Spatially Coupled Turbo-Like Codes

Moloudi, Saeedeh

2018

Spatially Coupled Turbo-Like Codes

Saeedeh Moloudi

Lund 2018

Series of licentiate and doctoral theses
No. 115
ISSN 1654-790X
ISBN 978-91-7753-647-5 (print)
ISBN 978-91-7753-648-2 (pdf)

Cover design by Saeedeh Moloudi

This thesis is produced by using the LaTeX Documentation System
Printed in Sweden by *Tryckeriet i E-huset*, Lund.

Department of Electrical and Information Technology
Lund University, Box 118, SE-221 00 Lund, Sweden

You are not a drop in the ocean
You are the entire ocean in a drop

Rumi (1207–1273)

Popular Science

Electronic devices have become indispensable parts of our lives. Every day, we send and receive lots of information, using these devices. Step by step, our societies are getting closer to what some theorists call a network society. In such a society, almost all humans and machines are connected to each other through extensive communication networks. Failure in these networks can lead to severe consequences in almost all societal functions. Reliable communication is one of the key characteristics of a well-functioning network. Therefore, it is important to ensure the reliability of the received data. In any communication system, there are at least three main elements: the sender, the receiver, and a physical medium (channel) that connects the sender to the receiver. Often, the data is corrupted or distorted along the channel. Sometimes, the distortion is so big that the receiver may not be able to recover the original correct data. In the late 1940s, an American mathematician, Claude Shannon, showed that the reliability could be enhanced by a technique called *channel coding*. To describe this technique in a simple way, let us use an example.

Consider a typical conversation between two persons. The person who is talking is the sender, the one who is listening is the receiver, and the air is the medium. Now imagine that this conversation is happening in a crowded and noisy room where it is hard for the listener to follow the conversation. She may easily miss some parts of the conversation or mishear them. In technical words, the channel of this communication system is noisy and the communication is not reliable. Let's assume that the speaker wants to tell three numbers to his listener: 6, 9, and 10. The room is crowded; therefore, the listener may miss one of the numbers. A simple solution to this problem is to send one extra number, which is the summation of the original three numbers. In this case, the transmitted numbers become 6, 9, 10, and 25. Now, if the listener misses a number, she can calculate it by using the other three numbers. A similar technique is used in digital communication systems.

In digital communications, any data is usually represented by a sequence of zeros and ones. Each of these zeros and ones is called *a bit*, and the whole sequence is referred to as the *information sequence*. To help mitigating the

noise effect that can alter some of the bits, some redundant bits are added to the information sequence. This redundancy is added in a controlled way and is introduced to provide reliable transmission of data over the channel. This process of protecting the message bits from errors is called error control coding.

There are many different classes of codes that can be used. The strength of a codes is measured by its capability in detecting and correcting the errors. This depends on the number of redundant bits, the algorithm that generates these bits and the recovering method that is used at the receiver side. Selection of a suitable code depends on the application and practical trade-offs. However, low latency, low error rate at the output of the decoder, low complexity, and high energy efficiency are generally considered as characteristics of a good code.

It is possible to concatenate two or more codes, to make a stronger code. A very famous class of concatenated codes are the so-called *turbo-like codes* (TCs). The studies on turbo-like codes demonstrate that these codes perform close to the theoretical limits. Therefore, these codes have been used in many communication standards. Low-density parity-check (LDPC) codes are another powerful class of codes. In the late 90s, it was shown that the performance of these codes could improve by introducing the concept of memory in their encoding procedure. Later, this technique was called *spatial coupling* and attracted the attention of many experts in the field of coding theory. However, spatial coupling is a general technique and is not limited to LDPC codes. In this thesis, we have proposed spatially coupled TCs and investigated the impact of spatial coupling on this class of codes. Our results confirm considerable improvements in the performance of TCs by spatial coupling.

The proposed codes in this thesis can contribute to finding new trade-offs between latency and error rate in many communication applications.

Abstract

The focus of this thesis is on proposing and analyzing a powerful class of codes on graphs—with trellis constraints—that can simultaneously approach capacity and achieve very low error floor. In particular, we propose the concept of spatial coupling for turbo-like code (SC-TC) ensembles and investigate the impact of coupling on the performance of these codes. The main elements of this study can be summarized by the following four major topics.

First, we considered the spatial coupling of parallel concatenated codes (PCCs), serially concatenated codes (SCCs), and hybrid concatenated codes (HCCs). We also proposed two extensions of braided convolutional codes (BCCs) to higher coupling memories.

Second, we investigated the impact of coupling on the asymptotic behavior of the proposed ensembles in term of the decoding thresholds. For that, we derived the exact density evolution (DE) equations of the proposed SC-TC ensembles over the binary erasure channel. Using the DE equations, we found the thresholds of the coupled and uncoupled ensembles under belief propagation (BP) decoding for a wide range of rates. We also computed the maximum a-posteriori (MAP) thresholds of the underlying uncoupled ensembles. Our numerical results confirm that TCs have excellent MAP thresholds, and for a large enough coupling memory, the BP threshold of an SC-TC ensemble improves to the MAP threshold of the underlying TC ensemble. This phenomenon is called threshold saturation and we proved its occurrence for SC-TCs by use of a proof technique based on the potential function of the ensembles.

Third, we investigated and discussed the performance of SC-TCs in the finite length regime. We proved that under certain conditions the minimum distance of an SC-TCs is either larger or equal to that of its underlying uncoupled ensemble. Based on this fact, we performed a weight enumerator (WE) analysis for the underlying uncoupled ensembles to investigate the error floor performance of the SC-TC ensembles. We computed bounds on the error rate performance and minimum distance of the TC ensembles. These bounds indicate very low error floor for SCC, HCC, and BCC ensembles, and show that for HCC, and BCC ensembles, the minimum distance grows linearly with the

input block length. The results from the DE and WE analysis demonstrate that the performance of TCs benefits from spatial coupling in both waterfall and error floor regions. While uncoupled TC ensembles with close-to-capacity performance exhibit a high error floor, our results show that SC-TCs can simultaneously approach capacity and achieve very low error floor.

Fourth, we proposed a unified ensemble of TCs that includes all the considered TC classes. We showed 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. This unified ensemble not only helps us to understand the connections and trade-offs between the TC ensembles but also can be considered as a bridge between TCs and generalized low-density parity check codes.

Preface

This doctoral thesis is comprised of two parts. The first part gives an overview of the research field in which I have been working during my Ph.D. studies and a brief summary of my contribution in the research field. The second part is composed of six papers that constitute my main scientific work, including:

Paper I S. Moloudi, M. Lentmaier, "Density evolution analysis of braided convolutional codes on the erasure channel," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Honolulu, HI, USA, 2014.

Paper II S. Moloudi, M. Lentmaier, and A. Graell i Amat, "Spatially coupled turbo codes," in *Proc. Int. Symp. on Turbo Codes and Iterative Inf. Processing (ISTC)*, Bremen, Germany, 2014.

Paper III S. Moloudi, M. Lentmaier, and A. Graell i Amat, "Spatially coupled turbo-like codes," in *IEEE Trans. Inf. Theory*, vol. 63, no. 10, pp. 6199-6215, Oct. 2017.

Paper IV S. Moloudi, M. Lentmaier, and A. Graell i Amat, "Spatially coupled hybrid concatenated codes," in *Proc. Int. ITG Conf. Systems, Commun. and Coding (SCC)*, Hamburg, Germany, 2017.

Paper V S. Moloudi, M. Lentmaier, and A. Graell i Amat, "Spatially coupled turbo-like codes: a new trade-off between waterfall and error floor," submitted to *IEEE Trans. Commun.*.

Paper VI S. Moloudi, M. Lentmaier, and A. Graell i Amat, "A unified ensemble of concatenated convolutional codes," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Aachen, Germany, 2017.

In all of the included papers, I have been the main author. Under supervision of my supervisors, I have carried out the main part of the work, including: analysis, numerical computations, simulations, and writing of the papers. During my Ph.D. studies, I have also contributed to the following publications that are not included in this thesis.

- S. Moloudi, M. Lentmaier, and A. Graell i Amat, "Threshold saturation for spatially coupled turbo-like codes over the binary erasure channel," in *Proc. IEEE Inf. Theory Workshop - Fall (ITW)*, Jeju, South Korea, 2015.
- S. Moloudi, M. Lentmaier, and A. Graell i Amat, "Finite length weight enumerator analysis of braided convolutional codes," in *Proc. Int. Symp. on Inf. Theory and Its App. (ISITA)*, Monterey, CA, USA, 2016.
- M. Lentmaier, S. Moloudi, and A. Graell i Amat, "Braided convolutional codes - a class of spatially coupled turbo-like codes," in *Proc. Int. Conf. on Signal Process. and Commun. (SPCOM)*, Bangalore, India, 2014.
- A. Graell i Amat, S. Moloudi, and M. Lentmaier, "Spatially coupled turbo codes: Principles and finite length performance," in *Proc. Int. Symp. on Wireless Commun. Syst. (ISWCS)*, Barcelona, Spain, 2014.
- M. U. Farooq, S. Moloudi, and M. Lentmaier, "Thresholds of Braided Convolutional Codes on the AWGN Channel," accepted to *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, 2018.

Acknowledgements

Along my Ph.D. studies, I have met many dear people who have added meaning and joy to this journey. Although I cannot name all of them here, I would like to express my sincere thanks to each.

I am grateful to my insightful supervisor and teacher, Michael Lentmaier, without his generous and professional supports this work would not have been possible. I am also highly indebted to Alexandre Graell i Amat who acted informally as my other supervisor, for all his supports and efforts to improve the quality of my work. I realize that almost every strong aspect of my thesis is a reflection of the guidance of my supervisors, but I myself stand for any of its shortcomings.

I would like to thank Ove Edfors, for his supports and care during my Ph.D. studies. I would also like to express my gratitude to all my colleagues at EIT, especially my officemate, Xuhong Li, who is an amazing friend and colleague.

I cannot thank my friends in Sweden enough. Their presence have kept my heart warm during the long winters and has been the most joyful gift during the beautiful Swedish summers.

I would like to express my deepest gratitude to my family for their kindness and endless supports. This thesis is dedicated to them; baba Ali, maman Mehri, and my lovely sisters, Sara, Azita, Fatemeh and Nazanin.

پدر، مادر و خواهرهای عزیزم، این رساله را به پاس همهٔ گذشت‌ها و محبت‌های بی‌دریغ‌تان به شما تقدیم می‌کنم. از این که همیشه در تصمیم‌های دشوار حمایتم کرده‌اید بی‌نهایت سپاسگزارم. بودن‌تان بزرگترین دلگرمی من است.

Finally, I would like to thank my dear husband whose love, patience and wisdom have added light and new colors to my life. Roozbeh! If this thesis represents any success, it is undoubtedly due to your unconditional supports and encouragements. Thank you!

Saeedeh

List of Acronyms and Abbreviations

APP	A-Posteriori Probability
AWGN	Additive White Gaussian Noise
BCC	Braided Convolutional Code
BEC	Binary Erasure Channel
BER	Bit Error Rate
BP	Belief Propagation
CC	Convolutional Code
CN	Check Node
DE	Density Evolution
EXIT	EXtrinsic Information Transfer
GLDPC	Generalize Low-Density Parity-Check
HCC	Hybrid Concatenated Code
IO-WEF	Input-Output Weight Enumerator Function
IP-WEF	Input-Parity Weight Enumerator Function
LDPC	Low-Density Parity-Check
LDPC-CC	Low-Density Parity-Check Convolutional Code
LLR	Log-Likelihood Ratio
MAP	Maximum A-Posteriori
ML	Maximum Likelihood
PCC	Parallel Concatenated Code
PC	Product Code
RA	Repeat-Accumulate Code
SBC	Sparsely Braided Code
SCC	Serially Concatenated Code
SC-HCC	Spatially Coupled Hybrid Concatenated Code
SC-LDPC	Spatially Coupled Low-Density Parity-Check
SC-PCC	Spatially Coupled Parallel Concatenated Code
SC-SCC	Spatially Coupled Serially Concatenated Code

SC-TC	Spatially Coupled Turbo-like Code
SISO	Soft-Input Soft-Output
SNR	Signal-to-Noise Ratio
TBC	Tightly Braided Code
TC	Turbo-like Code
VN	Variable Node
WE	Weight Enumerator
WEF	Weight Enumerator Function
URP	Uniformly Random Permutation

Contents

Part I

Overview of Research Field

Chapter 1

Introduction

Communication is an integral part of all human societies. The history is full of innovations related to how we communicate with each other, ranging from natural languages, to the invention of writing and more recent developments such as the printing press, the radio, the Internet and several other simple tools or sophisticated communication systems. Nowadays, the demand for communication is more than any other time in history. We not only communicate with each other on a global scale, but also increasingly communicate with our devices. In particular, data communication has become an inseparable part of our lives, witnessed by its outstanding expansion in the last decades. In all data communications, guaranteeing the reliability of the received data is among the most important concerns.

Any communication system consists of three main elements: a sender, a receiver, and a physical medium (channel) through which the sender transmits the data to the receiver. A typical channel is not ideal and its impairments, usually due to noise, cause uncertainty at the receiver side on the transmitted data. In digital data communication systems, a technique called *channel coding* is developed to increase the reliability of the received data.

To explain channel coding [1], [2], we consider a communication system in which the sender sends the information sequence $\boldsymbol{u}$ with K bits through a noisy channel (see Figure 1.1). At the transmitter side, the sequence $\boldsymbol{u}$ is mapped to a sequence, $\boldsymbol{v}$, with N bits, $N > K$, through the *encoding* procedure. The corresponding code $\mathcal{C}$ is the set of all possible sequences $\boldsymbol{v}$. Encoding $\boldsymbol{u}$ to a more extended length sequence $\boldsymbol{v}$ can be seen as adding redundancy to the information sequence in a controlled way. The amount of added redundancy can be represented by the ratio $R = K/N$, called *code rate*. As it is shown in Figure 1.1, the encoded sequence $\boldsymbol{v}$ is transmitted over the noisy channel, and the sequence $\boldsymbol{r}$ is received. At the receiver side, the sequence $\boldsymbol{r}$ is used

to recover the original information sequence, through the *decoding* procedure. The transmitted redundant bits help to recover the K original bits with higher certainty.

Figure 1.1: A communication system.

The idea of channel coding was first introduced by Shannon in his landmark paper [3] published in 1948. In this article, Shannon showed that, by proper encoding and decoding procedures, the transmission can be error-free over a noisy channel as long as K and N are large numbers, and the transmission rate is less than a fundamental limit called *channel capacity*. Since then, there has been a lot of research on finding code schemes that approach the Shannon limit [4]. In practice, the limitations on latency and computational complexity of the encoding and decoding procedures are also crucial criteria for many applications. Therefore, it may be more correct to say that, finding the right trade-offs between close-to-capacity performance, low latency and low complexity has been the primary goal of many of these studies. The invention of turbo codes —parallel concatenated codes (PCCs)— [5], [6] and rebirth of low-density parity-check (LDPC) codes [7], [8] have eventually been an important step toward this goal.

The conventional turbo codes or PCCs are built by concatenating two *convolutional codes* (CCs)[9] which are interconnected with permutations. The very good performance of PCCs has been the motivation for proposing a more general class of concatenated CCs called *turbo-like codes* (TCs) [10].

Nowadays, both TCs and LDPC codes are adopted in many communication standards. It has been shown that these classes of codes can perform close to the Shannon limit, for large block lengths. However, finding code schemes for short and moderate block lengths that can perform close to capacity is still a challenge due to the following issues.

First, even by using the optimal decoder, after a certain signal-to-noise ratio (SNR) the so-called *bit error rate* (BER) performance of these codes does not improve significantly by increasing SNR, and the error rate curve gets flat. This phenomenon—called *error floor*—is getting more pronounced for short and moderate block lengths.

Second, the efficient iterative decoders proposed for TCs and LDPCs are suboptimal, and there is a gap between their performance and theoretical limits. Although this gap may look acceptably small for large N, it increases as N decreases.

In [11], [12], [13], the authors have shown that by introducing memory in the encoding of LDPC codes and interconnecting the graphs of LDPC codes in

different time slots, the asymptotic performance of the resulting codes—called LDPC convolutional codes (LDPC-CCs)—improves under iterative decoding. Later, in [14], [15], [16], [17], it was proved that LDPC-CCs—known also as spatially coupled LDPC (SC-LDPC) codes—can asymptotically achieve the performance of the optimal maximum a-priori (MAP) decoder of the underlying LDPC codes with efficient iterative decoders. Since then, SC-LDPC codes have received a great deal of attention in the coding and information theory community. Although SC-LDPC codes show outstanding performance for very long block lengths, their performance for short to moderate block lengths is rather poor.

Spatial coupling is a general concept that can be applied to other codes such as TCs. TCs and LDPC codes share a lot of properties; both are classes of codes on graphs and can be constructed or analyzed by similar methods. However, considering the graph representations of these codes, it can be seen that LDPC codes have a lot of very simple components, connected by many edges; on the contrary, TCs have a small number of strong components, CCs, connected by a small number of edges. Having simple component codes can limit the error correction capability of LDPC codes in the short-to-moderate block length regime, despite their excellent performance for very long block lengths. However, the strong component codes of TCs makes them more robust for short-to-moderate block lengths. All these together make TCs an excellent potential candidate for applying spatial coupling. However, before this thesis, the spatial coupling of TCs had not been studied systematically.

The purpose of this thesis is to propose, design, and investigate the spatially coupled counterparts for different classes of TCs (SC-TCs), and to understand trade-offs of SC-TCs and their connection to SC-LDPC codes and SC generalized LDPC (SC-GLDPC) codes. This thesis includes six papers. In Paper I, the asymptotic limits on the performance of a class of SC-TCs—braided convolutional codes (BCCs) [18]—is investigated based on the density evolution (DE) technique. In Paper II and IV, some coupled ensembles are proposed for different classes of TCs and their asymptotic behavior is investigated by DE. In Paper III, two extensions of BCC are proposed and it is proved that different classes of SC-TCs can achieve the asymptotic performance of the underlying TCs by iterative decoding. In Paper V, the performance of SC-TCs for finite block lengths is investigated. Finally, a unified ensemble for different TCs is proposed. This ensemble can be used to find connections between different classes of SC-TCs and SC-LDPC codes.

1.1 Thesis Outline

This thesis is written in the paper collection format and consists of two main parts. The first part provides an introduction of the research field, while the

second part contains the research contributions in form of a selection of papers. The structure of the first part of this thesis is as follows. In Chapter 2, a few fundamental concepts that are relevant to this thesis, are presented. In Chapter 3, I discuss different classes of codes on graphs including LDPC codes, TCs, product codes (PCs) [9], and the corresponding iterative decoding algorithms. In Chapter 4, I explain a few tools for analyzing the performance of the codes on graphs. These tools are used in the included articles. This chapter includes threshold analysis methods and weight enumerator analysis. Chapter 5 is focusing on spatially coupled codes by describing SC-LDPC codes and braided codes. Finally, in Chapter 6, I provide a summary of the included papers, discuss the main conclusions of this thesis, and also suggest a few related areas for future research.

Chapter 2

Basic Concepts

In this chapter, two mathematical models for communication channels, and two main categories of codes are discussed.

2.1 Channels

The channel is the medium that connects the sender to the receiver. Depending on the characteristics of this medium and its effect on the transmitted bits, it can be mathematically modeled.

2.1.1 Binary Erasure Channel

The binary erasure channel (BEC) is one of the simplest mathematical models for a communication channel. Figure 2.1 shows the model for this channel. The input to this channel is a binary and random variable $x \in \mathcal{X} = \{-1, +1\}$. This channel does either transmit x perfectly with probability $1 - \varepsilon$ or erases x with probability ε. Therefore, the output of this channel is a random variable $y \in \mathcal{Y} = \{-1, +1, e\}$, where e denotes the erased symbol. The parameter ε is called *channel erasure probability*, and the capacity of the channel is directly related to it by $C = 1 - \varepsilon$.

The simplicity and mathematical appealing nature of the BEC makes it attractive for studying and investigating theories and statements in the coding theory field, where many of these theories and statements can be generalize to the other channels.

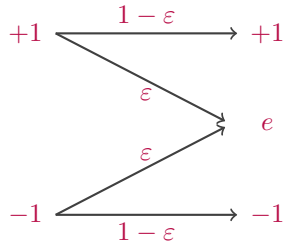


Figure 2.1: BEC with erasure probability ε.

2.1.2 Additive White Gaussian Noise Channel

The additive white Gaussian noise (AWGN) channel is widely used to model many communication mediums. The relation between input and output of this channel can simply described as

$$y = x + n,$$

where x and y are input and output of the channel, respectively. The parameter n denotes the additive noise and it is a random variable with Gaussian distribution $\mathcal{N}(0, \sigma^2)$ where its probability density distribution is

$$p(n) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-n^2/2\sigma^2}. \tag{2.1}$$

2.2 Block Codes

The term block code refers to any coding scheme that maps a block of input bits, $\boldsymbol{u} = (u_1, u_2, \ldots, u_K)$, into a block of codeword bits, $\boldsymbol{v} = (v_1, v_2, \ldots, v_N)$, where $u_i, v_j \in \mathbb{F}$, $i = 1, \ldots, K$, $j = 1, \ldots, N$, and $\mathbb{F}$ is an arbitrary field. In the case of binary codes, the field is $\mathbb{F}_2$, i.e., $u_i, v_j \in \{0, 1\}$.

A very important class of block codes is the class of linear block codes, and many well known codes such as *Hamming codes*[19], *Reed-Solomon codes*[20], LDPC codes and TCs belong to this class. A linear block code $\mathcal{C}(N, K)$ is a linear subspace of $\mathbb{F}^N$, i.e., any superposition of codewords is also a codeword. The encoding of linear block codes can be described by means of matrix multiplication as

$$\boldsymbol{v} = \boldsymbol{u}\boldsymbol{G}, \qquad \boldsymbol{G} \in \mathbb{F}^{K \times N},$$

where $\boldsymbol{G}$ is the code's *generator matrix*. A linear code can also be described by its *parity-check matrix*. This matrix shows the constraints on the code symbols as

$$\boldsymbol{v}\boldsymbol{H}^{\mathrm{T}} = \boldsymbol{0}, \qquad \boldsymbol{H} \in \mathbb{F}^{(N-K) \times N}.$$

2.3 Convolutional Codes

Unlike block codes, for CCs [21], the information and codeword sequences are considered to be infinite,

$$\boldsymbol{u} = \boldsymbol{u}_0, \boldsymbol{u}_1, \ldots, \boldsymbol{u}_t, \ldots$$
$$\boldsymbol{v} = \boldsymbol{v}_0, \boldsymbol{v}_1, \ldots, \boldsymbol{v}_t, \ldots \ .$$

At time t, a block of k bits of information, $\boldsymbol{u}_t = (u_t^{(1)}, \ldots, u_t^{(k)})$, is encoded to produce a block of n code symbols, $\boldsymbol{v}_t = (v_t^{(1)}, \ldots, v_t^{(n)})$. However, $\boldsymbol{v}_t$ does not only depend on $\boldsymbol{u}_t$, but also on information blocks, $\boldsymbol{u}_{t'}$, $t' = t - m, \ldots, t$, at previous m time slots, where m is the *memory* of encoding. The ratio $R = k/n$ defines the code rate.

This class of codes are called CCs as the relation between the codeword bits and the input bits can be expressed by

$$v_t^{(j)} = \sum_{i=1}^{k} \sum_{l=0}^{m} u_{t-l}^{(i)} g_{i,l}^{(j)}, \tag{2.2}$$

where $g_{i,l}^{(j)}$ is 1 if $v_t^{(j)}$ depends on $u_{t-l}^{(i)}$, otherwise it is 0. The vector $\boldsymbol{g}_i^{(j)} = \left(g_{i,0}^{(j)}, \ldots, g_{i,m}^{(j)}\right)$ is called *generator vector*.

By introducing delay elements, the encoder block diagram of a CC can be depicted. Figure 2.2 shows the encoder of CC with $R = 1/2$ and $m = 2$. This encoder has one input, two outputs, and two delay blocks.

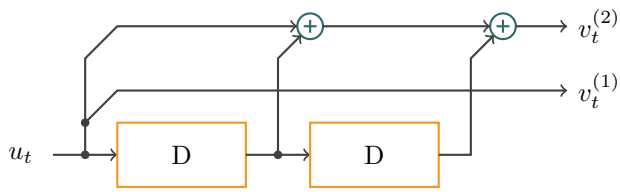


Figure 2.2: An encoder of a CC.

For this encoder

$$v_t^{(1)} = u_t \ , \qquad\qquad v_t^{(2)} = u_t + u_{t-1} + u_{t-2} \ .$$

Therefore, the generator vectors can be written as

$$\boldsymbol{g}_1^{(1)} = (1, 0, 0) \ , \qquad\qquad \boldsymbol{g}_1^{(2)} = (1, 1, 1) \ .$$

It is more convenient to describe the encoding procedure of CCs in transform domain by means of delay operator D. In D-domain, the information and codeword sequences can be written as

$$\begin{aligned}\boldsymbol{u}(D) &= \boldsymbol{u}_0 + \boldsymbol{u}_1 D + \boldsymbol{u}_2 D^2 + \dots \\ \boldsymbol{v}(D) &= \boldsymbol{v}_0 + \boldsymbol{v}_1 D + \boldsymbol{v}_2 D^2 + \dots \ .\end{aligned}$$

Then,

$$\boldsymbol{v}(D) = \boldsymbol{u}(D)\boldsymbol{G},$$

where $\boldsymbol{G}$ is the corresponding generator matrix in form of

$$\boldsymbol{G} = \begin{pmatrix} \boldsymbol{g}_1^{(1)}(D) & \boldsymbol{g}_1^{(2)}(D) & \dots & \boldsymbol{g}_1^{(n)}(D) \\ \boldsymbol{g}_2^{(1)}(D) & \boldsymbol{g}_2^{(2)}(D) & \dots & \boldsymbol{g}_2^{(n)}(D) \\ \dots & \dots & & \dots \\ \boldsymbol{g}_k^{(1)}(D) & \boldsymbol{g}_k^{(2)}(D) & \dots & \boldsymbol{g}_k^{(n)}(D) \end{pmatrix}_{k \times n} . \tag{2.3}$$

For example, the generator matrix corresponding to the CC encoder in Figure 2.2 is $\boldsymbol{G} = (1, 1 + D + D^2)$.

The encoder state of a CCs can be defined by the outputs of the delay elements. It is usually considered that the encoding of CCs starts from the zero state, i.e., all memory blocks contain zero. The encoder is terminated if it is forced to drive back to the zero state by adding proper bits at the end of the input sequence.

It can be seen that the output of the encoder $\boldsymbol{v}_t$ depends on the state of the encoder and the input $\boldsymbol{u}_t$. Therefore, it is possible to assign a state diagram to a CC encoders. Expanding the state diagram along the time instants t results in a trellis representation of the encoder. Efficient decoding algorithms for CCs—such as Viterbi [22] and BCJR [23] algorithm—are developed based on the trellis representation of these codes.

Chapter 3

Codes on Graphs

In [24], Michael Tanner showed that codes such as LDPC codes and PCs can be represented by graphs and be decoded by efficient iterative decoding algorithms in which the component decoders exchanging messages within the iterations [25]. The graph representation of these codes also simplifies their construction and analysis. In this chapter, a few classes of codes on graphs are explained by use of their graph representations.

3.1 Low-Density Parity-Check Codes

An LDPC code can be described either by a parity-check matrix $\boldsymbol{H}$ or by a bipartite graph. For a binary LDPC code, the parity-check matrix, $\boldsymbol{H}$, is a matrix with binary elements and has a low density of 1s. The set of codewords corresponding to an LDPC code defined by a matrix $\boldsymbol{H}$ is the set of all possible N-tuples $\boldsymbol{v}$ that satisfy

$$\boldsymbol{v}\boldsymbol{H}^{\mathrm{T}} = \mathbf{0}.$$

As an example, consider the parity-check matrix,

$$\boldsymbol{H} = \begin{pmatrix} 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \end{pmatrix}. \tag{3.1}$$

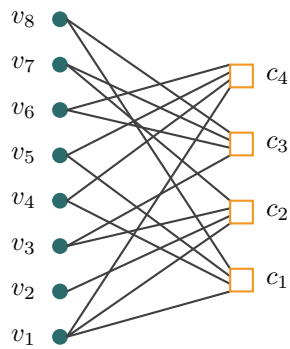


Figure 3.1: Tanner graph of an LDPC code corresponding to the parity-check matrix in (3.1).

The constraint $\boldsymbol{v}\boldsymbol{H}^{\mathrm{T}} = \mathbf{0}$ can be written as a set of parity-check equations,

$$\begin{aligned} c_1: &\quad v_2 + v_4 + v_5 + v_8 = 0 \\ c_2: &\quad v_1 + v_2 + v_3 + v_7 = 0 \\ c_3: &\quad v_3 + v_6 + v_7 + v_8 = 0 \\ c_4: &\quad v_1 + v_4 + v_5 + v_6 = 0, \end{aligned}$$

where v_i shows the ith element of codeword $\boldsymbol{v}$.

The Tanner graph corresponding to the $\boldsymbol{H}$ matrix in (3.1) is shown in Figure 3.1. This graph is a bipartite graph and consists of: 1) a set of *variable nodes* (VNs) corresponding to code symbols v_i, shown by filled circles 2) a set of *check nodes* (CNs) corresponding to the parity-check equations, shown by squares, and 3) a set of *edges* which connect the VNs to the CNs. Each edge connects a VN to CNs corresponding to equations in which the variable is involved.

A Tanner graph is based on the parity-check equations. Therefore, there is a one-to-one connection between an $\boldsymbol{H}$ matrix of an LDPC code and its Tanner graph. The columns and rows in the parity-check matrix correspond to the VNs and the CNs, respectively. If the element of $\boldsymbol{H}$ in the ith row and jth column is one, there is an edge in the corresponding Tanner graph which connects the ith CN to the jth VN.

The number of edges connected to a node defines the degree of that node. Based on the degree distribution, LDPC codes can be classified into two groups: regular and irregular codes. A regular (d_v, d_c) LDPC code is a code for which the degree of all VNs is d_v, and the degree of all CNs is d_c. The performance of the iterative decoder strongly depends on the degree distribution of the VNs and CNs.

3.1.1 Protograph-Based LDPC Codes

LDPC codes can be constructed based on *protographs* [26]. A protograph is a small Tanner graph, in which parallel edges are also allowed. This means there may be several edges that connect a VN to a CN. For a given protograph, the Tanner graph of a corresponding LDPC code can be obtained by the so-called *copy and permutation*—or *lifting*—operation. The resulting Tanner graph is called *derived graph*. The example in Figure 3.2 is provided to explain the steps from a given protograph to a derived graph, in a simple way. In part (a) of this figure, a protograph of an LDPC ensemble is shown. The term ensemble is used to emphasize that this protograph represents a family of LDPC codes. To obtain the derived graph for an LDPC code with $N = 12$, first this protograph is copied three times, $b = 3$ (see Figure 3.2 (b)). The parameter b is called *lifting factor*. These three graphs are still disconnected. Then, as shown in Figure 3.2 (c), the endpoints of the three copies of each edge in the protograph are permuted among the three copies of the corresponding VNs and CNs.

