(article starts on next page)

A Unified Ensemble of Concatenated Convolutional Codes

Saeedeh Moloudi[†], Michael Lentmaier[†], and Alexandre Graell i Amat[‡]
†Department of Electrical and Information Technology, Lund University, Lund, Sweden
‡Department of Signals and Systems, Chalmers University of Technology, Gothenburg, Sweden
{saeedeh.moloudi,michael.lentmaier}@eit.lth.se, alexandre.graell@chalmers.se

Abstract—We introduce a unified ensemble for turbo-like codes (TCs) that contains the four main classes of TCs: parallel concatenated codes, serially concatenated codes, hybrid concatenated codes, and braided convolutional codes. We show that for each of the original classes of TCs, it is possible to find an equivalent ensemble by proper selection of the design parameters in the unified ensemble. We also derive the density evolution (DE) equations for this ensemble over the binary erasure channel. The thresholds obtained from the DE indicate that the TC ensembles from the unified ensemble have similar asymptotic behavior to the original TC ensembles.

I. INTRODUCTION

Over the last few years, research on low-density parity-check (LDPC) convolutional codes [1], also known as spatially coupled LDPC (SC-LDPC) codes [2], has become very popular. It is proved that for these codes, the belief propagation (BP) decoder can achieve the threshold of the maximum-a-posteriori (MAP) decoder [2], [3]. This remarkable phenomenon is known as threshold saturation. Spatial coupling is a general concept that is not limited to LDPC codes. Recently, spatially coupled turbo-like codes (SC-TCs) were introduced in [4], [5], [6]. In these works, the spatial coupling of the four main classes of TCs was considered. These included parallel concatenated codes (PCCs) [7], serially concatenated codes (SCCs) [8], braided convolutional codes (BCCs)[9], and hybrid concatenated codes (HCCs) [10], [11]. The density evolution (DE) analysis performed in [4], [5], [6] suggests that SC-TCs have an excellent asymptotic behavior and for them, threshold saturation occurs. This gives a new perspective in designing a concatenated coding scheme: optimizing the uncoupled ensembles for achieving the best BP threshold may not necessarily lead to the best overall performance.

TCs are adopted in many communication standards. Each class of TCs exhibits a unique asymptotic behavior. While certain classes—such as PCCs—yield good BP thresholds, certain others—such as SCCs and BCCs—have excellent MAP thresholds. However, spatial coupling gives a new perspective in the designing of TCs; relying on threshold saturation, we can optimize the component codes of SC-TCs for higher MAP threshold. In this way, powerful ensembles with good distance properties can perform very close to capacity [6].

So far, the different classes of TCs have been considered separately. A unified ensemble which contains all main TC ensembles can unify the frameworks for analysis, and clarify the connections between the TC classes. In fact, this ensemble can lead to a better understanding of the similarities and differences between various TC classes and the possible trade-offs in the code design. In addition, the ensemble allows us to design new ensembles that do not belong to any of the original classes of TCs.

In [12], the authors introduced an ensemble which unifies PCCs and SCCs. This ensemble is based on concatenations of several component encoders and does not cover the BCC and HCC ensembles. In this paper, we introduce an ensemble of concatenated convolutional codes that encompasses all the above-mentioned four major classes of TCs. For simplicity, we only use a single rate-1 component code [1]. In other words, the different trellises are combined to a single self-concatenated trellis. Probably, the most famous class of self-concatenated convolutional codes are repeat accumulate (RA) codes, first introduced in [13]. This class of codes is covered by the proposed ensemble if the component code in the equivalent PCC ensemble is set as an accumulator. In order to find a self-concatenated equivalent for the other classes of TCs, some feedback path has to be introduced in the encoder structure. The differences between the various original TC ensembles are then reflected in the permutation structure and the amount of feedback in the unified ensemble.

We also derive the exact density evolution equations for the binary erasure channel (BEC). Using these equations, we compute the BP thresholds of the corresponding classes of TCs and we show that the obtained thresholds are very close to the thresholds of the original ensembles.

II. STEPS TOWARD THE SELF-CONCATENATED ENSEMBLE

In this section, for each class of TCs, we separately describe how to reduce the number of component codes in order to obtain the equivalent self-concatenated ensemble.

