Thresholds of Braided Convolutional Codes on the AWGN Channel

Farooq, Muhammad Umar; Moloudi, Saeedeh; Lentmaier, Michael

Published in:
2018 IEEE International Symposium on Information Theory, ISIT 2018

DOI:
10.1109/ISIT.2018.8437937

2018

Document Version:
Peer reviewed version (aka post-print)

Thresholds of Braided Convolutional Codes on the AWGN Channel

Muhammad Umar Farooq, Saeedeh Moloudi, and Michael Lentmaier
Dept. of Electrical and Information Technology,
Lund University, Sweden
Emails: {muhammad.umar_farooq, saeedeh.moloudi, michael.lentmaier}@eit.lth.se

***Abstract*—In this paper, we perform a threshold analysis of braided convolutional codes on the additive white Gaussian noise (AWGN) channel. The decoding thresholds are estimated by Monte-Carlo density evolution techniques and compared with approximate thresholds from an erasure channel prediction. The results show that, with spatial coupling, the predicted thresholds are very accurate and quickly approach capacity if the coupling memory is increased. For uncoupled ensembles with random puncturing, the prediction can be improved with help of the AWGN threshold of the unpunctured ensemble.**

I. Introduction

Braided convolutional codes (BCCs) [1], [2], [3] are a class of spatially-coupled (SC) turbo-like codes with regular graph structure. On the binary erasure channel (BEC), explicit density evolution (DE) equations have been derived for BCCs in [4], which can be used to efficiently compute exact decoding thresholds for that channel. The results show that BCCs have superior maximum-a-posteriori (MAP) decoding thresholds compared to parallel or serially concatenated codes on the BEC [5]. Furthermore it has been proven analytically in [5] that threshold saturation occurs, i.e., with spatial coupling a belief propagation (BP) decoder can achieve the same threshold as an optimal MAP decoder.

The aim of this paper is to compute the BP thresholds of BCCs on the additive white Gaussian noise (AWGN) channel. For this channel, exact DE equations are not available for turbo-like codes, and Monte Carlo (MC) methods are usually applied to estimate decoding thresholds. One of the difficulties of such an approach is that the graphs of spatially coupled systems contain a large number of edge types whose message densities have to be considered individually during DE. This requires significantly larger computational efforts than classical methods, like the single edge-type extrinsic information transfer characteristics (EXIT) chart analysis [6].

We use Monte-Carlo density evolution (MC-DE) to estimate the thresholds of uncoupled and coupled BCCs with and without puncturing. As an efficient alternative, we then consider the erasure channel approximation by Chung [7] for predicting the AWGN channel thresholds from those of the BEC and compare the results. Finally, we demonstrate that for randomly punctured ensembles, analogously to LDPC codes [8], the thresholds of BCCs of all higher rates can immediately be predicted from the unpunctured thresholds of the BEC and/or the AWGN channel. Some simulation results are also given.

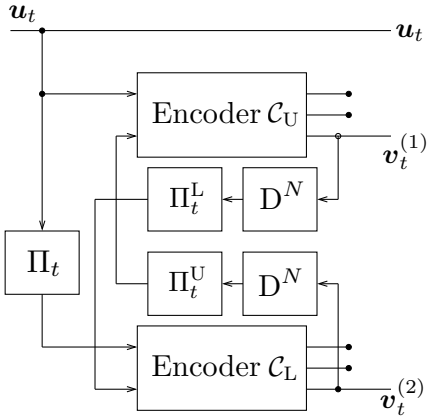

Fig. 1. Blockwise BCCs: turbo-like codes with parity feedback ($R = 1/3$).

II. Braided Convolutional Codes

BCCs were originally introduced in [1]. Their characteristics is that the parity symbols of one component encoder are used as information symbols of the other and vice versa. Due to this, both information and parity symbols are protected by both component codes in a symmetric way. In this paper we consider an example of BCCs of rate $R = 1/3$, which are defined by using two systematic component convolutional encoders of rate $R_{cc} = 2/3$ and three permutors of length N. The component code and the encoding method is same as used in [5]. In particular, we are using the blockwise BCCs, for which an encoder diagram is shown in Fig. 1. The parity symbols created by one encoder at time t pass a delay of one block, D^N and a permutor before entering the other encoder at time $t + 1$ [2].

