This document is a preprint of: M.B.S. Tavares, K.S. Zigangirov and G. Fettweis, “Tail-Biting LDPC Convolutional Codes,” in Proceedings of IEEE International Symposium on Information Theory (ISIT 2007), Nice, France, Jun 2007. DOI:10.1109/ISIT.2007.4557569

Tail-Biting LDPC Convolutional Codes

Marcos B.S. Tavares
Vodafone Chair
Technische Universität Dresden
01062 Dresden, Germany
Email: tavares@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

Gerhard P. Fettweis
Vodafone Chair
Technische Universität Dresden
01062 Dresden, Germany
Email: fettweis@ifn.et.tu-dresden.de

***Abstract*— In this paper, the *tail-biting* version of the low-density parity-check convolutional codes (LDPCCCs) is introduced and investigated. The definition of the tail-biting LDPCCCs (TB-LDPCCCs) is given through their parity-check matrices and the basic ideas behind the architectures of the encoders and decoders for these codes are also presented. In addition, the asymptotical lower bound for the minimum distance of these codes and simulation results are shown.**

I. INTRODUCTION

In the last years, the low-density parity-check (LDPC) codes invented by Gallager [1] and the turbo codes discovered by Berrou *et al.* [2] attracted great attention of the coding community because of their Shannon capacity approaching performance and low-complexity decoding. This paper is devoted to the study of the tail-biting version of the LDPC convolutional codes (LDPCCCs) introduced in [3][1].

LDPCCCs have some advantages in comparison with LDPC block codes, specially for transmitting streaming data [5]. An important feature of LDPCCCs is that the same encoder can be used to obtain codes sequences of varying lengths with quite good performance by choosing different termination lengths. However, the introduction of a zero-tail for termination results in the so-called *rate loss*, which is specially noticeable for small frame lengths. The introduction of tail-biting allows to avoid this loss.

In the following section, we define the transposed parity-check matrix of a tail-biting LDPCCC (TB-LDPCCC) as the wrapped syndrome former of the mother LDPCCC. Then, in section III, we describe the encoder and the concept for its implementation. Section IV is devoted to the description of the circular pipeline decoder for TB-LDPCCCs. In section V, we study the asymptotical properties of the minimal distance of TB-LDPCCCs as a function of the syndrome former memory of the mother LDPCCC and blocklength. Simulation results for regular $(3, 6)$ TB-LDPCCCs of different lengths are presented in section VI and also compared to random generated LDPC block codes (LDPCBCs). Finally, section VII concludes the paper.

II. TAIL-BITING LDPC CONVOLUTIONAL CODES

In order to simplify the notations, we will only give the description of rate $R = 1/2$ codes. Generalizations to other rates is straightforward. Let

[1]LDPCCCs were first described by Tanner in a patent application [4].

$$\mathbf{u}_{[0,t]} = (u_0, u_1, \cdots, u_t), \tag{1}$$

where $u_i \in GF(2)$, be the information sequence that enters the encoder. The encoded sequence is given by

$$\mathbf{v}_{[0,t]} = (\mathbf{v}_0, \mathbf{v}_1, \cdots, \mathbf{v}_t), \tag{2}$$

where $\mathbf{v}_i = (v_i^{(0)}, v_i^{(1)})$ and $v_i^{(\cdot)} \in GF(2)$. For the case of systematic encoders, we have $v_i^{(0)} = u_i$.

Now, we recall the definition of an LDPCCC. A time-varying LDPCCC is defined as the set of all sequences $\mathbf{v}_{[0,\infty]}$ satisfying the equation $\mathbf{v}_{[0,\infty]}\mathbf{H}^{\mathrm{T}}_{[0,\infty]} = \mathbf{0}$, where

$$\mathbf{H}^{\mathrm{T}}_{[0,\infty]} = \begin{pmatrix} \mathbf{H}^{\mathrm{T}}_0(0) & \cdots & \mathbf{H}^{\mathrm{T}}_{m_s}(m_s) & & \mathbf{0} \\ & \ddots & & \ddots & \\ \mathbf{0} & & \mathbf{H}^{\mathrm{T}}_0(t) & \cdots & \mathbf{H}^{\mathrm{T}}_{m_s}(t+m_s) \\ & & \ddots & & \ddots \end{pmatrix} \tag{3}$$