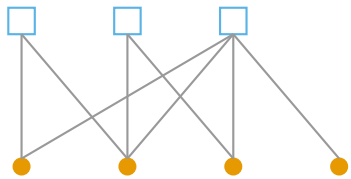
A similar matrix to $\boldsymbol{H}$ can be used to describe the protograph. For the protograph in Figure 3.2 (a) the *base matrix* is,

$$\boldsymbol{B} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}_{b_c \times b_v} . \tag{3.2}$$

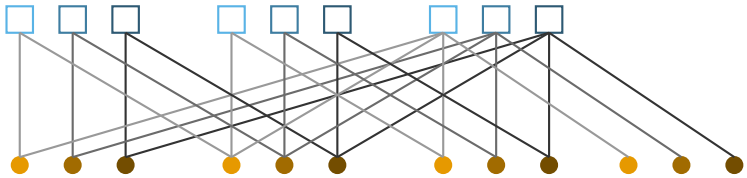
The $\boldsymbol{H}$ matrix corresponding to the derived graph can be obtained by replacing each element of $\boldsymbol{B}$ in the rth row and cth column, $[\boldsymbol{B}]_{r,c}$, by a sum of $[\boldsymbol{B}]_{r,c}$ distinct and non-overlapping randomly selected permutation matrices with size $b \times b$.

3.2 Generalized LDPC Codes

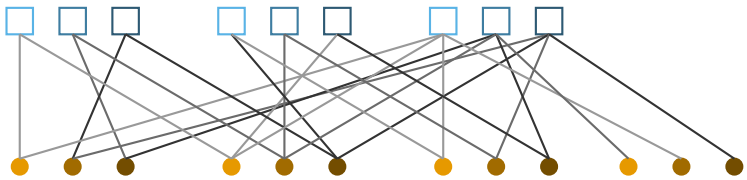
As mentioned, each of the CNs in a Tanner graph of an LDPC codes represents a single parity-check equation. A generalized LDPC (GLDPC) code [24] can be obtained by replacing the CNs of an LDPC code by more complex linear constraints referred to as *component codes*; The $\boldsymbol{H}_{\mathrm{G}}$ matrix of the GLDPC code is both related to the parity-check matrices of the mother LDPC code $\boldsymbol{H}$ and the component code $\boldsymbol{H}_{\mathrm{c}}$. More precisely, $\boldsymbol{H}_{\mathrm{G}}$ can be obtained by replacing each of nonzero elements of $\boldsymbol{H}$ with a column of $\boldsymbol{H}_{\mathrm{c}}$ that is uniformly random selected.



(a)



(b)



(c)

Figure 3.2: (a) A simple protograph (b) 3 copies of the protograph (c) A derived Tanner graph.

3.3 Turbo-Like Codes

The conventional way of describing concatenated CCs—TCs—is based on the block diagram of their encoders and decoders. However, these codes are also a class of codes on graphs and their graph representations help to simplify their analysis. In this section, first, the graph representation of CCs is discussed; then, some classes of TCs are introduced by their graph representations.

3.3.1 Convolutional Codes

In [27], [28], Wiberg showed that it is possible to illustrate trellis-based codes by introducing some hidden VNs. These hidden VNs are not corresponding to code bits, and they are used to give a suitable local structure. For the case of CC, the hidden VNs represent the states of the encoder—therefore called *state node*—, and are shown by double circles in the graph.

Consider a systematic CC with arbitrary rate $R = k/n$ and N trellis sections. Let $\boldsymbol{u} = (\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_N)$ and $\boldsymbol{v} = (\boldsymbol{v}_{p,1}, \boldsymbol{v}_{p,2}, \ldots, \boldsymbol{v}_{p,N})$ denote the input and parity sequences, respectively. More precisely, at each time instant $\tau = 1, \ldots, N$, $\boldsymbol{u}_\tau = (u_\tau^{(1)}, \ldots, u_\tau^{(k)})$ and $\boldsymbol{v}_{p,\tau} = (v_{p,\tau}^{(1)}, \ldots, v_{p,\tau}^{(n-k)})$ are the input and parity sequences, respectively. The corresponding *factor graph* [29] is shown in Figure 3.3. In this graph, the code bits are illustrated by filled circles and the code constraints by filled squares, and as mentioned, the double circles represent the state nodes.

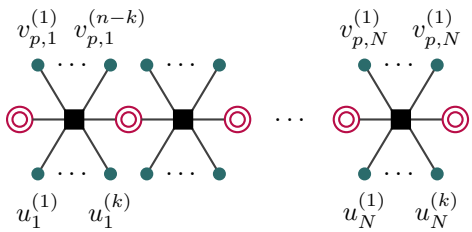


Figure 3.3: Factor graph representation of systematic rate-k/n convolutional codes.

3.3.2 Parallel Concatenated Codes

Turbo codes or PCCs are a class of TCs proposed in [5], [6]. The main idea of turbo codes is to encode the information sequence $\boldsymbol{u}$ with two or more component CC encoders. The encoder block diagram corresponding to a family of PCCs —an ensemble—, with two component encoders, is shown in Figure 3.4 (a). The information $\boldsymbol{u}$ is encoded by the upper encoder $\mathcal{C}^{\mathrm{U}}$ to produce the sequence $\boldsymbol{v}^{\mathrm{U}}$. Then, $\boldsymbol{u}$ is reordered by the permutation Π and the resulting sequence is encoded by the lower encoder to produce the sequence $\boldsymbol{v}^{\mathrm{L}}$. The output of this encoder is $\boldsymbol{v} = (\boldsymbol{u}, \boldsymbol{v}^{\mathrm{U}}, \boldsymbol{v}^{\mathrm{L}})$. Note that the overall rate of a PCC ensemble depends on the number of its component encoders and their rates. Different permutations result in different PCCs with different performances. Therefore, finding the characteristics of a permutation which results in a PCC with better performance has been the topic of many studies.

Using the graph representation of CCs, it is possible to represent PCC ensembles by factor graphs. For rate $R = 1/2$ component encoders, the corresponding graph representation is shown in Figure 3.4 (b).

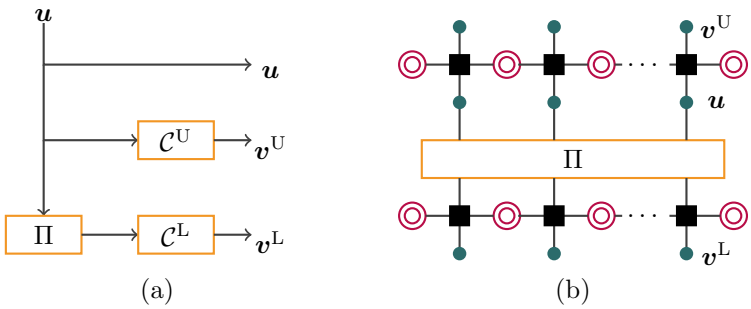


Figure 3.4: PCCs (a) encoder block diagram (b) factor graph representation.

3.3.3 Serially Concatenated Codes

Similar to PCCs, the idea of serially concatenated codes (SCCs) [30], [31] is to encode the information sequence with two or more component encoders. However, for SCCs the component encoders are serially interconnected by permutations, i.e., the output of one encoder is reordered and used as input to the next encoder. The encoder block diagram of an SCC ensemble with two component encoders is shown in Figure 2 (a).

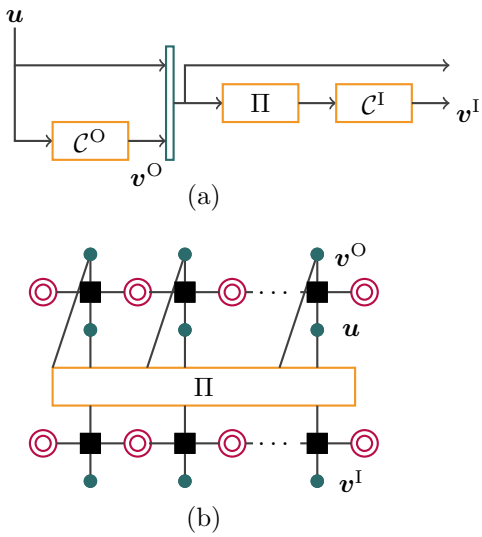


Figure 3.5: SCCs (a) encoder block diagram (b) factor graph representation.

The information sequence is encoded by the outer encoder $\mathcal{C}^{\mathrm{O}}$ to produce

the parity sequence $\boldsymbol{v}^{\mathrm{O}}$. Then, $\boldsymbol{u}$ and $\boldsymbol{v}^{\mathrm{O}}$ are multiplexed and reordered by permutation Π. Then, the resulting sequence is fed to the inner encoder to produce the parity sequence $\boldsymbol{v}^{\mathrm{I}}$. Finally, the output of the encoder is $\boldsymbol{v} = (\boldsymbol{u}, \boldsymbol{v}^{\mathrm{O}}, \boldsymbol{v}^{\mathrm{I}})$. Considering rate $R = 1/2$ component encoders the corresponding factor graph is shown in Figure 3.5 (b).

3.3.4 Hybrid Concatenated Codes

Hybrid concatenated codes (HCCs) [32], [33], [34] are a mixture of PCCs and SCCs, and consequently are built from three or more component encoders. An example of an HCC ensemble is shown in Figure 3.6 (a).

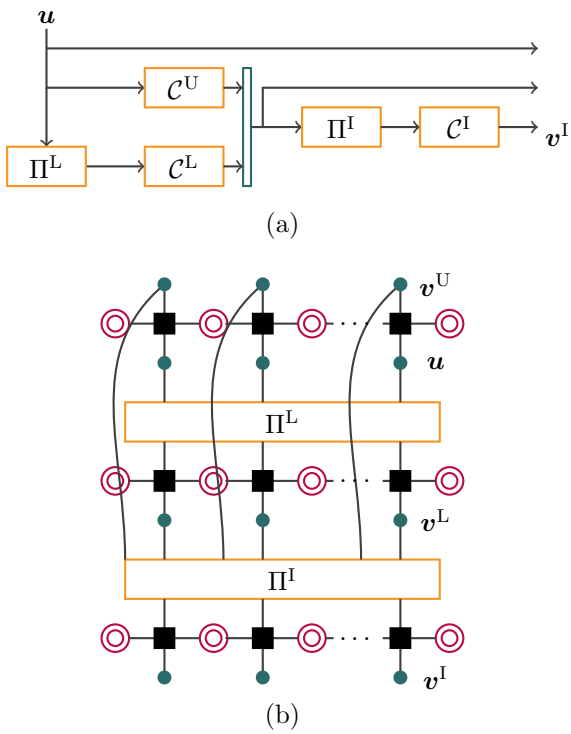


Figure 3.6: HCCs (a) encoder block diagram (b) factor graph representation.

This ensemble consists of a PCC ensemble serially connected to an inner encoder, and is very closed to so-called *3D-turbo codes* introduced in [33], [35]. As it can be seen from the figure, the information sequence $\boldsymbol{u}$ and a reordered copy of it are encoded by upper $\mathcal{C}^{\mathrm{U}}$ and lower $\mathcal{C}^{\mathrm{L}}$ encoders, respectively. The

corresponding parities are denoted by $\boldsymbol{v}^{\mathrm{U}}$ and $\boldsymbol{v}^{\mathrm{L}}$. Then, these parity sequences are multiplexed and reordered by permutation Π^{I}. The resulting sequence is fed to the inner encoder to produce the parity sequence $\boldsymbol{v}^{\mathrm{I}}$. Considering rate $R = 1/2$ component encoders, the corresponding graph representation is shown in 3.6 (b).

3.4 Repeat-Accumulate Codes

Repeat-accumulate (RA) codes, proposed by Divsalar et al. [10], interestingly, can be interpreted as a subclass of either LDPC codes or TCs. Being a subclass of TCs gives the RA-based LDPC codes the advantage of easy and efficient encoding. This fact has made RA-codes an attractive candidate to be used in communication standards. The encoder block diagram of a regular RA code ensemble is shown in Figure 3.7 (a).

This ensemble is a serial concatenation of a repetition block encoder $\mathcal{R}$ and a CC encoder. The information sequence $\boldsymbol{u}$ is first encoded by $\mathcal{R}$ to make q copies of $\boldsymbol{u}$. Then, the resulting sequence is permuted by Π—to create the sequence $\boldsymbol{z}$—and fed to a simple two-state CC—called *accumulator*—with generator matrix $\boldsymbol{G} = (1, 1/(1 + D))$. An irregular RA code can be obtained by puncturing the sequence $\boldsymbol{z}$.

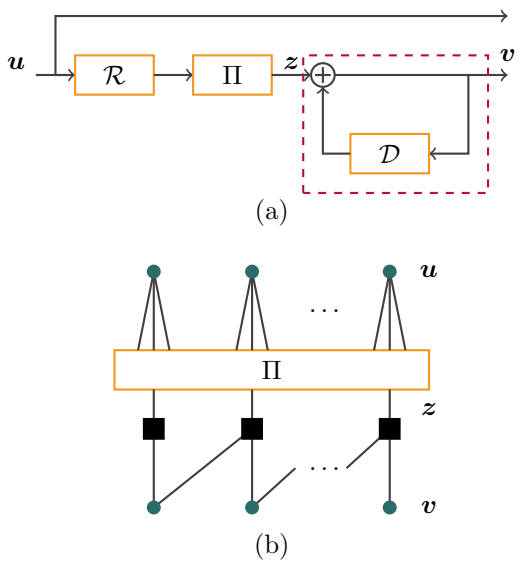


Figure 3.7: RA codes: (a) encoder block diagram (b) factor graph.

Figure 3.7 (b) shows the Tanner graph representation of an RA code en-

semble with $q = 3$. It is worthy to mention that RA codes are closely related to PCCs. For $q = 2$, RA codes can be seen as the self-concatenated counterpart of PCCs with two identical component encoders.

3.5 Product Codes

The idea of PCs [9] is to build powerful codes from smaller component codes. To obtain a PC, $\mathcal{C}(N, K)$, consider two linear block codes $\mathcal{C}_1(N_1, K_1)$ and $\mathcal{C}_2(N_2, K_2)$, where $N = N_1N_2$ and $K = K_1K_2$. The encoding procedure of PCs can be described by use of the array represented in Figure 3.8. The K information bits are written in the $K_1 \times K_2$ block A at the upper right side of the array. Then, each row of A is encoded by $\mathcal{C}_2$, and the parity bits are written at the corresponding row of block B. Then, columns of A and B are encoded by $\mathcal{C}_1$, and the parity bits are written at corresponding columns of block C and D, respectively.

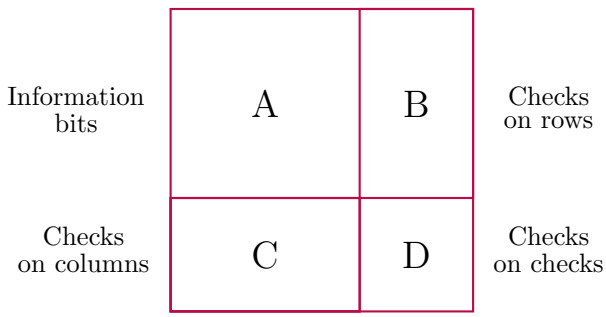


Figure 3.8: Product code array.

3.6 Iterative Decoders

The excellent performance of codes on graphs, and their efficiency in terms of computational complexity are rooted in the message passing decoding algorithms such as belief propagation (BP). In these iterative algorithms, the extrinsic informations are exchanged within the graph. The iterative decoders are more efficient if their component decoders can exchange soft messages, i.e., the component decoders can receive soft input and deliver soft output (SISO).

In this section, before describing the iterative decoders, some basic concepts are discussed.

3.6.1 Maximum A-Posteriori Decoding

A bit-wise maximum a-posteriori decoder is a decoder that delivers the estimated bit $\hat{u}_i$ by

$$\hat{u}_i = \underset{u_i \in \{0,1\}}{\arg\max} \quad P(u_i|\boldsymbol{r})$$

where $\boldsymbol{r}$ is the received vector and $p(u_i|\boldsymbol{r})$ is the *a-posteriori probability* (APP) for the ith bit of the input sequence, u_i, $i = 1, \ldots, K$.

Instead of a hard decision of each bit, a decoder can provide the APP for the bit. The corresponding decoder is called *APP decoder*. Having soft outputs, the APP decoders are widely used as component decoders in the iterative decoding procedure in which the soft output of one decoder is passed to the input of the other decoders.

3.6.2 Likelihood Ratio

For the binary cases, the bit u_i can take the two values 0 or 1 with an a-priori probability $P(u_i)$. However, instead of working with probabilities, from an implementation perspective, it is better to work with *log-likelihood ratio* (LLR) defined as

$$L(u_i) = \log \frac{P(u_i = 0)}{P(u_i = 1)}.$$

Likewise, LLRs can be used instead of the APPs, $p(u_i|\boldsymbol{r})$,

$$L(u_i|\boldsymbol{r}) = \log \frac{P(u_i = 0|\boldsymbol{r})}{P(u_i = 1|\boldsymbol{r})}.$$

Also, the channel output can be presented in LLR format; as an example, for the BEC the LLRs at the output of the channel can have one of the following values,

$$L_{\text{ch}}(u_i|\boldsymbol{r}) = \log \frac{P(u_i = 0|\boldsymbol{r})}{P(u_i = 1|\boldsymbol{r})} = \begin{cases} +\infty & r_i = +1 \\ -\infty & r_i = -1 \\ 0 & r_i = e \end{cases},$$

and $L_{\text{ch}}(v_i|r_i) = \frac{2}{\sigma^2} r_i$ for the case of the AWGN channel.

3.6.3 Belief Propagation Decoding

For the ith bit of the codeword $\boldsymbol{v}$, the soft output $L(v_i)$ can be written as the summation of three LLRs; a-priori L-value $L_a(v_i)$, channel L-value $L_{ch}(v_i)$, and extrinsic L-value $L_e(v_i)$ that is corresponding to the estimation of v_i based on other bits v_j, $j \neq i$,

$$L(v_i) = L_a(v_i) + L_{ch}(v_i) + L_e(v_i).$$

Note that in the iterative decoders only the extrinsic informations are exchanged, i.e., the extrinsic output of one component is passed as a-priori message to the other decoder.

LDPC Codes

For LDPC codes, the decoding process is based on the Tanner graph representation of the code; through that, the extrinsic L-values are exchanged between VNs and CNs. Let $L_v^{(i)}(e_j)$ denote the extrinsic L-value from the jth edge of the VN v—with degree d_v—in ith iteration. Likewise, let $L_c^{(i)}(e_j)$ denote the extrinsic L-value from the jth edge of the CN c—with degree d_c—in the ith iteration. The decoding process can be initiated by considering $L_c^{(0)}(e_j) = 0$, $j = 1, \ldots, d_c$, for all CNs.

Figure 3.9: Iterative decoder of LDPC codes.

At the VN side (Figure 3.9 (a)), the extrinsic L-value from VN v through its jth edge can be updated by

$$L_v^{(i)}(e_j) = L_{ch}(v) + \sum_{j' \neq j} L_c^{(i-1)}(e_{j'}).$$

At the CN side (Figure 3.9 (b)), the extrinsic L-value from CN c through its jth edge can be updated by

$$L_c^{(i)}(e_j) = \boxplus_{j' \neq j} L_v^{(i)}(e'_j),$$

where the operation $\boxplus$ is called *box-plus*, and defined as

$$L(v_1) \boxplus L(v_2) \triangleq L(v_1 + v_2) = \log \frac{1 + e^{L(v_1)} e^{L(v_2)}}{e^{L(v_1)} + e^{L(v_2)}}.$$

Box-plus can be approximated by

$$L(v_1) \boxplus L(v_2) \approx sign(L(v_1) \cdot L(v_2)) \cdot \min(|L(v_1)|, |L(v_2)|).$$

Turbo-Like Codes

Iterative decoders of TCs use SISO component decoders based on the BCJR algorithm. As an example, the block diagram of an iterative decoder with two component BCJR decoders, A and B, is shown in Figure 3.10. At the beginning of the decoding procedure, decoder A receives the channel L-values and no a-priori L-values ($L_a(\boldsymbol{v}_A) = 0$). Then, the BCJR algorithm is run to find the corresponding extrinsic L-values $L_e(\boldsymbol{v}_A)$. This extrinsic message is properly permuted and used as a-priori message to decoder B. A very similar procedure is running in decoder B. After a certain number of the iteration the final output of the decoder is,

$$L(v_i) = L_a(v_i) + L_{ch}(v_i) + L_e(v_{i,A}) + L_e(v_{i,B})$$

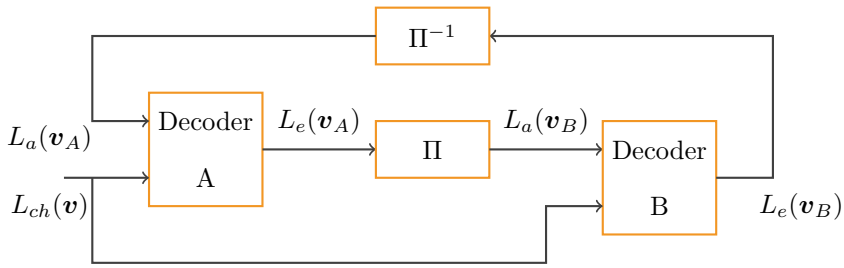


Figure 3.10: Iterative decoder for TCs with two component decoders.

Chapter 4

Tools for Performance Analysis

The performance of codes can be evaluated by their error rate performance curves. For that, one can use Monte Carlo simulations to obtain the corresponding error rate curve as a function of SNR. The error rate curve of graph-based codes consists of two regions; 1) *waterfall* at low SNRs where this curve has a steep slope, 2) *error floor* at higher SNRs where the error rate curve has flatter slope, and by increasing the SNR, the error rate does not improve significantly.

Since the invention of TCs, considerable effort has been devoted to the study of the error rate performance of these codes. In this chapter, the tools for investigating the performance of TCs in both waterfall and error floor regions are discussed. To study the waterfall region performance, in the first section, the provided tools can be used to find a parameter called *decoding threshold* of the considered code ensembles. The decoding threshold is the channel parameter that divides the channel parameter spectrum into two regions regarding reliable decoding, as the block length tends to infinity. For the AWGN channel, the threshold is the smallest SNR, and for BEC, it is the largest channel erasure probability, for which reliable decoding is possible.

Then, in the next section, to investigate the performance of a decoder in the error floor region, the so-called *weight enumerator* (WE) analysis of TCs is discussed. The WE analysis can be used to find bounds on the error rate performance and the minimum distance of the TC ensembles.

The results from the threshold and WE analysis together can be used to design code ensembles that can be simultaneously powerful in both waterfall and error floor regions.

4.1 Threshold Analysis

4.1.1 Density Evolution

It is possible to investigate the performance of iterative decoders by *density evolution* (DE). For the BEC, DE can be formalized as a set of equations that tracks how the erasure probability of the message evolves during the iterations. The simplicity of the BEC allows us to derive the exact DE equations.

As an example, consider a regular (d_v, d_c) LDPC ensemble and transmission over the BEC with erasure probability ε. Considering the BP decoder, the exact DE equation, corresponding to this ensemble, can be easily derived as follows. The symmetric structure of the regular LDPC ensemble, allows us to consider that in each iteration the erasure probabilities of the exchanged messages to all CNs are equal. The same assumption can be considered for VNs.

An outgoing message from a VN through a specific edge is erased if all incoming messages to that VN, through other edges—including the message from the channel— are erased (see Figure 4.1 (a)). Therefore, for incoming erasure probability x to the VN, the outgoing erasure probability is

$$y = \varepsilon \cdot x^{d_v - 1} \ .$$

At the CN side, the outgoing message through a specific edge is erased if at least one of the incoming messages to the corresponding CN is erased. For incoming erasure probability y to the CN, the outgoing erasure probability is

$$x = 1 - (1 - y)^{d_c - 1} \ .$$

Considering the exchange of messages between VNs and CNs through iterations, the DE equation of the considered LDPC ensemble in the ith iteration is

$$x^{(i)} = \varepsilon \cdot \left(1 - (1 - x^{(i-1)})^{d_c - 1}\right)^{d_v - 1}. \tag{4.1}$$

For the BEC, the BP threshold $\varepsilon_{\mathrm{BP}}$ is the largest channel erasure probability for which the only fixed point of the DE recursion is $x = 0$.

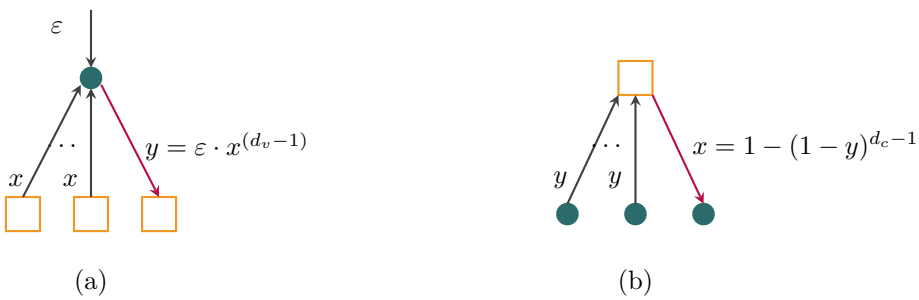


Figure 4.1: Iterative decoder of LDPC codes.

For TCs, the component decoders use a more complex decoding algorithm than that of LDPC codes. Considering the BCJR decoding algorithm, for the transmission over the BEC, it is possible to compute the exact transfer function between the input erasure probabilities—including channel erasure probability and a-priori erasure probability—and extrinsic output erasure probabilities of the decoder by the methods described in [36], [37].

Consider the iterative decoder of the TC ensemble in Figure 4.2 and transmission over the BEC with erasure probability ε. The extrinsic erasure probability at the output of Decoder A, x can be computed as a function of the a-priori erasure probability y and channel parameter,

$$x = f_{\mathrm{A}}(y; \varepsilon),$$

where f_{A} is the transfer function of Decoder A.

Figure 4.2: Input-output erasure probabilities of BCJR Decoder A.

A similar equation can be written for Decoder B. As the extrinsic output of Decoder A is passed to the input of Decoder B, the corresponding DE equation can be written as

$$y = f_{\mathrm{B}}(x; \varepsilon).$$

where f_{B} is the transfer function of Decoder B, and y and x are the extrinsic and a-priori erasure probability of Decoder B, respectively. Finally, by considering the exchanging of the extrinsic messages the DE of the PCC ensemble at ith iteration can be written as

$$x^{(i)} = f_{\mathrm{A}}(f_{\mathrm{B}}(x^{(i-1)}; \varepsilon); \varepsilon).$$

4.1.2 EXIT Chart

An *extrinsic information transfer* (EXIT) chart [38] is a graphical tool which visualizes the convergence behavior of iterative decoding algorithms. To understand the EXIT chart properly, we need to know the definition of mutual information. For two random variables X and Y, the mutual information is defined as

$$I(X;Y) = H(Y) - H(Y|X),$$

where $H(Y)$ is the entropy of the variable y, and

$$H(Y|X) = \int\int f(x,y) \log \frac{1}{f(y|x)} dxdy \ .$$

The EXIT chart tracks the evolution of mutual information between the codeword bits and the messages that are exchanged.

Consider the iterative decoder of the TC ensemble in Figure 3.10. We denote the mutual information between the a-priori message at the input of Decoder A and the transmitted codeword by $I_{\mathrm{a}}^{\mathrm{A}}$, and the mutual information between the extrinsic message at the output of Decoder A and the transmitted codeword by $I_{\mathrm{e}}^{\mathrm{A}}$. Likewise, we define the mutual information $I_{\mathrm{a}}^{\mathrm{B}}$ and $I_{\mathrm{e}}^{\mathrm{B}}$ at the input and output of Decoder B. The EXIT functions for Decoder A can be defined as

$$I_e^{\mathrm{A}} = f^{\mathrm{A}}(I_a^{\mathrm{A}}) \ .$$

During the iterative decoding procedure the extrinsic output values of one decoder become the a-priori values of the other decoder. Therefore, in the EXIT chart, the EXIT function for Decoder A and the inverse of the EXIT function for Decoder B are depicted to follow the decoding trajectory. Starting from $I_a^{\mathrm{A}} = 0$, finding the I_e^{A} on the corresponding function, then, by considering $I_a^{\mathrm{B}} = I_e^{\mathrm{A}}$ and finding I_e^{B}, and continuing so on, the decoding trajectory can be depicted on the EXIT chart. Note that the decoding is successful if the trajectory can achieve 1 in one of the axis.

For the transmission over the BEC, the mutual information is closely related to the erasure permeability by $I_e = 1 - p_e$ and $I_a = 1 - p_a$. Therefore, it is possible to compute the exact EXIT functions. As an example, consider the PCC ensemble in Figure 3.4, with identical component encoders and $\boldsymbol{G} = (1, 5/7)$ in octal notation. The corresponding functions to Decoder A and B are computed and for the channel erasure probability $\varepsilon = 0.6$, the corresponding EXIT chart is shown in Figure 4.3.

As it can be seen in the figure, starting from $I_a^{\mathrm{A}} = 0$, the decoding trajectory can achieve 1 after a few iterations. In this case, the tunnel between two EXIT functions is open, and the decoding is successful. However, by increasing ε, the two EXIT functions change their shapes and get close in some parts. Finally, the tunnel gets close for any $\varepsilon > \varepsilon_{\mathrm{BP}}$. Thus, in terms of EXIT charts, $\varepsilon_{\mathrm{BP}}$ is the largest channel erasure probability for that the tunnel is still open. For the considered PCC ensemble the BP threshold is equal to $\varepsilon_{\mathrm{BP}} = 0.642$, and for this channel erasure probability the corresponding EXIT chart is shown in Figure 4.4.

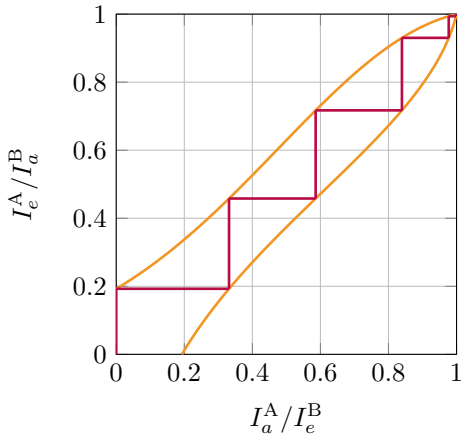


Figure 4.3: EXIT chart of a PCC ensemble at $\varepsilon = 0.6$.

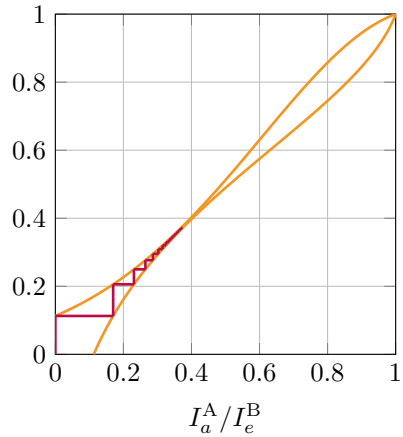


Figure 4.4: EXIT chart of a PCC ensemble at $\varepsilon_{\mathrm{BP}} = 0.642$.