A. Parallel Concatenated Codes

Fig. 1(a) and (b) show the encoder block diagram and compact graph representation [6] of a PCC, respectively. The

---

[1]In particular, we use a single rate-1/2 component encoder for which the systematic bits are punctured, thus obtaining a rate-1 encoder. We remark that it is also possible to build the unified ensemble based on a component encoder with general rate R and considering proper puncturing of the encoder.
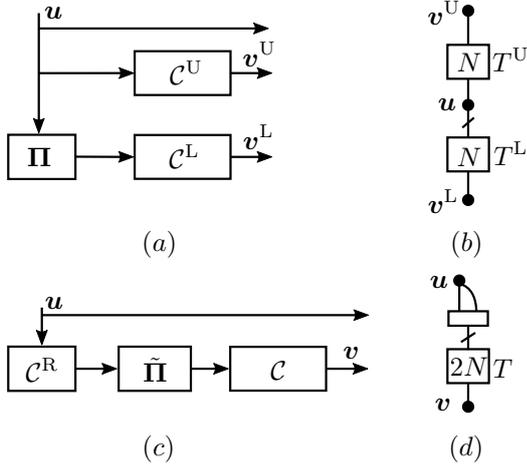
Fig. 1. (a) Block diagram of a PCC, (b) Compact graph of a PCC, (c) Block diagram of a self-concatenated PCC, and (d) Compact graph of a self-concatenated PCC
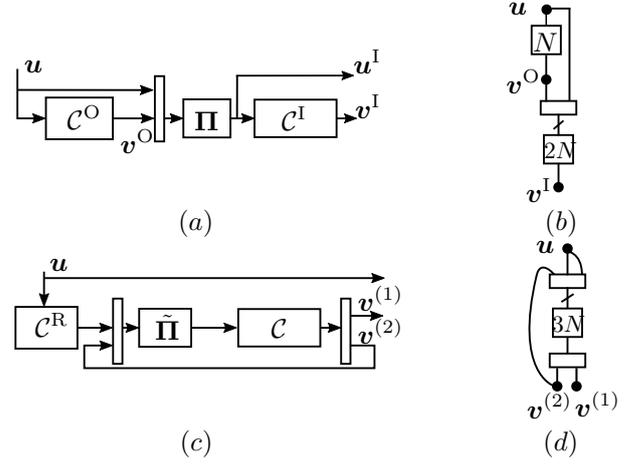

Fig. 2. (a) Block diagram of a SCC, (b) Compact graph of a SCC, (c) Block diagram of a self-concatenated SCC, and (d) Compact graph of a self-concatenated SCC

considered PCC ensemble is built of two identical rate-1 component encoders, called upper and lower encoders, and shown by $\mathcal{C}^{\mathrm{U}}$ and $\mathcal{C}^{\mathrm{L}}$, respectively. The information sequence $\boldsymbol{u}$ is connected to $\mathcal{C}^{\mathrm{U}}$ to produce the parity sequence $\boldsymbol{v}^{\mathrm{U}}$. Likewise, a reordered copy of $\boldsymbol{u}$ is connected to $\mathcal{C}^{\mathrm{L}}$ to produce the parity sequence $\boldsymbol{v}^{\mathrm{L}}$. The output of the PCC encoder is the tuple $(\boldsymbol{u}, \boldsymbol{v}^{\mathrm{U}}, \boldsymbol{v}^{\mathrm{L}})$. In the compact graph representation (see Fig. 1(b)), the trellises corresponding to $\mathcal{C}^{\mathrm{U}}$ and $\mathcal{C}^{\mathrm{L}}$, are depicted by squares (factor nodes) and denoted by T^{U} and T^{L}, respectively. These factor nodes are labeled with the length of the corresponding trellises. Each of the sequences $\boldsymbol{u}$, $\boldsymbol{v}^{\mathrm{U}}$, and $\boldsymbol{v}^{\mathrm{L}}$ is represented by a black circle, called variable node. The permutation $\boldsymbol{\Pi}$ in the block diagram is replaced in Fig. 1(b) by a line that crosses the edge between $\boldsymbol{u}$ and $\boldsymbol{v}^{\mathrm{L}}$.