To compare BCCs with parallel or serially concatenated codes, it sometimes can be useful to define an uncoupled equivalent of BCCs. This can be achieved by removing the delay in the encoder depicted in Fig. 1. Since the feedback now occurs without the delay, a straightforward encoding by means of the trellis is not possible. But the code is still well defined by the trellis constraints that the code symbols have to satisfy.

Turbo-like codes, like LDPC codes, can be described by factor graphs [9], [10]. This way of expressing a code is useful in describing the exchange of messages in an iterative BP decoder, as well as for the corresponding DE analysis.

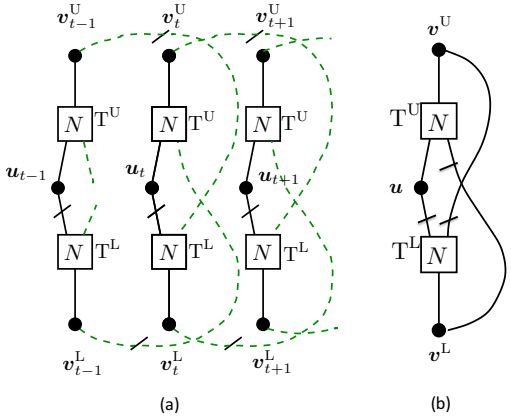

Fig. 2. Compact graph representation.

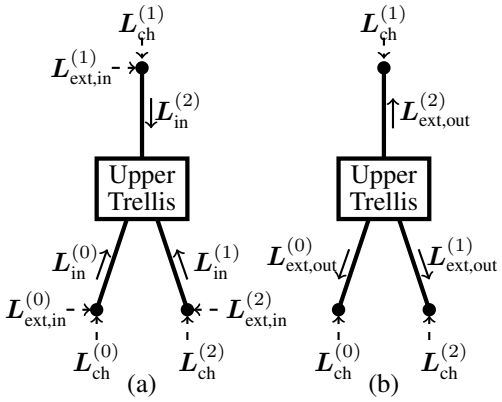

Fig. 3. Monte Carlo density evolution process.

Instead of a conventional factor graph, we use a compact graph representation as introduced in [5]. The compact graph of a BCC and its uncoupled equivalent is shown in Fig. 2. Each block of symbols is represented by a variable node and each trellis by a factor node. A permutor is indicated by a short line that crosses an edge in the graph.

Similar to SC-LDPC codes, a BCC code can be obtained by repeating the graph of an uncoupled code and spreading some edges across $m+1$ neighboring blocks, where m denotes the coupling memory. The original BCCs shown in Fig. 2(a) have coupling memory $m = 1$.

III. Monte Carlo Density Evolution

In order to describe the MC-DE process, let us consider the upper decoder of the uncoupled (UC) BCC shown in Fig. 2(b). The exchange of messages with the upper decoder is depicted in Fig. 3. Due to symmetry, the processing at the lower decoder follows analogously. In each iteration of MC-DE, assuming a flooding schedule, the decoding at the upper and lower decoder is performed independently in parallel and the *densities* of updated output messages are exchanged. Iterative exchange of densities continues in this fashion until the decoding error probability converges. The key points of the MC-DE process can be summarized in the following three major steps:

1) *Variable node update:* Each variable node $k = 0, 1, 2$ in Fig. 3(a) takes the sequence $\boldsymbol{L}_{\text{ch}}^{(k)}$ of channel LLRs and the sequence $\boldsymbol{L}_{\text{ext,in}}^{(j)}$ of incoming extrinsic LLRs received from output j of the connected lower trellis node and combines them to the updated message sequence $\boldsymbol{L}_{\text{in}}^{(i)} = \boldsymbol{L}_{\text{ch}}^{(k)} + \boldsymbol{L}_{\text{ext,in}}^{(j)}$, which forms the input i of the upper trellis node. All sequences are of equal length N.
2) *Trellis node update:* The trellis node receives the three input sequences from the different variable nodes corresponding to symbol blocks $\boldsymbol{u}$, $\boldsymbol{v}^{\text{U}}$ and $\boldsymbol{v}^{\text{L}}$. The node performs decoding and produces the updated sequences $\boldsymbol{L}_{\text{ext,out}}^{(i)}$ of extrinsic LLRs at each output $i = 0, 1, 2$ of the trellis node in Fig. 3(b). Finally, these output message sequences are used to estimate the message densities $f(L_{\text{ext,out}}^{(i)})$.