4.1.3 Area Theorem

For sparse-graph codes, it is possible to connect their performance under BP decoding to that of the MAP decoding by the area theorem [38], [39], [40], [41]. According to this theorem, for transmission over the BEC, an upper bound on the MAP threshold, $\varepsilon_{\mathrm{MAP}}$, can be computed by

$$\int_{\varepsilon_{\mathrm{MAP}}}^{1} \bar{p}_e(\varepsilon) d\varepsilon = R, \tag{4.2}$$

where R is the code rate, and $\bar{p}_e(\varepsilon)$ is the average extrinsic erasure probability at the output of the iterative decoder.

A modified version of EXIT chart—called *BP-EXIT chart*—can also be used to visualize the asymptotic behavior of the decoding algorithm, and obtain the BP and MAP decoding thresholds. For the BEC, $\bar{p}_e(\varepsilon)$ is considered as the BP-EXIT function and is used to obtain the corresponding BP-EXIT chart. Consider the PCC and SCC ensembles in Chapter 3, with identical component encoders, and $\boldsymbol{G} = (1, 5/7)$ in octal notation. The BP-EXIT charts corresponding to these ensembles are shown in Figure 4.5 and 4.6, respectively. Based on this chart, the BP threshold is the largest channel erasure probability for which $\bar{p}_e(\varepsilon) = 0$. The upper bound on the MAP threshold $\varepsilon_{\mathrm{MAP}}$ is the channel erasure probability from which, the area under the BP-EXIT chart is equal to the code rate R.

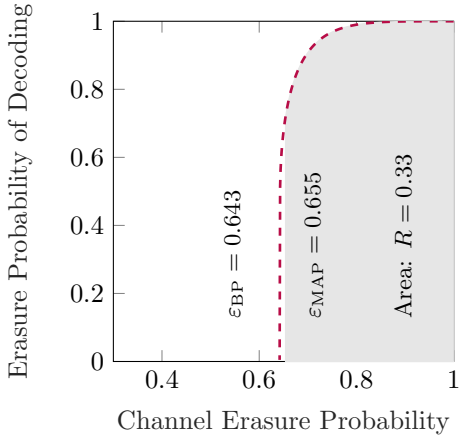


Figure 4.5: BP-EXIT chart of a rate 1/3 PCC ensemble.

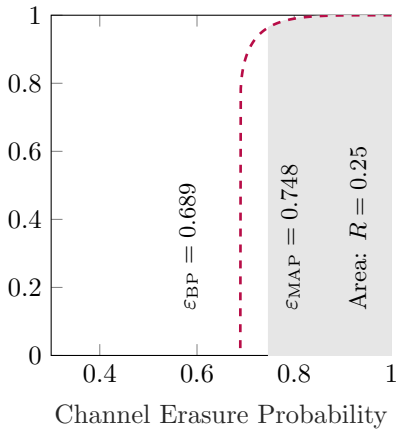


Figure 4.6: BP-EXIT chart of a rate 1/4 SCC ensemble.

4.2 Weight Enumerator Analysis

The weight enumerator function (WEF) of a linear (N, K) block encoder is a polynomial that represents the number of codewords of each possible Hamming weight. The input-output WEF (IO-WEF) of the encoder can be written as

$$A(I, W) = \sum_{w=1}^{N} \sum_{i=1}^{K} A_{i,w} I^i W^w,$$

where $A_{i,w}$ denotes the number of codewords with input Hamming weight i and output Hamming weight w. Given the IO-WEF of the encoder, it is possible to find an upper bound on the error probability for the ML decoder. Based on the pairwise error probability p_w, the frame error probability is upper bounded by,

$$P_F \leq \sum_{w=1}^{N} \sum_{i=1}^{K} A_{i,w}\ p_w. \tag{4.3}$$

For the AWGN channel p_w is equal to

$$\mathsf{Q}\left(\sqrt{2wR\frac{E_\mathrm{b}}{N_0}}\right),$$

where $\mathsf{Q}(.)$ is the Q-function and E_b/N_0 is the signal-to-noise ratio.

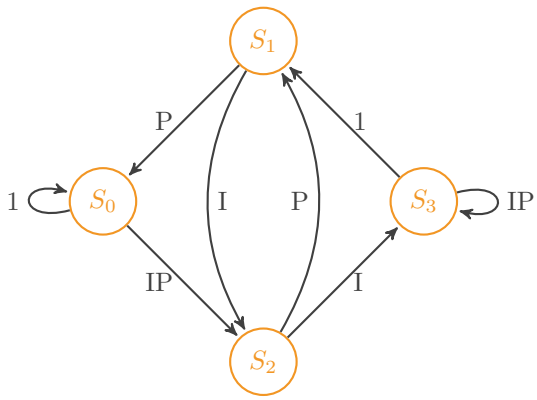


Figure 4.7: State diagram corresponding to CC encoder in Figure 2.2.

The upper bound on bit error probability can be obtained by,

$$P_b \leq \sum_{w=1}^{N} \sum_{i=1}^{K} \frac{i}{K} A_{i,w}\, p_w. \tag{4.4}$$

Equations (4.3) and (4.4) show that the IO-WEF plays a key role in computing bounds on error performance of codes. To find the IO-WEF of TCs, the first step is to compute the IO-WEF of the corresponding CC component encoders. In the following subsection, a method for computing the IO-WEF of CCs is provided.

4.2.1 Weight Enumerator Function of Convolutional Codes

The input-parity WEF (IP-WEF) of a CC, $A(I, P)$[1],can be obtained by use of its state diagram and its corresponding transition matrix. As an example consider the state diagram corresponding to the CC encoder in Figure 2.2. This state diagram is shown in Figure 4.7. However, in this state diagram, the edge labels are replaced with monomials $I^i P^p$, where i and p are either 0 or 1 depending on the weight of the corresponding input and parity bits, respectively.

The state transition diagram with s states, can be summarized in an $s \times s$ matrix $\boldsymbol{M}$, called *state transition matrix*. The element of M in the rth row and cth column, $[\boldsymbol{M}]_{r,c}$, is the label on the edge that connects the rth state

[1]The coefficient of IP-WEF, $A_{i,p}$, specifies the number of codewords in $\mathcal{C}$ with input Hamming weight i and parity Hamming weight p

to the cth state. As an example, for the state diagram in Figure 4.7, the state transition matrix can be written as

$$\boldsymbol{M} = \begin{bmatrix} 1 & 0 & IP & 0 \\ P & 0 & I & 0 \\ 0 & P & 0 & I \\ 0 & 1 & 0 & IP \end{bmatrix}_{s \times s} .$$

For a trellis with N sections, the overall transition matrix can be obtained by $\boldsymbol{M}^N$. Then, by considering that the trellis is initialized and terminated to s_0, the IO-WEF is equal to $[\boldsymbol{M}^N]_{1,1}$.

4.2.2 Average Weight Enumerator of the Concatenated Convolutional Codes

The average IO-WEF of a TC ensemble can be obtained from the IO-WEF of its component encoders, by considering uniformly random permutations (URPs) [42]. A URP with size N' can be seen as a probabilistic device that gets a vector with size N' and Hamming weight w as input and generates one of all the possible permutations of this vector at the output with probability $1/\binom{N'}{w}$. Using the URP for analysis is equivalent to averaging over all possible permutations.

As an example, consider the PCC ensemble in Figure 3.4. The IP-WEFs corresponding to component encoders can be obtained by the method explained. Let $A_{i,p}^{\mathrm{T_U}}$ and $A_{i,p}^{\mathrm{T_L}}$ denote the IP-WEF of the upper and lower encoder, respectively. For a given input with weight i, the coefficients of the average IP-WEF of the ensemble $\bar{A}_{i,p}^{\mathrm{PCC}}$, can be obtained as [42]

$$\bar{A}_{i,p}^{\mathrm{PCC}} = \frac{\sum_{p_1} A_{i,p_1}^{\mathrm{T_U}} \cdot A_{i,p-p_1}^{\mathrm{T_L}}}{\binom{N}{i}} . \tag{4.5}$$

Similar to the PCC ensemble, by considering URP, the WEFs of other TC ensembles can be obtained [31].

Using the obtained average IP-WEF of the TC ensembles in equations (4.3) and (4.4), it is possible to compute bounds on the average error rate performance of the ensembles. These bounds diverge for small SNRs. Therefore, they are usually used to estimate the error floor of an ML decoder. Also, it is worthy to mention that to obtain bounds on the performance of the BP decoder, more investigations on the corresponding absorbing sets [43] and pseudo-codewords [44] have to be done.

Chapter 5

Spatially Coupled Codes

Spatial coupling [45] is a technique that can be applied to codes on graphs, and leads to an improvement in the performance of the codes under iterative decoding. In this chapter, two classes of spatially coupled codes are discussed.

5.1 SC-LDPC Codes

LDPC convolutional codes (LDPC-CCs) were proposed in [11] by introducing memory into the encoding procedure of LDPC codes. These codes —also known as SC-LDPC codes [14]— can be constructed from LDPC codes by use of edge spreading over their graph representation. Consider a graph representation of an LDPC code, either Tanner graph or protograph [46]. Then, make L copies of this graph, where L is called *coupling length*, and assign a time index t, $t = 1, \ldots, L$, to each of these copies. To obtain a SC-LDPC ensemble with coupling memory m, connect the graph at time t to its neighboring copies at time slots $t + 1$ to $t + m$. For that, we can randomly select one edge of the graph, keep the VN end but connect the other end randomly to one of the corresponding CN of the next m copies. The graphs at the boundaries do not have either right or left neighbors, so the overall graph is slightly irregular at the boundaries of the chain. This structured irregularity makes the code more powerful at the boundaries. Therefore, on the decoder side, the original bits can be recovered with a lower error rate at the boundaries. Then, the boundary copies also help their neighboring copies in the decoding, and in turn, this effect is cascading over the whole chain.

As an example, consider L copies of the protograph of a regular (3,6) LDPC ensemble shown in Figure 5.1. For this ensemble,

$$\boldsymbol{B} = [3, 3].$$

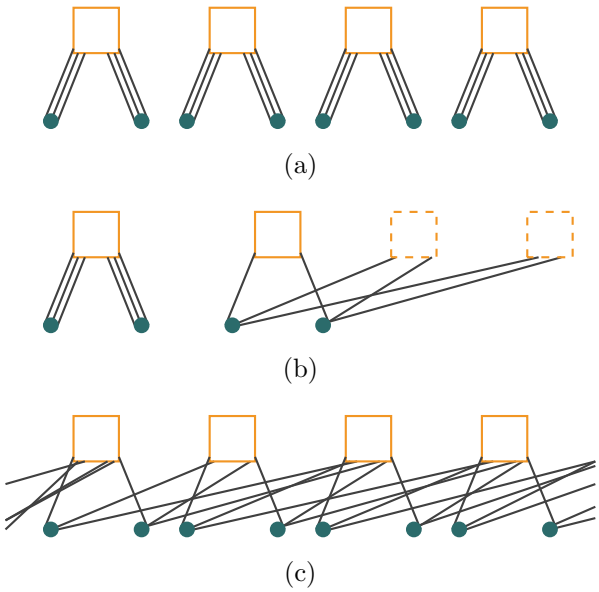


Figure 5.1: SC-LDPC codes.

For coupling memory $m = 2$, the edges at the tth copy are spread over the copies at positions $t + 1$ to $t + m$, Figure 5.1 (b). A chain of an SC-LDPC ensemble can be obtained by applying the edge-spreading procedure for all positions $t = 1, \ldots, L$ (Figure 5.1 (c)).

This edge-spreading procedure —also called *unwrapping*— can be described by splitting the corresponding matrix $\boldsymbol{B}$ (or $\boldsymbol{H}$) into m sub-matrices $\boldsymbol{B}_0, \ldots, \boldsymbol{B}_m$ ($\boldsymbol{H}_0, \ldots, \boldsymbol{H}_m$), such that $\boldsymbol{B} = \boldsymbol{B}_0 + \cdots + \boldsymbol{B}_m$ ($\boldsymbol{H} = \boldsymbol{H}_0 + \cdots + \boldsymbol{H}_m$). Then the $\boldsymbol{B}_{\text{sc}}$ matrix of the corresponding SC-LDPC ensemble is

$$\boldsymbol{B}_{\text{sc}} = \begin{bmatrix} \boldsymbol{B}_0 & & & \\ \boldsymbol{B}_1 & \boldsymbol{B}_0 & & \\ \vdots & \boldsymbol{B}_1 & \ddots & \\ \boldsymbol{B}_m & \vdots & \ddots & \boldsymbol{B}_0 \\ & \boldsymbol{B}_m & & \boldsymbol{B}_1 \\ & & \ddots & \vdots \\ & & & \boldsymbol{B}_m \end{bmatrix}_{(L+m)b_c \times Lb_v} . \tag{5.1}$$

For the example in Figure 5.1, $\boldsymbol{B}_0 = \boldsymbol{B}_1 = \boldsymbol{B}_2 = [1, 1]$, and the corresponding $\boldsymbol{B}_{\text{sc}}$ can be obtained by replacing these sub-matrices in equation (5.1).

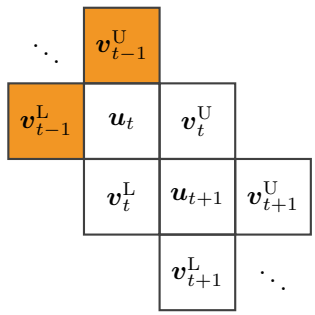


Figure 5.2: Array representation of braided codes.

It has been shown that the BP threshold of the SC-LDPC codes improves to the MAP threshold of the underlying LDPC codes. This remarkable phenomenon is called *threshold saturation* and has been proved for SC-LDPC codes [15], [16], [47], [48].

Although the complexity and latency of BP decoding increases by the coupling length for the spatially coupled ensembles, this problem can be solved by the use of sliding window decoding [49], [50], [51], [52]. For that, over the spatially coupled chain of decoders, only the decoders within a window frame are getting active. The decoders iterate for a certain number of times. Then, the window is shifted, and the decoders within the new window are iterating. This procedure is continuing until the whole chain has been processed by the window. By this way, while the excellent performance of the BP decoder is preserved, the complexity and latency of the decoding are reducing significantly.

5.2 Braided Codes

Braided codes, introduced in [18], [53], [54], can be seen as convolutional—or spatially coupled— version of PCs. To explain the structure of these codes, one can use a two dimensional infinite array, consisting of three diagonal ribbons (see Figure 5.2). Based on the density of the storage array, braided codes can be categorized into two classes: tightly and sparesly braided codes. A tightly braided code (TBC) has a dense array of the information and parity symbols. Sparsely braided codes (SBC) have low density, and show better performance under iterative decoding. As shown in Figure 5.2, similarly to product codes, the information and parity symbols are written in cells, and the information symbols are encoded twice by vertical and horizontal component encoders. However, for braided codes at time t, the parity of the horizontal encoder $\boldsymbol{v}^{\mathrm{U}}_t$ depends on the current information symbol $\boldsymbol{u}$ and the parity of the vertical

encoder at $t-1$, $\boldsymbol{v}_{t-1}^{\mathrm{L}}$. Likewise, the vertical encoder gets $\boldsymbol{u}$ and $\boldsymbol{v}_{t-1}^{\mathrm{U}}$ as inputs to produce $\boldsymbol{v}_t^{\mathrm{L}}$. It is considered that $\boldsymbol{v}_t^{\mathrm{L}} = 0$ and $\boldsymbol{v}_t^{\mathrm{U}} = 0$ for $t \leq 0$.

Depending on their component encoders, braided codes are also categorized into two main categories: 1) braided block codes (BCCs) which have block component encoders 2) braided convolutional codes (BCCs) which have convolutional component encoders. BBCs with Bose-Chaudhuri- Hocqenghem (BCH) component codes are closely related to so-called *staircase codes* [55] and have been investigated for high speed optical communications, and it has been shown that they have an excellent performance under iterative hard decision decoding [56], [57].

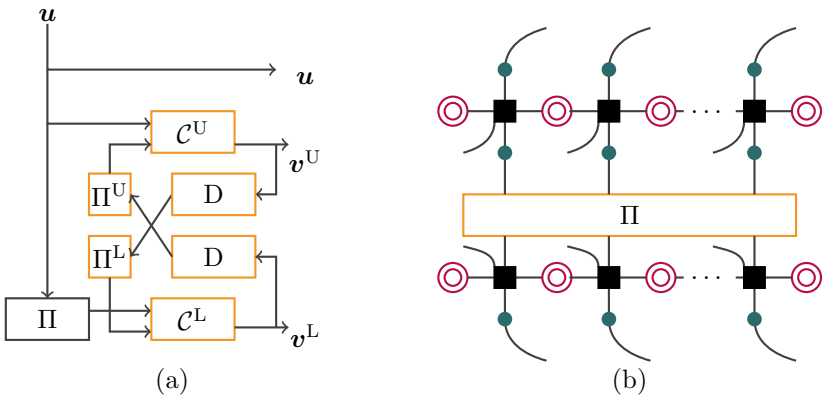


Figure 5.3: BCCs: (a) encoder block diagram (b) factor graph.

BCCs are closely related to PCCs and can be considered as a class of TCs. However, for BCCs, through a feedback the parity sequence of one component encoder is used as future input of the other component encoder. Therefore, the information bits and parity bits are equally protected due this symmetric type of concatenation. The encoder block diagram of a BCC ensemble with $R = 1/3$ is shown in Figure 5.3 (a). As it is shown, the ensemble is built of two rate-2/3 convolutional encoders, called upper and lower encoders. At time t, the input to this encoder is $\boldsymbol{u}_t$ and the output is $\boldsymbol{v} = (\boldsymbol{u}_t, \boldsymbol{v}_t^{\mathrm{U}}, \boldsymbol{v}_t^{\mathrm{L}})$. The upper encoder gets $\boldsymbol{u}_t$, and a reordered copy of $\boldsymbol{v}_{t-1}^{\mathrm{L}}$ as first and second inputs, respectively. Likewise, the lower encoder gets a reordered copy of $\boldsymbol{u}_t$, and a reordered copy of $\boldsymbol{v}_{t-1}^{\mathrm{U}}$ as first and second inputs, respectively. The corresponding factor graph representation of the BCC ensemble is shown in Figure 5.3 (b).

The delay block, introduced in the encoder block diagram of the BCC ensemble, makes it inherently a spatially coupled ensemble with coupling memory $m = 1$. Figure 5.4 shows the encoder block diagram of the BCC ensemble which is expanded over the time. The corresponding uncoupled BCC ensemble can

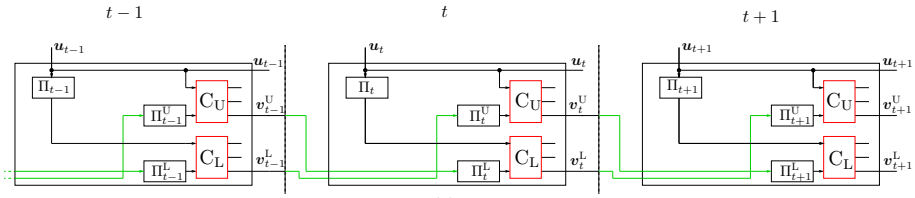


Figure 5.4: Block diagram of the BCC encoder.

be obtained by considering tail biting of a coupled chain with $L = 1$.

BCC can be seen as a class of SC-GLDPC codes with trellis constraints. Therefore, it is expected that threshold saturation also occurs for them. Investigating the impact of coupling on the BP threshold of BCCs has been done in the Paper I.

Chapter 6

Summary and Contributions

In this chapter, I summarize the main contributions and results of the included papers. I also discuss the overall conclusion of the papers and provide comments on the possible future research.

6.1 Research Contributions

In the following, the contribution of each paper is discussed separately.

6.1.1 Paper I: Density Evolution Analysis of Braided Convolutional Codes on the Erasure Channel

S. Moloudi, M. Lentmaier,

Proc. IEEE Int. Symp. Inf. Theory (ISIT), Honolulu, HI, USA, 2014.

As mentioned in the previous chapter, BCCs are a class of concatenated CCs that are inherently spatially coupled. However, it is possible to obtain the uncoupled counterpart of a BCC by considering tail-biting of a coupled chain with length $L = 1$. Knowing BCCs as the first class of spatially coupled codes with trellis constraints, we decided to start our studies on SC-TCs with them.

In this paper, based on DE analysis, we investigated the decoding thresholds of BCCs over the BEC and the impact of spatial coupling on their BP thresholds. For that, we extended the Markov chain analysis of the decoding metrics— ini-

tially developed for rate $R = 1/2$ component encoders—, to derive the explicit input/output transfer functions of the $R = 2/3$ component decoders of the considered BCC ensemble. Then, using the obtained transfer functions, we formulated the exact DE equations for block-wise coupled and uncoupled BCCs and computed the corresponding BP thresholds. We also used the area theorem to compute the MAP threshold of the considered uncoupled BCC ensemble.

Our results suggest that for the uncoupled BCC ensemble, the gap between the BP and MAP threshold is more significant than that of PCCs investigated in the literature; BCCs have better MAP threshold but worse BP threshold than PCCs. However, spatial coupling improves the BP threshold of BCCs significantly, such that that the BP threshold of the resulting coupled BCC is getting even superior to the MAP threshold of the PCCs. To compute the BP thresholds, we used a sliding window starting at time instant $t = 1$. The numerical results show that for some component encoders, the BP threshold of the BCC ensemble with sliding window decoding is worse than the BP threshold of the ensemble. This observation is the result of the fact that the error probability over the chain converges to zero from the end of the coupled chain.

For the original BCC ensemble, the coupling memory is $m = 1$. Therefore, we have not observed the threshold saturation phenomenon in this paper. However, this has been our motivation to propose and investigate some modified BCC ensembles with larger coupling memories in later papers.

6.1.2 Paper II: Spatially Coupled Turbo Codes

S. Moloudi, M. Lentmaier, and A. Graell i Amat,
Proc. Int. Symp. on Turbo Codes and Iterative Inf. Processing (ISTC), Bremen, Germany, 2014.

In this paper, we proposed SC-TCs as a class of spatially coupled codes and studied them by a systematic approach. In particular, we proposed some block-wise spatially coupled counterparts for PCCs and SCCs with general coupling memory m and investigated the impact of coupling on their BP thresholds. For the proposed coupled ensembles, we derived the exact DE equations over the BEC. Then, using these equations, we obtained the corresponding BP thresholds by numerical methods. The computed BP thresholds were compared with the corresponding MAP thresholds computed by the area theorem. Based on the results obtained in this paper, we observed that for both SC-PCCs and SC-SCCs, the coupling BP threshold approaches the MAP threshold of the underlying ensemble by increasing coupling memory. In other words, our nu-

merical results suggest the occurrence of threshold saturation for the proposed coupled ensembles. To compare the thresholds of SC-PCC and SC-SCC ensembles, some ensembles with random puncturing were also considered. The reported thresholds for two rates, $R = 1/2$ and $R = 1/3$, confirm that for equal rate, SCCs have worse BP but better MAP thresholds than PCCs; by spatial coupling, the improvement of the BP threshold is more considerable for the SCC ensemble. Therefore, the BP threshold of the SC-SCC ensemble surpasses that of the SC-PCC ensemble.

So far, PCCs have been more attractive for many applications than SCCs because of their better BP thresholds. The results of this paper demonstrate that SC-SCCs can easily compete with SC-PCCs regarding decoding threshold.

6.1.3 Paper III: Spatially Coupled Turbo-Like Codes

S. Moloudi, M. Lentmaier, and A. Graell i Amat,

IEEE Trans. Inf. Theory, vol. 63, no. 10, pp. 6199-6215, Oct. 2017.

In this paper, we followed two main purposes: First, we investigated the issues raised in the first two papers—investigating the impact of coupling on the TC ensembles—in more detail; second, we provided a mathematical proof for threshold saturation for the SC-TCs.

In the very beginning of the paper, we introduced a graphical representation for convolutional codes that can be considered as a compact form of the corresponding factor graph. This new graph representation not only simplifies the graph representation of TCs and SC-TCs but also makes the DE analysis of these codes easier. Using the compact graph representation, besides describing the proposed coupled ensembles for SC-PCC, and SC-SCC with general coupling memory m, we suggested two extensions of BCC ensembles—which are inherently spatially coupled with $m = 1$—to higher coupling memories $m > 1$. We also assumed random puncturing and performed a DE analysis to compute the BP thresholds of the considered ensembles over the BEC for a wide range of rates. Similar to the first two papers, we used the area theorem to compute the corresponding MAP thresholds of the punctured ensembles.

Our numerical results demonstrate that spatial coupling improves the decoding thresholds of TCs for all rates. This improvement is more considerable for SCCs and BCCs whose uncoupled ensembles suffer from poor BP thresholds. The uncoupled BCC ensemble has the worst BP threshold but the best MAP threshold among the considered TC ensembles, and by spatial coupling, the BP threshold of the BCC ensemble improves to a value close to the corresponding close-to-capacity MAP threshold. In general, our numerical results show

that an SC-TC ensemble can achieve the MAP threshold of the underlying uncoupled ensemble for large enough coupling memory, i.e., threshold saturation happens for SC-TCs.

Motivated by this observation, we proved threshold saturation for SC-TCs analytically. In general, the DE equations of TCs fall within a vector recursion format, but we showed that under certain conditions, it is possible to rewrite the DE equations of the TC ensembles as scalar recursions. This allowed us to use a proof technique based on the potential function developed by Yedla et al.. For the case of PCCs, considering the proof technique for vector recursions, we also generalized the proof to non-symmetric ensembles with different component encoders.

The results from this paper confirm that TCs, like LDPC codes, can benefit from spatial coupling. So far, TCs have been optimized to have better thresholds for BP decoding. Threshold saturation guarantees that by spatial coupling, the MAP threshold of a TC ensemble is achievable with low complexity BP decoding. This suggests optimizing TC ensembles for better MAP threshold. By spatial coupling of the powerful code ensembles with strong distance properties such as SCCs and BCCs, the resulting ensembles can then perform close to capacity with low-complexity BP decoding.

6.1.4 Paper IV: Spatially Coupled Hybrid Concatenated Codes

S. Moloudi, M. Lentmaier, and A. Graell i Amat,

Proc. Int. ITG Conf. Systems, Commun. and Coding (SCC), Hamburg, Germany, 2017.

Any mixture of PCC and SCC ensembles can be considered as an HCC ensemble. This fact makes HCC ensembles more flexible for the trade-offs in the code design. Depending on the structure of the ensemble, an HCC ensemble can have trade-offs between close-to-capacity threshold and very low error probabilities in the error floor region. Moreover, for some HCC ensembles, the minimum distance grows linearly with input block length. All these make HCCs a powerful class of TCs.

In this paper, we tried to make further investigation on SC-TCs by studying the impact of spatial coupling on the thresholds of HCCs. In particular, as an HCC ensemble, we considered a serially concatenation of a PCC ensemble with an inner encoder; this ensemble is very close to the 3-D turbo code ensemble. Then, we introduced two spatially coupled ensembles of HCCs, referred to as Type-I SC-HCCs and Type-II SC-HCCs. Considering different component

encoders, we computed the thresholds of SC-HCC ensembles and compared them with the thresholds of BCCs for a range of different rates.

The numerical results show that the MAP threshold of HCCs can be even better than that of BCCs, and it is achievable by the threshold saturation effect of coupling. Moreover, the results confirm that, by selecting the component encoders suitably, HCCs can be either optimized for BP threshold or MAP threshold. Although HCCs can have the best MAP threshold among the considered TC ensembles, for a given coupling memory m, their spatially coupled counterparts do not have necessarily the best coupling threshold. The reported threshold of Table IV confirm this fact; for small values of m, BCCs have better coupled threshold than corresponding HCCs.

6.1.5 Paper V: Spatially coupled turbo-like codes: a new trade-off between waterfall and error floor

S. Moloudi, M. Lentmaier, and A. Graell i Amat,
submitted to *IEEE Trans. Commun.*.

In the previous articles, we showed that TCs have excellent MAP thresholds, and relying on threshold saturation, these thresholds are achievable by spatial coupling. Motivated by these excellent decoding thresholds, in this paper, we discussed the impact of spatial coupling on the performance of TCs in the finite block-length regime. We investigated the performance of the SC-TCs in both the waterfall and error floor region over the AWGN channel. To discuss the waterfall performance, we used the computed decoding thresholds for the BEC to predict the corresponding decoding thresholds for the AWGN channel. We also provided simulation results for TCs and SC-TCs. The simulation results confirm that spatial coupling significantly improves the BER performance of TCs in the waterfall region, like the decoding thresholds.

Then, we investigated the impact of coupling on the error-floor performance of TCs. For that, we established conditions under which spatial coupling either preserves or improves the minimum distance of TCs. The provided conditions can be seen as a guideline for unwrapping the TC ensembles. To investigate the minimum distance and error floor of SC-TCs, it is possible to perform a WEF analysis for the coupled ensembles. However, by increasing the coupling length and the coupling memory the complexity of computing the average WEF of the coupled ensemble increases significantly. Instead, relying on the established connection, we simply performed the WEF analysis of the corresponding uncoupled ensembles to investigate the minimum distance and error floor of the coupled ensembles. Using the WEFs for a given block length, we computed

bounds on the error performance, and bounds on the minimum distance of TCs. These bounds indicate very low error floor for the SCC, BCC, and HCC ensembles. Moreover, for the BCC and HCC ensembles, the minimum distance grows linearly with the block length.

So far, among all the considered TC ensembles, only PCCs have been used in various standards because of their good BP thresholds. However, other TC ensembles have better MAP threshold and distance properties than PCCs. The results from this paper and all previous papers confirm that the BP thresholds of these ensembles can be significantly improved by applying coupling. Regarding the finite length regime, while their error floor stays at very low error probabilities, their waterfall performance gets much closer to capacity. Interestingly, it can be seen that by coupling of a TC ensemble with close to capacity MAP threshold and low error floor, such as SCCs, BCCs, and HCCs, the resulting spatially coupled ensemble is very promising and can perform close-to-capacity, yet achieving low error floor, with a low complexity BP decoder.

6.1.6 Paper VI: A Unified Ensemble of Concatenated Convolutional Codes

S. Moloudi, M. Lentmaier, and A. Graell i Amat,

Proc. IEEE Int. Symp. Inf. Theory (ISIT), Aachen, Germany, 2017.

So far, the different classes of TCs have been considered separately. In this article, we proposed a unified ensemble that contains all four major classes of TCs; PCCS, SCCs, HCCs, and BCCs. We believe this unified ensemble can unify the frameworks for the analysis of TCs, and moreover, can lead us to a better understanding of the similarities and differences between various TC classes and the possible trade-offs in the code design. This ensemble also allows us to design new ensembles that do not belong to any of the original classes of TCs. By proper selection of the design parameters in the unified ensemble, it is possible to find equivalent ensembles for different TC classes. This ensemble is based on a single self-concatenated trellis. In particular, we used a single rate-1/2 component encoder, but 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. In this paper, we introduced two elementary steps to find the self-concatenated equivalents of PCCs and SCCs. Then, we used these elementary steps to find the self-concatenated equivalents of HCCs and BCCs. It is worthy to mention that these elementary steps can also be applied to other concatenated CCs to find the corresponding self-concatenated codes. Then, we derived the exact DE equations for this unified ensemble over the

BEC. Using these DE equations, we computed the decoding thresholds of the equivalent TC ensembles. Our results confirm that the obtained thresholds are very close to those of the original ensembles.

