained codes are decoded using the BCJR algorithm.

TABLE I
THRESHOLDS FOR BRAIDED LDPC-CC ENSEMBLES IN AWGN CHANNEL

Ensemble	Unconstrained Codes	Constrained Codes
Ψ_1	1.68 dB	1.10 dB
Ψ_2	1.43 dB	0.86 dB

VI. PERFORMANCE ANALYSIS

In this section, we present the performance results for several braided LDPC-CCs and tail-biting braided LDPC-CCs belonging to the ensembles Ψ_1 and Ψ_2. All simulations were performed in the BPSK modulated AWGN channel and a maximum of 100 decoding iterations were allowed. For the braided LDPC-CCs at least 500000 bits were decoded and at least 5000 bit errors were detected in each simulation point.

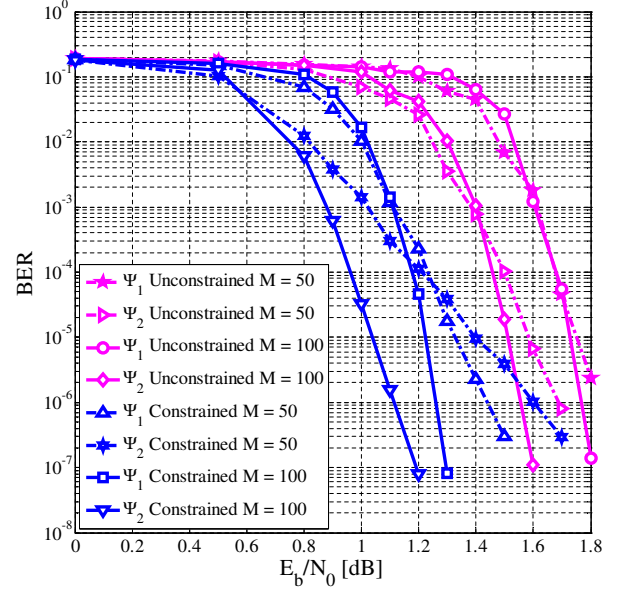[2]This is actually defined by the amount of memory available in the decoder.

Fig. 4.   BERs of braided LDPC-CCs.

On the other hand, for the tail-biting braided LDPC-CCs and *parallel concatenated convolutional codes* (PCCCs) [2] at least 1000 frames were decoded and at least 50 frame errors were detected in each point. In addition, all permutation matrices and interleavers were generated completely random and the periods T of the syndrome formers and parity-check matrices of the braided LDPC-CCs and their tail-biting versions are equal to $T = 10$. An exception is done for codes with $M = 50$. In this case, cycles of length 4 were avoided and $T = 20$.

Fig. 4 shows the bit error rate performances of different braided LDPC-CCs from ensembles Ψ_1 and Ψ_2. The decoding occurs in a streaming fashion using the pipeline decoders presented in Section IV-A. As we can observe, the constrained codes always show better results than the unconstrained ones in the low to moderate SNR region. We can also note that some additional gain can be achieved if codes from ensemble Ψ_2 are used. Moreover, all codes have the *waterfall regions* of their BER curves close to the corresponding iterative thresholds shown in Section V and no *error-floors* can be observed. Finally, we observe that by increasing the size M of the permutation matrices, the BER curves become steeper.

The performances of unconstrained tail-biting braided LDPC-CCs from ensembles Ψ_1 and Ψ_2 are shown in Fig. 5. We can observe that the codes from the ensemble Ψ_2 perform better than the codes from Ψ_1 when similar parameters are considered. For comparison purposes, we also have in the same plot the BER curves of rate $R = 1/3$ PCCCs with 4-state component encoders defined by the generator matrix $\mathbf{G}(D) = [1, (1 + D^2)/(1 + D + D^2)]$ and block lengths $N = 3000$ and $N = 6000$. It is evident that the PCCCs have better performances at low SNR, however, they also have poor minimum distance properties and high error-floors appear. On the other hand, the BER curves of the tail-biting codes are steep and cross the BER curves of the PCCCs. Moreover, in spite of the relatively short block lengths and sizes M of the permutation matrices, no error-floors can be observed for the
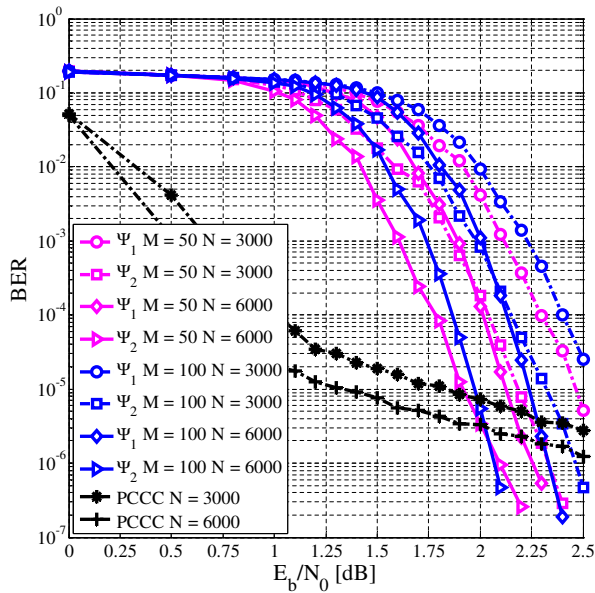
Fig. 5. BERs of unconstrained tail-biting braided LDPC-CCs versus PCCCs.