Fig. 1(c) and (d) show respectively the encoder block diagram and compact graph representation of the self-concatenated coding ensemble corresponding to a PCC. In this ensemble, the two component encoders of the PCC ensemble are replaced by a repetition encoder $\mathcal{C}^{\mathrm{R}}$ (with repeating factor 2) followed by a rate-1 convolutional encoder $\mathcal{C}$. The information sequence $\boldsymbol{u}$ is connected to $\mathcal{C}^{\mathrm{R}}$ to produce the sequence $(\boldsymbol{u}, \boldsymbol{u})$. The resulting sequence is reordered by the permutation $\tilde{\boldsymbol{\Pi}}$ and used as input to $\mathcal{C}$. The parity sequence $\boldsymbol{v}$ has length $2N$ and corresponds to both $\boldsymbol{v}^{\mathrm{U}}$ and $\boldsymbol{v}^{\mathrm{L}}$ in the original ensemble. Note that, by replacing $\mathcal{C}$ by an accumulator in Fig. 1(c), an RA code can be obtained. However, the ensemble in Fig. 1(c) is more general as $\mathcal{C}$ can be any convolutional encoder.

In the compact graph representation (see Fig. 1(d)), the repetition of the information sequence is shown by increasing the degree of the corresponding variable node. As it is shown in the figure, the sequence $\boldsymbol{u}$ and its repetition are multiplexed to produce the sequence $(\boldsymbol{u}, \boldsymbol{u})$. The multiplexer is represented by a rectangle. The resulting sequence is connected to trellis T to produce the parity sequence $\boldsymbol{v}$. Note that the length of the trellis in the self-concatenated ensemble is twice of the length of T^{U} and T^{L} in the original ensemble.

In this paper we consider block-wise multiplexers. By

selecting

$$\tilde{\boldsymbol{\Pi}} = \begin{bmatrix} \boldsymbol{I} & 0 \\ 0 & \boldsymbol{\Pi} \end{bmatrix}, \tag{1}$$

where $\boldsymbol{I}$ is the $N \times N$ identity matrix, the self-concatenated ensemble is equivalent to the original ensemble.

B. Serially Concatenated Codes

The encoder block diagram and the compact graph representation of the SCC ensemble are shown in Fig. 2(a) and (b), respectively. This ensemble is built of two identical rate-1 component encoders called outer and inner encoders and shown by $\mathcal{C}^{\mathrm{O}}$ and $\mathcal{C}^{\mathrm{I}}$, respectively. The length-N information sequence, $\boldsymbol{u}$, is connected to $\mathcal{C}^{\mathrm{O}}$ to produce the parity sequence $\boldsymbol{v}^{\mathrm{O}}$. Then, the sequences $\boldsymbol{u}$ and $\boldsymbol{v}^{\mathrm{O}}$ are multiplexed and reordered. The resulting sequence is used as input for $\mathcal{C}^{\mathrm{I}}$ to produce the parity sequence $\boldsymbol{v}^{\mathrm{I}}$.

The encoder block diagram and compact graph representation of the equivalent self-concatenated ensemble are shown in Fig. 2(c) and (d), respectively. In this ensemble, the trellises of the outer and inner encoders are combined to make a trellis with length $3N$. Similarly to the self-concatenated ensemble for PCCs, $\boldsymbol{u}$ is connected to $\mathcal{C}^{\mathrm{R}}$ with repetition factor 2 to produce the sequence $\tilde{\boldsymbol{u}} = (\boldsymbol{u}, \boldsymbol{u})$. In the original ensemble $\boldsymbol{v}^{\mathrm{O}}$ is used as part of the input to $\mathcal{C}^{\mathrm{I}}$. To satisfy this condition with only one component encoder, the overall parity sequence of the self-concatenated ensemble, $\boldsymbol{v}$, is divided into two sequences $\boldsymbol{v}^{(1)}$ and $\boldsymbol{v}^{(2)}$, of length $2N$ and N, respectively. Then, $\boldsymbol{v}^{(2)}$ is used as a part of the input sequence through a feedback path. The feedback path connects $\boldsymbol{v}^{(2)}$ to a multiplexer. Then, this sequence is multiplexed with sequence $\tilde{\boldsymbol{u}}$. The resulting sequence is reordered by $\tilde{\boldsymbol{\Pi}}$ and used as input to $\mathcal{C}$. Note that sequences $\boldsymbol{v}^{(1)}$ and $\boldsymbol{v}^{(2)}$ correspond to $\boldsymbol{v}^{\mathrm{O}}$ and $\boldsymbol{v}^{\mathrm{I}}$ in the original ensemble, respectively.