is a semi-infinite transposed parity-check matrix, called *syndrome former*. For a rate $R = 1/2$ code, the elements of $\mathbf{H}^{\mathrm{T}}_{[0,\infty]}$ are submatrices of dimension 2×1 given by

$$\mathbf{H}^{\mathrm{T}}_i(t) = \begin{pmatrix} h_i^{(0)}(t) \\ h_i^{(1)}(t) \end{pmatrix}, \qquad i = 0, \cdots, m_s, \tag{4}$$

where $h_0^{(k)}(t) = 1$, $k = 0, 1$, and, at least for one time-instant t, we have $\mathbf{H}^{\mathrm{T}}_{m_s}(t) \neq \mathbf{0}$. Moreover, the syndrome former defining a regular (J, K) LDPCCC has exactly J ones in each row and K ones in each column, starting from the m_s-th column. The value m_s is the *syndrome former memory*. For practical applications, periodic syndrome former matrices with period T are used, i.e., $\mathbf{H}^{\mathrm{T}}_i(t) = \mathbf{H}^{\mathrm{T}}_i(t+T)$. In [3], the methods for constructing periodical LDPCCCs based on the unwrapping procedure are described.

The code sequences $\tilde{\mathbf{v}}_{[0,N-1]}$ of the tail-biting LDPCCC satisfy the equality

$$\tilde{\mathbf{v}}_{[0,N-1]}\tilde{\mathbf{H}}^{\mathrm{T}}_{[0,N-1]} = \mathbf{0}. \tag{5}$$

Here, the transposed parity-check matrix $\tilde{\mathbf{H}}^{\mathrm{T}}_{[0,N-1]}$ of a rate $R = 1/2$ TB-LDPCCC with blocklength $2N$ is obtained by wrapping the last m_s columns of submatrices of the syndrome

This document is a preprint of: M.B.S. Tavares, K.S. Zigangirov and G. Fettweis, "Tail-Biting LDPC Convolutional Codes," in Proceedings of IEEE International Symposium on Information Theory (ISIT 2007), Nice, France, Jun 2007. DOI:10.1109/ISIT.2007.4557569

$$\tilde{\mathbf{H}}^{\mathrm{T}}_{[0,N-1]} = \left(\begin{array}{cccccccccc}
\mathbf{H}^{\mathrm{T}}_0(0) & \mathbf{H}^{\mathrm{T}}_1(1) & \cdots & \mathbf{H}^{\mathrm{T}}_{m_s}(m_s) & \mathbf{0} & & \cdots & & & \mathbf{0} \\
\mathbf{0} & \mathbf{H}^{\mathrm{T}}_0(1) & \cdots & \mathbf{H}^{\mathrm{T}}_{m_s-1}(m_s) & \mathbf{H}^{\mathrm{T}}_{m_s}(m_s+1) & \mathbf{0} & \cdots & & & \mathbf{0} \\
 & \ddots & & & \ddots & & & \ddots & & \\
\mathbf{H}^{\mathrm{T}}_{m_s}(N) & \mathbf{0} & & \cdots & & \mathbf{0} & \mathbf{H}^{\mathrm{T}}_0(N-m_s) & \cdots & & \mathbf{H}^{\mathrm{T}}_{m_s-1}(N-1) \\
\mathbf{H}^{\mathrm{T}}_{m_s-1}(N) & \mathbf{H}^{\mathrm{T}}_{m_s}(N+1) & \mathbf{0} & & & & & & & \vdots \\
\vdots & & & & & & \mathbf{0} & \mathbf{H}^{\mathrm{T}}_0(N-2) & & \mathbf{H}^{\mathrm{T}}_1(N-1) \\
\mathbf{H}^{\mathrm{T}}_1(N) & \mathbf{H}^{\mathrm{T}}_2(N+1) & \cdots & \mathbf{H}^{\mathrm{T}}_{m_s}(N+m_s-1) & \mathbf{0} & & \cdots & \mathbf{0} & & \mathbf{H}^{\mathrm{T}}_0(N-1)
\end{array}\right) \quad (6)$$