TABLE I
Uncoupled BCC Thresholds

Thresholds E_b/N_0 (dB)	Pattern	Rate 1/3	Rate 1/2	Rate 2/3
MC	P1	1.003	2.408	-
GA	P1	1.023	2.399	-
GA SE	P1	1.050	2.708	-
MC	P2	1.003	2.121	4.151
GA	P2	1.018	2.128	4.062
GA SE	P2	1.048	2.161	4.131

3) *Drawing samples from message densities:* In this step independent LLR sequences $\boldsymbol{L}_{\text{ch}}^{(k)}$, $k = 0, 1, 2$ and $\boldsymbol{L}_{\text{ext,in}}^{(j)}$, $j = 0, 1, 2$ are created from the channel density and the densities $f(L_{\text{ext,out}}^{(j)})$ received from the lower trellis node. These sequences are used in the next MC-DE iteration.

In the literature, the message densities $f(L_{\text{ext,out}}^{(i)})$ are often *approximated* by Gaussian densities as well in MC-DE. In this case, it is not necessary to estimate the exact $f(L_{\text{ext,out}}^{(i)})$ in step 2. Instead, some parameter like the mean, the variance or the mutual information corresponding to the LLR sequences is computed to characterize the Gaussian density.

Table I shows the uncoupled BCC thresholds obtained via MC-DE with two different puncturing methods. P2 puncturing refers to when uniform random puncturing is applied on both information and parity bits, whereas in P1 random puncturing is applied only on parity bits. MC refers to the threshold obtained via the MC-DE algorithm and GA refers to the threshold when $f(L_{\text{ext,out}}^{(i)})$ is approximated by a Gaussian density and $\boldsymbol{L}_{\text{in}}^{(i)}$ are drawn from it.

In determining MC and GA thresholds, the statistics along the incoming edges to the trellis nodes are different. It means that for a code with multi edge-types (ME), such as BCCs, the $f(L_{\text{in}}^{(i)})$ densities are different along each incoming edge. In the literature, these densities $f(L_{\text{in}}^{(i)})$ are often averaged to obtain a single density $f(L_{\text{in}})$, which then is used along all incoming edges to the trellis nodes. This reduces the complexity of DE analysis, but the corresponding single edge-type (SE) ensemble can have a different threshold.

In Table I it can be observed that the MC thresholds are closer to the GA thresholds for low rates. Whereas, the difference in these thresholds becomes high as the fraction of puncturing increases. With punctured bits, $f(L_{\text{in}}^{(i)})$ will be a mixture of the Gaussian and erasure densities. The erasure density will be dominant in this mixture at higher rate. Therefore we expect that the Gaussian approximation of the densities at higher rate results in inaccuracies. Furthermore, the difference from GA SE thresholds to other thresholds is relatively larger. The GA SE case is equivalent to the classical EXIT chart technique. In case of ME ensembles, the MC-DE process with GA of densities $f(L_{\text{in}}^{(i)})$ is a generalization of the protograph LDPC EXIT analysis technique [11] to component codes with trellis constraints.

In order to get a high accurancy of estimated thresholds via MC-DE, a large number of trellis node output messages are collected before computing the densities within each iteration. In our experiments, the statistics are considered stabilized when the bit error rate (BER) as a function of the number of simulated blocks within an iteration reaches a steady state with a difference of .0001. This requirement of accuracy control makes MC-DE take days to compute UC-BCCs thresholds and it takes even much longer for the SC-BCCs. The accuracy of the MC-DE thresholds can be increased by running MC-DE for a longer time.