We remark that, in general, the encoder of the self-concatenated ensemble is not causal. However, this problem can be solved by proper selection of $\tilde{\boldsymbol{\Pi}}$ or by spatial coupling. By selecting $\tilde{\boldsymbol{\Pi}}$ as in (1) the corresponding trellis is split
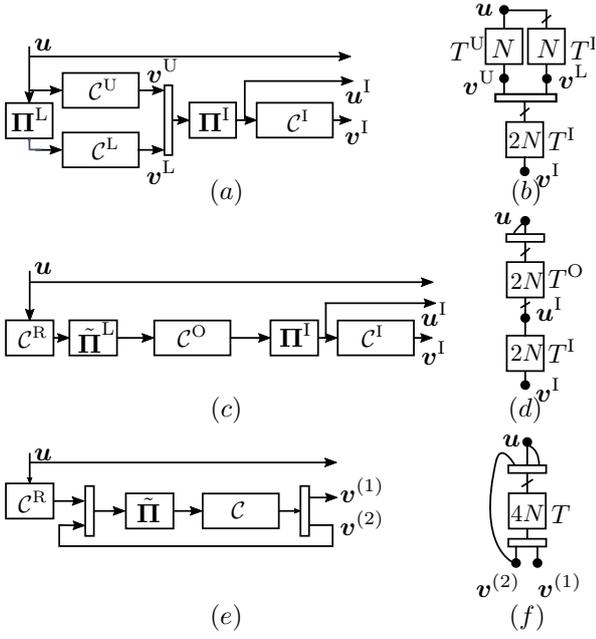
Fig. 3. (a) Block diagram of a HCC, (b) Compact graph a HCC, (c) Block diagram of a self-concatenated HCC (step 1), (d) Compact graph of a self-concatenated HCC (step 1), (e) Block diagram of a self-concatenated HCC (step 2), and (f) Compact graph of a self-concatenated HCC (step 2)

into two parts. The information sequence $\boldsymbol{u}$ is connected to the first part to produce $\boldsymbol{v}^{(2)}$. Then, a reordered copy of the sequence $(\boldsymbol{u}, \boldsymbol{v}^{(2)})$ is connected to the second part of the trellis to produce $\boldsymbol{v}^{(1)}$. By spatial coupling, the feedback path can be fed forward to the corresponding multiplexer in the next time slots.

C. Hybrid Concatenated Codes

Fig. 3(a) shows the encoder block diagram of an HCC ensemble built from three identical rate-1 component encoders. The corresponding compact graph representation is also shown in Fig. 3(b). The considered HCC ensemble is a serial concatenation of a parallel ensemble with an inner encoder. The information sequence $\boldsymbol{u}$ and a reordered copy of it are fed to two encoders, referred to as upper and lower encoders, and denoted by $\mathcal{C}^{\mathrm{U}}$ and $\mathcal{C}^{\mathrm{L}}$, to produce parity sequences $\boldsymbol{v}^{\mathrm{U}}$ and $\boldsymbol{v}^{\mathrm{L}}$, respectively. Then, $\boldsymbol{v}^{\mathrm{U}}$ and $\boldsymbol{v}^{\mathrm{L}}$ are multiplexed and reordered. The resulting sequence is used as an input to the inner encoder $\mathcal{C}^{\mathrm{I}}$.

The corresponding self-concatenated ensemble can be obtained in two steps. First, as it is shown in Fig. 3(c)(d), the upper and lower trellises can be unified into a single trellis with length $2N$ by the method described for PCCs. Then, the resulting trellis can be connected to the inner trellis using the method described for SCCs.

The self-concatenated ensemble for HCCs is shown in Fig. 3(e)(f). In this ensemble, the overall trellis has length $4N$. The parity sequence $\boldsymbol{v}$ is divided into two equal-size sequences $\boldsymbol{v}^{(1)}$ and $\boldsymbol{v}^{(2)}$ of length $2N$. Then, $\boldsymbol{v}^{(2)}$ is multiplexed with sequence $(\boldsymbol{u}, \boldsymbol{u})$ generated by a repetition encoder $\mathcal{C}^{\mathrm{R}}$. The resulting sequence is reordered and used as an input to a rate-
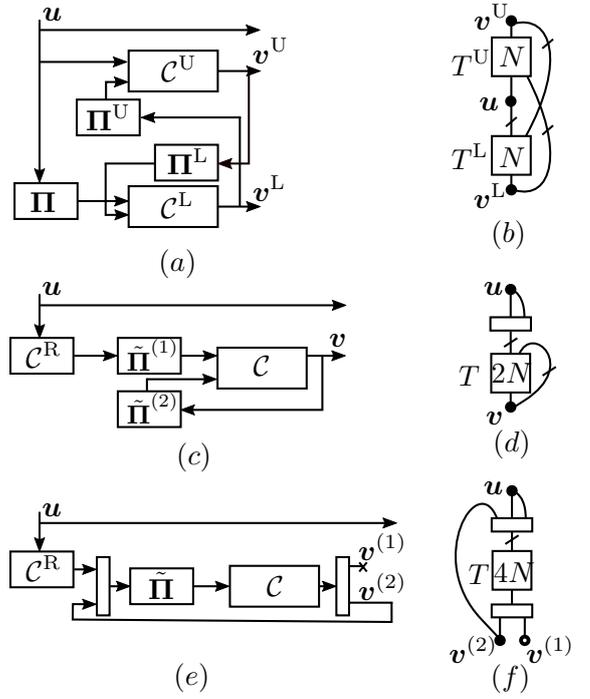