$$\left[\tilde{\mathbf{H}}^{(k)}_{[0,N-1]}\right]^{\mathrm{T}} = \left(\begin{array}{cccccccccc}
h^{(k)}_0(0) & h^{(k)}_1(1) & \cdots & h^{(k)}_{m_s}(m_s) & 0 & & \cdots & & & 0 \\
0 & h^{(k)}_0(1) & \cdots & h^{(k)}_{m_s-1}(m_s) & h^{(k)}_{m_s}(m_s+1) & 0 & \cdots & & & 0 \\
 & \ddots & & & \ddots & & & \ddots & & \\
h^{(k)}_{m_s}(N) & 0 & & \cdots & & 0 & h^{(k)}_0(N-m_s) & \cdots & & h^{(k)}_{m_s-1}(N-1) \\
h^{(k)}_{m_s-1}(N) & h^{(k)}_{m_s}(N+1) & 0 & & & & & & & \vdots \\
\vdots & & & & & & 0 & h^{(k)}_0(N-2) & & h^{(k)}_1(N-1) \\
h^{(k)}_1(N) & h^{(k)}_2(N+1) & \cdots & h^{(k)}_{m_s}(N+m_s-1) & 0 & & \cdots & 0 & & h^{(k)}_0(N-1)
\end{array}\right) \quad (7)$$

former in (3) after N time instants. Thus, $\tilde{\mathbf{H}}^{\mathrm{T}}_{[0,N-1]}$ can be written as (6).

III. Encoding of Tail-Biting LDPCCCs

The rows of the parity-check matrix $\tilde{\mathbf{H}}^{\mathrm{T}}_{[0,N-1]}$ can be reordered to facilitate the encoding procedure. For this purpose, we separate the even rows of $\tilde{\mathbf{H}}^{\mathrm{T}}_{[0,N-1]}$ in $[\tilde{\mathbf{H}}^{(0)}_{[0,N-1]}]^{\mathrm{T}}$ and the odd rows in $[\tilde{\mathbf{H}}^{(1)}_{[0,N-1]}]^{\mathrm{T}}$, where $[\tilde{\mathbf{H}}^{(k)}_{[0,N-1]}]^{\mathrm{T}}$, for $k = 0, 1$, is given by (7).

Analogously, if we reorder the elements of $\tilde{\mathbf{v}}_{[0,N-1]}$ into two subvectors $\tilde{\mathbf{v}}^{(k)}_{[0,N-1]} = (v^{(k)}_0, \cdots, v^{(k)}_{N-1})$, $k = 0, 1$; then we can rewrite equation (5) as

$$\tilde{\mathbf{v}}^{(0)}_{[0,N-1]}\left[\tilde{\mathbf{H}}^{(0)}_{[0,N-1]}\right]^{\mathrm{T}} + \tilde{\mathbf{v}}^{(1)}_{[0,N-1]}\left[\tilde{\mathbf{H}}^{(1)}_{[0,N-1]}\right]^{\mathrm{T}} = \mathbf{0} \quad (8)$$

Furthermore, if the matrix $[\tilde{\mathbf{H}}^{(1)}_{[0,N-1]}]^{\mathrm{T}}$ has full rank, it has an $N \times N$ inverse, which we call $\mathbf{G}_{[0,N-1]}$. Then, we can obtain the following encoding equation for systematic codes

$$\tilde{\mathbf{v}}^{(1)}_{[0,N-1]} = \mathbf{u}_{[0,N-1]}\left[\tilde{\mathbf{H}}^{(0)}_{[0,N-1]}\right]^{\mathrm{T}}\mathbf{G}_{[0,N-1]}, \quad (9)$$

where $\mathbf{u}_{[0,N-1]} = (u_0, \cdots, u_{N-1}) = \tilde{\mathbf{v}}^{(0)}_{[0,N-1]}$ is the information sequence. Note that the $N \times N$ matrix $\tilde{\mathbf{G}}_{[0,N-1]} = [\tilde{\mathbf{H}}^{(0)}_{[0,N-1]}]^{\mathrm{T}}\mathbf{G}_{[0,N-1]}$ can be calculated in advance and stored. In this case, the encoder can be implemented using a linear matrix multiplication circuitry with operations defined on $GF(2)$ with complexity of the order $\mathcal{O}(N^2)$.