Fig. 6. BERs of constrained tail-biting braided LDPC-CCs versus PCCCs.

tail-biting codes under consideration.

We also obtained performance results for the constrained tail-biting braided LDPC-CCs. This is shown in Fig. 6 in companion with the BER curves of the PCCCs from Fig. 5. As it was expected, the constrained codes from ensemble Ψ_2 have better performances than the ones from Ψ_1. Comparing Fig. 5 to Fig. 6, we can also observe that the constrained codes always perform better than the unconstrained ones, even if they have shorter block lengths. Additionally, the curves are not showing error-floors in the SNR intervals considered. In Fig. 5 and Fig. 6, we can also note that the codes with smaller M perform slightly better in the low to moderate SNR region. We expect this situation to be reversed for high SNRs. An indication of this are the curves (Ψ_2 $M = 50$ $N = 6000$) and (Ψ_2 $M = 100$ $N = 6000$) in Fig. 5 and Fig. 6.

VII. CONCLUSION

In this paper, we introduced a new class of LDPC convolutional codes derived from braided convolutional codes [4] which we call *braided LDPC-CCs*. We also have shown how tail-biting codes can be derived from braided LDPC-CCs. Moreover, the basic ideas behind the encoding and decoding architectures were presented. Considering the decoding, the fact that the braided LDPC-CCs can be decoded using Gallager's algorithm [1] or the BCJR algorithm [7] in a turbo-like fashion is particularly interesting. Iterative thresholds for two ensembles of braided LDPC-CCs were also computed. Finally, simulations results have shown the error correction capabilities of our codes, where we could observe that the tail-biting braided LDPC-CCs have potentially better minimum distance properties than parallel concatenated convolutional codes with similar component codes.

Our further research on this topic includes the understanding of the impact that the component convolutional codes and the structure of the permutation matrices have on the minimum distances and th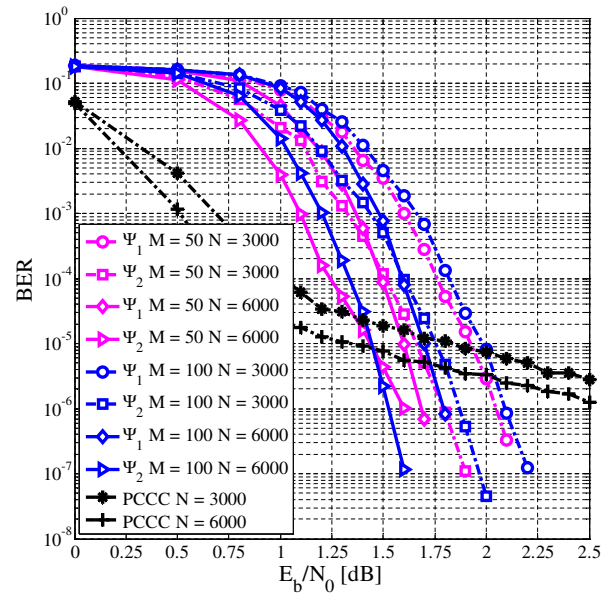resholds of the braided LDPC-CCs. It is also interesting to apply puncturing to obtain other coding rates. In other words, our future efforts will focus on deriving some practical design rules for the codes presented in this paper.

REFERENCES

[1] R. Gallager, *Low-Density Parity-Check Codes*, MIT Press, Cambridge, MA, 1963.
[2] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: turbo-codes (1)," in *Proc. IEEE International Conference on Communications*, Geneva, Switzerland, May 1993, vol. 2, pp. 1064–1070.
[3] 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.
[4] 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.
[5] J. Thorpe, "Low-density parity-check (LDPC) codes constructed from protographs," in *IPN Progress Report 42-154, JPL*, Aug. 2005.
[6] 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.
[7] 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.
[8] 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.
[9] M.B.S. Tavares, K.Sh. Zigangirov, and G.P. Fettweis, "Tail-biting LDPC convolutional codes based on protographs," in *Proc. of IEEE Vehicular Technology Conference (VTC'07)*, Baltimore, USA, Sept. 2007.
[10] W.W. Peterson and E.J. Weldon, Jr., *Error-Correcting Codes*, MIT Press, Cambridge, MA, 1972.
[11] T.J. Richardson and R.L. Urbanke, "Efficient encoding of low-density parity-check codes," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 638–656, Feb. 2001.
[12] S. Benedetto, G. Montorsi, D. Divsalar, and F. Pollara, "A soft-input soft-output maximum a posteriori (MAP) module to decode parallel and serial concatenated codes," *JPL TDA Progress Report*, vol. 42, no. 127, pp. 1–20, Nov. 1996.
[13] J.B. Anderson and S.M. Hladik, "Tailbiting MAP decoders," *IEEE J. Select. Areas Commun.*, vol. 16, no. 2, pp. 297–302, Feb. 1998.