Fig. 4. (a) Block diagram of a BCC, (b) Compact graph a BCC, (c) Block diagram of a self-concatenated BCC (step 1), (d) Compact graph of a self-concatenated BCC (step 1), (e) Block diagram of a self-concatenated BCC (step 2), and (f) Compact graph of a self-concatenated BCC (step 2)

1 convolutional encoder $\mathcal{C}$. Note that $\boldsymbol{v}^{(1)}$ and $\boldsymbol{v}^{(2)}$ correspond to $(\boldsymbol{v}^{\mathrm{U}}, \boldsymbol{v}^{\mathrm{L}})$ and $\boldsymbol{v}^{\mathrm{I}}$ of the original ensemble. By selecting

$$\tilde{\boldsymbol{\Pi}} = \begin{bmatrix} \boldsymbol{I} & 0 & 0 \\ 0 & \boldsymbol{\Pi}^{\mathrm{L}} & 0 \\ 0 & 0 & \boldsymbol{\Pi}^{\mathrm{I}} \end{bmatrix}, \tag{2}$$

the self-concatenated ensemble is equivalent to the original ensemble.

D. Braided Convolutional Codes

Fig. 4(a) shows the encoder block diagram of a BCC ensemble. This ensemble is similar to the PCC ensemble but the BCC ensemble is built of two rate-2 [2] convolutional encoders and the parity sequence of each encoder is fed back to the input of the other encoder. Similarly to PCCs, the component encoders are denoted by $\mathcal{C}^{\mathrm{U}}$ and $\mathcal{C}^{\mathrm{L}}$ and called upper and lower encoders, respectively. The compact graph representation of the ensemble is shown in Fig. 4(b). The information sequence $\boldsymbol{u}$ and a reordered copy of $\boldsymbol{v}^{\mathrm{L}}$ are used as the first and second input of $\mathcal{C}^{\mathrm{U}}$, respectively to produce the parity sequence $\boldsymbol{v}^{\mathrm{U}}$. Likewise, a reordered copy of $\boldsymbol{u}$ and a reordered copy of $\boldsymbol{v}^{\mathrm{U}}$ are used as the first and second input of $\mathcal{C}^{\mathrm{L}}$ respectively, to produce $\boldsymbol{v}^{\mathrm{L}}$.

In order to obtain the self-concatenated ensemble for BCCs, we can use the method described for PCCs. Fig. 4(c) and (d) show the encoder block diagram and compact graph representation of the corresponding self-concatenated ensemble.

[2]The component encoders are rate-2/3 convolutional encoders with all systematic bits punctured.
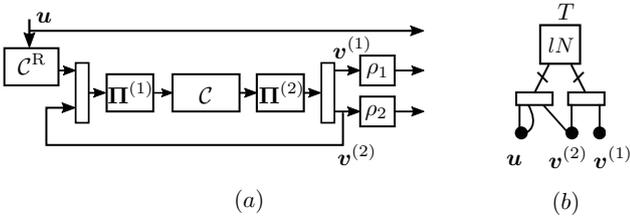
Fig. 5. (a) Block diagram of encoder (b) Compact graph representation of the unified ensemble

TABLE I
PARAMETERS OF PCCS, SCCS, BCCS AND HCCS

Ensemble	R	ρ_1	ρ_2	l	l_1	l_2	l_2/l_1
PCC	1/3	1	-	2	2	0	0
SCC	1/4	1	1	3	2	1	1/2
BCC	1/3	0	1	4	2	2	1
HCC	1/5	1	1	4	2	2	1