We believe that the proposed unified ensemble also establishes a bridge between TCs and protograph-based GLDPC codes with convolutional constraints. In fact, this unified ensemble can be considered as the first step toward understanding of the connections between TC ensembles and LDPC code ensembles.

6.2 General Conclusions

By selecting a TC ensemble and its component convolutional encoders properly, one can optimize the ensemble for higher BP or MAP thresholds. Among all the concatenated ensembles, only the PCC ensemble have had the opportunity to be used in various standards because of their good BP thresholds and their performance in the waterfall region. However, only for properly designed permutations a PCC can achieve low error rates in the error floor region. The other TC ensembles —SCCs, BCCs, and HCCs— have better MAP thresholds and distance properties than PCCs, but because of their poor BP thresholds, they have got less commercial attention.

The main purpose of this thesis was to propose, investigate and understand a powerful class of codes on graphs with trellis constraints, which can achieve both close-to-capacity performance and very low error rates for moderate block lengths. In particular, we have proposed spatial coupling for concatenated convolutional codes (SC-TCs). Then, we performed DE analysis to investigate the decoding thresholds of the TCs and the impact of coupling on their thresholds.

Our results confirm excellent decoding thresholds for TC ensembles. These results also show that the ensemble with the better BP threshold does not have necessarily the better MAP threshold. More precisely, SCCs, BCCs, and HCCs have excellent MAP thresholds but worse BP thresholds than the PCC ensemble. We have proved that by spatial coupling of TCs the resulting ensembles can achieve the MAP threshold of the underlying TC ensemble (threshold saturation). This brings a new perspective in designing concatenated codes: a TC scheme can be optimized for the best MAP threshold, and relying on threshold saturation, this MAP threshold can be achieved by BP decoder.

Based on the WEFs, we also computed bounds on the error rate performance and the minimum distance of TC ensembles. These bounds show very low error floor for the SCC, BCC and HCC ensembles. Also, the bounds on the minimum distance of BCCs and HCCs, demonstrate that their minimum distance grows linearly with block length. We proved that under certain conditions SC-TCs can have either equal or better minimum distance than the underlying TC ensemble (preserving the minimum distance).

The DE and WEF analysis demonstrates that the TC ensembles with better MAP thresholds —SCCs, BCCs, and HCCs— also have better minimum distance and lower error floor. Then, relying on threshold saturation and preserving the minimum distance, by spatial coupling of these ensembles, the resulting ensembles perform close-to-capacity, yet achieving low error floor, with a low complexity BP decoder.

So far, the different classes of TCs have been studied and investigated separately. However, its not completely clear how the connection of the component encoders affects the performance of the TCs. We proposed an unified ensembles for the considered TC ensembles which is based on self concatenation of a convolutional code. The graph representation of this ensemble confirms that the differences between the TC ensembles are manifested in the proportion of degree-1, and degree-2 VNs in the graph, or the puncturing of part of the parity sequence. We believe that this unified ensemble establishes a connection between concatenated code ensembles and LDPC code ensembles.

6.3 Future Research

This thesis can be seen as one of the beginning steps toward understanding SC-TCs. Therefore, there are many potentially interesting topics for future research on SC-TCs. From them I can mention:

- Extending our threshold analysis to the AWGN channel by using Monte Carlo methods.
- Investigating the impact of increasing memory of component encoders of SC-TCs or increasing the degree of VNs in the graph.
- Obtaining bounds on the performance of the BP decoder, by investigating the corresponding absorbing sets, trapping sets and pseudo-codewords.
- Analyzing the benefits of SC-TCs in practical scenarios.

Finally, I believe that my studies on SC-TCs may attract some new interest in turbo-like coding structures for a wider range of applications.

References

[1] S. Lin and D. Costello, *Error-correcting codes*. Prentice-Hall, Inc, 1983.

[2] W. Ryan and S. Lin, *Channel codes: Classical and modern*. Cambridge University Press, 2009.

[3] C. E. Shannon, "A mathematical theory of communication", *The Bell System Technical Journal*, vol. 27, no. 4, pp. 623–656, Oct. 1948.

[4] D. J. Costello and G. D. Forney, "Channel coding: The road to channel capacity", *Proceedings of the IEEE*, vol. 95, no. 6, pp. 1150–1177, 2007.

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

[6] C. Berrou and A. Glavieux, "Near optimum error correcting coding and decoding: Turbo-codes", *IEEE Trans. Commun.*, vol. 44, no. 10, pp. 1261–1271, 1996.

[7] R. Gallager, *Low-density parity-check codes*. Cambridge, MA: MIT Press, 1963.

[8] D. MacKay, "Good error-correcting codes based on very sparse matrices", *IEEE Trans. Inf. Theory*, vol. 45, no. 2, pp. 399–431, Mar. 1999.

[9] P. Elias, "Error free coding", *IRE Trans. Inf. Theory*, vol. PGIT-4, pp. 29–37, Sep. 1954.

[10] D. Divsalar, H. Jin and R. J. McEliece, "Coding theorems for turbo-like codes", in *Proc. of the Annual Allerton Conf. on Commun. Control and Computing*, Illinois, USA, 1998.

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

[12] M. Lentmaier, A. Sridharan, D. J. Costello and K. S. Zigangirov, "Iterative decoding threshold analysis for LDPC convolutional codes", *IEEE Trans. Inf. Theory*, vol. 56, no. 10, pp. 5274–5289, Oct. 2010.

[13] M. Lentmaier, A. Sridharan, K. S. Zigangirov and D. J. Costello Jr., "Terminated LDPC convolutional codes with thresholds close to capacity", in *Proc. IEEE Int. Symp. on Inf. Theory*, Adelaide, Australia, Sep. 2005.

[14] S. Kudekar, T. Richardson and R. 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.

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

[16] A. Yedla, Y. Y. Jian, P. S. Nguyen and H. D. Pfister, "A simple proof of threshold saturation for coupled scalar recursions", in *Proc. Int. Symp. Turbo Codes and Iterative Inf. Process. (ISTC)*, Gothenburg, Sweden, Aug. 2012.

[17] A. Yedla, Y. Y. Jian, P. S. Nguyen and H. D. Pfister, "A simple proof of threshold saturation for coupled vector recursions", in *Proc. IEEE Inf. Theory Work. (ITW)*, Lausanne, Switzerland, Sep. 2012.

[18] 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.

[19] R. W. Hamming, "Error detecting and error correcting codes", *Bell Labs Tech. J.*, vol. 29, no. 2, pp. 147–160, 1950.

[20] I. S. Reed and G. Solomon, "Polynomial codes over certain finite fields", *J. of the Soc. for Ind. and Appl. Math.*, vol. 8, no. 2, pp. 300–304, 1960.

[21] R. Johannesson and K. S. Zigangirov, *Fundamentals of convolutional coding*. John Wiley & Sons, 2015.

[22] A. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm", *IEEE Trans. Inf. Theory*, vol. 13, no. 2, pp. 260–269, Apr. 1967.

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

[24] R. Tanner, "A recursive approach to low complexity codes", *IEEE Trans. Inf. Theory*, vol. 27, no. 5, pp. 533–547, Sep. 1981.

[25] G. D. Forney, "Codes on graphs: Normal realizations", *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 520–548, Feb. 2001.

[26] J. Thorpe, "Low-density parity-check (LDPC) codes constructed from protographs", in *IPN Progress Report*, vol. 42, Agu. 2003, pp. 42–154.

[27] N. Wiberg, "Codes and decoding on general graphs", PhD thesis, Linköping University, 1996.

[28] N. Wiberg, H. A. Loeliger and R. Kotter, "Codes and iterative decoding on general graphs", *Trans. on Emerging Telecommun. Tech.*, vol. 6, no. 5, pp. 513–525, 1995.

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

[30] S. Benedetto, G. Montorsi, D. Divsalar and F. Pollara, "Serial concatenation of interleaved codes: Performance analysisdesign, and iterative decoding", *Telecommun. and Data Acquisition Progress Report*, vol. 126, pp. 1–26, 1996.

[31] 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.

[32] D. Divsalar and F. Pollara, "Hybrid concatenated codes and iterative decoding", in *Proc. IEEE Int. Symp. on Inf. Theory*, Jun. 1997.

[33] E. Rosnes and A. Graell i Amat, "Performance analysis of 3-D turbo codes", *IEEE Trans. Inf. Theory*, vol. 57, no. 6, pp. 3707–3720, Jun. 2011.

[34] C. Koller, A. Graell i Amat, J. Kliewer, F. Vatta and D. J. Costello, "Hybrid concatenated codes with asymptotically good distance growth", in *Proc. Int. Symp. Turbo Codes and Related Topics (ISTC)*, Sep. 2008.

[35] C. Berrou, A. Graell i Amat, Y. Ould-Cheikh-Mouhamedou and Y. Saouter, "Improving the distance properties of turbo codes using a third component code: 3D turbo codes", *IEEE Trans. Commun.*, vol. 57, no. 9, pp. 2505–2509, Sep. 2009.

[36] B. Kurkoski, P. Siegel and J. Wolf, "Exact probability of erasure and a decoding algorithm for convolutional codes on the binary erasure channel", in *Proc. IEEE Global Telecommun. Conf. (GLOBECOM),*, 2003.

[37] J. Shi and S. ten Brink, "Exact EXIT functions for convolutional codes over the binary erasure channel", in *Proc. Allerton Conf. Commun., Control, and Computing*, Monticello, IL, USA, 2006.

[38] A. Ashikhmin, G. Kramer and S. ten Brink, "Extrinsic information transfer functions: Model and erasure channel properties", *IEEE Trans. Inf. Theory*, vol. 50, no. 11, pp. 2657–2673, Nov. 2004.

[39] C. Méasson, A. Montanari and R. Urbanke, "Maxwell construction: The hidden bridge between iterative and maximum a posteriori decoding", *IEEE Trans. Inf. Theory*, vol. 54, no. 12, pp. 5277–5307, Dec. 2008.

[40] C. Méasson, "Conservation laws for coding", PhD thesis, École Polytechnique Fédérale de Lausanne, 2006.

[41] C. Measson, A. Montanari, T. Richardson and R. Urbanke, "The generalized area theorem and some of its consequences", *IEEE Trans. Inf. Theory*, vol. 55, no. 11, pp. 4793–4821, Nov. 2009.

[42] S. Benedetto and G. Montorsi, "Unveiling turbo codes: Some results on parallel concatenated coding schemes", *IEEE Trans. Inf. Theory*, vol. 42, no. 2, pp. 409–428, Mar. 1996.

[43] D. G. M. Mitchell, L. Dolecek and D. J. Costello, "Absorbing set characterization of array-based spatially coupled LDPC codes", in *IEEE Int. Symp. Inf. Theory (ISIT)*, Honolulu, HI, USA, Jun. 2014.

[44] E. Rosnes, M. Helmling and A. Graell i Amat, "Minimum pseudoweight analysis of 3-dimensional turbo codes", *IEEE Trans. Commun.*, vol. 62, no. 7, pp. 2170–2182, Jul. 2014.

[45] D. J. Costello, L. Dolecek, T. E. Fuja, J. Kliewer, D. G. M. Mitchell and R. Smarandache, "Spatially coupled sparse codes on graphs: Theory and practice", *IEEE Commun. Mag.*, vol. 52, no. 7, pp. 168–176, Jul. 2014.

[46] D. G. Mitchell, M. Lentmaier and D. J. Costello, "Spatially coupled LDPC codes constructed from protographs", *IEEE Trans. Inf. Theory*, vol. 61, no. 9, pp. 4866–4889, Sep. 2015.

[47] S. Kudekar, T. J. Richardson and R. L. Urbanke, "Wave-like solutions of general 1-d spatially coupled systems", *IEEE Trans. Inf. Theory*, vol. 61, no. 8, pp. 4117–4157, Aug. 2015.

[48] I. Andriyanova and A. Graell i Amat, "Threshold saturation for non-binary sc-ldpc codes on the binary erasure channel", *IEEE Trans. Inf. Theory*, vol. 62, no. 5, pp. 2622–2638, May 2016.

[49] N. ul Hassan, A. E. Pusane, M. Lentmaier, G. P. Fettweis and D. J. Costello, "Non-uniform windowed decoding schedules for spatially coupled codes", in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2013.

[50] N. U. Hassan, A. E. Pusane, M. Lentmaier, G. P. Fettweis and D. J. Costello, "Non-uniform window decoding schedules for spatially coupled LDPC codes", *IEEE Trans. Commun.*, vol. 65, no. 2, pp. 501–510, Feb. 2017.

[51] M. Zhu, D. G. M. Mitchell, M. Lentmaier, D. J. Costello and B. Bai, "Window decoding of braided convolutional codes", in *Proc. Inf. Theory Work. (ITW)*, Jeju, South Korea, Oct. 2015.

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

[53] A. Feltström, D. Truhachev, M. Lentmaier and K. Zigangirov, "Braided block codes", *IEEE Trans. Inf. Theory*, vol. 55, no. 6, pp. 2640–2658, Jun. 2009.

[54] D. Truhachev, M. Lentmaier and K. Zigangirov, "On braided block codes", in *Proc. IEEE Int. Symp. on Inf. Theory (ISIT)*, Jun. 2003.

[55] B. Smith, A. Farhood, A. Hunt, F. Kschischang and J. Lodge, "Staircase codes: FEC for 100 Gb/s OTN", *J. Lightw. Technol.*, vol. 30, no. 1, pp. 110–117, Jan. 2012.

[56] Y. Jian, H. Pfister, K. Narayanan, R. Rao and R. Mazahreh, "Iterative hard-decision decoding of braided BCH codes for high-speed optical communication", in *Proc. IEEE Global Telecommun. Conf. (GLOBECOM)*, Atlanta, GA, USA, Dec. 2013.

[57] J. Justesen, K. Larsen and L. Pedersen, "Error correcting coding for OTN", *IEEE Commun. Mag.*, vol. 48, no. 9, pp. 70–75, Sep. 2010.

Part II

Included Papers

Paper I

Density Evolution Analysis of Braided Convolutional Codes on the Erasure Channel

Braided convolutional codes (BCCs) are a class of spatially coupled turbo-like codes with a structure that is similar to product codes or generalized LDPC codes. We derive explicit input/output transfer functions of the component convolutional decoders for the binary erasure channel (BEC). These are then used to formulate exact density evolution equations for blockwise BCCs under belief propagation (BP) decoding with optimal component APP decoders. Thresholds are computed for the coupled and uncoupled case, which is equivalent to tailbiting. Due to the relatively high rate of the component codes a significant threshold improvement by spatial coupling can be observed.

S. Moloudi, M. Lentmaier,
"Density evolution analysis of braided convolutional codes on the erasure channel,"
in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Honolulu, HI, USA, 2014.

1 Introduction

Braided block codes (BBCs) [1] are a class of generalized low-density parity-check (LDPC) convolutional codes that can be viewed as a spatially coupled version of Elias' product codes [2]. Similar to LDPC codes, sparsity can be introduced into their structure, without changing the component codes, in order to construct codes of arbitrary length or memory. BBCs with BCH component codes were recently considered for high-speed optical communications in [3], where product-like codes are commonly applied [4]. Like the closely related staircase codes [5], they show an excellent performance together with low-complexity iterative hard decision decoding.

In this paper we consider a counterpart of BBCs called braided convolutional codes (BCCs) [6]. Again the encoding can be described by a two-dimensional sliding array in which each symbol is protected by a horizontal and a vertical component code. But now the component codes are convolutional codes, resulting in a class of spatially coupled turbo-like codes with a structure similar to generalized LDPC codes. Unlike parallel or serially concatenated convolutional codes all information and parity symbols are protected by both component codes in a symmetric fashion.

For a random ensemble of BCCs with Markov permutors it was shown in [6] that the minimum distance of typical codes grows linearly with their constraint length, i.e., BCCs are asymptotically good. Although a formal proof is still open, it is expected that this is also the case for the slightly different blockwise BCC construction that we consider here. Indeed, the simulation results for blockwise BCCs in [6, Figure 12] indicate superior distance properties compared to parallel concatenated codes since no error floor is visible for comparable permutor sizes. At the same time, unlike serial concatenated codes, the BCCs can compete with the parallel concatenation in the waterfall region. Interestingly, the simulated codes in [6, Figure 12] performed significantly better than the tailbiting case in [6, Figure 13]. For the latter case the AWGN channel threshold was estimated by Monte Carlo techniques. It was conjectured that this performance improvement can be prescribed to a similar effect as the threshold saturation phenomenon known for coupled LDPC codes [7].

The aim of this paper is to confirm this conjecture by performing a threshold analysis of blockwise BCCs. After introducing BCCs and their decoding in Section II and Section III, we derive explicit input/output transfer functions that characterize the *a posteriori* probability (APP) decoders of their component codes in Section IV. Considering the binary erasure channel (BEC), these transfer functions can be computed analytically by means of a Markov chain analysis of the decoder metrics, as presented in [8] for rate $R = 1/2$ encoders. We apply the technique from [8] to rate $R = 2/3$ encoders with different input

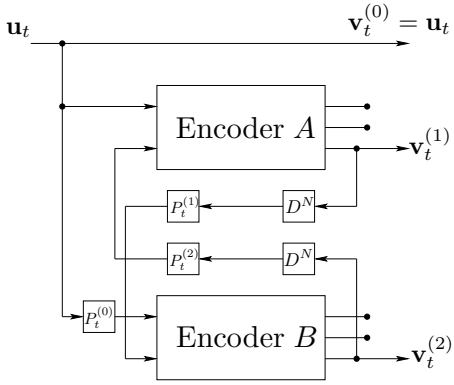


Figure 1: Encoder of BCC.

and output probabilities for each symbol type, resulting in a three-dimensional transfer function for each of the output symbols. These transfer functions are then used in Section V to formulate exact density evolution recursions for the blockwise BCCs and compute belief propagation (BP) thresholds for the coupled and tailbiting (or uncoupled) case. Due to the higher rate of the component codes the tailbiting/uncoupled threshold is worse than the thresholds of typical parallel concatenated codes. However, as expected, the coupled ensemble has a significantly better threshold.

2 Braided Convolutional Codes

Similar to turbo codes, BCCs have convolutional codes as component codes but the most important difference between turbo codes and braided codes is that, in BCCs, the parity symbols of one component encoder are used as future inputs of the other component encoder. Throughout this paper we limit ourselves to the example of rate $R = 1/3$ blockwise BCCs as illustrated in Figure1. They consist of two systematic convolutional component encoders of rate $R = 2/3$. At time t, a block of N information symbols $\boldsymbol{u}_t$ and a block of N parity symbols $\boldsymbol{v}_{t-1}^{(2)}$ (there is a delay D^N of one block) enter Encoder A directly and through the block permutor $P^{(2)}$, respectively. Encoder B has permuted information symbols through block permutor $P^{(0)}$ and permuted parity symbols $\boldsymbol{v}_{t-1}^{(1)}$ from Encoder A through block permutor $P^{(1)}$ and delay block, as inputs. The output of the encoder at time t is $\boldsymbol{v}_t = (\boldsymbol{v}_t^{(0)}, \boldsymbol{v}_t^{(1)}, \boldsymbol{v}_t^{(2)})$, where $\boldsymbol{v}_t^{(0)} = \boldsymbol{u}_t$. It follows from the encoding procedure that BCCs are a class of spatially coupled codes because the encoded blocks $\boldsymbol{v}_t$ depend on blocks from previous time instants.

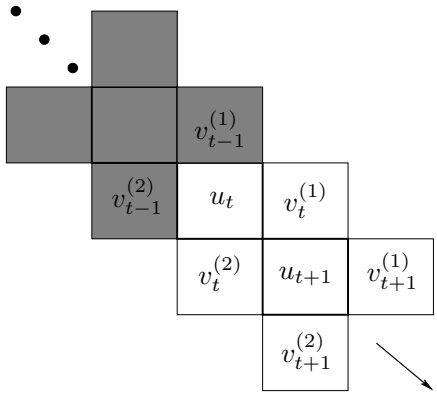


Figure 2: Array representation of TBC codes.

An uncoupled braided code can be defined by omitting the delay blocks. It is also possible to use block length $N = 1$ and not to use the permutors. In this case the codes are called tightly braided convolutional codes (TBCCs).

BCCs are closely related to classic product codes, in which the data is written in an infinite two-dimentional array and the rows and columns are encoded by separate component codes. Moreover, the horizontal and vertical encoders are linked for BCCs through parity feedback. The array of a TBCC is illustrated in Figure 2. It consists of three diagonal ribbons and the information symbols are placed in the center ribbon. The parity symbols of the horizontal and vertical encoder are stored in the upper and lower ribbons, respectively. At time t, the output of the horizontal encoder, which is shown by $v_t^{(1)}$, depends on the current information symbol u_t, its left neighbor $v_{t-1}^{(2)}$ and the encoder state. $v_t^{(1)}$ will be placed as the right neighbor of u_t in the array. Shaded squares of the array contain the previous inputs and outputs, which are assumed to be known. The operation of the vertical encoder is analogous to the horizontal one. Finally, at time t, the coded symbols $\boldsymbol{v}_t = (u_t, v_t^{(1)}, v_t^{(2)})$ are sent over the channel.

Throughout this paper we consider transmission of a sequence of L coupled blocks $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_L$, and distinguish between encoders with termination or tailbiting. In the first case the encoder is terminated and the blocks at times $t < 1$ and $t > L$ are equal to $\boldsymbol{v}_t = 0$. In the second case we have a circular structure. Uncoupled BCCs can be defined by using tailbiting with $L = 1$.

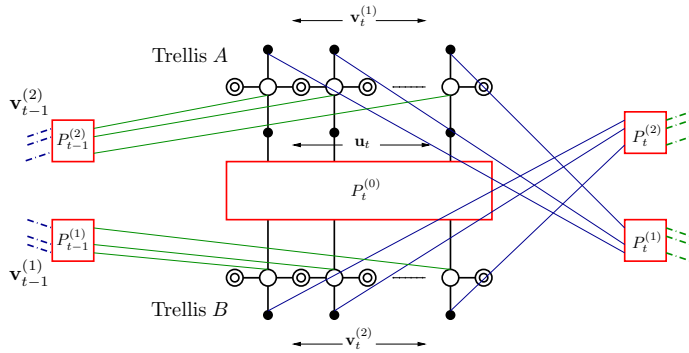


Figure 3: Factor graph of a blockwise BCC at time t.

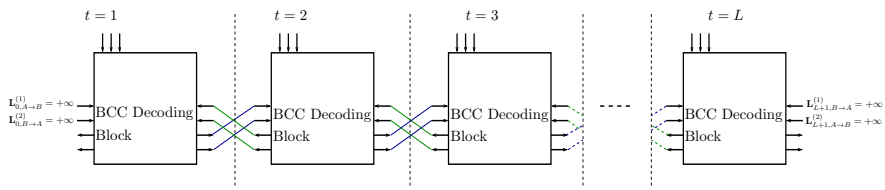


Figure 4: Block diagram of the iterative message passing decoder of a blockwise BCC.

3 Iterative Decoding

A factor graph representation of a blockwise BCC is shown in Figure 3. We consider BP decoding, i.e., an iterative message passing decoder in which the trellises of the component codes are decoded by the BCJR algorithm. In every iteration each decoding block at time t, $t = 1, \ldots, L$ receives log-likelihood ratios (LLRs) from the channel and the decoders at the same time t and the neighboring blocks at time $t-1$ and $t+1$, resulting from previous iteration. Figure 4 shows the connection of the decoders at different time instants and how they exchange LLRs between time slots for coupled BCCs. Note that we omit the permutations in the block diagram in order to simplify the illustration. LLRs coming from time $t < 1$ and $t > L$ are set to $+\infty$, since the corresponding symbols are equal to zero by definition.

Based on the input values $\boldsymbol{L}_{\text{in},t}^{(k)}$, $k = 0, 1, 2$, the BCJR decoders create new extrinsic output values $\boldsymbol{L}_{\text{out},t}^{(k)}$ that are passed back to the other BCJR decoder in the same and neighboring time instants. An illustration of the decoding block at time t is given in Figure 5. Here $\boldsymbol{L}_{ch,t}^{(k)}$ denotes the channel

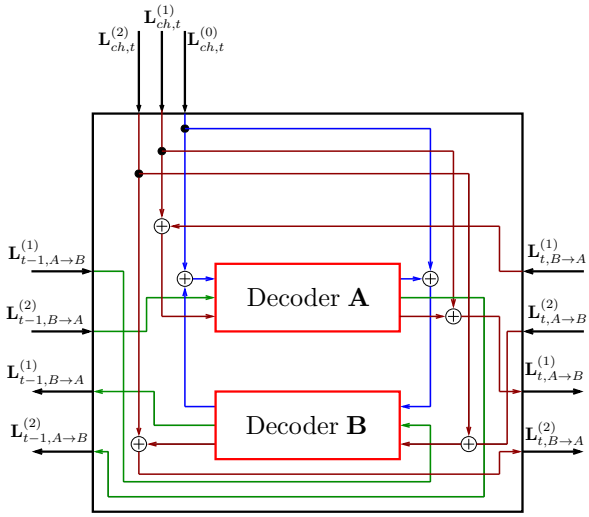


Figure 5: BCC Decoding Block at time t.

LLRs of the kth symbol block $\boldsymbol{v}_t^{(k)}$ and $\boldsymbol{L}^{(k)}_{t,A\to B}$ defines the extrinsic outputs $\boldsymbol{L}^{(k)}_{\text{out},t}$ of Decoder A, which contributes to the input of Decoder B. $\boldsymbol{L}^{(k)}_{t,B\to A}$ is defined analogously. In the first decoding iteration the values from the previous iteration are initialized as erasures, which means that the respective LLRs are defined as zero.

In the BCC decoding block of Figure 5, green lines show the LLRs that are exchanged with the past, and LLRs exchanged with the future are illustrated in brown lines. Some LLRs are produced at the current time instant and only used at the current time, namely those which are related to the first inputs of the component decoders. Blue lines show this kind of LLRs.

As the considered channel is the BEC, the LLRs from the channel and from the outputs of the decoders can only be one of the values $+\infty$, 0 and $-\infty$ for 0, erasure and 1 and the only combinations that are possible to happen are as follows:

$$+\infty + \infty = +\infty \quad 0 + \infty = +\infty$$
$$-\infty - \infty = -\infty \quad 0 - \infty = -\infty$$

4 Probability of Bit Erasure for Component Convolutional Codes

In order to derive an analytical expression for the probability of erasure of BCCs, the probabilities of erasure of the component codes are required. To catch this goal, we use the method proposed in [8] and apply it to convolutional codes with rate $R = 2/3$.

The extrinsic output erasure probabilities are functions of the input erasure probabilities p_0, p_1 and p_2,

$$\begin{aligned} p_{e,0} &= f_0(p_0, p_1, p_2) \\ p_{e,1} &= f_1(p_0, p_1, p_2) \\ p_{e,2} &= f_2(p_0, p_1, p_2) \end{aligned} \tag{1}$$

For each component code, there is a BCJR decoder and its lth input at trellis section n is denoted by $L^{(l)}_{\text{in},n}$.[1]

Moving over the trellis, forward and backward state metric values are obtained from the following equations[2]:

$$\alpha_n(\sigma) = \max^*_{\sigma'}(\gamma_n(\sigma', \sigma) + \alpha_{n-1}(\sigma')) \tag{2}$$

$$\beta_{n-1}(\sigma') = \max^*_{\sigma}(\gamma_n(\sigma', \sigma) + \beta_n(\sigma)) \tag{3}$$

where σ and σ' denote the states at time n and $n-1$, respectively, and

$$\gamma_n(\sigma', \sigma) = \sum_{l=1}^{3} L^{(l)}_{\text{in},n} \cdot \left(\frac{1}{2} - v^{(l)}_n\right)$$

Finally, the lth output (extrinsic information) can be calculated as

$$\begin{aligned} L^{(l)}_{\text{out},n} = &\max^*_{(\sigma',\sigma):v^{(l)}_n=0}\big(\alpha_{n-1}(\sigma') + \gamma_n(\sigma', \sigma) + \beta_n(\sigma)\big) \\ &- \max^*_{(\sigma',\sigma):v^{(l)}_n=1}\big(\alpha_{n-1}(\sigma') + \gamma_n(\sigma', \sigma) + \beta_n(\sigma)\big) \end{aligned}$$

We define the metric vectors $\boldsymbol{\mu}_{\alpha,n}$ and $\boldsymbol{\mu}_{\beta,n}$, whose length is equal to the number of the states. $\boldsymbol{\mu}_{\alpha,n}(i)$ is the forward metric of ith state at time n and $\boldsymbol{\mu}_{\beta,n}$ has the backward metric of the ith state as the ith element. Since for

[1] We use l to denote the symbol index from the perspective of the component decoder, whereas k denotes the index from the perspective of the overall code. Likewise, t is the time index of the code sequence $\boldsymbol{v}_t$, while n denotes the index within the trellis of a component decoder at a given time instant t.

[2] max* denotes the Jacobian logarithm.

both $\boldsymbol{\mu}_{\alpha,n}$ and $\boldsymbol{\mu}_{\beta,n}$ nonzero elements are always equal, we can normalize these entires to 1.

Due to the linearity of the code, we can assume in the analysis that the all-zero codeword has been transmitted. Let $\mathcal{M}_\alpha = \{\boldsymbol{\sigma}_\alpha^{(1)}, \boldsymbol{\sigma}_\alpha^{(2)}, \dots \boldsymbol{\sigma}_\alpha^{(|\mathcal{M}_\alpha|)}\}$ and $\mathcal{M}_\beta$ denote the set of all possible $\boldsymbol{\mu}_{\alpha,n}$ and $\boldsymbol{\mu}_{\beta,n}$, respectively. $\boldsymbol{\mu}_{\alpha,n}$ is one of the elements of $\mathcal{M}_\alpha$.

Example 1 *Consider the rate $R = 2/3$ convolutional code with generator matrix*

$$\boldsymbol{G}(D) = \begin{pmatrix} 1 & 0 & \frac{1}{1+D+D^2} \\ 0 & 1 & \frac{1+D^2}{1+D+D^2} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 1/7 \\ 0 & 1 & 5/7 \end{pmatrix} \tag{4}$$

For this code, using observer canonical form, there are four states and in this case, $\mathcal{M}_\alpha$ and $\mathcal{M}_\beta$ are equal and have finite number of elements.

$$\mathcal{M}_\alpha = \mathcal{M}_\beta =$$

$$\{(1,0,0,0), (1,1,0,0), (1,0,0,1), (1,0,1,0), (1,1,1,1)\}$$

The sequence $\dots, \boldsymbol{\mu}_{\alpha,n-1}, \boldsymbol{\mu}_{\alpha,n}, \boldsymbol{\mu}_{\alpha,n+1}, \dots$ forms a Markov chain with transition matrix $\boldsymbol{M}_\alpha$, in which $\boldsymbol{M}_\alpha(j,k)$ is the probability of coming from state $\boldsymbol{\sigma}_\alpha^{(j)}$ to sate $\boldsymbol{\sigma}_\alpha^{(k)}$. This probability depends on the input erasure probabilities $p_l,\ l = 0, 1, 2$. Using the following formula, we can obtain the steady state distribution of the Markov chain,

$$\pi_\alpha = \boldsymbol{M}_\alpha \pi_\alpha \ . \tag{5}$$

With the same method, $\boldsymbol{M}_\beta$ and π_β are obtained.