IV. Decoding of Tail-Biting LDPCCCs

The decoding of TB-LDPCCCs can be performed using the pipelined version [3] of the belief propagation decoding of LDPC codes [1]. Particularly, it is convenient to use a pipeline decoder with a circular topology [6] that reflects the wrap-around in the parity-check matrix in (6). The circular pipeline decoder will consist of $1 \leq I_P \leq N/(m_s+1)$ parallel processors $\mathcal{D}_i$ (Fig. 1).

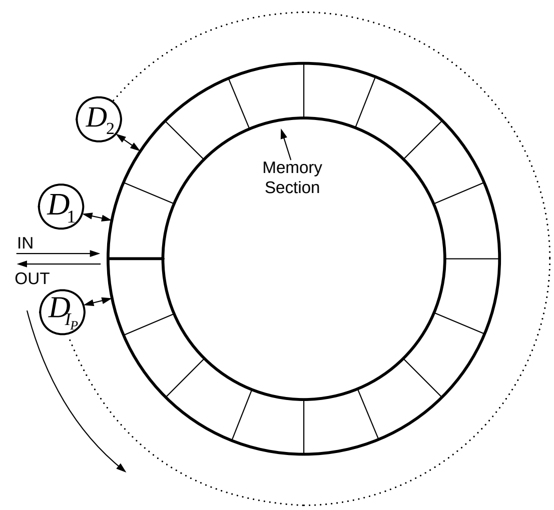

Fig. 1. Circular pipeline decoder.

The operation of the circular pipeline decoder is as follows. The code sequence of a TB-LDPCCC corresponding to N time instants enters the decoder, so that its memory is completely written. Each parallel processor $\mathcal{D}_i$ is attributed a memory section (Fig. 1), which is large enough to avoid the memory access collisions between the processors and, thus, enables their parallel operation. Considering the graph representation of the TB-LDPCCC, we can easily note that the size of the memory sections should be at least $m_s + 1$ time instants to permit simultaneous operation of all processors. Then, the processors $\mathcal{D}_i$ hop over the memory addresses in steps corresponding to the amount of data transmitted in one time instant of the TB-LDPCCC. After each processor has completed I_R rounds over the complete memory, a number $I = I_R \cdot I_P$

$$z_t^{(k)}(i,l) = z_t^{(k)}(0) + \sum_{l' \in \mathcal{L}(t,k)\setminus l} \log \frac{1 + \left(\prod_{t' \in \mathcal{N}(l',k)\setminus t} \tanh(z_{t'}^{(k)}(i-1,l')/2)\right) \cdot \left(\prod_{t' \in \mathcal{N}(l',\bar{k})} \tanh(z_{t'}^{(\bar{k})}(i-1,l')/2)\right)}{1 - \left(\prod_{t' \in \mathcal{N}(l',k)\setminus t} \tanh(z_{t'}^{(k)}(i-1,l')/2)\right) \cdot \left(\prod_{t' \in \mathcal{N}(l',\bar{k})} \tanh(z_{t'}^{(\bar{k})}(i-1,l')/2)\right)}. \quad (11)$$

of decoding iterations has been performed. Clearly, a circular pipeline decoder with $I'_P = N'/(m_s+1)$ parallel processors can also decode TB-LDPCCCs with blocklengths smaller than N' time instants. In this case, the processor that is performing the operations at the borders of the active memory region must neglect the inactive memory region in between by performing a wrap-around in the address generation. Moreover, the processors that are located in the inactive memory region can be turned off.

Now, we consider the decoding from the belief propagation viewpoint. We suppose that the rows of the matrix $[\tilde{\mathbf{H}}^{(k)}_{[0,N-1]}]^{\mathrm{T}}$ are numerated by the pair (t,k), where $t = 0,\cdots,N-1$ and $k = 0,1$. We also numerate the columns of $[\tilde{\mathbf{H}}^{(k)}_{[0,N-1]}]^{\mathrm{T}}$ by l, where $l = 0,\cdots,N-1$.