Life can be much easier if we can predict the thresholds on the Gaussian channel reliably and much faster than the MC-DE does. One way would be to use the exact DE equations as is used in [5] for BCCs over the BEC. However, the DE equations for the BCCs on the AWGN channel are not available. Therefore, we have to look for an alternative solution to find the thresholds of BCCs much faster than MC-DE does.

IV. Erasure Channel Prediction of AWGN Channel Thresholds

It has been observed in [7] that the thresholds of LDPC codes on the AWGN channel can be *approximated* by the corresponding thresholds on the BEC. Such an erasure channel prediction (ECP) is computationally attractive for turbo-like code ensembles, since BEC thresholds can be computed exactly with relatively small efforts. In this section we will apply this approach to uncoupled and coupled BCC ensembles and compare the resulting thresholds with those obtained from MC-DE.

Let $C_E(\varepsilon) = 1 - \varepsilon$ denote the capacity of a BEC with erasure probability ε and $C_G(\sigma)$ denote the capacity of a binary-input AWGN channel with noise variance σ^2. Let ε^* and σ^* denote the DE thresholds computed on the two types of channels for a given code ensemble. The ECP is based on the observation that $C_E(\varepsilon^*) \approx C_G(\sigma^*)$, which suggests the approximation

$$\sigma^* \approx C_G^{-1}\left(C_E(\varepsilon^*)\right) = C_G^{-1}\left(1-\varepsilon^*\right) \ . \tag{1}$$

Using (1), the AWGN threshold of a given ensemble can now be predicted from the corresponding BEC threshold ε^*.

TABLE II
Predicted vs Estimated Thresholds (E_b/N_0) of BCC

		Erasure		UC BCC		SC BCC	
Rate	Pattern	ϵ_{SC}^1	ϵ_{UC}	ECP	MC	ECP	MC
1/3	P1	0.6609	0.5541	1.21	1.00	-0.39	-0.39
1/2	P1	0.4932	0.3013	2.71	2.40	0.27	0.25
2/3	P1	0.3257	-	-	-	1.15	1.12
3/4	P1	0.2411	-	-	-	1.74	1.70
4/5	P1	0.1915	-	-	-	2.16	2.12
1/3	P2	0.6609	0.5541	1.21	1.00	-0.40	-0.39
1/2	P2	0.4914	0.3312	2.33	2.12	0.30	0.29
2/3	P2	0.3219	0.1083	4.33	4.15	1.20	1.18
3/4	P2	0.2371	-	-	-	1.80	1.77
4/5	P2	0.1862	-	-	-	2.24	2.21

Table II shows the resulting predicted thresholds for UC-BCCs and SC-BCCs along with the corresponding MC-DE thresholds. The original BEC thresholds are also given, where the values of puncturing pattern P1 are identical to Table II of [5]. We can see from Table II that the ECP thresholds of SC-BCCs are quite close to the MC thresholds. However, the difference is larger on higher rates than it is on lower rates as we have observed in the UC-BCCs thresholds. The ECP and MC thresholds of the UC-BCCs have bigger difference than the SC-BCCs. Furthermore, the ECP and the MC thresholds are much closer to each other on P2 puncturing compared to P1 puncturing.

V. Thresholds of Randomly Punctured Ensembles

Can we predict the thresholds such that the difference of the predicted and the estimated threshold does not increase as the amount of puncturing increases? In [8], it has been shown that for the P2 punctured LDPC codes, it is possible to closely predict the thresholds on the AWGN channel, given just the threshold of the unpunctured code ensemble and the design rate R. In the following, we will apply the methods discussed in [8] to the randomly punctured BCC ensembles and discuss the results obtained by it.

A. θ_E Predictions

Consider random puncturing with a fraction $\alpha = p/n$, where p denotes the number of punctured bits and n the total number of bits before puncturing. For the BEC, the threshold $\epsilon^*(\alpha)$ of the punctured ensemble is equal to

$$\epsilon^*(\alpha) = 1 - \theta_E R(\alpha) \ , \tag{2}$$

where

$$R(\alpha) = \frac{R}{1-\alpha} \tag{3}$$

denotes the desired rate after puncturing and

$$\theta_E = \frac{1-\epsilon^*}{R} \tag{4}$$

and is a parameter that follows from the threshold ϵ^* and rate R of the unpunctured ensemble.