For the encoder defined in (58) we get

$$\boldsymbol{M}_\alpha = \begin{bmatrix} (1-p)^2(2p+1) & (1-p)^2 & (1-p)^3 & 0 & 0 \\ p^2(1-p) & 0 & p(1-p)^2 & p^3-2p+1 & (1-p)^2 \\ p^2(1-p) & p(1-p) & p(1-p)^2 & 0 & 0 \\ p^2(1-p) & p(1-p) & p(1-p)^2 & 0 & 0 \\ p^3 & p^2 & p^2(3-2p) & p^2(2-p) & p(2-p) \end{bmatrix}$$

For simpler presentation we have asssumed that p_0, p_1 and p_2 are equal to p, however in general the elements of this matrix are calculated as a function of these three variables.

Define the matrices $\boldsymbol{T}^{(l)}$

$$T_{i,j}^{(l)} = P\left(L_{\text{out},n}^{(l)} = 0 | \boldsymbol{\mu}_{\alpha,n} = \boldsymbol{\sigma}_\alpha^{(i)}, \boldsymbol{\mu}_{\beta,n+1} = \boldsymbol{\sigma}_\beta^{(j)}\right)$$

The probability of erasure is equal to

$$\begin{aligned} p_e^{(l)} &= P\left(L_{\text{out},n}^{(l)} = 0\right) = \\ &\sum_{i=1}^{|\mathcal{M}_\alpha|}\sum_{j=1}^{|\mathcal{M}_\beta|} P\left(L_{\text{out},n}^{(l)} = 0 | \boldsymbol{\mu}_{\alpha,n} = \boldsymbol{\sigma}_\alpha^{(i)}, \boldsymbol{\mu}_{\beta,n+1} = \boldsymbol{\sigma}_\beta^{(j)}\right) \\ &\cdot P\left(\boldsymbol{\mu}_{\alpha,n} = \boldsymbol{\sigma}_\alpha^{(i)}\right) \cdot P\left(\boldsymbol{\mu}_{\beta,n+1} = \boldsymbol{\sigma}_\beta^{(j)}\right) \\ &= \pi_\alpha \cdot \boldsymbol{T}^{(l)} \cdot \pi_\beta \ . \end{aligned} \tag{6}$$

Using the above formula, the desired transfer functions of (1) are acquired.

5 Analysis of Iterative Decoding

5.1 Density Evolution for BCCs

By means of the erasure probability of the component decoders, we are able to calculate the evolution of the erasure probability during the decoding procedure.[3] As the decoder is the same in all iterations we can use the transfer functions obtained in the previous section recursively to obtain the exact decoding probability of erasure after a certain number of iterations.

For coupled BCCs, the decoding probability of erasure for symbol $l = 0, 1, 2$ of decoder A at time t after i iterations can be obtained as

$$p_{D_A,0}^{(i,t)} = f_{D_A,0}\left(q_{D_B,0}^{(i-1)}, q_{D_B,1}^{(i-1)}, q_{D_B,2}^{(i-1)}\right) \tag{7}$$

$$p_{D_A,1}^{(i,t)} = f_{D_A,1}\left(q_{D_B,0}^{(i-1)}, q_{D_B,1}^{(i-1)}, q_{D_B,2}^{(i-1)}\right) \tag{8}$$

$$p_{D_A,2}^{(i,t)} = f_{D_A,2}\left(q_{D_B,0}^{(i-1)}, q_{D_B,1}^{(i-1)}, q_{D_B,2}^{(i-1)}\right) \ , \tag{9}$$

where

$$q_{D_B,0}^{(i-1)} = \varepsilon \cdot p_{D_B,0}^{(i-1,t)} \tag{10}$$

$$q_{D_B,1}^{(i-1)} = \varepsilon \cdot p_{D_B,2}^{(i-1,t-1)} \tag{11}$$

$$q_{D_B,2}^{(i-1)} = \varepsilon \cdot p_{D_B,1}^{(i-1,t+1)} \ . \tag{12}$$

Here $f_{D_A,l}$ is the transfer function for the lth symbol of the component decoder D_A and ε denotes the erasure probability of the channel. Because of the symmetric design, the update equations for decoder D_B are identical to those of decoder D_A after interchanging D_A and D_B in equations (7)–(12).

[3] An assumption in density evolution is that the messages exchanged by the decoders are independent. For turbo codes it has been shown in [9] that this can be achieved by considering a windowed BCJR decoder. A formal proof for BCCs is still an open problem, but we expect that the technique in [9] can be generalized to the ensembles considered here.

Table 1: Thresholds of blockwise BCCs.

$\boldsymbol{G}(D)$	$\varepsilon_{\mathrm{BP}}$	$\varepsilon_{\mathrm{MAP}}$	$\varepsilon_{\mathrm{SC}}$	$\varepsilon_{\mathrm{SC}}^{\mathrm{W}}$
$(1/7, 5/7)$	0.5541	0.6654	0.6609	0.6554
$(1/5, 7/5)$	0.5541	0.6654	0.6609	0.6609

The initial LLRs from before the first iteration are assumed to be set to zero, i.e., $p_{D_A,l}^{(i=0,t)} = p_{D_B,l}^{(i=0,t)} = 1$ for $l = 0, 1, 2$. However, for coupling length L, all messages which come from time $t < 1$ or $t > L$ are assumed to be known, i.e., all probabilities with time index $t < 1$ or $t > L$ are equal to zero. The decoding probability of erasure at time t for blockwise BCCs after i iterations is:

$$p_{e,t} = \varepsilon \cdot p_{D_A,0}^{(i,t)} \cdot p_{D_B,0}^{(i,t)}$$

As a special case, an uncoupled BCC can be achieved by tailbiting and the assumption of $L = 1$. The transfer function for the uncoupled case can be achieved from the above mentioned equations for coupled BCCs by omitting the time index t.

5.2 Results and Discussion

We want to evaluate the largest probability of erasure of the channel ε for which the probability of erasure of BP decoding $p_{e,t}$ converges to zero for all t. To obtain such a threshold, $p_{e,t}$ is evaluated as a function of the number of iterations for different values of ε (density evolution). We consider two examples of blockwise BCC ensembles of rate $R = 1/3$ with identical component encoders. The first case corresponds to component encoders with $\boldsymbol{G}(D)$ as defined in (58), i.e., with generators (1/7, 5/7) in octal form. In the second case we consider the generators (1/5, 7/5), i.e., the feedback polynomial is exchanged. The thresholds $\varepsilon_{\mathrm{BP}}$ for the uncoupled case (tailbiting with $L = 1$) and $\varepsilon_{\mathrm{SC}}$ for the coupled (terminated) case are shown in Table 1[4]. It can be observed that spatial coupling leads to a significantly better BP decoding threshold. The value $\varepsilon_{\mathrm{SC}}^{\mathrm{W}}$ denotes the threshold that can be achieved with a sliding window decoder that starts at time instant $t = 1$. For the first encoder we see that $\varepsilon_{\mathrm{SC}}^{\mathrm{W}}$ is worse than $\varepsilon_{\mathrm{SC}}$, which is due to the fact that the decoder converges better from the end of the coupled sequence. This suboptimality of the window decoding threshold can be avoided by exchanging the generator polynomials.

Table 1 shows also an upper bound $\varepsilon_{\mathrm{MAP}}$ on the threshold of an optimal MAP decoder of the uncoupled codes. This upper bound can be obtained

[4]It should be noted that the threshold for tailbiting, due to the circular structure, is equal to the threshold for the uncoupled case for any value of L.

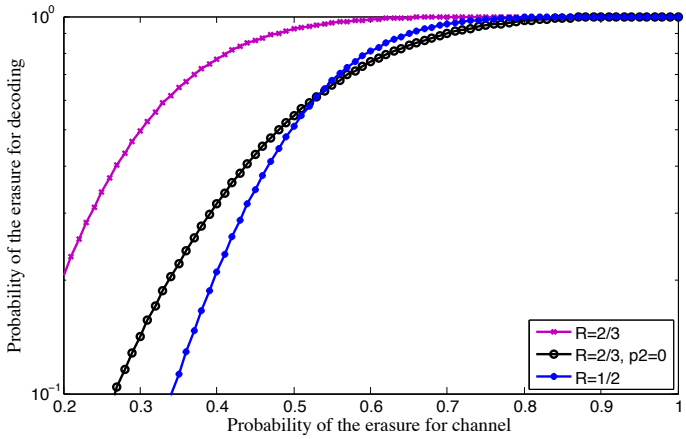


Figure 6: Probability of erasure of the component convolutional codes.

according to the area theorem [10] as solution to the following equation:

$$\int_{\varepsilon_{\mathrm{MAP}}}^{1} \bar{p}_{\mathrm{extr}}(p)dp = R \ . \tag{13}$$

Here R is the rate of the BCC and

$$\bar{p}_{\mathrm{extr}}(p) = \frac{1}{3}\left(p_{D_A,0}^{(\infty)} \cdot p_{D_B,0}^{(\infty)} + p_{D_A,2}^{(\infty)} \cdot p_{D_B,1}^{(\infty)} + p_{D_A,1}^{(\infty)} \cdot p_{D_B,2}^{(\infty)}\right)$$

denotes the extrinsic probability of erasure of uncoupled BCCs, which is a function of the channel parameter p. To solve this equation we compute the area under the curve $\bar{p}_{\mathrm{extr}}(p)$ for a sufficiently large number of decoding iterations.

We see that the BP threshold of the coupled codes is close to $\varepsilon_{\mathrm{MAP}}$. However, some gap is still visible for the considered ensemble, which is equivalent to the one introduced in [6]. As shown in [11], this gap to the MAP threshold vanishes if the original ensemble is generalized to larger coupling memories $m_{\mathrm{BCC}} > 1$.

Another interesting observation is that for BCCs the coupling gain, i.e., the gap between $\varepsilon_{\mathrm{SC}}$ and $\varepsilon_{\mathrm{BP}}$ appears to be significantly larger than for conventional turbo codes (i.e., parallel concatenated convolutional codes). Although thresholds for spatially coupled turbo codes have not yet been investigated in the literature, this follows from the gap between their uncoupled BP threshold and MAP threshold, which have been determined in [10]. The large threshold improvement for the coupled case can be justified by looking at the transfer function of the component codes, as illustrated in Figure 6.

On one hand, since a rate $R = 2/3$ encoder performs worse than a $R = 1/2$ encoder it is natural that the BP threshold of an uncoupled turbo code is better than the BP threshold of an uncoupled BCC code. However, assuming that the second input of the $R = 2/3$ encoder is known (i.e., $p_2 = 0$) we effectively obtain an equivalent encoder of $R = 1/2$ whose performance is considerably improved (see black curve in Figure 6). This effect appears for the BCC ensemble at time $t = 1$ and $t = L$, and it propagates further to the other time instants during the iterative decoding procedure, resulting in a threshold improvement.

6 Conclusions

In this paper, we derived exact density evolution equations for blockwise BCCs under BP decoding over the BEC. Considering component encoders of memory $m = 2$ we computed BP thresholds for the coupled (terminated) and uncoupled (tailbiting) case and compared them with an upper bound on the MAP threshold. Our threshold analysis confirms the conjecture made in [6] that terminated BCCs can have much better thresholds than their tailbiting counterparts. The threshold of the considered original BCC ensemble is already close to the MAP threshold but for a vanishing gap some generalization to larger coupling memories is required. A major advantage of the BCC construction compared to other turbo-like codes are the superior minimum distance properties in combination with the capacity approaching thresholds.

References

[1] A. Feltström, D. Truhachev, M. Lentmaier and K. Zigangirov, "Braided block codes", *IEEE Trans. Inf. Theory*, vol. 55, no. 6, pp. 2640–2658, Jun. 2009.

[2] P. Elias, "Error free coding", *IRE Trans. Inf. Theory*, vol. PGIT-4, pp. 29–37, Sep. 1954.

[3] Y. Jian, H. Pfister, K. Narayanan, R. Rao and R. Mazahreh, "Iterative hard-decision decoding of braided BCH codes for high-speed optical communication", in *Proc. IEEE Global Telecommun. Conf. (GLOBECOM)*, Atlanta, GA, USA, Dec. 2013.

[4] J. Justesen, K. Larsen and L. Pedersen, "Error correcting coding for OTN", *IEEE Commun. Mag.*, vol. 48, no. 9, pp. 70–75, Sep. 2010.

[5] B. Smith, A. Farhood, A. Hunt, F. Kschischang and J. Lodge, "Staircase codes: FEC for 100 Gb/s OTN", *J. Lightw. Technol.*, vol. 30, no. 1, pp. 110–117, Jan. 2012.

[6] 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.

[7] S. Kudekar, T. Richardson and R. 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.

[8] B. Kurkoski, P. Siegel and J. Wolf, "Exact probability of erasure and a decoding algorithm for convolutional codes on the binary erasure channel", in *Proc. IEEE Global Telecommun. Conf. (GLOBECOM)*, San Francisco, CA, USA, Dec. 2003.

[9] M. Lentmaier, D. V. Truhachev, K. S. Zigangirov and D. J. Costello, "An analysis of the block error probability performance of iterative decoding", *IEEE Trans. Inf. Theory*, vol. 51, no. 11, pp. 3834–3855, Nov. 2005.

[10] C. Méasson, "Conservation laws for coding", PhD thesis, École Polytechnique Fédérale de Lausanne, 2006.

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

Paper II

Spatially Coupled Turbo Codes

In this paper, we introduce the concept of spatially coupled turbo codes (SC-TCs), as the turbo codes counterpart of spatially coupled low-density parity-check codes. We describe spatial coupling for both Berrou *et al.* and Benedetto *et al.* parallel and serially concatenated codes. For the binary erasure channel, we derive the exact density evolution (DE) equations of SC-TCs by using the method proposed by Kurkoski *et al.* to compute the decoding erasure probability of convolutional encoders. Using DE, we then analyze the asymptotic behavior of SC-TCs. We observe that the belief propagation (BP) threshold of SC-TCs improves with respect to that of the uncoupled ensemble and approaches its maximum a posteriori threshold. This phenomenon is especially significant for serially concatenated codes, whose uncoupled ensemble suffers from a poor BP threshold.

S. Moloudi, M. Lentmaier, and A. Graell i Amat,
"Spatially coupled turbo codes,"
in *Proc. Int. Symp. on Turbo Codes and Iterative Inf. Processing (ISTC)*, Bremen, Germany, 2014.

1 Introduction

Low-density parity-check (LDPC) convolutional codes [1], also known as spatially coupled LDPC (SC-LDPC) codes [2], can be obtained from a sequence of individual LDPC block codes by distributing the edges of their Tanner graphs over several adjacent blocks [3]. The resulting spatially coupled codes exhibit a threshold saturation phenomenon, which has attracted a lot of interest in the past few years: the threshold of an iterative belief propagation (BP) decoder, obtained by density evolution (DE), is improved to that of the optimal maximum-a-posteriori (MAP) decoder [2], [3]. As a consequence, it is possible to achieve capacity with simple regular LDPC codes, which show without spatial coupling a significant gap between BP and MAP threshold.

The concept of spatial coupling is not limited to LDPC codes. Spatially coupled turbo-like codes, for example, can be obtained by replacing the block-wise permutation of a turbo code by a convolutional permutation [4]. In combination with a windowed decoder for the component code, a continuous streaming implementation is possible [5]. The self-concatenated convolutional codes in [6] are closely related structures as well. A variant of spatially coupled self-concatenated codes with block-wise processing, called laminated codes was considered in [7]. They have the advantage that an implementation similar to uncoupled turbo codes is possible, without the need for a streaming implementation of the decoder. A block-wise version of braided convolutional codes [8], a class of spatially coupled codes with convolutional components, has recently been analyzed in [9].

The aim of this paper is to investigate the impact of spatial coupling on the thresholds of *classical* turbo codes. For this purpose we introduce some special block-wise spatially coupled ensembles of parallel concatenated codes (SC-PCCs) and serially concatenated codes (SC-SCCs), which are spatially coupled versions of the ensembles by Berrou *et al.* [10] and Benedetto *et al.* [11], respectively. With a slight abuse of the term, we call both parallel and serial ensembles spatially coupled turbo codes (SC-TCs). For these ensembles we derive exact DE equations from the transfer functions of the component decoders [12], [13] and perform a threshold analysis for the binary erasure channel (BEC), analogously to [3], [9]. To compare the results for SC-PCC and SC-SCC ensembles with each other some ensembles with puncturing are also considered. The BP thresholds of the different ensembles are presented and compared to the MAP thresholds for different coupling memories.

2 Spatially Coupled Turbo Codes

In this section, we introduce spatially coupled turbo codes. We first describe spatial coupling for both parallel and serially concatenated codes, and then

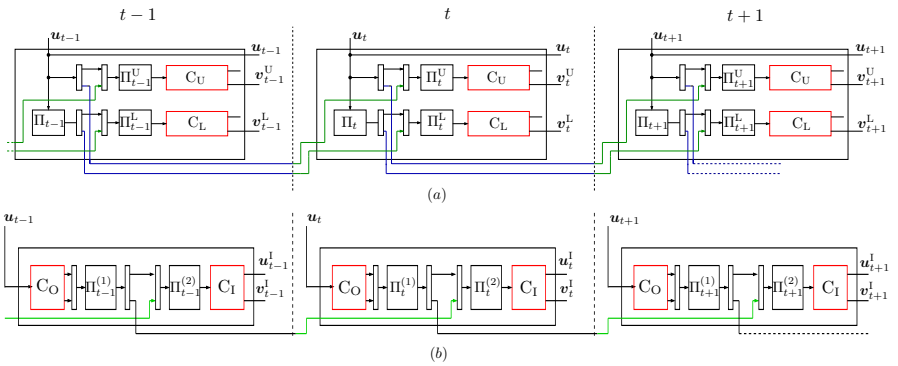


Figure 1: Block diagram of the encoder of a spatially coupled turbo code for coupling memory $m = 1$. (a) parallel concatenation (b) serial concatenation.

address their iterative decoding.

2.1 Spatially Coupled Parallel Concatenated Codes

We consider the spatial coupling of $R = 1/3$ parallel concatenated codes, built from the parallel concatenation of two rate-1/2 recursive systematic convolutional encoders, denoted by C_{U} and C_{L} (see Figure 1). For simplicity, we describe spatial coupling with coupling memory $m = 1$. Consider a collection of L turbo encoders at time instants $t = 1, \ldots, L$, as illustrated in Figure 1(a). L is called the coupling length. We denote by $\boldsymbol{u}_t$ the information sequence, and by $\boldsymbol{v}_t^{\mathrm{U}}$ and $\boldsymbol{v}_t^{\mathrm{L}}$ the code sequences of C_{U} and C_{L}, respectively, at time t. The output of the turbo encoder is given by the tuple $\boldsymbol{v}_t = (\boldsymbol{u}_t, \boldsymbol{v}_t^{\mathrm{U}}, \boldsymbol{v}_t^{\mathrm{L}})$. A SC-PCC ensemble (with $m = 1$) is obtained by connecting each turbo code in the chain to the one on the left and to the one on the right as follows. Divide the information sequence $\boldsymbol{u}_t$ into two sequences, $\boldsymbol{u}_{t,\mathrm{A}}$ and $\boldsymbol{u}_{t,\mathrm{B}}$ by a demultiplexer. Also divide a copy of the information sequence, which is properly reordered by the permutation Π_t, into two sequences, $\boldsymbol{u}_{t,\mathrm{A}'}$ and $\boldsymbol{u}_{t,\mathrm{B}'}$ by another demultiplexer. At time t, the information sequence at the input of encoder C_{U} is $(\boldsymbol{u}_{t,\mathrm{A}}, \boldsymbol{u}_{t-1,\mathrm{B}})$, properly reordered by a permutation Π_t^{U}. Likewise, the information sequence at the input of encoder C_{L} is $(\boldsymbol{u}_{t,\mathrm{A}'}, \boldsymbol{u}_{t-1,\mathrm{B}'})$, properly reordered by the permutation Π_t^{L}. In Figure 1 the blue lines represent the information bits from the current time slot t that are used in the next time slot $t+1$ and the green lines represent the information bits from the previous time slot $t-1$. In order to terminate the encoder of the SC-PCC to the zero state, the information sequences at the end of the chain are chosen in such a way that the output sequence at time $t = L + 1$ becomes $\boldsymbol{v}_{L+1} = \mathbf{0}$. Analogously to conventional convolutional

codes this results in a rate loss that becomes smaller as L increases.

Using the procedure described above a coupled chain (a convolutional structure over time) of L turbo encoders with coupling memory $m = 1$ is obtained. An extension to larger coupling memories $m > 1$ is presented in Section 4.

2.2 Spatially Coupled Serially Concatenated Codes

We consider the coupling of serially concatenated codes (SCCs) built from the serial concatenation of two rate-1/2 recursive systematic convolutional encoders. The overall code rate of the uncoupled ensemble is therefore $R = 1/4$. A block diagram of the encoder is depicted in Figure 1(b) for coupling memory $m = 1$. As for SC-PCCs, let $\boldsymbol{u}_t$ be the information sequence at time t. Also, denote by $\boldsymbol{v}_t^{\mathrm{O}} = (\boldsymbol{v}_t^{\mathrm{O,s}}, \boldsymbol{v}_t^{\mathrm{O,p}}) = (\boldsymbol{u}_t, \boldsymbol{v}_t^{\mathrm{O,p}})$ and $\boldsymbol{v}_t^{\mathrm{I}}$ the encoded sequence at the output of the outer and inner encoder, respectively, and by $\tilde{\boldsymbol{v}}_t^{\mathrm{O}}$ the sequence $\boldsymbol{v}_t^{\mathrm{O}}$ after permutation. The SC-SCC with $m = 1$ is constructed as follows. Consider a collection of L SCCs at time instants $t = 1, \ldots, L$. Divide the sequence $\tilde{\boldsymbol{v}}_t^{\mathrm{O}}$ into two parts, $\tilde{\boldsymbol{v}}_{t,\mathrm{A}}^{\mathrm{O}}$ and $\tilde{\boldsymbol{v}}_{t,\mathrm{B}}^{\mathrm{O}}$. Then, at time t the sequence at the input of the inner encoder C_{I} is $(\tilde{\boldsymbol{v}}_{t,\mathrm{A}}^{\mathrm{O}}, \tilde{\boldsymbol{v}}_{t-1,\mathrm{B}}^{\mathrm{O}})$. In order to terminate the encoder of the SC-SCC to the zero state, the information sequences at the end of the chain are chosen in such a way that the output sequence at time $t = L + 1$ becomes $\boldsymbol{v}_{L+1}^{\mathrm{I}} = \mathbf{0}$.

Using this construction method, a coupled chain of L SCCs with coupling memory $m = 1$ is obtained. An extension to larger coupling memories $m > 1$ is presented in Section 4.

2.3 Iterative decoding

As standard turbo codes, SC-TCs can be decoded using iterative message passing (belief propagation) decoding, where the component encoders of each turbo code are decoded using the BCJR algorithm. The BP decoding of SC-PCCs can be easily visualized with the help of Figure 2, which shows the factor graph of a single section of the SC-PCC. We denote by D_{U} and D_{L} the decoder of the upper and lower encoder, respectively.

The decoder D_{U} receives at its input information from the channel for both systematic and parity bits. Furthermore, it also receives a-priori information on the systematic bits from other decoders. As described above, at time t the information sequence at the input of C_{U} consists of two parts, $\boldsymbol{u}_{t,\mathrm{A}}$ and $\boldsymbol{u}_{t-1,\mathrm{B}}$. Correspondingly, D_{U} at time instant t receives a priori information from D_{L} at time instants $t-1$, t and $t+1$. Based on the information from the channel and from the companion decoders, D_{U} computes the extrinsic information on the systematic bits using the BCJR algorithm. Since the structure of SC-PCCs

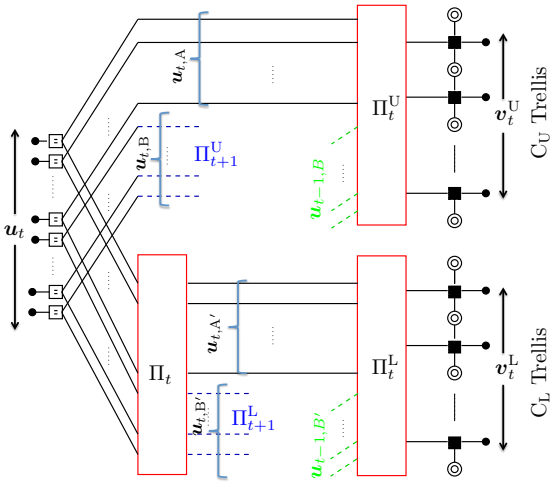


Figure 2: Factor graph of a single section of a SC-PCC.

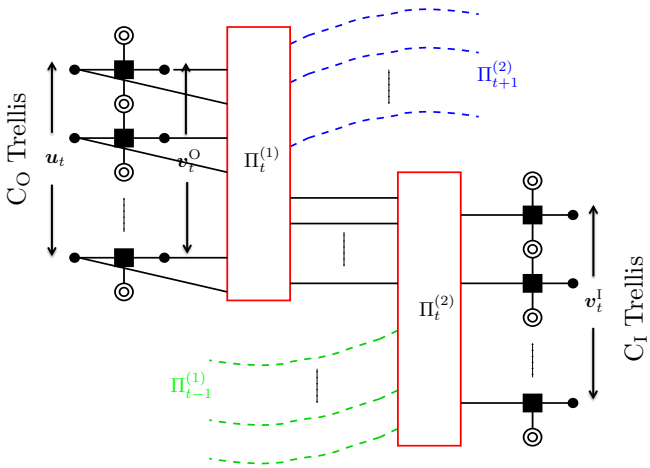


Figure 3: Factor graph of a single section of a SC-SCC.

is symmetric, the decoding of the lower encoder is performed in an identical manner.

Similarly to SC-PCCs, the decoding SC-SCCs can also be described with the help of a factor graph. The factor graph of a section of a SC-SCC with $m = 1$ is shown in Figure 3.

3 Density Evolution Analysis on the BEC

In this section, we analyze the asymptotic performance of SC-TCs using DE. We consider transmission over a BEC with erasure probability ε, denoted by BEC(ε). We derive the exact DE equations for both (unpunctured) SC-PCCs and SC-SCCs and discuss the modification of the equations when puncturing is applied for achieving higher rates.

3.1 Spatially Coupled Parallel Concatenated Codes

Let $p_{\mathrm{U,s}}$ and $p_{\mathrm{L,s}}$ be the average (extrinsic) erasure probability on the systematic bits at the output of the upper and lower decoder, respectively. Likewise, we define $p_{\mathrm{U,p}}$ and $p_{\mathrm{L,p}}$ for the parity bits.

The erasure probabilities $p_{\mathrm{U,s}}$ and $p_{\mathrm{U,p}}$ at iteration i and time instant t can be written as

$$p_{\mathrm{U,s}}^{(i,t)} = f_{\mathrm{U,s}}\left(q_{\mathrm{L}}^{(i-1)}, \varepsilon\right) \tag{1}$$

$$p_{\mathrm{U,p}}^{(i,t)} = f_{\mathrm{U,p}}\left(q_{\mathrm{L}}^{(i-1)}, \varepsilon\right), \tag{2}$$

where

$$q_{\mathrm{L}}^{(i-1)} = \varepsilon \cdot \frac{2p_{\mathrm{L,s}}^{(i-1,t)} + p_{\mathrm{L,s}}^{(i-1,t-1)} + p_{\mathrm{L,s}}^{(i-1,t+1)}}{4}, \tag{3}$$

and $f_{\mathrm{U,s}}$ and $f_{\mathrm{U,p}}$ denote the upper decoder transfer functions for the systematic and parity bits, respectively.

Note that the upper decoder transfer function at time t depends on both the channel erasure probability and the extrinsic erasure probability on the systematic bits from the lower decoder at time instants t, $t-1$ and $t+1$, due to the coupling. Because of the symmetric design, the lower decoder update is identical to that of the upper decoder by interchanging p_{U} and p_{L}, and substituting $q_{\mathrm{L}} \leftarrow q_{\mathrm{U}}$ in (1)–(3).

Finally, the a-posteriori erasure probability on the information bits at time t and iteration i is[5]

$$p_{\mathrm{a}}^{(i,t)} = \varepsilon \cdot \frac{p_{\mathrm{U,s}}^{(i,t)} p_{\mathrm{L,s}}^{(i,t)} + p_{\mathrm{U,s}}^{(i,t)} p_{\mathrm{L,s}}^{(i,t+1)} + p_{\mathrm{U,s}}^{(i,t+1)} p_{\mathrm{L,s}}^{(i,t)} + p_{\mathrm{U,s}}^{(i,t+1)} p_{\mathrm{L,s}}^{(i,t+1)}}{4} \tag{4}$$

For the BEC it is possible to compute analytic expressions for the exact (extrinsic) probability of erasure of convolutional encoders, using the method proposed in [12] and [13]. Here, we use this method to derive the exact expressions for the transfer functions of the component decoders. DE is then

[5] We remark that although (2) is not applied within the DE recursion it is required for the computation of the area bound on the MAP threshold.

performed by tracking the evolution of $p_{\mathrm{a}}^{(i,t)}$ with the number of iterations, with the initialization $p_{\mathrm{U,s}}^{(0,t)} = p_{\mathrm{U,p}}^{(0,t)} = p_{\mathrm{L,s}}^{(0,t)} = p_{\mathrm{L,p}}^{(0,t)} = 0$ for $t = 0$ and $t > L$, and 1 otherwise. The BP threshold corresponds to the maximum channel parameter ε for which successful decoding is achieved, i.e., $p_{\mathrm{a}}^{(i,t)}$ tends to zero for all time instants t as i tends to infinity.

3.2 Spatially Coupled Serially Concatenated Codes

Similarly to the parallel case, DE equations can be derived for SC-SCCs. Let $p_{\mathrm{O,s}}$ and $p_{\mathrm{I,s}}$ be the average (extrinsic) erasure probability on the systematic bits at the output of the outer and inner decoder, respectively. Likewise, we define $p_{\mathrm{O,p}}$ and $p_{\mathrm{I,p}}$ for the parity bits at the output of the outer and inner decoder, respectively.

The erasure probabilities $p_{\mathrm{I,s}}$ and $p_{\mathrm{I,p}}$ can be written as

$$p_{\mathrm{I,s}}^{(i,t)} = f_{\mathrm{I,s}}\left(q_{\mathrm{O}}^{(i-1)}, \varepsilon\right) \tag{5}$$

$$p_{\mathrm{I,p}}^{(i,t)} = f_{\mathrm{I,p}}\left(q_{\mathrm{O}}^{(i-1)}, \varepsilon\right), \tag{6}$$

where

$$q_{\mathrm{O}}^{(i-1)} = \varepsilon \cdot \frac{p_{\mathrm{O,s}}^{(i-1,t)} + p_{\mathrm{O,p}}^{(i-1,t)} + p_{\mathrm{O,s}}^{(i-1,t-1)} + p_{\mathrm{O,p}}^{(i-1,t-1)}}{4}, \tag{7}$$

and $f_{\mathrm{I,s}}$ and $f_{\mathrm{I,p}}$ denote the inner decoder transfer functions for the systematic and parity bits, respectively.