Let $\mathbf{r}_{[0,N-1]} = (\mathbf{r}_0,\cdots,\mathbf{r}_{N-1})$, where $\mathbf{r}_t = (r_t^{(0)}, r_t^{(1)})$, be the received sequence corresponding to the transmitted sequence $\tilde{\mathbf{v}}_{[0,N-1]} = (\mathbf{v}_0,\cdots,\mathbf{v}_{N-1})$. The decoder uses the received symbols of $\mathbf{r}_{[0,N-1]}$ to calculate the initial *log-likelihood ratios* (LLRs) $z_t^{(k)}(0)$, called intrinsic information, for all symbols of the sequence $\tilde{\mathbf{v}}_{[0,N-1]}$. This is expressed as:

$$z_t^{(k)}(0) = \log \frac{\Pr(v_t^{(k)} = 0 | r_t^{(k)})}{\Pr(v_t^{(k)} = 1 | r_t^{(k)})}, \quad t = 0,1,\cdots,N-1, \quad k = 0,1. \quad (10)$$

Let $\mathcal{L}(t,k)$ be the set of column numbers of $[\tilde{\mathbf{H}}^{(k)}_{[0,N-1]}]^{\mathrm{T}}$ having a one in the t-th row of $[\tilde{\mathbf{H}}^{(k)}_{[0,N-1]}]^{\mathrm{T}}$. Correspondingly, let $\mathcal{N}(l,k)$ be the set of row numbers having a one in the l-th column of $[\tilde{\mathbf{H}}^{(k)}_{[0,N-1]}]^{\mathrm{T}}$. As we can note, these two sets represent the graph connectivity of the code.

Based on the intrinsic information, the decoder performs I iterations. In the i-th iteration, J LLRs $z_t^{(k)}(i,l)$, where $t = 0,\cdots,N-1$; $k = 0,1$ and $l \in \mathcal{L}(t,k)$ are updated for each one of the $2N$ symbols. It is worth to mention that the number of columns of the matrix $[\tilde{\mathbf{H}}^{(k)}_{[0,N-1]}]^{\mathrm{T}}$ having a one in the t-th row is equal to J.

Hence, the LLR $z_t^{(k)}(i,j)$ of the i-th iteration is connected with the LLRs $z_{t'}^{(k')}(i-1,j')$ of the previous iteration according to the expression given by (11), where $\bar{k} = 1$ for $k = 0$ and $\bar{k} = 0$ for $k = 1$.

The first term in (11) represents the intrisic values from (10), which are derived from the received symbols $r_t^{(k)}$. The second term in (11) represents the extrinsic information that the parity-check equation corresponding to the columns of (6) deliver to the symbol $v_t^{(k)}$ on the i-th iteration. In the last I-th iteration, all parity-check sets participate on the calculation of the final LLR, which is used to compute the hard-decision estimate given by

$$\hat{v}_t^{(k)} = \begin{cases} 1, & \text{if } z_t^{(k)} < 0 \\ 0, & \text{otherwise} \end{cases}. \quad (12)$$

V. Distance Bounds for Tail-Biting LDPCCCs

Tail-biting convolutional codes have dual nature. Namely, they have simultaneously properties of convolutional and block codes. Particularly, the minimum distance of these codes, which is a function of the code blocklength, depends on the free distance of the mother convolutional code.

To lowerbound the minimum distance of a tail-biting LDPCCC, we consider the same random ensemble of the time-varying LDPCCCs as in [7], namely, an ensemble $\mathcal{C}_{\text{LDPCCC}}(J,K,M)$ of (J,K) LDPCCCs with syndrome formers composed of $M \times M$ block permutation matrices. The main result of [7] can be formulated as the following theorem.

Theorem 1: In the ensemble $\mathcal{C}_{\text{LDPCCC}}(3,6,M)$, there exists a convolutional code with free distance d_{free} lowerbounded by

$$d_{\text{free}} \geq \alpha_{\text{LDPCCC}}(3,6) \cdot \nu + o(\nu), \quad (13)$$

where $\nu = 6M$ is the constraint length of the code, and the coefficient $\alpha_{\text{LDPCCC}}(3,6)$ is equal to 0.083. ■

Note that the coefficient $\alpha_{\text{LDPCCC}}(3,6)$ is about 4.7 times less than the coefficient $\alpha_{\text{C}} = 0.39$ of Costello's bound for the free distance of conventional convolutional codes.