TABLE III
UNCOUPLED BCC THRESHOLDS

Thresholds	**Pattern**	**Rate**		
E_b/N_0 (dB)		1/3	1/2	2/3
MC-DE	P2	1.00	2.12	4.15
θ_E Predicted	P2	1.21	2.33	4.33
θ_G Predicted	P2	1.00	2.05	3.79
MP (θ_E and θ_G)	P2	1.00	2.19	4.19
Erasure Probability	P2	0.5541	0.3312	0.1083

The elegance of this method is that it will provide the exact BEC thresholds for all rates $R(\alpha)$, $R \leq R(\alpha) \leq 1/\theta_E$, based on a single parameter θ_E and hence it is not required to perform DE on all different rates. However, this method is limited to the puncturing pattern P2 only.

Once we know the BEC thresholds, we can apply the ECP method discussed in Section IV. Equivalently, we can write

$$h_G(\sigma^*(\alpha)) \approx h_E(\varepsilon^*(\alpha)) = \epsilon^*(\alpha) = 1 - \theta_E R(\alpha) \ , \quad (5)$$

where $h_G(\sigma) = 1 - C_G(\sigma)$ and $h_E(\varepsilon) = 1 - C_E(\varepsilon) = \varepsilon$ denote the conditional entropy of the AWGN channel and the BEC, respectively. From (5) we can compute the AWGN thresholds in terms of standard deviation σ or signal-to-noise ratio E_b/N_0. The results obtained via this method for UC-BCCs are given in Table III.

B. θ_G Predictions

While the θ_E prediction works especially well for high code rates, it can be further improved for low rates if the unpunctured AWGN threshold σ^* is available. Then

$$h_G(\sigma^*(\alpha)) \approx 1 - \theta_G R(\alpha) \ , \quad (6)$$

where

$$\theta_G = \frac{1 - h_G(\sigma^*)}{R} \ . \quad (7)$$

It can be seen in Table III that for the lower code rates, the MC thresholds and the θ_G predictions are quite close. However, for a higher fraction of punctured bits, the MC thresholds are closer to the θ_E predictions.

C. Mixed Predictions

To account for the higher puncturing fraction in the prediction of thresholds on the AWGN channel, a mixed prediction (MP) method is suggested in [8], where both θ_E and $h_G(\sigma^*)$ are used.

The important idea behind the mixed prediction method is that all the punctured code ensemble thresholds on the BEC, when viewed in the entropy domain, lie on a straight line. The slope of this line is characterized by θ_E. It is further demonstrated in [8] that for P2 punctured LDPC codes on the AWGN channel the estimated entropies, corresponding to the E_b/N_0 thresholds, are observed to follow a straight line.

From the observation that at lower rate the estimated thresholds are closer to the θ_G predictions and at higher rates are closer to the θ_E predictions, we obtain the following mixed prediction

$$h_G(\sigma_{BP}(\alpha)) \approx \frac{(R(\alpha) - R) \cdot (1 - \theta_E - h_G(\sigma^*))}{1 - R} + h_G(\sigma^*), \quad (8)$$

where $R \leq R(\alpha) \leq R_{\text{max}}$. R_{max} is an intercept that can be obtained by setting $h_G(\sigma_{BP}(\alpha)) = 0$ in (8). This straight line prediction passes through the unpunctured rate $R(0)$ and the associated threshold (entropy) on the AWGN channel and $R(\alpha) = 1$ and its associated threshold (entropy) on the BEC. The results of the mixed predictions are shown in Fig. 4 for the P2 punctured UC-BCCs ensembles. The estimated MC-DE thresholds almost follow the mixed prediction line in this figure. However, the estimated thresholds do not strictly lie on a straight line as was the case in [8]. The related E_b/N_0 thresholds of UC-BCCs are also given in Table III.