The two component encoders, with N trellis sections, in the original ensembles are combined to a component encoder with length-$2N$ trellis. The sequence $\boldsymbol{u}$ is connected to a repetition encoder $\mathcal{C}^{\mathrm{R}}$ to produce the sequence $(\boldsymbol{u}, \boldsymbol{u})$. The resulting sequence is reordered by permutation $\tilde{\boldsymbol{\Pi}}^{(1)}$ and used as the first input of a rate-2 convolutional encoder $\mathcal{C}$. The second input of the encoder is a copy of the parity sequence $\boldsymbol{v}$ that is reordered by the permutation $\tilde{\boldsymbol{\Pi}}^{(2)}$. By selecting the permutations as

$$\tilde{\boldsymbol{\Pi}}^{(1)} = \begin{bmatrix} \boldsymbol{I} & 0 \\ 0 & \boldsymbol{\Pi} \end{bmatrix},$$

$$\tilde{\boldsymbol{\Pi}}^{(2)} = \begin{bmatrix} 0 & \boldsymbol{\Pi}^{\mathrm{U}} \\ \boldsymbol{\Pi}^{\mathrm{L}} & 0 \end{bmatrix},$$

the encoders in Fig. 4(a) and (c) are equivalent.

The encoder ensembles in Fig. 4(a) and (c) are not causal. For the original ensemble of BCCs introduced in [9], this problem was solved by spatial coupling. In the block-wise BCC ensemble, the parity sequences $\boldsymbol{v}^{\mathrm{U}}$ and $\boldsymbol{v}^{\mathrm{L}}$, (or $\boldsymbol{v}$ in the self-concatenated ensemble), are connected to the inputs of the corresponding encoders after passing delay blocks. This makes the encoder causal.

It is also possible to find a self-concatenated ensemble for BCCs that is very close to those for the other TC classes. We can replace the rate-2 component encoder in the self-concatenated ensemble by a rate-1 encoder for that half of its output sequence is punctured. The encoder block diagram for this ensemble is shown in Fig. 4(e) and its corresponding compact graph representation is depicted in Fig. 4(f). As it is shown in these figures, the parity sequence is divided into two parts $\boldsymbol{v}^{(1)}$ and $\boldsymbol{v}^{(2)}$. Sequence $\boldsymbol{v}^{(1)}$ is fully punctured and $\boldsymbol{v}^{(2)}$ is multiplexed with the sequence $(\boldsymbol{u}, \boldsymbol{u})$ at the output of the repetition encoder $\mathcal{C}^{\mathrm{R}}$. The resulting sequence is reordered and fed to the convolutional encoder $\mathcal{C}$ with corresponding trellis of length $4N$.

III. THE UNIFIED ENSEMBLE

Comparing the obtained self-concatenated ensembles introduced in the previous section for the considered classes of TCs, we note similarities between them (see Fig. 1-Fig. 4). Based on these similarities, we develop a unified ensemble. The block diagram of the encoder and the compact graph representation of this ensemble is shown in Fig. 5. In this ensemble, the component convolutional encoder is a rate-1 convolutional encoder. The information sequence $\boldsymbol{u}$, of length N, is connected to a repetition encoder $\mathcal{C}^{\mathrm{R}}$ with repetition factor 2 to produce the sequence $\tilde{\boldsymbol{u}} = (\boldsymbol{u}, \boldsymbol{u})$. This sequence

is multiplexed with $\boldsymbol{v}^{(2)}$ which is a part of the overall parity sequence $\boldsymbol{v}$. Then, the resulting sequence is reordered by the permutation $\boldsymbol{\Pi}^{(1)}$, of size lN, and fed to the convolutional encoder $\mathcal{C}$. Parameter l is a design parameter which can be tuned to yield a specific TC class (PCC, SCC, BCC, or HCC). The values of l for the different classes of TCs are presented in Table I.

The sequence $\boldsymbol{v}$ is divided into two sequences $\boldsymbol{v}^{(1)}$ and $\boldsymbol{v}^{(2)}$ of length l_1N and l_2N, respectively. In order to guarantee random division of $\boldsymbol{v}$, first, this sequence is reordered by a permutation. Then, it is divided into the two sequences $\boldsymbol{v}^{(1)}$ and $\boldsymbol{v}^{(2)}$. The values of l_1 and l_2 for each class of TCs, are provided in Table I. Note that $l = l_1 + l_2$.