The minimum distance d_{min} of (J,K) LDPC block codes is lower-bounded by Gallager's bound [1]:

$$d_{\text{min}} \geq \alpha_{\text{G}}(J,K) \cdot \mu + o(\mu), \quad (14)$$

where μ is the blocklength and the coefficient $\alpha_{\text{G}}(J,K)$ is equal to 0.023 for $J = 3$ and $K = 6$.

We have considered an ensemble $\tilde{\mathcal{C}}_{\text{LDPCCC}}(3,6,M,N)$ of tail-biting LDPCCCs constructed by unwrapping the last m_s columns of submatrices of the syndrome former of a code from ensemble $\mathcal{C}_{\text{LDPCCC}}(3,6,M)$ after N time instants.

Theorem 2: In the ensemble $\tilde{\mathcal{C}}_{\text{LDPCCC}}(3,6,M,N)$, there exists a tail-biting convolutional code with minimum distance lower-bounded by

$$d_{\text{min}} \geq \min\left\{\alpha_{\text{G}}(J,K) \cdot \mu + o(\mu), \alpha_{\text{LDPCCC}}(3,6) \cdot \nu + o(\nu)\right\}, \quad (15)$$

where $\mu = 2N$ is the blocklength, $\nu = 6M$ is the constraint

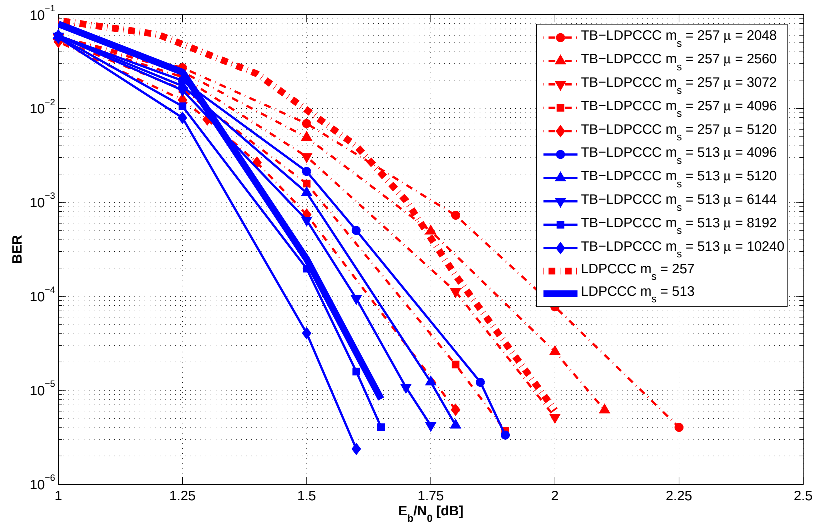

Fig. 2. BER performance of the tail-biting LDPCCCs obtained from convolutional codes with syndrome-former memories $m_s = 257$ and $m_s = 513$. The BER curves of the original convolutional codes (thick lines) are also shown for comparison purposes.

length of the mother LDPCCC, $\alpha_{\mathrm{G}}(J,K) = 0.023$ and $\alpha_{\mathrm{LDPCCC}}(3,6) = 0.083$. ■

From (15), it follows that, from the minimum distance viewpoint, it is reasonable to choose blocklengths for tail-biting LDPCCCs such that

$$2N = \mu \geq \frac{\alpha_{\mathrm{LDPCCC}}(3,6)}{\alpha_{\mathrm{G}}(J,K)} \cdot \nu \approx 3.6\nu. \qquad (16)$$

VI. Simulation Results

The performance of rate $R = 1/2$ regular $(3,6)$ TB-LDPCCCs is considered in this section. For this purpose, we constructed several TB-LDPCCCs with blocklengths $\mu = 2N$ from different convolutional codes with syndrome-former memories m_s. The construction of the mother LDPCCCs is based on the unwrapping procedure presented in [3]. To construct the TB-LDPCCCs, we use the wrapping of the last m_s columns of submatrices that was shown in section II. Additionally, the full-rank conditions mentioned in section III were satisfied to enable the encoding procedure expressed by (9).