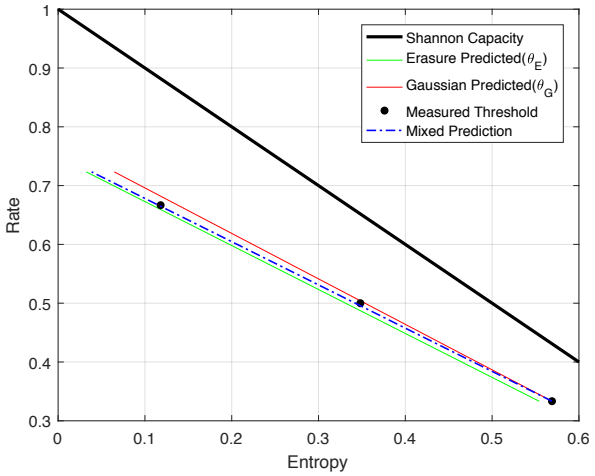

Fig. 4. Entropy vs rate of UC-BCCs.

It can be seen in Table II that for SC-BCCs, the MC-DE and predicted thresholds are almost identical. As a result of threshold saturation, the SC-BCCs performance is much better compared to the performance of UC-BCCs and the gap to capacity is much smaller for SC-BCCs, which can be seen in Fig. 5. θ_E and θ_G predictions for the punctured UC-BCCs have been made by using $\theta_E = 1.337$ and $\theta_G = 1.293$ respectively. Similarly, θ_E predictions for the punctured $m = 1$ SC-BCCs have been made by using $\theta_E = 1.017$. Since the θ_E predictions are very close to the MC-DE thresholds, θ_G and mixed predictions are hard to distinguish from them.

Consider now a larger coupling memory $m = 3$. The predicted thresholds are given in Table IV, where it can be seen that $m = 3$ SC-BCCs outperform $m = 1$ SC-BCCs and operate very close to the Shannon limit. Again, the θ_E predictions and MC-DE thresholds are very close to each other and we have only provided the θ_E predictions in this table.

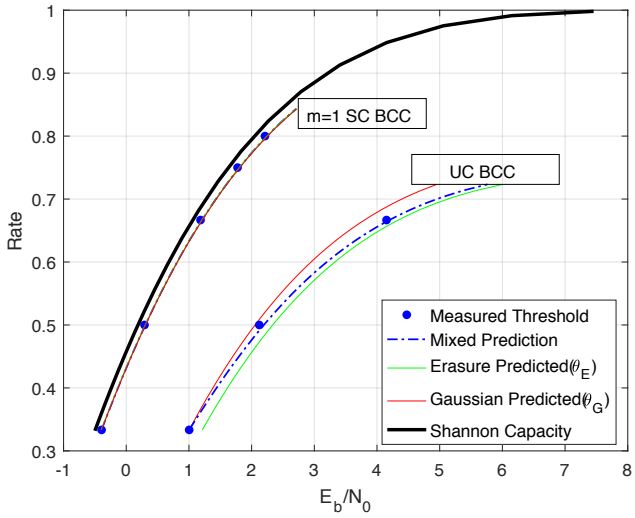

Fig. 5. E_b/N_0 vs rate of SC and UC-BCCs.

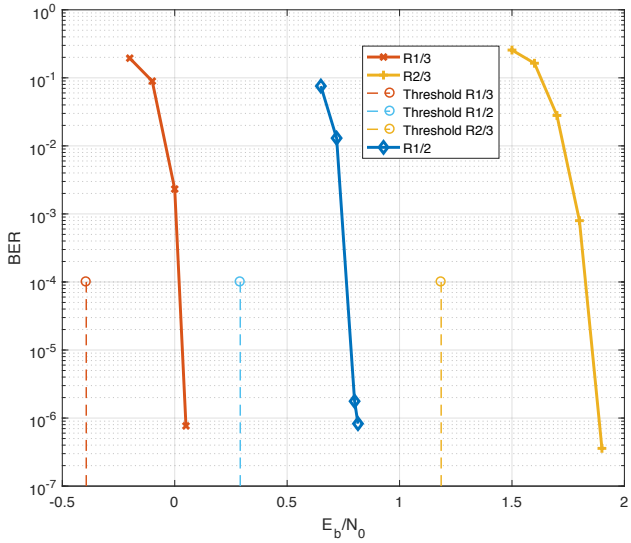

Fig. 6. BER vs E_b/N_0 thresholds of P2 punctured SC-BCCs.