Likewise, $p_{\mathrm{O,s}}$ and $p_{\mathrm{O,p}}$ are

$$p_{\mathrm{O,s}}^{(i,t)} = f_{\mathrm{O,s}}\left(q_{\mathrm{I}}^{(i-1)}, q_{\mathrm{I}}^{(i-1)}\right) \tag{8}$$

$$p_{\mathrm{O,p}}^{(i,t)} = f_{\mathrm{O,p}}\left(q_{\mathrm{I}}^{(i-1)}, q_{\mathrm{I}}^{(i-1)}\right), \tag{9}$$

where

$$q_{\mathrm{I}}^{(i-1)} = \varepsilon \cdot \frac{p_{\mathrm{I,s}}^{(i-1,t)} + p_{\mathrm{I,s}}^{(i-1,t+1)}}{2}\ . \tag{10}$$

The a-posteriori erasure probability on the information bits at time t after i iterations is

$$p_{\mathrm{a}}^{(i,t)} = \varepsilon \cdot p_{\mathrm{O,s}}^{(i,t)} \cdot \frac{p_{\mathrm{I,s}}^{(i,t)} + p_{\mathrm{I,s}}^{(i,t+1)}}{2}\ . \tag{11}$$

DE is then performed by tracking the evolution of $p_{\mathrm{a}}^{(i,t)}$ with the number of iterations, with the initialization $p_{\mathrm{I,s}}^{(0,t)} = p_{\mathrm{I,p}}^{(0,t)} = p_{\mathrm{O,s}}^{(0,t)} = p_{\mathrm{O,p}}^{(0,t)} = 0$ for $t = 0$ and $t > L$ and 1 otherwise.

3.3 Spatially Coupled Turbo Codes with Random Puncturing

Higher rates can be obtained by applying puncturing. Here, we consider random puncturing. Assume that a code sequence $\boldsymbol{x}$ is randomly punctured such that a fraction $\rho \in [0,1]$ of the coded bits survive after puncturing, and then transmitted over a BEC(ε). ρ will be referred to as the *permeability rate*. For the BEC, puncturing is equivalent to transmitting $\boldsymbol{x}$ through a BEC(ε_ρ) resulting from the concatenation of two BECs, BEC(ε) and BEC($1-\rho$), where $\varepsilon_\rho = 1-(1-\varepsilon)\rho$. The DE equations derived in the previous subsections can be easily modified to account for puncturing. Consider first the case of SC-PCCs. We consider only puncturing of the parity bits, and that both C_U and C_L are equally punctured with permeability rate ρ. The code rate of the (uncoupled) punctured parallel concatenated code (PCC) is $R = \frac{1}{1+2\rho}$. This results in a slight modification of the DE equations, substituting $\varepsilon \leftarrow \varepsilon_\rho$ in (1), (2).

For SC-SCCs we consider puncturing as proposed in [14], [15], which results in better SCCs as compared to standard SCCs. Let ρ_0 and ρ_1 be the permeability rate of the systematic and parity bits, respectively, of C_O sent directly to the channel (see [15, Figure 1]), and ρ_2 the permeability rate of the parity bits of C_I. The code rate of the (uncoupled) punctured[6] SCC is $R = \frac{1}{\rho_0+\rho_1+2\rho_2}$. The DE for punctured SC-SCCs is obtained by substituting $\varepsilon \leftarrow \varepsilon_{\rho_2}$ in (5), (6), and modifying (7) to

$$q_\mathrm{O}^{(i-1)} = \frac{\varepsilon \cdot \left(p_{\mathrm{O,s}}^{(i-1,t)} + p_{\mathrm{O,s}}^{(i-1,t-1)}\right) + \varepsilon_{\rho_1} \cdot \left(p_{\mathrm{O,p}}^{(i-1,t)} + p_{\mathrm{O,p}}^{(i-1,t-1)}\right)}{4},$$

and (8), (9) to

$$p_{\mathrm{O,s}}^{(i,t)} = f_{\mathrm{O,s}}\left(q_\mathrm{I}^{(i-1)}, \tilde{q}_\mathrm{I}^{(i-1)}\right) \tag{12}$$

$$p_{\mathrm{O,p}}^{(i,t)} = f_{\mathrm{O,p}}\left(q_\mathrm{I}^{(i-1)}, \tilde{q}_\mathrm{I}^{(i-1)}\right), \tag{13}$$

where $q_\mathrm{I}^{(i-1)}$ is given in (10) and

$$\tilde{q}_\mathrm{I}^{(i-1)} = \varepsilon_{\rho_1} \cdot \frac{p_{\mathrm{I,s}}^{(i-1,t)} + p_{\mathrm{I,s}}^{(i-1,t+1)}}{2}\,. \tag{14}$$

4 Extension to Larger Coupling Memories

The results from the previous sections can easily be generalized to larger coupling memories $m > 1$.

[6] In this paper we consider $\rho_0 = 1$, i.e., the overall code is systematic.

Let us first consider SC-PCCs. In the general case the information sequences $\boldsymbol{u}_t, \boldsymbol{u}_{t-1}, \ldots, \boldsymbol{u}_{t-m}$ from $m+1$ different time instances are used by the encoders at time t. This is achieved by dividing the information sequence $\boldsymbol{u}_t$ into the sequences $\boldsymbol{u}_{t,j}$, $j = 0, \ldots, m$ by a multiplexer, and also dividing a properly reordered copy of the information bits into $\boldsymbol{u}_{t,j'}$, $j' = 0, \ldots, m$, which can be accomplished by permutation Π_t followed by a multiplexer. At the input of the upper encoder C_{U} at time t the sequences $\boldsymbol{u}_{t-j,j}$ are multiplexed and reordered by the permutation Π_t^{U}. The lower encoder C_{L} receives the information sequences $\boldsymbol{u}_{t-j',j'}$, multiplexed and reordered by Π_t^{L}. The encoder in Figure 1(a) corresponds to the special case $m = 1$.

In the DE recursion we now have to modify (3) to

$$q_{\mathrm{L}}^{(i-1)} = \varepsilon \cdot \frac{\sum_{j=0}^{m} \sum_{k=0}^{m} p_{\mathrm{L,s}}^{(i,t+j-k)}}{(m+1)^2} \ ,$$

and the a-posteriori erasure probability on the information bits at time t and iteration i (4) becomes

$$p_{\mathrm{a}}^{(i,t)} = \varepsilon \cdot \frac{\sum_{j=0}^{m} \sum_{k=0}^{m} p_{\mathrm{U,s}}^{(i,t+j)} p_{\mathrm{L,s}}^{(i,t+k)}}{(m+1)^2} \ .$$

Likewise, for SC-SCCs the code sequence $\boldsymbol{v}_t^{\mathrm{O}}$ of C_{O} is divided randomly into the sequences $\tilde{\boldsymbol{v}}_{t,j}^{\mathrm{O}}$, $j = 0, \ldots, m$. C_{I} receives at time t the sequences $\tilde{\boldsymbol{v}}_{t-j,j}^{\mathrm{O}}$ after passing a multiplexer and a permutation. The encoder in Figure 1(b) corresponds to the special case $m = 1$.

Equations (7) and (10) in the DE recursion are modified accordingly to

$$q_{\mathrm{O}}^{(i-1)} = \varepsilon \cdot \frac{\sum_{j=0}^{m} p_{\mathrm{O,s}}^{(i-1,t-j)} + p_{\mathrm{O,p}}^{(i-1,t-j)}}{2(m+1)}$$

and

$$q_{\mathrm{I}}^{(i-1)} = \varepsilon \cdot \frac{\sum_{j=0}^{m} p_{\mathrm{I,s}}^{(i-1,t+j)}}{m+1} \ .$$

The a-posteriori erasure probability on the information bits at time t after i iterations (11) becomes

$$p_{\mathrm{a}}^{(i,t)} = \varepsilon \cdot p_{\mathrm{O,s}}^{(i,t)} \cdot \frac{\sum_{j=0}^{m} p_{\mathrm{I,s}}^{(i,t+j)}}{m+1} \ .$$

5 Results and Discussion

In this section, we give numerical results for some SC-TCs, using the DE described in Section 3 and 4. In our examples we consider SC-TCs with identical

Table 1: Thresholds for SC-TCs.

Ensemble	Rate	$\varepsilon_{\mathrm{BP}}$	$\varepsilon_{\mathrm{MAP}}$	$\varepsilon_{\mathrm{SC}}^{1}$	$\varepsilon_{\mathrm{SC}}^{3}$
$\mathcal{C}_{\mathrm{PCC}}/\mathcal{C}_{\mathrm{SC-PCC}}$	1/3	0.6428	0.6553	0.6553	0.6553
$\mathcal{C}_{\mathrm{SCC}}/\mathcal{C}_{\mathrm{SC-SCC}}$	1/4	0.6896	0.7483	0.7378	0.7482

rate-1/2, 4-states component encoders. In particular, we consider component encoders with generator polynomials $(1, 5/7)$ in octal notation. For notational simplicity, we denote the uncoupled PCC ensemble by $\mathcal{C}_{\mathrm{PCC}}$ and the corresponding coupled ensemble by $\mathcal{C}_{\mathrm{SC-PCC}}$. For SC-SCCs, we denote by $\mathcal{C}_{\mathrm{SCC}}$, and $\mathcal{C}_{\mathrm{SC-SCC}}$ the uncoupled and coupled ensembles, respectively. Note that since the two component encoders are identical, $f_{\mathrm{U,s}}(x, y) = f_{\mathrm{L,s}}(x, y)$ and $f_{\mathrm{U,p}}(x, y) = f_{\mathrm{L,p}}(x, y)$ for SC-PCCs, and $f_{\mathrm{I,s}}(x, y) = f_{\mathrm{O,s}}(x, y)$ and $f_{\mathrm{I,p}}(x, y) = f_{\mathrm{O,p}}(x, y)$ for SC-SCCs. All presented thresholds correspond to the stationary case $L \to \infty$, which lower bounds the thresholds for finite L. For small L the threshold can be considerably larger but at the expense of a higher rate loss.

In Table 1 we give the BP threshold for several SC-TCs and coupling memory $m = 1$ and 3, denoted by $\varepsilon_{\mathrm{SC}}^{1}$ and $\varepsilon_{\mathrm{SC}}^{3}$. We also report in the table the BP threshold ($\varepsilon_{\mathrm{BP}}$) and the MAP threshold ($\varepsilon_{\mathrm{MAP}}$) of the uncoupled ensembles. The MAP threshold was computed applying the area theorem [16]. In all cases we observe an improvement of the BP threshold when coupling is applied. We remark that for $\mathcal{C}_{\mathrm{SC-PCC}}$ the BP threshold of the uncoupled ensemble is already close to the MAP threshold, therefore the potential gain with coupling is limited. However, it is interesting to observe that the BP threshold of $\mathcal{C}_{\mathrm{SC-PCC}}$ with $m = 1$ is very close to $\varepsilon_{\mathrm{MAP}}$, suggesting threshold saturation. The results for the ensemble $\mathcal{C}_{\mathrm{SC-SCC}}$ are also given in Table 1 for coupling memory $m = 1$ and 3. We observe that the ensemble $\mathcal{C}_{\mathrm{SCC}}$ has a poor BP threshold as compared to the MAP threshold. This is a well-known phenomenon for SCCs, for which the gap between the BP and the MAP threshold is large. A significant improvement is obtained by applying coupling with $m = 1$. However, there is still a gap between $\varepsilon_{\mathrm{BP}}$ and $\varepsilon_{\mathrm{MAP}}$, meaning that threshold saturation has not occurred. The BP threshold can be further improved by increasing the coupling memory to $m = 3$. In this case the BP threshold is very close to the MAP threshold, suggesting that threshold saturation occurs for large enough coupling memory. This behavior is similar to the threshold saturation phenomenon of SC-LDPC codes, which occurs for smoothing parameter $w \to \infty$ [2].

In Table 2 we show the BP thresholds of punctured SC-TCs, in order to compare SC-PCCs and SC-SCCs for a given code rate. We consider $R = 1/3$

Table 2: Thresholds for punctured SC-TCs.

Ensemble	Rate	$\varepsilon_{\mathrm{BP}}$	$\varepsilon_{\mathrm{MAP}}$	$\varepsilon_{\mathrm{SC}}^{1}$	$\varepsilon_{\mathrm{SC}}^{3}$
$\mathcal{C}_{\mathrm{PCC}}/\mathcal{C}_{\mathrm{SC-PCC}}$	1/3	0.6428	0.6553	0.6553	0.6553
$\mathcal{C}_{\mathrm{SCC}}/\mathcal{C}_{\mathrm{SC-SCC}}$	1/3	0.6118	0.6615	0.6519	0.6614
$\mathcal{C}_{\mathrm{PCC}}/\mathcal{C}_{\mathrm{SC-PCC}}$	1/2	0.4606	0.4689	0.4689	0.4689
$\mathcal{C}_{\mathrm{SCC}}/\mathcal{C}_{\mathrm{SC-SCC}}$	1/2	0.4010	0.4973	0.4773	0.4969

and $R = 1/2$, and coupling memory 1.[7] For the SC-SCC we used $\rho_1 = 1$ and $\rho_2 = 0.5$ for $R = 1/3$ and $\rho_1 = 0.2$ and $\rho_2 = 0.4$ for $R = 1/2$. Again, in all cases an improvement of the BP threshold is observed when coupling is applied. As expected, for a given rate the PCC ensemble shows a better threshold than the SCC ensemble. However, the improvement in the BP threshold due to coupling for the latter is very significant. For $R = 1/3$ and $m = 1$ the BP threshold of $\mathcal{C}_{\mathrm{SC-SCC}}$ is very close to that of the (unpunctured) ensemble $\mathcal{C}_{\mathrm{SC-PCC}}$, while a large gap is observed for the uncoupled ensembles. For $m = 3$, $\mathcal{C}_{\mathrm{SC-SCC}}$ achieves a better BP threshold than $\mathcal{C}_{\mathrm{SC-PCC}}$. The result is even more remarkable for $R = 1/2$. In this case, while the uncoupled SCC ensemble shows a very poor threshold, $\mathcal{C}_{\mathrm{SC-SCC}}$ shows a superior threshold than $\mathcal{C}_{\mathrm{SC-PCC}}$ already for $m = 1$.

6 Conclusions

In this paper we have introduced some block-wise spatially coupled ensembles of parallel and serially concatenated convolutional codes and performed a density evolution analysis on the BEC. In all considered cases spatial coupling results in an improvement of the BP threshold and our numerical results suggest that threshold saturation occurs if the coupling memory is chosen significantly large. The threshold improvement is larger for the serial ensembles, which are known to have poor BP thresholds without coupling but are stronger regarding the distance spectrum. Puncturing the serial and parallel ensembles to equal code rates, we observe that the threshold of the serial ensemble can surpass the one of the parallel ensemble.

[7] For $R = 1/3$ the SC-PCC is not punctured.

References

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

[2] S. Kudekar, T. Richardson and R. 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] M. Lentmaier, A. Sridharan, D.J. Costello, Jr. and K.Sh. Zigangirov, "Iterative decoding threshold analysis for LDPC convolutional codes", *IEEE Trans. Inf. Theory*, vol. 56, no. 10, pp. 5274–5289, Oct. 2010.

[4] M. Lentmaier, D. Truhachev and K. S. Zigangirov, "To the theory of low density convolutional codes II", *Problems of Inform. Transmission*, vol. 37, pp. 15–35, Oct.-Dec. 2001.

[5] E. Hall and S. Wilson, "Stream-oriented turbo codes", *IEEE Trans. Inf. Theory*, vol. 47, no. 5, pp. 1813–1831, Jul. 2001.

[6] K. Engdahl, M. Lentmaier and K. S. Zigangirov, "On the theory of low-density convolutional codes", *Lecture Notes in Computer Science (AAECC-13)*, vol. 1719, pp. 77–86, Springer-Verlag, New York 1999.

[7] A. Huebner, K. S. Zigangirov and D. J. Costello Jr., "Laminated turbo codes: A new class of block-convolutional codes", *IEEE Trans. Inf. Theory*, vol. 54, no. 7, pp. 3024–3034, Jul. 2008.

[8] 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.

[9] S. Moloudi and M. Lentmaier, "Density evolution analysis of braided convolutional codes on the erasure channel", in *Proc. IEEE Int. Symp. on Inf. Theory*, Honolulu, HI, USA, Jun. 2014.

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

[11] 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.

[12] B. Kurkoski, P. Siegel and J. Wolf, "Exact probability of erasure and a decoding algorithm for convolutional codes on the binary erasure channel", in *Proc. IEEE Global Telecommun. Conf. (GLOBECOM)*,, 2003.

[13] J. Shi and S. ten Brink, "Exact EXIT functions for convolutional codes over the binary erasure channel", in *Proc. Allerton Conf. Commun., Control, and Computing*, Monticello, IL, USA, 2006.

[14] A. Graell i Amat, G. Montorsi and F. Vatta, "Design and performance analysis of a new class of rate compatible serially concatenated convolutional codes", *IEEE Trans. Commun.*, vol. 57, no. 8, pp. 2280–2289, Aug. 2009.

[15] A. Graell i Amat, L. Rasmussen and F. Brännström, "Unifying analysis and design of rate-compatible concatenated codes", *IEEE Trans. Commun.*, vol. 59, no. 2, pp. 343–351, Feb. 2011.

[16] C. Measson, A. Montanari, T. Richardson and R. Urbanke, "The generalized area theorem and some of its consequences", *IEEE Trans. Inf. Theory*, vol. 55, no. 11, pp. 4793–4821, Nov. 2009.

Paper III

Spatially Coupled Turbo-Like Codes

In this paper, we introduce the concept of spatially coupled turbo-like codes (SC-TCs) as the spatial coupling of a number of turbo-like code ensembles. In particular, we consider the spatial coupling of parallel concatenated codes (PCCs), introduced by Berrou *et al.*, and that of serially concatenated codes (SCCs), introduced by Benedetto *et al.*. Furthermore, we propose two extensions of braided convolutional codes (BCCs), a class of turbo-like codes which have an inherent spatially coupled structure, to higher coupling memories, and show that these yield improved belief propagation (BP) thresholds as compared to the original BCC ensemble. We derive the exact density evolution (DE) equations for SC-TCs and analyze their asymptotic behavior on the binary erasure channel. We also consider the construction of families of rate-compatible SC-TC ensembles. Our numerical results show that threshold saturation of the belief propagation (BP) decoding threshold to the maximum a-posteriori threshold of the underlying uncoupled ensembles occurs for large enough coupling memory. The improvement of the BP threshold is especially significant for SCCs and BCCs, whose uncoupled ensembles suffer from a poor BP threshold. For a wide range of code rates, SC-TCs show close-to-capacity performance as the coupling memory increases. We further give a proof of threshold saturation for SC-TC ensembles with identical component encoders. In particular, we show that the DE of SC-TC ensembles with identical component encoders can be properly rewritten as a scalar recursion. This allows us to define potential functions and prove threshold saturation using the proof technique recently introduced by Yedla *et al.*.

S. Moloudi, M. Lentmaier, and A. Graell i Amat,
"Spatially coupled turbo-like codes,"
in *IEEE Trans. Inf. Theory*, vol. 63, no. 10, pp. 6199-6215, Oct. 2017.

1 Introduction

Low-density parity-check (LDPC) convolutional codes [1], also known as spatially coupled LDPC (SC-LDPC) codes [2], can be obtained from a sequence of individual LDPC block codes by distributing the edges of their Tanner graphs over several adjacent blocks [3]. The resulting spatially coupled codes exhibit a *threshold saturation* phenomenon, which has attracted a lot of interest in the past few years: The threshold of an iterative belief propagation (BP) decoder, obtained by density evolution (DE), can be improved to that of the optimal maximum-a-posteriori (MAP) decoder, for properly chosen parameters. It follows from threshold saturation that it is possible to achieve capacity by spatial coupling of simple regular LDPC codes, which show a significant gap between BP and MAP threshold in the uncoupled case. A first analytical proof of threshold saturation was given in [2] for the binary erasure channel (BEC), considering a specific ensemble with uniform random coupling. An alternative proof based on potential functions was then presented in [4], [5], [6], which was extended from scalar recursions to vector recursions in [7]. By means of vector recursions, the proof of threshold saturation can be extended to spatially coupled ensembles with structure, such as SC-LDPC codes based on protographs [8].

The concept of spatial coupling is not limited to LDPC codes. Also codes on graphs with stronger component codes can be considered. In this case the structure of the component codes has to be taken into account in a DE analysis. Instead of a simple check node update, a constraint node update within BP decoding of a generalized LDPC code involves an a-posteriori probability (APP) decoder applied to the associated component encoder. In general, the input/output transfer functions of the APP decoder are multi-dimensional because the output bits of the component encoder have different protection. For the BEC, however, it is possible to analytically derive explicit transfer functions [9] by means of a Markov chain analysis of the decoder metric values in a trellis representation of the considered code [10]. This technique was applied in [11], [12] to perform a DE analysis of braided block codes (BBCs) [13] and other spatially coupled generalized LDPC codes. Threshold saturation could be observed numerically in all the considered cases. BBCs can be seen as a spatially coupled version of product codes, and are closely related to staircase codes [14], which have been proposed for high-speed optical communications. It was demonstrated in [15], [16] that BBCs show excellent performance even with the iterative hard decision decoding that is proposed for such scenarios. The recently presented spatially coupled split-component codes [17] demonstrate the connections between BBCs and staircase codes.

In this paper, we study codes on graphs whose constraint nodes represent convolutional codes [18], [19], [20]. We denote such codes as turbo-like

codes (TCs). We consider three particular concatenated convolutional coding schemes: Parallel concatenated codes (PCCs) [21], serially concatenated codes (SCCs) [22], and braided convolutional codes (BCCs) [23]. Our aim is to investigate the impact of spatial coupling on the BP threshold of these TCs. For this purpose we introduce some special block-wise spatially coupled ensembles of PCCs (SC-PCCs) and SCCs (SC-SCCs) [24]. In the case of BCCs, which are inherently spatially coupled, we consider the original block-wise ensemble from [23], [25] and generalize it to larger coupling memories. Furthermore, we introduce a novel BCC ensemble in which not only the parity bits but also the information bits are coupled over several time instants [26].

For these spatially coupled turbo-like codes (SC-TCs), we perform a threshold analysis for the BEC analogously to [3], [11], [12]. We derive their exact DE equations from the transfer functions of the convolutional component decoders [27], [28], whose computation is similar to that for generalized LDPC codes in [10]. In order to evaluate and compare the ensembles at different rates, we also derive DE equations for the punctured ensembles. Using these equations, we compute BP thresholds for both coupled and uncoupled TCs [29] and compare them with the corresponding MAP thresholds [30], [31]. Our numerical results indicate that threshold saturation occurs if the coupling memory is chosen sufficiently large. The improvement of the BP threshold is specially significant for SCCs and BCCs, whose uncoupled ensembles suffer from a poor BP threshold. We then consider the construction of families of rate-compatible SC-TCs which achieve close-to-capacity performance for a wide range of code rates.

Motivated by the numerical results, we prove threshold saturation analytically. We show that, by few assumptions in the ensembles of uncoupled TCs, in particular considering identical component encoders, it is possible to rewrite their DE recursions in a form that corresponds to the recursion of a scalar admissible system. This representation allows us to apply the proof technique based on potential functions for scalar admissible systems proposed in [4], [5], which simplifies the analysis. For the general case, the analysis is significantly more complicated and requires the coupled vector recursion framework of [7]. Finally, for the example of PCCs, we generalize the proof to non-symmetric ensembles with different component encoders by using the framework in [7].

The remainder of the paper is organized as follows. In Section 2, we introduce a compact graph representation for the trellis of a convolutional code that is amenable for a DE analysis. Furthermore, we derive explicit input/output transfer functions of the BCJR decoder for transmission over the BEC. Then, in Section 3, we describe uncoupled ensembles of PCCs, SCCs and BCCs by means of the compact graph representation. SC-TCs, their spatially coupled counterparts, are introduced in Section 4. In Section 5, we derive exact DE equations for uncoupled and coupled ensembles of TCs. In Section 6, we consider random puncturing and derive the corresponding DE equations and analyze SC-TCs as

a family of rate compatible codes. Numerical results are presented and discussed in Section 7. Threshold saturation, which is observed numerically in the results section, is proved analytically in Section 8. Finally, the paper is concluded in Section 9.

2 Compact Graph Representation and Transfer Functions of Convolutional Codes

In this section, we introduce a graphical representation of a convolutional code, which can be seen as a compact form of its corresponding factor graph [20]. This compact graph representation makes the illustration of SC-TCs simpler and is convenient for the DE analysis. We also generalize the method in [27], [28] to derive explicit input/output transfer functions of the BCJR decoder of rate-k/n convolutional codes on the BEC, which will be used in Section 5 to derive the exact DE for SC-TCs.

2.1 Compact Graph Representation

Consider a rate-k/n systematic convolutional encoder of code length nN bits, i.e., its corresponding trellis has N trellis sections. At each time instant $\tau = 1, \ldots, N$, corresponding to a trellis section, the encoder encodes k input bits and generates $n-k$ parity bits. Let $\boldsymbol{u}^{(i)} = (u_1^{(i)}, u_2^{(i)}, \ldots, u_N^{(i)})$, $i = 1, \ldots, k$, and $\boldsymbol{v}_{\mathrm{p}}^{(i)} = (v_{\mathrm{p},1}^{(i)}, v_{\mathrm{p},2}^{(i)}, \ldots, v_{\mathrm{p},N}^{(i)})$, $i = 1, \ldots, n-k$, denote the k input sequences and the $n-k$ parity sequences, respectively. We also denote by $\boldsymbol{v}^{(i)} = (v_1^{(i)}, v_2^{(i)}, \ldots, v_N^{(i)})$, $i = 1, \ldots, n$, the ith code sequence, with $\boldsymbol{v}^{(i)} = \boldsymbol{u}^{(i)}$ for $i = 1, \ldots, k$ and $\boldsymbol{v}^{(i)} = \boldsymbol{v}_{\mathrm{p}}^{(i-k)}$ for $i = k+1, \ldots, n$. The conventional factor graph of a convolutional encoder is shown in Figure 1(a), where black circles represent code bits, each black square corresponds to the code constraints (allowed combinations of input state, input bits, output bits, and output state) of one trellis section, and the double circles are (hidden) state variable nodes.

For convenience, we will represent a convolutional encoder with the more compact graph representation depicted in Figure 1(b). In this compact graph representation, each input sequence $\boldsymbol{u}^{(i)}$ and each parity sequence $\boldsymbol{v}_{\mathrm{p}}^{(i)}$ is represented by a single black circle, referred to as variable node, i.e., each circle represents N bits. Furthermore, the code trellis is represented by a single empty square, referred to as factor node. The factor node is labeled by the length N of the trellis. Each node in the compact graph represents a sequence of nodes belonging to the same type, similar to the nodes in a protograph of an LDPC code. Variable nodes in the original factor graph may represent different bit

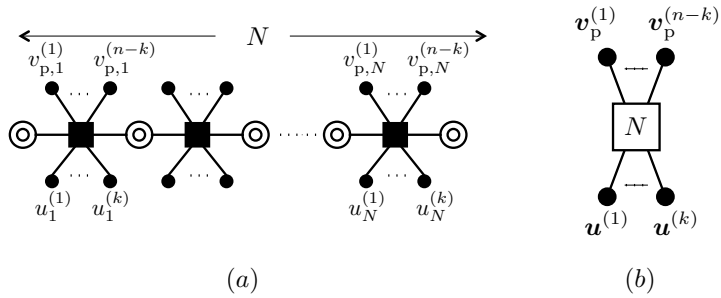


Figure 1: (a) Factor graph representation of a rate-k/n systematic convolutional code. (b) Compact graph representation of the same code.

values, even if they belong to the same type in the compact graph. However, assuming a tailbiting trellis, the probability distribution of these values after decoding will be equal for all variables that correspond to the same node type. As a consequence, a DE analysis can be performed in the compact graph, independently of the trellis length N, which plays a similar role as the lifting factor of a protograph ensemble. If a terminated convolutional encoder, which starts and ends in the zero state, is used instead, the bits that are close to the start and end of the trellis will have a slightly stronger protection. Since this effect will not have a significant impact on the performance, we will neglect this throughout this paper and assume equal output distributions for all bits of the trellis, even when termination is used.

2.2 Transfer Function of the BCJR Decoder of a Convolutional Code

Consider the BCJR decoder of a memory ν, rate$-k/n$ convolutional encoder and transmission over the BEC. Without loss of generality, we restrict ourselves within this paper to encoders with $k = 1$ or $n - k = 1$, which can be implemented with 2^ν states in controller canonical form or observer canonical form, respectively. We would like to characterize the transfer function between the input erasure probabilities (i.e., prior to decoding) and output erasure probabilities (i.e., after decoding) on both the input bits and the output bits of the convolutional encoder. Note that the erasure probabilities at the input of the decoder depend on both the channel erasure probability and the a-priori erasure probabilities on systematic and parity bits (provided, for example, by another decoder). Thus, in the more general case, we consider non-equal erasure probabilities at the input of the decoder.

Consider the extrinsic erasure probability of the lth code bit, $l = 1, 2, \ldots, n$,

which is the erasure probability of the lth code bit when it is estimated based on the other code bits[8]. This extrinsic erasure probability, at the output of the decoder, is denoted by p_l^{ext}. The probabilities p_l^{ext} depend on the erasure probabilities of all code bits (systematic and parity) at the input of the decoder,

$$p_l^{\text{ext}} = f_l(p_1, p_2, \ldots, p_n), \tag{1}$$

where p_l is the erasure probability of the lth code bit at the input of the decoder and $f_l(p_1, p_2, \ldots, p_n)$ is the transfer function of the BCJR decoder for the lth code bit. For notational simplicity, we will often omit the argument of $f_l(p_1, p_2, \ldots, p_n)$ and write simply f_l.

Let $\boldsymbol{r}^{(i)} = (r_1^{(i)}, r_2^{(i)}, \ldots, r_N^{(i)})$, $i = 1, \ldots, n$, be the vectors of received symbols at the output of the channel, with $r_j^{(i)} \in \{0, 1, ?\}$, where ? denotes an erasure. The branch metric of the trellis edge departing from state σ' at time $\tau - 1$ and ending to state σ at time τ, $\tau = 1, \ldots, N$, is

$$\gamma_\tau(\sigma', \sigma) = \prod_{l=1}^{n} p\left(r_\tau^{(l)} \mid v_\tau^{(l)}\right) \cdot p\left(v_\tau^{(l)}\right), \tag{2}$$

where $p\big(v_\tau^{(l)}\big)$ is the a-priori probability on symbol $v_\tau^{(l)}$.