Parameters ρ_1 and ρ_2 in Table I are the permeability rates for sequences $\boldsymbol{v}^{(1)}$ and $\boldsymbol{v}^{(2)}$, respectively, giving the fraction of surviving bits of $\boldsymbol{v}^{(1)}$ and $\boldsymbol{v}^{(2)}$ after puncturing. For example, to obtain the equivalent ensemble for BCCs, $\rho_1 = 0$, as $\boldsymbol{v}^{(1)}$ is fully punctured. We remark that by selecting ρ_1 and ρ_2 properly, the obtained ensemble covers a family of rate compatible TCs.

An inspection of Fig. 5(b) reveals that the compact graph representation of the unified ensemble is very close to the protograph of an irregular LDPC code. The factor node is a trellis with length lN, where its degree is fixed to two. The variable nodes are classified into three groups as follows: $\boldsymbol{u}$ is an information variable node with degree 2, $\boldsymbol{v}^{(1)}$ is a parity variable node with degree 1, and $\boldsymbol{v}^{(2)}$ is a parity variable node with degree 2. The length of theses variable nodes are not equal. Considering length N for $\boldsymbol{u}$, $\boldsymbol{v}^{(1)}$ and $\boldsymbol{v}^{(2)}$ have length l_1N and $l_2N = 2N$, respectively. According to Table I, different ensembles of TCs can be obtained by changing the ratio l_2/l_1 which is the proportion of degree-1 and degree-2 parity variable nodes. This is very close to defining the variable node degree distribution for LDPC codes.

A. Density Evolution

Considering transmission over a BEC with channel parameter ε, we can analyze the asymptotic behavior of the unified ensemble by tracking the evolution of the erasure probability with the number of iterations of the decoding procedure. This evolution can be shown as a set of equations called DE equations, and for the BEC, it is possible to derive an exact expression for them. In the ith iteration, the extrinsic erasure probabilities from factor node T toward variable nodes are denoted by $x_1^{(i)}$ and $x_2^{(i)}$, respectively, for the first and second

TABLE II
THRESHOLDS OF PCCs, SCCs, BCCs AND HCCs

Ensemble	R	ε_{BP}	ε_{BP}^{U}	ε_{MAP}	ε_{MAP}^{U}
PCC	1/3	0.6428	0.6428	0.6553	0.6552
SCC	1/4	0.6895	0.6863	0.7481	0.7482
BCC	1/3	0.5541	0.5603	0.6653	0.6646
HCC	1/5	0.7261	0.6997	0.7995	0.7994

edge connected to it. Then, the DE equations can be written as,

$$x_1^{(i+1)} = f_1(y_1^{(i)}, y_2^{(i)}), \tag{3}$$
$$x_2^{(i+1)} = f_2(y_1^{(i)}, y_2^{(i)}) \tag{4}$$

where

$$y_1^{(i)} = \frac{2\varepsilon x_1^{(i)} + l_2(\rho_2 \varepsilon x_2^{(i)} + (1-\rho_2) x_2^{(i)})}{2 + l_2}, \tag{5}$$

$$y_2^{(i)} = \frac{l_2(\rho_2 \varepsilon x_1^{(i)} + (1-\rho_2) x_1^{(i)}) + l_1(\rho_1 \varepsilon + (1-\rho_1))}{2 + l_2}. \tag{6}$$

Here, f_1 and f_2 are the transfer functions of T for the systematic and parity bits, respectively. The a-posteriori erasure probability of bits in the information sequence $\boldsymbol{u}$ at the ith iteration is,

$$p_a^{(i)} = \varepsilon \cdot (x_1^{(i)})^2.$$

The decoding thresholds obtained by DE are reported in Table II. The table shows the BP threshold ε_{BP} and the MAP threshold ε_{MAP} of the original ensembles. To obtain ε_{BP} and ε_{MAP}, we used the corresponding DE equations and the area theorem, respectively. We also report in the table the BP threshold ε_{BP}^{U} and the MAP threshold ε_{MAP}^{U} of the proposed equivalent ensembles. From the results in the table, it can be seen that in the PCC case there is a good match between the thresholds of the original ensemble and the corresponding values of the equivalent ensemble. For the other cases, ε_{BP} and ε_{BP}^{U} are similar. However, there is a small gap between these thresholds. This gap can be explained as follows. In the DE analysis of the unified ensemble, we consider that the permutations are chosen randomly. Therefore, in equations (5) and (6), we average over all possible cases. However, the original TC ensembles are more structured and, in consequence, except for BCCs, ε_{BP} is larger than ε_{BP}^{U}. For BCCs, replacing the rate-2 component encoder by a rate-1 component encoder with puncturing, is another reason for observing the gap between ε_{BP} and ε_{BP}^{U}. We also computed the thresholds for the self-concatenated ensemble in Fig. 4(c). The obtained BP and MAP thresholds for this ensemble are identical to those of the original BCC ensemble.