All simulations were performed in an AWGN channel with BPSK modulation and the maximum number of decoding iterations was chosen to be 50. Moreover, for each simulation point, at least 5000 frames were decoded and at least 100 frame errors have occurred.

In Fig. 2, we can observe the BER performance of TB-LDPCCCs with different blocklengths μ that were originated from mother codes with syndrome-former memories $m_s = 257$ and $m_s = 513$. In the considered interval of blocklengths, the correction capabilities of the codes are improved by increasing the blocklengths μ. Additionally, we can observe that, at least in the low SNR regime, the performance of the codes is mainly determined by the blocklength μ. For instance, the codes with blocklengths $\mu = 4096$ and $\mu = 5120$ and different memories m_s have almost the same performance.

Another important comment that we can make based on Fig. 2 is about the behavior of the TB-LDPCCCs in comparison to their mother LDPCCCs (thick curves in Fig. 2). The LDPCCCs were decoded using pipeline decoders [3] in a streaming fashion. The pipeline decoders consist of 50 equal processors, which correspond to the 50 decoding iterations. As we can observe, the mother LDPCCCs have similar performances to their tail-biting codes with blocklengths given by $\mu \approx 12(m_s+1)$, which are the constraint lengths of the LDPCCCs times 6.

The figures 3, 4 and 5 show a comparison between TB-LDPCCCs and random LDPC block codes (LDPCBCs). All LDPCBCs used for comparison have the degrees $(3,6)$, rate $R = 1/2$ and they are free of cycles of length 4. They were decoded using a maximum of 50 decoding iterations in the same conditions as the TB-LDPCCCs. As we can observe in the figures, TB-LDPCCCs and random LDPCBCs have similar performances. In some cases, the TB-LDPCCCs are slightly better. For instance, the random LDPCBC with blocklength $\mu = 2048$ in Fig. 3 shows an error-floor while the TB-LDPCCCs are still in their waterfall region. In Fig. 5, the TB-LDPCCCC with memory $m_s = 513$ has a better performance then the random LDPCBC with the same blocklength.

Let us now consider the implementation complexity of the decoder for TB-LDPCCCs. As we know, the hardware complexity of LDPC codes is mainly determined by the graph connectivity necessary for the message passing. Recalling the section IV, we see that the circular pipeline decoder consists of I_P identical processors $\mathcal{D}_i$, where each of them have a graph connectivity in the order of $m_s + 1$. Furthermore, the design of the decoders for TB-LDPCCCs is very simplified because

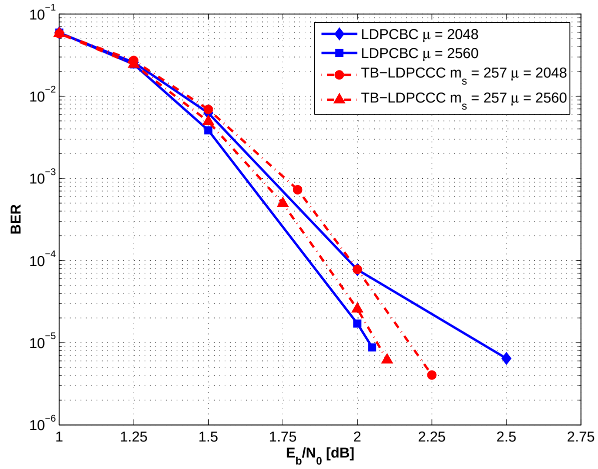

Fig. 3. BER performances of tail-biting LDPCCCs and random LDPC block codes for the blocklengths $\mu = 2048$ and $\mu = 2560$.