The forward and backward metrics of the BCJR decoder are

$$\alpha_\tau(\sigma) = \sum_{\sigma'} \gamma_\tau(\sigma', \sigma) \cdot \alpha_{\tau-1}(\sigma') \tag{3}$$

$$\beta_{\tau-1}(\sigma') = \sum_{\sigma} \gamma_\tau(\sigma', \sigma) \cdot \beta_\tau(\sigma'). \tag{4}$$

Finally, the extrinsic output likelihood ratio is given by

$$L_{\text{out},\tau}^{(l)} = \frac{\sum\limits_{(\sigma',\sigma):v_\tau^{(l)}=0} \alpha_{\tau-1}(\sigma') \cdot \gamma_\tau(\sigma', \sigma) \cdot \beta_\tau(\sigma)}{\sum\limits_{(\sigma',\sigma):v_\tau^{(l)}=1} \big(\alpha_{\tau-1}(\sigma') \cdot \gamma_\tau(\sigma', \sigma) \cdot \beta_\tau(\sigma)} \cdot \frac{p\left(v_\tau^{(l)} = 1\right)}{p\left(v_\tau^{(l)} = 0\right)}.$$

Let the 2^ν trellis states be $s_1, s_2, \ldots, s_{2^\nu}$. Then, we define the forward and backward metric vectors as

$$\boldsymbol{\alpha}_\tau = (\alpha_\tau(s_1), \ldots, \alpha_\tau(s_{2^\nu})),$$

and

$$\boldsymbol{\beta}_\tau = (\beta_\tau(s_1), \ldots, \beta_\tau(s_{2^\nu})),$$

[8]Without loss of generality we assume that the first k bits are the systematic bits.

respectively. For transmission on the BEC, the nonzero entries of vectors $\boldsymbol{\alpha}_\tau$ and $\boldsymbol{\beta}_\tau$ are all equal. Thus, we can normalize them to 1.

We consider transmission of the all-zero codeword. The sets of values that vectors $\boldsymbol{\alpha}_\tau$ and $\boldsymbol{\beta}_\tau$ can take on are denoted by

$$\mathcal{M}_\alpha = \{\boldsymbol{m}_\alpha^{(1)}, \ldots, \boldsymbol{m}_\alpha^{(|\mathcal{M}_\alpha|)}\},$$

and

$$\mathcal{M}_\beta = \{\boldsymbol{m}_\beta^{(1)}, \ldots, \boldsymbol{m}_\beta^{(|\mathcal{M}_\beta|)}\},$$

respectively. It is important to remark that these sets are finite. Furthermore, the sequence $\ldots, \boldsymbol{\alpha}_{\tau-1}, \boldsymbol{\alpha}_\tau, \boldsymbol{\alpha}_{\tau+1}, \ldots$ forms a Markov chain, which can be properly described by a probability transition matrix, denoted by $\boldsymbol{M}_\alpha$. The (i, j) entry of $\boldsymbol{M}_\alpha$ is the probability of transition from state $\boldsymbol{m}_\alpha^{(i)}$ to state $\boldsymbol{m}_\alpha^{(j)}$. Denote the steady state distribution vector of the Markov chain by $\boldsymbol{\pi}_\alpha$, which can be computed as the solution to

$$\boldsymbol{\pi}_\alpha = \boldsymbol{M}_\alpha \cdot \boldsymbol{\pi}_\alpha. \tag{5}$$

Similarly, we can define the transition matrix for the sequence of backward metrics $\ldots, \boldsymbol{\beta}_{\tau+1}, \boldsymbol{\beta}_\tau, \boldsymbol{\beta}_{\tau-1}, \ldots$, denoted by $\boldsymbol{M}_\beta$, and compute the steady state distribution vector $\boldsymbol{\pi}_\beta$.

Example 2 *Consider the rate-2/3, 4-state convolutional encoder with generator matrix*

$$\boldsymbol{G}(D) = \begin{pmatrix} 1 & 0 & \frac{1}{1+D+D^2} \\ 0 & 1 & \frac{1+D^2}{1+D+D^2} \end{pmatrix}.$$

$\mathcal{M}_\alpha$ and $\mathcal{M}_\beta$ are equal and have cardinality 5,

$$\mathcal{M}_\alpha = \mathcal{M}_\beta = \{(1,0,0,0), (1,1,0,0), (1,0,0,1), (1,0,1,0), (1,1,1,1)\}.$$

Consider equal erasure probability for all code bits at the input of the decoder, i.e., $p_1 = p_2 = p_3 = p$. Then,

$$\boldsymbol{M}_\alpha = \begin{bmatrix} (1-p)^2(2p+1) & (1-p)^2 & (1-p)^3 & 0 & 0 \\ p^2(1-p) & 0 & p(1-p)^2 & p^3-2p+1 & (1-p)^2 \\ p^2(1-p) & p(1-p) & p(1-p)^2 & 0 & 0 \\ p^2(1-p) & p(1-p) & p(1-p)^2 & 0 & 0 \\ p^3 & p^2 & p^2(3-2p) & p^2(2-p) & p(2-p) \end{bmatrix}.$$

△

In order to compute the erasure probability of the lth bit at the output of the decoder, we have to compute the probability of $L^{(l)}_{\text{out},\tau} = 1$. Define the matrices $\boldsymbol{T}_l$, $l = 1, 2, \ldots, n$, where the (i, j) entry of $\boldsymbol{T}_l$ is computed as

$$T_l(i,j) = p\left(L^{(l)}_{\text{out},\tau} = 1 \mid \boldsymbol{\alpha}_\tau = \boldsymbol{m}^{(i)}_\alpha, \boldsymbol{\beta}_{\tau+1} = \boldsymbol{m}^{(j)}_\beta\right).$$

Thenhresho, the extrinsic erasure probability of the lth output, p^{ext}_l, introduced in (1), is obtained as

$$\begin{aligned} p^{\text{ext}}_l &= f_l(p_1, p_2, \ldots, p_n) = p\left(L^{(l)}_{\text{out},\tau} = 1\right) \\ &= \sum_{i=1}^{|\mathcal{M}_\alpha|} \sum_{j=1}^{|\mathcal{M}_\beta|} p\left(L^{(l)}_{\text{out},\tau} = 1 \mid \boldsymbol{\alpha}_\tau = \boldsymbol{m}^{(i)}_\alpha, \boldsymbol{\beta}_{\tau+1} = \boldsymbol{m}^{(j)}_\beta\right) \\ &\qquad \cdot p\left(\boldsymbol{\alpha}_\tau = \boldsymbol{m}^{(i)}_\alpha\right) \cdot p\left(\boldsymbol{\beta}_{\tau+1} = \boldsymbol{m}^{(j)}_\beta\right) \\ &= \boldsymbol{\pi}_\alpha \cdot \boldsymbol{T}_l \cdot \boldsymbol{\pi}_\beta. \end{aligned} \tag{6}$$

Example 3 *Consider the rate−2/3 convolutional encoder with generator matrix*

$$\boldsymbol{G}(D) = \begin{pmatrix} 1 & 0 & \frac{1}{1+D} \\ 0 & 1 & \frac{D}{1+D} \end{pmatrix}.$$

Assuming $p_1 = p_2 = p_3 \triangleq p$, the transfer functions for the corresponding decoder are

$$f_1 = f_2 = \frac{p(p^5 - 4p^4 + 6p^3 - 5p^2 + 2p + 1)}{p^6 - 4p^5 + 6p^4 - 6p^3 + 5p^2 - 2p + 1},$$

$$f_3 = \frac{p^2(p^2 - 4p + 4)}{p^6 - 4p^5 + 6p^4 - 6p^3 + 5p^2 - 2p + 1}.$$

△

Lemma 1 *Consider a terminated convolutional encoder where all distinct input sequences have distinct encoded sequences. For such a system, the transfer function $f(p_1, p_2, \ldots, p_n)$ of a BCJR decoder with input erasure probabilities $p_1, p_2, \ldots, p_n$, or any convex combination of such transfer functions, is increasing in all its arguments.*

Proof 1 *We prove the statement by contradiction. Recall that the BCJR decoder is an optimal APP decoder. Now, consider the transmission of the same codeword over two channels, called channel 1 and 2. The erasure probabilities of the ith bit at the input of the decoder are denoted by $p^{(1)}_i$ and $p^{(2)}_i$ for transmission over channel 1 and 2, respectively. These erasure probabilities are*

equal for all $i = 1, \ldots, n$ *except for the* j*th bit, for which* $p_j^{(1)} < p_j^{(2)}$*. Assume that the transfer function* f *is non-increasing in its* j*th argument,*

$$f(p_1^{(1)}, \ldots, p_j^{(1)}, \ldots, p_n^{(1)}) \geq f(p_1^{(2)}, \ldots, p_j^{(2)}, \ldots, p_n^{(2)}). \tag{7}$$

Puncture the j*th bit sequence of the codeword transmitted over channel 1 such that* $p_{j,punc}^{(1)} = p_j^{(2)}$*. Since puncturing can only make the output of the decoder worse (otherwise we could replace our encoder with the punctured one and achieve a higher rate),*

$$f(p_1^{(1)}, \ldots, p_{j,punc}^{(1)}, \ldots, p_n^{(1)}) > f(p_1^{(1)}, \ldots, p_j^{(1)}, \ldots, p_n^{(1)}), \tag{8}$$

Since after puncturing $p_i^{(1)}$ *and* $p_i^{(2)}$ *are equal for all* i*, then* $f(p_1^{(1)}, \ldots, p_{j,punc}^{(1)}, \ldots, p_n^{(1)}) = f(p_1^{(2)}, \ldots, p_j^{(2)}, \ldots, p_n^{(2)})$*. Then, we can rewrite the inequality* (8) *as*

$$f(p_1^{(2)}, \ldots, p_j^{(2)}, \ldots, p_n^{(2)}) > f(p_1^{(1)}, \ldots, p_j^{(1)}, \ldots, p_n^{(1)}). \tag{9}$$

However, the inequality (9) *is in contradiction with* (7)*.*

3 Compact Graph Representation of Uncoupled Turbo-Like Codes

In this section, we describe PCCs, SCCs and BCCs using the compact graph representation introduced in the previous section. In Section 4, we then introduce the corresponding spatially coupled ensembles.

3.1 Parallel Concatenated Codes

We consider a rate $R = 1/3$ PCC built from two rate-1/2 recursive systematic convolutional encoders, referred to as the upper and lower component encoder. Its conventional factor graph is shown in Figure 2(a), where Π denotes the permutation. The trellises corresponding to the upper and lower encoders are denoted by T^{U} and T^{L}, respectively. The information sequence $\boldsymbol{u}$, of length N bits, and a reordered copy are encoded by the upper and lower encoder, respectively, to produce the parity sequences $\boldsymbol{v}^{\mathrm{U}}$ and $\boldsymbol{v}^{\mathrm{L}}$. The code sequence is denoted by $\boldsymbol{v} = (\boldsymbol{u}, \boldsymbol{v}^{\mathrm{U}}, \boldsymbol{v}^{\mathrm{L}})$. The compact graph representation of the PCC is shown in Figure 2(b), where each of the sequences $\boldsymbol{u}$, $\boldsymbol{v}^{\mathrm{U}}$ and $\boldsymbol{v}^{\mathrm{L}}$ is represented by a single variable node and the trellises are replaced by factor nodes T^{U} and T^{L} (cf. Figure 1). In order to emphasize that a reordered copy of the input sequence is used in T^{L}, the permutation is depicted by a line that crosses the edge which connects $\boldsymbol{u}$ to T^{L}.

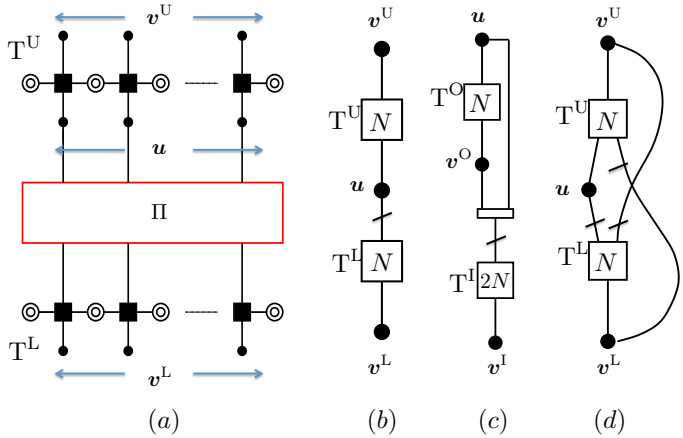


Figure 2: (a) Conventional factor graph of a PCC. Compact graph representation of a (b) PCC, (c) SCC, (d) BCC.

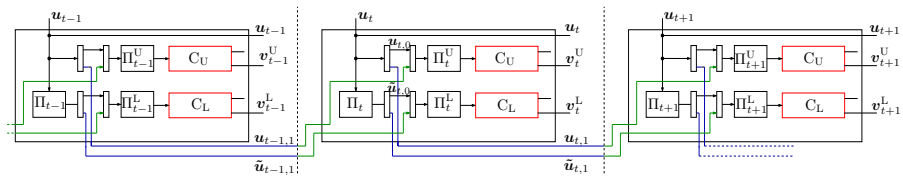


Figure 3: Block diagram of the encoder of a SC-PCC ensemble with $m = 1$.

3.2 Serially Concatenated Codes

We consider a rate $R = 1/4$ SCC built from the serial concatenation of two rate-1/2 recursive systematic component encoders, referred to as the outer and inner component encoder. Its compact graph representation is shown in Figure 2(c), where T^{O} and T^{I} are the factor nodes corresponding to the outer and inner encoder, respectively, and the rectangle illustrates a multiplexer/demultiplexer. The information sequence $\boldsymbol{u}$, of length N, is encoded by the outer encoder to produce the parity sequence $\boldsymbol{v}^{\mathrm{O}}$. Then, the sequences $\boldsymbol{u}$ and $\boldsymbol{v}^{\mathrm{O}}$ are multiplexed and reordered to create the intermediate sequence $\tilde{\boldsymbol{v}}^{\mathrm{O}}$, of length $2N$ (not shown in the graph). Finally, $\tilde{\boldsymbol{v}}^{\mathrm{O}}$ is encoded by the inner encoder to produce the parity sequence $\boldsymbol{v}^{\mathrm{I}}$. The transmitted sequence is $\boldsymbol{v} = (\boldsymbol{u}, \boldsymbol{v}^{\mathrm{O}}, \boldsymbol{v}^{\mathrm{I}})$.

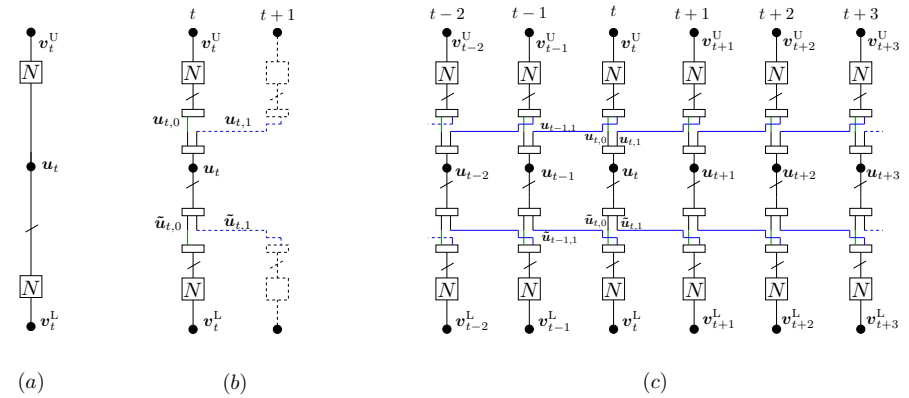


Figure 4: Compact graph representation of (a) PCC, (b) SC-PCC at time instant t, (c) SC-PCC.

3.3 Braided Convolutional Codes

We consider a rate $R = 1/3$ BCC built from two rate-2/3 recursive systematic convolutional encoders, referred to as upper and lower encoders. The corresponding trellises are denoted by T^{U} and T^{L}. The compact graph representation of this code is shown in Figure 2(d). The parity sequences of the upper and lower encoder are denoted by $\boldsymbol{v}^{\mathrm{U}}$ and $\boldsymbol{v}^{\mathrm{L}}$, respectively. To produce the parity sequence $\boldsymbol{v}^{\mathrm{U}}$, the information sequence $\boldsymbol{u}$ and a reordered copy of $\boldsymbol{v}^{\mathrm{L}}$ are encoded by T^{U}. Likewise, a reordered copy of $\boldsymbol{u}$ and a reordered copy of $\boldsymbol{v}^{\mathrm{U}}$ are encoded by T^{L} in order to produce the parity sequence $\boldsymbol{v}^{\mathrm{L}}$. Similarly to PCCs, the transmitted sequence is $\boldsymbol{v} = (\boldsymbol{u}, \boldsymbol{v}^{\mathrm{U}}, \boldsymbol{v}^{\mathrm{L}})$.

4 Spatially Coupled Turbo-Like Codes

In this section, we introduce SC-TCs. We first describe the spatial coupling for both PCCs and SCCs. Then, we generalize the original block-wise BCC ensemble [23] in order to obtain ensembles with larger coupling memories.

4.1 Spatially Coupled Parallel Concatenated Codes

We consider the spatial coupling of rate-1/3 PCCs, described in the previous section. For simplicity, we first describe the SC-PCC ensemble with coupling memory $m = 1$. Then we show the coupling for higher coupling memories. The block diagram of the encoder for the SC-PCC ensemble is shown in Figure 3.

In addition, its compact graph representation and the coupling are illustrated in Figure 4.

As it is shown in Figure 3 and Figure 4(a) we denote by $\boldsymbol{u}_t$ the information sequence, and by $\boldsymbol{v}_t^{\mathrm{U}}$ and $\boldsymbol{v}_t^{\mathrm{L}}$ the parity sequence of the upper and lower encoder, respectively, at time t. The code sequence of the PCC at time t is given by the triple $\boldsymbol{v}_t = (\boldsymbol{u}_t, \boldsymbol{v}_t^{\mathrm{U}}, \boldsymbol{v}_t^{\mathrm{L}})$. With reference to Figure 3 and Figure 4(b), in order to obtain the coupled sequence, the information sequence, $\boldsymbol{u}_t$, is divided into two sequences of equal size, $\boldsymbol{u}_{t,0}$ and $\boldsymbol{u}_{t,1}$ by a multiplexer. Then, the sequence $\boldsymbol{u}_{t,0}$ is used as a part of the input to the upper encoder at time t and $\boldsymbol{u}_{t,1}$ is used as a part of the input to the upper encoder at time $t+1$. Likewise, a reordered copy of the information sequence, $\tilde{\boldsymbol{u}}_t$, is divided into two sequences $\tilde{\boldsymbol{u}}_{t,0}$ and $\tilde{\boldsymbol{u}}_{t,1}$.

Therefore, the input to the upper encoder at time t is a reordered copy of $(\boldsymbol{u}_{t,0}, \boldsymbol{u}_{t-1,1})$, and likewise the input to the lower encoder at time t is a reordered copy of $(\tilde{\boldsymbol{u}}_{t,0}, \tilde{\boldsymbol{u}}_{t-1,1})$. In this ensemble, the coupling memory is $m = 1$ as $\boldsymbol{u}_t$ is used only at the time instants t and $t+1$.

Finally, an SC-PCC with $m = 1$ is obtained by considering a collection of L PCCs at time instants $t = 1, \ldots, L$, where L is referred to as the coupling length, and coupling them as described above, see Figure 4(c).

An SC-PCC ensemble with coupling memory m is obtained by dividing each of the sequences $\boldsymbol{u}_t$ and $\tilde{\boldsymbol{u}}_t$ into $m+1$ sequences of equal size and spread these sequences respectively to the input of the upper and the lower encoder at time slots t to $t+m$. The compact graph representation of the SC-PCC with coupling memory m is shown in Figure 5(a) for a given time instant t. The coupling is performed as follows. Divide the information sequence $\boldsymbol{u}_t$ into $m+1$ sequences of equal size $N/(m+1)$, denoted by $\boldsymbol{u}_{t,j}$, $j = 0, \ldots, m$. Likewise, divide $\tilde{\boldsymbol{u}}_t$, the information sequence $\boldsymbol{u}_t$ reordered by a permutation, into $m+1$ sequences of equal size, denoted by $\tilde{\boldsymbol{u}}_{t,j}$, $j = 0, \ldots, m$. At time t, the information sequence at the input of the upper encoder is $(\boldsymbol{u}_{t,0}, \boldsymbol{u}_{t-1,1}, \ldots, \boldsymbol{u}_{t-m,m})$, properly reordered by a permutation. Likewise, the information sequence at the input of the lower encoder is $(\tilde{\boldsymbol{u}}_{t,0}, \tilde{\boldsymbol{u}}_{t-1,1}, \ldots, \tilde{\boldsymbol{u}}_{t-m,m})$, reordered by a permutation. Using the procedure described above, a coupled chain (a convolutional structure over time) of L PCCs with coupling memory m is obtained.

In order to terminate the encoder of the SC-PCC to the zero state, the information sequences at the end of the chain are chosen in such a way that the code sequences become $\boldsymbol{v}_t = \mathbf{0}$ at time $t = L+1, \ldots, L+m$, and $\boldsymbol{u}_t$ is set to $\mathbf{0}$ for $t > L$. Analogously to conventional convolutional codes, this results in a rate loss that becomes smaller as L increases.

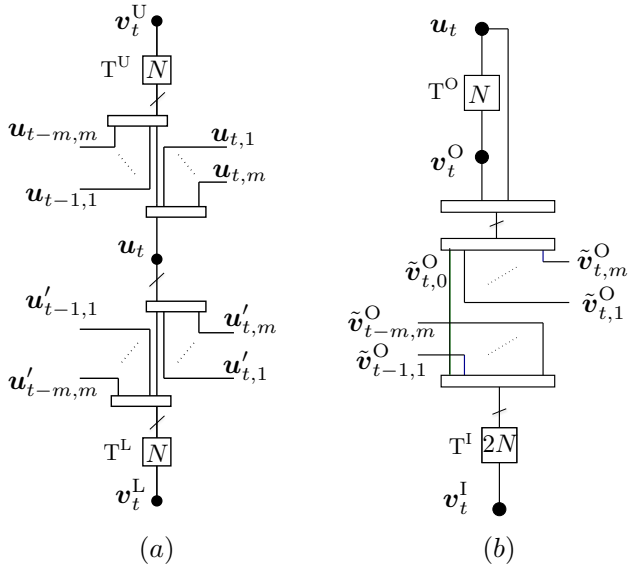


Figure 5: Compact graph representation of (a) SC-PCCs, and (b) SC-SCCs of coupling memory m for time instant t.

4.2 Spatially Coupled Serially Concatenated Codes

An SC-SCC is constructed similarly to SC-PCCs. Consider a collection of L SCCs at time instants $t = 1, \ldots, L$, and let $\boldsymbol{u}_t$ be the information sequence at time t. Also, denote by $\boldsymbol{v}_t^{\mathrm{O}}$ and $\boldsymbol{v}_t^{\mathrm{I}}$ the parity sequence at the output of the outer and inner encoder, respectively. The information sequence $\boldsymbol{u}_t$ and the parity sequence $\boldsymbol{v}_t^{\mathrm{O}}$ are multiplexed and reordered into the sequence $\tilde{\boldsymbol{v}}_t^{\mathrm{O}}$. The sequence $\tilde{\boldsymbol{v}}_t^{\mathrm{O}}$ is divided into $m+1$ sequences of equal length, denoted by $\tilde{\boldsymbol{v}}_{t,j}^{\mathrm{O}}$, $j = 0, \ldots, m$. Then, at time instant t, the sequence at the input of the inner encoder is $(\tilde{\boldsymbol{v}}_{t-j,0}^{\mathrm{O}}, \tilde{\boldsymbol{v}}_{t-1,1}^{\mathrm{O}} \ldots, \tilde{\boldsymbol{v}}_{t-m,m}^{\mathrm{O}})$, properly reordered by a permutation. This sequence is encoded by the inner encoder into $\boldsymbol{v}_t^{\mathrm{I}}$. Finally, the code sequence at time t is $\boldsymbol{v} = (\boldsymbol{u}_t, \boldsymbol{v}_t^{\mathrm{O}}, \boldsymbol{v}_t^{\mathrm{I}})$. Using this construction method, a coupled chain of L SCCs with coupling memory m is obtained. The compact graph representation of SC-SCCs with coupling memory m is shown in Figure 5(b) for time instant t.

In order to terminate the encoder of the SC-SCC, the information sequences at the end of the chain are chosen in such a way that the code sequences become $\boldsymbol{v}_t = \mathbf{0}$ at time $t = L+1, \ldots, L+m$. A simple and practical way to terminate SC-SCCs is to set $\boldsymbol{u}_t = \mathbf{0}$ for $t = L-m+1, \ldots, L$. This enforces $\boldsymbol{v}_t = \mathbf{0}$ for $t = L+1, \ldots, L+m$, since we can assume that $\boldsymbol{u}_t = \mathbf{0}$ for $t > L$. Using this

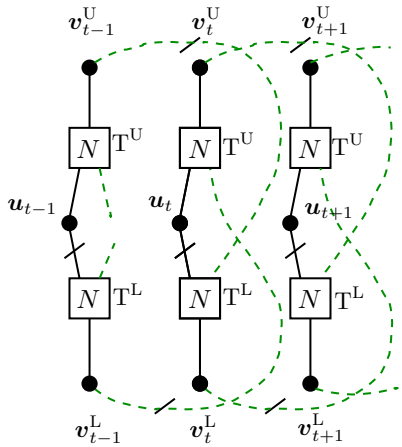


Figure 6: Compact graph representation of the original BCCs.

termination technique, only the parity sequence $\boldsymbol{v}^{\mathrm{I}}_t$ needs to be transmitted at time instants $t = L - m + 1, \ldots, L$.

4.3 Braided Convolutional Codes

The compact graph representation of the original BCCs is depicted in Fig 6. As for SC-PCCs, let $\boldsymbol{u}_t$, $\boldsymbol{v}^{\mathrm{U}}_t$ and $\boldsymbol{v}^{\mathrm{L}}_t$ denote the information sequence, the parity sequence at the output of the upper encoder, and the parity sequence at the output of the lower encoder, respectively, at time t. At time t, the information sequence $\boldsymbol{u}_t$ and a reordered copy of $\boldsymbol{v}^{\mathrm{L}}_{t-1}$ are encoded by the upper encoder to generate the parity sequence $\boldsymbol{v}^{\mathrm{U}}_t$. Likewise, a reordered copy of the information sequence, denoted by $\tilde{\boldsymbol{u}}_t$, and a reordered copy of $\boldsymbol{v}^{\mathrm{L}}_{t-1}$ are encoded by the lower encoder to produce the parity sequence $\boldsymbol{v}^{\mathrm{L}}_t$. The code sequence at time t is $\boldsymbol{v} = (\boldsymbol{u}_t, \boldsymbol{v}^{\mathrm{U}}_t, \boldsymbol{v}^{\mathrm{L}}_t)$.

As it can be seen from Fig 6, the original BCCs are inherently spatially coupled codes[9] with coupling memory one. In the following, we introduce two extensions of BCCs, referred to as Type-I and Type-II, with increased coupling memory, $m > 1$.

The compact graph of Type-I BCCs is shown in Figure 7(a) for time instant t. The parity sequence $\boldsymbol{v}^{\mathrm{U}}_t$ is randomly divided into m sequences $\boldsymbol{v}^{\mathrm{U}}_{t,j}$, $j = 1, \ldots, m$, of the same length. Likewise, the parity sequence $\boldsymbol{v}^{\mathrm{L}}_t$ is randomly divided into m sequences $\boldsymbol{v}^{\mathrm{L}}_{t,j}$, $j = 1, \ldots, m$. At time t, the information se-

[9]The uncoupled ensemble, discussed in the previous section, can be defined by tailbiting a coupled chain of length $L = 1$.

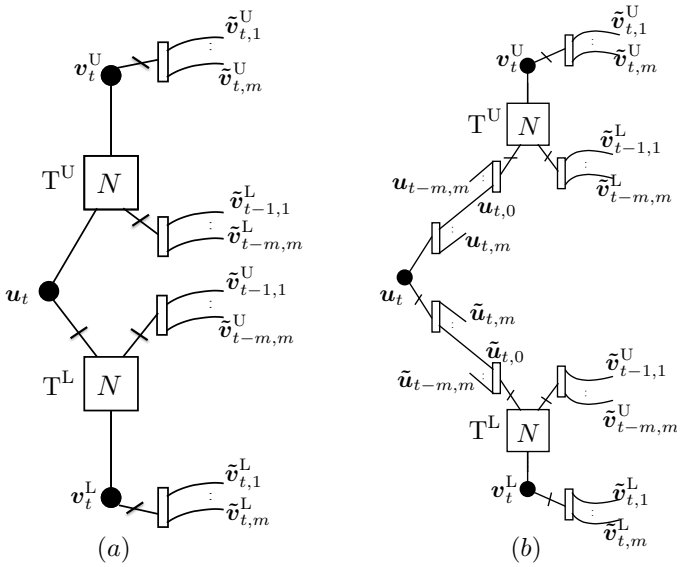


Figure 7: Compact graph representation of (a) Type-I BCCs, and (b) Type-II BCCs of coupling memory m at time instant t.

quence $\boldsymbol{u}_t$ and the sequence $(\boldsymbol{v}^{\mathrm{L}}_{t-1,1}, \boldsymbol{v}^{\mathrm{L}}_{t-2,2}, \ldots, \boldsymbol{v}^{\mathrm{L}}_{t-m,m})$, properly reordered, are used as input sequences to the upper encoder to produce the parity sequence $\boldsymbol{v}^{\mathrm{U}}_t$. Likewise, a reordered copy of the information sequence $\boldsymbol{u}_t$ and the sequence $(\boldsymbol{v}^{\mathrm{U}}_{t-1,1}, \boldsymbol{v}^{\mathrm{U}}_{t-2,2}, \ldots, \boldsymbol{v}^{\mathrm{U}}_{t-m,m})$, properly reordered, are encoded by the lower encoder to produce the parity sequence $\boldsymbol{v}^{\mathrm{L}}_t$.

The compact graph of Type-II BCCs is shown in Figure 7(b) for time instant t. Contrary to Type-I BCCs, in addition to the coupling of parity bits, for Type-II BCCs information bits are also coupled. At time t, divide the information sequence $\boldsymbol{u}_t$ into $m+1$ sequences $\boldsymbol{u}_{t,j}$, $j = 0, \ldots, m$ of equal length. Furthermore, divide the reordered copy of the information sequence, $\tilde{\boldsymbol{u}}_t$, into $m+1$ sequences $\tilde{\boldsymbol{u}}_{t,j}$, $j = 0, \ldots, m$. The first input of the upper and lower encoders are now the sequences $(\boldsymbol{u}_{t-0,0}, \boldsymbol{u}_{t-1,1}, \ldots, \boldsymbol{u}_{t-m,m})$ and $(\tilde{\boldsymbol{u}}_{t-0,0}, \tilde{\boldsymbol{u}}_{t-1,1}, \ldots, \tilde{\boldsymbol{u}}_{t-m,m})$, respectively, properly reordered.