TABLE IV
BCC COUPLED - $m = 3$

Thresholds			**Rate**		
E_b/N_0 (dB)	1/3	1/2	2/3	3/4	4/5
Shannon Capacity	-0.50	0.18	1.06	1.63	2.05
θ_E prediction	-0.4585	0.2307	1.1151	1.6924	2.1166
Erasure Probability	0.6644	0.4967	0.3289	0.2450	0.1947

D. Simulation Results

The simulated BER performance of P2 punctured $m = 1$ SC-BCCs on the AWGN channel is shown in Fig. 6. The simulations are obtained with a sliding window decoder [3] with window size $w = 5$, and 20 iterations at each window position. We can see from Fig. 6 that for all the considered rates $R = 1/3, 1/2, 2/3$ the BER curves are in accordance with their corresponding thresholds.

VI. Conclusions

In this paper, have presented MC-DE as a technique to compute the AWGN channel thresholds of spatially-coupled turbo-like codes. Furthermore, we have applied the threshold prediction methods presented for LDPC codes in [8] for predicting BCC thresholds. The θ_E and θ_G predictions of the UC and SC-BCC ensembles have been compared with the estimated thresholds obtained by MC-DE. The results show that, with spatial coupling, the predicted thresholds are very accurate and quickly approach capacity if the coupling memory is increased. For uncoupled ensembles with random puncturing, the prediction can be improved with help of the AWGN threshold of the unpunctured ensemble.

References

[1] W. Zhang, M. Lentmaier, K. S. Zigangirov, and D. J. Costello, Jr., "Braided convolutional codes: A new class of turbo-like codes," *IEEE Transactions on Information Theory*, vol. 56, no. 1, pp. 316–331, Jan. 2010.

[2] D. J. Costello, Jr., M. Lentmaier, and D. G. M. Mitchell, "New perspectives on braided convolutional codes," in *Proc. IEEE 9th International Symposium on Turbo codes and Iterative Information Processing*, Brest, France, Sep. 2016.

[3] M. Zhu, D. G. M. Mitchell, M. Lentmaier, D. J. Costello, Jr., and B. Bai, "Braided convolutional codes with sliding window decoding," *IEEE Transactions on Communications*, vol. 65, no. 9, pp. 3645–3658, Sep. 2017.

[4] S. Moloudi and M. Lentmaier, "Density evolution analysis of braided convolutional codes on the erasure channel," in *Proc. IEEE International Symposium on Information Theory*, Honolulu, HI, July 2014.

[5] S. Moloudi, M. Lentmaier, and A. Graell i Amat, "Spatially coupled turbo-liked codes," *IEEE Transactions on Information Theory*, vol. 63, no. 10, pp. 6199–6215, Jan. 2017.

[6] S. ten Brink, "Convergence behavior of iteratively decoded parallel concatenated codes," *IEEE Transactions on Communications*, vol. 40, no. 10, pp. 1727–1737, Oct. 2001.

[7] S.-Y. Chung, "On the construction of some capacity-approaching coding schemes," Ph.D. dissertation, Dept. of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA, Sept. 2000.

[8] D. G. M. Mitchell, M. Lentmaier, A. E. Pusane, and D. J. Costello, Jr., "Randomly punctured LDPC codes," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 2, pp. 408–421, Feb. 2016.

[9] N. Wiberg, H. Loeliger, and R. Kötter, "Codes and iterative decoding on general graphs," *European Transactions on Telecommunications*, vol. 6, no. 5, pp. 513–525, Sep. 1995.

[10] F. Kschischang, B. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 498–519, Feb. 2001.

[11] G. Liva and M. Chiani, "Protograph LDPC codes design based on EXIT analysis," in *Proc. IEEE Globecom*, Washington, DC, Nov. 2007.