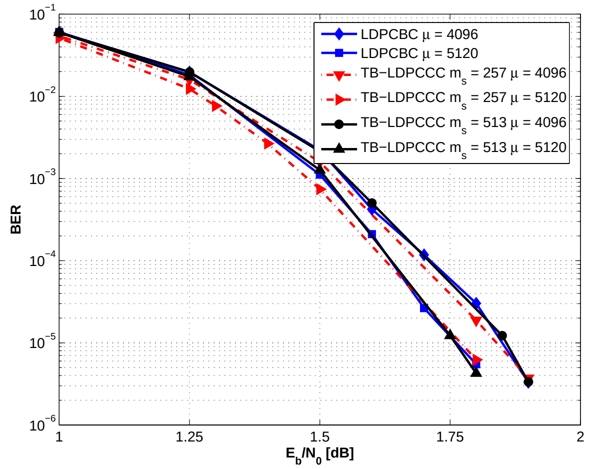

Fig. 4. BER performances of tail-biting LDPCCCs and random LDPC block codes for the blocklengths $\mu = 4096$ and $\mu = 5120$.

all the parallel processors are identical. These facts can be summarized to the point that, from the viewpoint of hardware implementation, an LDPCBC with the same complexity as the TB-LDPCCC would have a blocklength of $\mu = 2(m_s + 1)$. Thus, the random LDPCBCs of the simulations are about 5, 10 or even 20 times more complex than the TB-LDPCCCs.

VII. Conclusion

In this paper, we presented and analyzed a new class of iteratively decodable codes called *tail-biting LDPC convolutional codes*. We showed how to obtain these codes from a mother LDPC convolutional code by wrapping the last few columns of its syndrome former. The principles behind the encoder and decoder algorithms/architectures for these codes were also presented. In addition, distance bounds for these new codes were proposed and we could recognize the dual nature of these codes. Namely, their minimum distance depends on the free

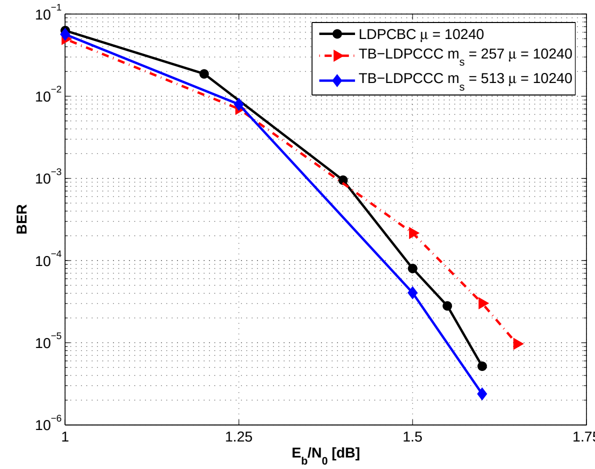

Fig. 5. BER performances of tail-biting LDPCCCs and random LDPC block codes for the blocklength $\mu = 10240$.

distance d_{free} of the mother LDPC convolutional codes and on the blocklength. Finally, we analyze the BER performance of these codes by means of simulations, where we could realize the performance vs. complexity advantages of the tail-biting LDPC convolutional codes in comparison to the conventional random LDPC block codes.

Our further research on this topic includes the study of irregular TB-LDPCCCs, as well as, efficient high-speed VLSI designs of the encoders and decoders for these codes.

Acknowledgment

M.B.S. Tavares and G.P. Fettweis were supported in part by the BMBF within the project WIGWAM under grant 01 BU 370. K.Sh. Zigangirov was supported in part by the NSF grant CCR02-05310 and NASA grant NNG05GH73G.

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] 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.

[4] R. M. Tanner, "Error correcting coding system," U.S. Patent No. 4295218, Oct. 1981.

[5] D. J. Costello, Jr., A. E. Pusane, S. Bates, and K. Sh. Zigangirov, "A comparison between LDPC block and convolutional codes," in *Proc. Information Theory and Applications Workshop*, San Diego, USA, 2006.

[6] A. Jimenéz Feltström, *Iteratively Decodable Convolutional Codes: Analysis and Implementation Aspects*, Lund University, Lund, Sweden, 2006.

[7] A. Sridharan, D. Truhachev, M. Lentmaier, D. J. Costello, Jr., and K. Sh. Zigangirov, "Distance bounds for an ensemble of LDPC convolutional codes," submitted to *IEEE Trans. on Inform. Theory*, July 2004.