The results in Table II demonstrate that thresholds similar to those of the original TC classes can be obtained by changing the design parameters in the unified ensemble.

IV. CONCLUSIONS

In this paper, a unified ensemble for various classes of TCs is introduced. This ensemble is based on a single trellis with self-concatenation. We introduced two elementary steps to find the self-concatenated equivalent of PCCs and SCCs. We also used these elementary steps to find the self-concatenated HCCs and BCCs. These elementary steps can also be applied to more general concatenations.

Then, by considering the similarities between the self-concatenated ensembles for different TC classes, we found a unified ensemble. By changing the proportion of degree-1, and degree-2 variable nodes in the graph or puncturing a part of the parity sequence, the original TC ensembles can be obtained. The compact graph representation of our ensemble establishes a bridge between TCs and protograph based LDPC codes, where the check nodes are replaced by trellis constraints.

We believe that the unified ensemble may help in better understanding the connections between concatenated code ensembles and LDPC code ensembles.

REFERENCES

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

[2] S. Kudekar, T.J. Richardson, and R.L. Urbanke, "Threshold saturation via spatial coupling: Why convolutional LDPC ensembles perform so well over the BEC," *IEEE Trans. Inf. Theory*, vol. 57, no. 2, pp. 803–834, Feb. 2011.

[3] A. Yedla, Yung-Yih Jian, P.S. Nguyen, and H.D. Pfister, "A simple proof of Maxwell saturation for coupled scalar recursions," *IEEE Trans. Inf. Theory*, vol. 60, no. 11, pp. 6943–6965, Nov. 2014.

[4] S. Moloudi, M. Lentmaier, and A. Graell i Amat, "Spatially coupled turbo codes," in *Proc. 8th Int. Symp. on Turbo Codes and Iterative Inform. Process. (ISTC)*, Bremen, Germany, 2014.

[5] M. Lentmaier, S. Moloudi, and A. Graell i Amat, "Braided convolutional codes - a class of spatially coupled turbo-like codes," in *Proc. Int. Conference on Signal Process. and Commun. (SPCOM)*, Bangalore, India, 2014.

[6] S. Moloudi, M. Lentmaier, and A. Graell i Amat, "Spatially coupled turbo-liked codes," submitted to *IEEE Trans. Inf. Theory*, available online at https://arxiv.org/pdf/1604.07315v1.pdf, Dec. 2015.

[7] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: turbo-codes (1)," in *Proc. IEEE Int. Conference on Commun. (ICC)*, Geneva, Switzerland, 1993.

[8] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara, "Serial concatenation of interleaved codes: performance analysis, design, and iterative decoding," *IEEE Trans. Inf. Theory*, vol. 44, no. 3, pp. 909–926, May 1998.

[9] W. Zhang, M. Lentmaier, K.Sh. Zigangirov, and D.J. Costello, Jr., "Braided convolutional codes: a new class of turbo-like codes," *IEEE Trans. Inf. Theory*, vol. 56, no. 1, pp. 316–331, Jan. 2010.

[10] D. Divsalar and F. Pollara, "Hybrid concatenated codes and iterative decoding," in *Proceedings of IEEE Int. Symp. Inform. Theory*, Ulm, Germany, Jun 1997.

[11] C. Koller, A. Graell i Amat, J. Kliewer, F. Vatta, and D. J. Costello, "Hybrid concatenated codes with asymptotically good distance growth," in *Proc. 5th Inte. Symp. on Turbo Codes and Related Topics*, Lausanne, Switzerland, 2008.

[12] A. Graell i Amat, L. K. Rasmussen, and F. Brannstrom, "Unifying analysis and design of rate-compatible concatenated codes," *IEEE Trans. on Comm.*, vol. 59, no. 2, pp. 343–351, February 2011.

[13] D. Divsalar, J. Jin, and R.J. McEliece, "Coding theorems for turbo-like codes," *Proc. 36th Annu. Allerton Conf. Communication, Control, and Computing*, Sep. 1998.