5 Density Evolution Analysis for SC-TCs over the Binary Erasure Channel

In this section we derive the exact DE for SC-TCs. For the three considered code ensembles, we first derive the DE equations for the uncoupled ensembles

and then extend them to the coupled ones.

5.1 Density Evolution Equations and Decoding Thresholds

For transmission over the BEC, it is possible to analyze the asymptotic behavior of TCs and SC-TCs by tracking the evolution of the erasure probability with the number of decoding iterations. This evolution can be formalized in a compact way as a set of equations called DE equations. For the BEC, it is possible to derive a exact DE equations for TCs and SC-TCs. By use of these equations, the BP decoding threshold can be computed. The BP threshold is the largest channel erasure probability ε for which the erasure probability at the output of the BP decoder converges to zero as the block length and number of iterations grow to infinity.

It is also possible to compute the threshold of the MAP decoder, $\varepsilon_{\mathrm{MAP}}$, by the use of the area theorem [31]. According to the area theorem, the MAP threshold[10] can be obtained from the following equation,

$$\int_{\varepsilon_{\mathrm{MAP}}}^{1} \bar{p}_{\mathrm{extr}}(\varepsilon) d\varepsilon = R \ ,$$

where R is the rate of the code and $\bar{p}_{\mathrm{extr}}(\varepsilon)$ is the average extrinsic erasure probability for all transmitted bits.

5.2 Parallel Concatenated Codes

Uncoupled

Consider the compact graph of a PCC in Figure 2(b). Let $p_{\mathrm{U,s}}^{(i)}$ and $p_{\mathrm{U,p}}^{(i)}$ denote the average extrinsic erasure probability from factor node T^{U} to $\boldsymbol{u}$ and $\boldsymbol{v}^{\mathrm{U}}$, respectively, in the ith iteration.[11] Likewise, denote by $p_{\mathrm{L,s}}^{(i)}$ and $p_{\mathrm{L,p}}^{(i)}$ the extrinsic erasure probabilities from T^{L} to $\boldsymbol{u}$ and $\boldsymbol{v}^{\mathrm{L}}$, respectively. It is easy to see that the erasure probability from $\boldsymbol{u}_t$ and $\boldsymbol{v}_t^{\mathrm{U}}$ to T^{U} is $\varepsilon \cdot p_{\mathrm{L,s}}^{(i-1)}$ and ε, respectively. Therefore, the DE updates for T^{U} can be written as

$$p_{\mathrm{U,s}}^{(i)} = f_{\mathrm{U,s}}\left(q_{\mathrm{L}}^{(i)}, \varepsilon\right), \tag{10}$$

$$p_{\mathrm{U,p}}^{(i)} = f_{\mathrm{U,p}}\left(q_{\mathrm{L}}^{(i)}, \varepsilon\right), \tag{11}$$

[10] The threshold given by the area theorem is actually an upper bound on the MAP threshold. However, the numerical results show that the thresholds of the coupled ensembles converge to this upper bound. This indicates that the upper bound on the MAP threshold is a tight bound.

[11] With some abuse of language, we sometimes refer to a variable node representing a sequence (e.g., $\boldsymbol{u}$) as the sequence itself ($\boldsymbol{u}$ in this case).

where

$$q_{\mathrm{L}}^{(i)} = \varepsilon \cdot p_{\mathrm{L,s}}^{(i-1)}, \tag{12}$$

and $f_{\mathrm{U,s}}$ and $f_{\mathrm{U,p}}$ denote the transfer function of T^{U} for the systematic and parity bits, respectively.

Similarly, the DE update for T^{L} can be written as

$$p_{\mathrm{L,s}}^{(i)} = f_{\mathrm{L,s}}\left(q_{\mathrm{U}}^{(i)}, \varepsilon\right), \tag{13}$$

$$p_{\mathrm{L,p}}^{(i)} = f_{\mathrm{L,p}}\left(q_{\mathrm{U}}^{(i)}, \varepsilon\right), \tag{14}$$

where

$$q_{\mathrm{U}}^{(i)} = \varepsilon \cdot p_{\mathrm{U,s}}^{(i-1)}, \tag{15}$$

and $f_{\mathrm{L,s}}$ and $f_{\mathrm{L,p}}$ are the transfer functions of T^{L} for the systematic and parity bits, respectively.

Coupled

Consider the compact graph of a SC-PCC ensemble in Figure 5(a). The variable node $\boldsymbol{u}_t$ is connected to factor nodes $T_{t'}^{\mathrm{U}}$ and $T_{t'}^{\mathrm{L}}$, at time instants $t' = t, \ldots, t+m$. We denote by $p_{\mathrm{U,s}}^{(i,t')}$ and $p_{\mathrm{U,p}}^{(i,t')}$ the average extrinsic erasure probability from factor node $T_{t'}^{\mathrm{U}}$ at time instant t' to $\boldsymbol{u}$ and $\boldsymbol{v}^{\mathrm{U}}$, respectively, computed in the ith iteration. We also denote by $\bar{q}_{\mathrm{U}}^{(i-1,t)}$ the input erasure probability to variable node $\boldsymbol{u}_t$ in the ith iteration, received from its neighbors $T_{t'}^{\mathrm{U}}$. It can be written as

$$\bar{q}_{\mathrm{U}}^{(i-1,t)} = \frac{1}{m+1} \sum_{j=0}^{m} p_{\mathrm{U,s}}^{(i-1,t+j)}. \tag{16}$$

Similarly, the average erasure probability from factor nodes $T_{t'}^{\mathrm{L}}$, $t' = t, \ldots, t+m$, to $\boldsymbol{u}_t$, denoted by $\bar{q}_{\mathrm{L}}^{(i-1,t)}$, can be written as

$$\bar{q}_{\mathrm{L}}^{(i-1,t)} = \frac{1}{m+1} \sum_{j=0}^{m} p_{\mathrm{L,s}}^{(i-1,t+j)}. \tag{17}$$

The erasure probabilities from variable node $\boldsymbol{u}_t$ to its neighbors $T_{t'}^{\mathrm{U}}$ and $T_{t'}^{\mathrm{L}}$ are $\varepsilon \cdot \bar{q}_{\mathrm{L}}^{(i-1,t)}$ and $\varepsilon \cdot \bar{q}_{\mathrm{U}}^{(i-1,t)}$, respectively.

On the other hand, T_t^{U} at time t is connected to the set of $\boldsymbol{u}_{t'}$s for $t' = t-m, \ldots, t$. The erasure probability to T_t^{U} from this set, denoted by $q_{\mathrm{L}}^{(i,t)}$, is

given by

$$q_{\mathrm{L}}^{(i,t)} = \varepsilon \cdot \frac{1}{m+1} \sum_{k=0}^{m} \bar{q}_{\mathrm{L}}^{(i-1,t-k)}$$
$$= \varepsilon \cdot \frac{1}{(m+1)^2} \sum_{k=0}^{m} \sum_{j=0}^{m} p_{\mathrm{L,s}}^{(i-1,t+j-k)}. \tag{18}$$

Thus, the DE updates of T_t^{U} are

$$p_{\mathrm{U,s}}^{(i,t)} = f_{\mathrm{U,s}}\left(q_{\mathrm{L}}^{(i,t)}, \varepsilon\right), \tag{19}$$
$$p_{\mathrm{U,p}}^{(i,t)} = f_{\mathrm{U,p}}\left(q_{\mathrm{L}}^{(i,t)}, \varepsilon\right). \tag{20}$$

Similarly, the input erasure probability to T_t^{L} from the set of connected $\boldsymbol{u}_{t'}$s at time instants $t' = t-m, \ldots, t$, is

$$q_{\mathrm{U}}^{(i,t)} = \varepsilon \cdot \frac{1}{m+1} \sum_{k=0}^{m} \bar{q}_{\mathrm{U}}^{(i-1,t-k)}$$
$$= \varepsilon \cdot \frac{1}{(m+1)^2} \sum_{k=0}^{m} \sum_{j=0}^{m} p_{\mathrm{U,s}}^{(i-1,t+j-k)}, \tag{21}$$

and the DE updates of T_t^{L} are

$$p_{\mathrm{L,s}}^{(i,t)} = f_{\mathrm{L,s}}\left(q_{\mathrm{U}}^{(i,t)}, \varepsilon\right), \tag{22}$$
$$p_{\mathrm{L,p}}^{(i,t)} = f_{\mathrm{L,p}}\left(q_{\mathrm{U}}^{(i,t)}, \varepsilon\right). \tag{23}$$

Finally the a-posteriori erasure probability on $\boldsymbol{u}_t$ at time t and iteration i is

$$p_a^{(i,t)} = \varepsilon \cdot \bar{q}_{\mathrm{U}}^{(i,t)} \cdot \bar{q}_{\mathrm{L}}^{(i,t)}. \tag{24}$$

DE is performed by tracking the evolution of the a-posteriori erasure probability with the number of iterations.

5.3 Serially Concatenated Codes

Uncoupled

Consider the compact graph of the SCC ensemble in Figure 2(c). Let $p_{\mathrm{O,s}}^{(i)}$ and $p_{\mathrm{O,p}}^{(i)}$ denote the erasure probability from T^{O} to $\boldsymbol{u}$ and $\boldsymbol{v}^{\mathrm{O}}$, respectively, computed in the ith iteration. Likewise, $p_{\mathrm{I,s}}^{(i)}$ and $p_{\mathrm{I,p}}^{(i)}$ denote the extrinsic erasure probability from T^{I} to $\tilde{\boldsymbol{v}}^{\mathrm{O}} = (\boldsymbol{u}, \boldsymbol{v}^{\mathrm{O}})$ and $\boldsymbol{v}^{\mathrm{I}}$.

Both $\boldsymbol{u}$ and $\boldsymbol{v}^{\mathrm{O}}$ receive the same erasure probability, $p_{\mathrm{I,s}}^{(i-1)}$, from T^{I}. Therefore, the erasure probabilities that T^{O} receives from these two variable nodes are equal and given by

$$q_{\mathrm{I}}^{(i)} = \varepsilon \cdot p_{\mathrm{I,s}}^{(i-1)}. \tag{25}$$

The DE equations for T^{O} can then be written as

$$p_{\mathrm{O,s}}^{(i)} = f_{\mathrm{O,s}}\left(q_{\mathrm{I}}^{(i)}, q_{\mathrm{I}}^{(i)}\right), \tag{26}$$

$$p_{\mathrm{O,p}}^{(i)} = f_{\mathrm{O,p}}\left(q_{\mathrm{I}}^{(i)}, q_{\mathrm{I}}^{(i)}\right), \tag{27}$$

where $f_{\mathrm{O,s}}$ and $f_{\mathrm{O,p}}$ are the transfer functions of T^{O} for the systematic and parity bits, respectively.

The erasure probability that T^{I} receives from $\tilde{\boldsymbol{v}}^{\mathrm{O}} = (\boldsymbol{u}, \boldsymbol{v}^{\mathrm{O}})$ is the average of the erasure probabilities from $\boldsymbol{u}$ and $\boldsymbol{v}^{\mathrm{O}}$,

$$q_{\mathrm{O}}^{(i)} = \varepsilon \cdot \frac{p_{\mathrm{O,s}}^{(i)} + p_{\mathrm{O,p}}^{(i)}}{2}. \tag{28}$$

On the other hand, the erasure probability to T^{I} from $\boldsymbol{v}^{\mathrm{I}}$ is ε. Therefore, the DE equations for T^{I} can be written as

$$p_{\mathrm{I,s}}^{(i)} = f_{\mathrm{I,s}}\left(q_{\mathrm{O}}^{(i)}, \varepsilon\right), \tag{29}$$

$$p_{\mathrm{I,p}}^{(i)} = f_{\mathrm{I,p}}\left(q_{\mathrm{O}}^{(i)}, \varepsilon\right). \tag{30}$$

Coupled

Consider the compact graph representation of SC-SCCs in Figure 5(b). Variable nodes $\boldsymbol{u}_t$ and $\boldsymbol{v}_t^{\mathrm{O}}$ are connected to factor nodes $T_{t'}^{\mathrm{I}}$ at time instants $t' = t, \ldots, t+m$. The input erasure probability to variable nodes $\boldsymbol{u}_t$ and $\boldsymbol{v}_t^{\mathrm{O}}$ from these factor nodes, denoted by $\bar{q}_{\mathrm{I}}^{(i-1,t)}$, is the same for both $\boldsymbol{u}_t$ and $\boldsymbol{v}_t^{\mathrm{O}}$ and is obtained as the average of the erasure probabilities from each of the factor nodes $T_{t'}^{\mathrm{I}}$,

$$\bar{q}_{\mathrm{I}}^{(i-1,t)} = \frac{1}{m+1} \sum_{j=0}^{m} p_{\mathrm{I,s}}^{(i-1,t+j)} \ . \tag{31}$$

The erasure probability to T_t^{O} from $\boldsymbol{u}_t$ and $\boldsymbol{v}_t^{\mathrm{O}}$ is

$$q_{\mathrm{I}}^{(i,t)} = \varepsilon \cdot \bar{q}_{\mathrm{I}}^{(i-1,t)} = \frac{\varepsilon}{m+1} \sum_{j=0}^{m} p_{\mathrm{I,s}}^{(i-1,t+j)} \ . \tag{32}$$

Thus, the DE updates of T_t^{O} are

$$p_{\mathrm{O,s}}^{(i,t)} = f_{\mathrm{O,s}}\left(q_{\mathrm{I}}^{(i,t)}, q_{\mathrm{I}}^{(i,t)}\right) , \tag{33}$$

$$p_{\mathrm{O,p}}^{(i,t)} = f_{\mathrm{O,p}}\left(q_{\mathrm{I}}^{(i,t)}, q_{\mathrm{I}}^{(i,t)}\right) . \tag{34}$$

At time t, T_t^{I} is connected to a set of $\tilde{\boldsymbol{v}}_{t'}^{\mathrm{O}}$s at time instants $t' = t-m, \ldots, t$. The erasure probability that T_t^{I} receives from this set is the average of the erasure probabilities of all $\boldsymbol{u}_{t'}$s and $\boldsymbol{v}_{t'}^{\mathrm{O}}$s at times $t' = t-m \ldots, t$. This erasure probability can be written as

$$q_{\mathrm{O}}^{(i,t)} = \frac{\varepsilon}{m+1} \sum_{k=0}^{m} \frac{p_{\mathrm{O,s}}^{(i,t-k)} + p_{\mathrm{O,p}}^{(i,t-k)}}{2} . \tag{35}$$

Hence, the DE updates for the inner encoder are given by

$$p_{\mathrm{I,s}}^{(i,t)} = f_{\mathrm{I,s}}\left(q_{\mathrm{O}}^{(i,t)}, \varepsilon\right) , \tag{36}$$

$$p_{\mathrm{I,p}}^{(i,t)} = f_{\mathrm{I,p}}\left(q_{\mathrm{O}}^{(i,t)}, \varepsilon\right) . \tag{37}$$

Finally, the a-posteriori erasure probability on information bits at time t and iteration i is

$$p_a^{(i,t)} = \varepsilon \cdot p_{\mathrm{O,s}}^{(i,t)} \cdot \bar{q}_{\mathrm{I}}^{(i,t)} . \tag{38}$$

5.4 Braided Convolutional Codes

Uncoupled

Consider the compact graph of uncoupled BCCs in Figure 2(c). These can be obtained by tailbiting BCCs, as shown in Figure 6, with coupling length $L = 1$. Let $p_{\mathrm{U},k}^{(i)}$ and $p_{\mathrm{L},k}^{(i)}$ denote the erasure probabilities of messages from T^{U} and T^{L} through their kth connected edge, $k = 1, 2, 3$, respectively. The erasure probability of messages that T^{U} receives through its edges are

$$q_{\mathrm{L},1}^{(i)} = \varepsilon \cdot p_{\mathrm{L},1}^{(i-1)} , \tag{39}$$

$$q_{\mathrm{L},2}^{(i)} = \varepsilon \cdot p_{\mathrm{L},3}^{(i-1)} , \tag{40}$$

$$q_{\mathrm{L},3}^{(i)} = \varepsilon \cdot p_{\mathrm{L},2}^{(i-1)} . \tag{41}$$

The exact DE equations of T^{U} can be written as

$$p_{\mathrm{U},1}^{(i)} = f_{\mathrm{U},1}\left(q_{\mathrm{L},1}^{(i)}, q_{\mathrm{L},2}^{(i)}, q_{\mathrm{L},3}^{(i)}\right) , \tag{42}$$

$$p_{\mathrm{U},2}^{(i)} = f_{\mathrm{U},2}\left(q_{\mathrm{L},1}^{(i)}, q_{\mathrm{L},2}^{(i)}, q_{\mathrm{L},3}^{(i)}\right) , \tag{43}$$

$$p_{\mathrm{U},3}^{(i)} = f_{\mathrm{U},3}\left(q_{\mathrm{L},1}^{(i)}, q_{\mathrm{L},2}^{(i)}, q_{\mathrm{L},3}^{(i)}\right) , \tag{44}$$

where $f_{\mathrm{U},k}$ denotes the transfer function of T^{U} for its kth connected edge. Similarly, the DE equations for T^{L} can be written by swapping indexes U and L in (39)–(44).

Coupled

Consider the compact graph representation of Type-I BCCs in Figure 7(a). As in the uncoupled case, the DE updates of factor nodes T_t^{U} and T_t^{L} are similar due to the symmetric structure of the coupled construction. Therefore, for simplicity, we only describe the DE equations of T_t^{U} and the equations for T_t^{L} are obtained by swapping indexes U and L in the equations.

The first edge of T_t^{U} is connected to $\boldsymbol{u}_t$. Thus, the erasure probability that T_t^{U} receives through this edge is

$$q_{\mathrm{L},1}^{(i,t)} = \varepsilon \cdot p_{\mathrm{L},1}^{(i-1,t)}. \tag{45}$$

The second edge of T_t^{U} is connected to variable nodes $\boldsymbol{v}_{t'}^{\mathrm{L}}$ at time instants $t' = t-m, \ldots, t-1$. The erasure probability that T_t^{U} receives through its second edge is therefore the average of the erasure probabilities from the variable nodes $\boldsymbol{v}_{t'}^{\mathrm{L}}$ that are connected to this edge. This erasure probability can be written as

$$q_{\mathrm{L},2}^{(i,t)} = \frac{\varepsilon}{m} \sum_{j=1}^{m} p_{\mathrm{L},3}^{(i-1,t-j)}. \tag{46}$$

The third edge of T_t^{U} is connected to $\boldsymbol{v}_t^{\mathrm{U}}$, which is in turn connected to the second edges of factor nodes $T_{t'}^{\mathrm{L}}$ at time instants $t' = t+1, \ldots, t+m$. The erasure probability that $\boldsymbol{v}_t^{\mathrm{U}}$ receives from the set of connected nodes $T_{t'}^{\mathrm{L}}$ is the average of erasure probabilities from these nodes through their second edges. The erasure probability from $\boldsymbol{v}_t^{\mathrm{U}}$ to T_t^{U} is

$$q_{\mathrm{L},3}^{(i,t)} = \frac{\varepsilon}{m} \sum_{j=1}^{m} p_{\mathrm{L},2}^{(i-1,t+j)}. \tag{47}$$

The DE equations of T_t^{U} can then be written as[12]

$$p_{\mathrm{U},1}^{(i,t)} = f_{\mathrm{U},1}\left(q_{\mathrm{L},1}^{(i,t)}, q_{\mathrm{L},2}^{(i,t)}, q_{\mathrm{L},3}^{(i,t)}\right), \tag{48}$$

$$p_{\mathrm{U},2}^{(i,t)} = f_{\mathrm{U},2}\left(q_{\mathrm{L},1}^{(i,t)}, q_{\mathrm{L},2}^{(i,t)}, q_{\mathrm{L},3}^{(i,t)}\right), \tag{49}$$

$$p_{\mathrm{U},3}^{(i,t)} = f_{\mathrm{U},3}\left(q_{\mathrm{L},1}^{(i,t)}, q_{\mathrm{L},2}^{(i,t)}, q_{\mathrm{L},3}^{(i,t)}\right). \tag{50}$$

[12]The DE equations of the original BCCs are obtained by setting $m = 1$ in the DE equations of Type-I BCCs.

The a-posteriori erasure probability on $\boldsymbol{u}_t$ at time t and iteration i for Type-I BCCs is

$$p_a^{(i,t)} = \varepsilon \cdot p_{\mathrm{U},1}^{(i,t)} \cdot p_{\mathrm{L},1}^{(i,t)}. \tag{51}$$

As we discussed in the previous section, the difference between Type-I and Type-II BCCs is that $\boldsymbol{u}_t$ is also coupled in the latter. Variable node $\boldsymbol{u}_t$ in Type-II BCCs is connected to a set of factor nodes $T_{t'}^{\mathrm{U}}$ and $T_{t'}^{\mathrm{L}}$ at time instants $t' = t, \ldots, t+m$. The DE equations of Type-II BCCs are identical to those of Type-I BCCs except for equation (45). Denote by $\bar{q}_{\mathrm{L},1}^{(i-1,t)}$ the input erasure probability to $\boldsymbol{u}_t$ from the connected factor nodes $T_{t'}^{\mathrm{L}}$ in the ith iteration. According to Figure 7(b), $\bar{q}_{\mathrm{L},1}^{(i-1,t)}$ is the average of erasure probabilities from $T_{t'}^{\mathrm{L}}$ at time instants $t' = t, \ldots, t+m$,

$$\bar{q}_{\mathrm{L},1}^{(i-1,t)} = \frac{1}{m+1} \sum_{j=0}^{m} p_{\mathrm{L},1}^{(i-1,t+j)}. \tag{52}$$

Factor node T_t^{U} is connected to variable nodes $\boldsymbol{u}_{t'}$ at time instants $t' = t-m, \ldots, t$. The incoming erasure probability to T_t^{U} through its first edge, denoted by $q_{\mathrm{L},1}^{(i,t)}$, is therefore the average of the erasure probabilities from $\boldsymbol{u}_{t'}$ at times $t' = t-m, \ldots, t$,

$$\begin{aligned} q_{\mathrm{L},1}^{(i,t)} &= \varepsilon \cdot \frac{1}{m+1} \sum_{k=0}^{m} \bar{q}_{\mathrm{L},1}^{(i-1,t-k)} \\ &= \varepsilon \cdot \frac{1}{(m+1)^2} \sum_{k=0}^{m} \sum_{j=0}^{m} p_{\mathrm{L},1}^{(i-1,t+j-k)}. \end{aligned} \tag{53}$$

Finally, the a-posteriori erasure probability on $\boldsymbol{u}_t$ at time t and iteration i for Type-II BCCs is

$$p_a^{(i,t)} = \varepsilon \cdot \bar{q}_{\mathrm{U}}^{(i,t)} \cdot \bar{q}_{\mathrm{L}}^{(i,t)}. \tag{54}$$

6 Rate-Compatible SC-TCs via Random Puncturing

Higher rate codes can be obtained by applying puncturing. For analysis purposes, we consider random puncturing. Random puncturing has been considered, e.g., for LDPC codes in [32], [33] and for turbo-like codes in [34], [35]. In [33], the authors introduced a parameter called θ which allows comparing the strengths of the codes with different rates. In this section, we consider the construction of rate-compatible SC-TCs by means of random puncturing.

We denote by $\rho \in [0,1]$ the fraction of surviving bits after puncturing, referred to as the permeability rate. Consider that a code sequence $\boldsymbol{v}$ is randomly punctured with permeability rate ρ and transmitted over a BEC with erasure probability ε, BEC(ε). For the BEC, applying puncturing is equivalent to transmitting $\boldsymbol{v}$ over a BEC with erasure probability $\varepsilon_\rho = 1-(1-\varepsilon)\rho$, resulting from the concatenation of two BECs, BEC(ε) and BEC(ε_ρ). The DE equations of SC-TCs in the previous section can then be easily modified to account for random puncturing.

For SC-PCCs, we consider puncturing of parity bits only, i.e., the overall code is systematic. The rate of the punctured code (without considering termination of the coupled chain) is $R = \frac{1}{1+2\rho}$. The DE equations of punctured SC-PCCs are obtained by substituting $\varepsilon \leftarrow \varepsilon_\rho$ in (19), (20), (22) and (23).

For punctured SC-SCCs, we consider the coupling of the punctured SCCs proposed in [34], [36][13], where ρ_0 and ρ_1 are the permeability rates of the systematic and parity bits, respectively, of the outer code (see [36, Figure 1]), and ρ_2 is the permeability rate of the parity bits of the inner code. The code rate of the punctured SC-SCC is $R = \frac{1}{\rho_0+\rho_1+2\rho_2}$ (neglecting the rate loss due to termination). The DE for punctured SC-SCCs is obtained by substituting $\varepsilon \leftarrow \varepsilon_{\rho_2}$ in (36) and (37), and modifying (35) to

$$q_{\mathrm{O}}^{(i,t)} = \frac{1}{m+1}\sum_{k=0}^{m} \frac{\varepsilon \cdot p_{\mathrm{O,s}}^{(i,t-k)} + \varepsilon_{\rho_1} \cdot p_{\mathrm{O,p}}^{(i,t-k)}}{2}$$

and (33), (34) to

$$p_{\mathrm{O,s}}^{(i,t)} = f_{\mathrm{O,s}}\left(q_{\mathrm{I}}^{(i,t)}, \tilde{q}_{\mathrm{I}}^{(i,t)}\right), \tag{55}$$

$$p_{\mathrm{O,p}}^{(i)} = f_{\mathrm{O,p}}\left(q_{\mathrm{I}}^{(i,t)}, \tilde{q}_{\mathrm{I}}^{(i,t)}\right), \tag{56}$$

where $q_{\mathrm{I}}^{(i,t)}$ is given in (32) and

$$\tilde{q}_{\mathrm{I}}^{(i,t)} = \frac{\varepsilon_{\rho_1}}{m+1}\sum_{j=0}^{m} p_{\mathrm{I,s}}^{(i-1,t+j)}. \tag{57}$$

For both Type-I and Type-II BCCs, similarly to SC-PCCs, we consider only puncturing of parity bits with permeability rate ρ. The DE equations of

[13] In contrast to standard SCCs, characterized by a rate-1 inner code and for which to achieve higher rates the outer code is heavily punctured, the SCCs proposed in [34], [36] achieve higher rates by moving the puncturing of the outer code to the inner code, which is punctured beyond the unitary rate. This allows to preserve the interleaving gain for high rates and yields a larger minimum distance, which results in codes that significantly outperform standard SCCs, especially for high rates. Furthermore, the SCCs in [34], [36] yield better MAP thresholds than standard SCCs.

Table 1: Thresholds for rate-1/2 TCs, and SC-TCs.

Ensemble	states	$\varepsilon_{\mathrm{BP}}$	$\varepsilon_{\mathrm{MAP}}$	$\varepsilon^{1}_{\mathrm{SC}}$
$\mathcal{C}_{\mathrm{PCC}}/\mathcal{C}_{\mathrm{SC-PCC}}$	4	0.4606	0.4689	0.4689
$\mathcal{C}_{\mathrm{SCC}}/\mathcal{C}_{\mathrm{SC-SCC}}$	4	0.3594	0.4981	0.4708
$\mathcal{C}_{\mathrm{PCC}}/\mathcal{C}_{\mathrm{SC-PCC}}$	8	0.4651	0.4863	0.4862
$\mathcal{C}_{\mathrm{SCC}}/\mathcal{C}_{\mathrm{SC-SCC}}$	8	0.3120	0.4993	0.4507
Type-I $\mathcal{C}_{\mathrm{BCC}}$	4	0.3013	0.4993	0.4932
Type-II $\mathcal{C}_{\mathrm{BCC}}$	4	0.3013	0.4993	0.4988

punctured SC-BCCs are obtained by substituting $\varepsilon \leftarrow \varepsilon_\rho$ in (46) and (47) and the corresponding equations for $q^{(i,t)}_{\mathrm{U},2}$ and $q^{(i,t)}_{\mathrm{U},3}$.

7 Numerical Results

In Table 1, we give DE results for the SC-TC ensembles, and their uncoupled ensembles for rate $R = 1/2$. In particular, we consider SC-PCC and SC-SCC ensembles with identical 4-state and 8-state component encoders with generator matrix $\mathbf{G} = (1, 5/7)$ and $\mathbf{G} = (1, 11/13)$, respectively, in octal notation. For the BCC ensemble, we consider two identical 4-state component encoders and generator matrix

$$\boldsymbol{G}_1(D) = \begin{pmatrix} 1 & 0 & 1/7 \\ 0 & 1 & 5/7 \end{pmatrix} . \tag{58}$$

The BP thresholds ($\varepsilon_{\mathrm{BP}}$) and MAP thresholds ($\varepsilon_{\mathrm{MAP}}$) for the uncoupled ensembles are reported in Table 1. The MAP threshold is obtained using the area theorem [9], [30]. We also give the BP thresholds of SC-TCs for coupling memory $m = 1$, denoted by $\varepsilon^{1}_{\mathrm{SC}}$.

As expected, PCC ensembles yield better BP thresholds than SCC ensembles. However, SCCs have better MAP threshold. The BP decoder works poorly for uncoupled BCCs and the BP thresholds are worse than those of PCCs and SCCs. On the other hand, the MAP thresholds of BCCs are better than those of both PCCs and SCCs. By applying coupling, the BP threshold improves and for $m = 1$, the Type-II BCC ensemble has the best coupling threshold.

Table 2 shows the thresholds of TCs and SC-TCs for several rates. In the table, for the ensembles $\mathcal{C}_{\mathrm{PCC}}/\mathcal{C}_{\mathrm{SC-PCC}}$, ρ_2 is the permeability rate of the parity bits of the upper encoder and the lower encoder. For example, $\rho_2 = 0.5$ means that half of the bits of $\boldsymbol{v}^{\mathrm{U}}$ and $\boldsymbol{v}^{\mathrm{L}}$ are punctured (thus, the resulting code rate is $R = 1/2$). Note that ρ_2 corresponds to permeability ρ defined in Section 6.

Here, we use ρ_2 instead to unify notation with that of SCCs. For the ensembles $\mathcal{C}_{\mathrm{SCC}}/\mathcal{C}_{\mathrm{SC-SCC}}$ (based on the SCCs introduced in [34], [36]), for a given code rate R the puncturing rates ρ_0, ρ_1 and ρ_2 (see Section 6) may be optimized. In this paper, we consider $\rho_0 = 1$, i.e., the overall code is systematic, and we optimize ρ_1 and ρ_2 such that the MAP threshold of the (uncoupled) SCC is maximized.[14] Note that, if $\rho_0 = 1$, for a given R the optimization simplifies to the optimization of a single parameter, say ρ_2, since ρ_1 and ρ_2 are related by $\rho_1 = \frac{1}{R} - 1 - 2\rho_2$.[15] Rate-compatibility can be guaranteed by choosing ρ_1 and ρ_2 to be decreasing functions of R. In the table, we report the coupling thresholds for coupling memory $m = 1, 2, 3$, denoted by $\varepsilon_{\mathrm{SC}}^{1}$, $\varepsilon_{\mathrm{SC}}^{2}$, and $\varepsilon_{\mathrm{SC}}^{3}$, respectively. The gap to the Shannon limit is shown by $\delta_{\mathrm{SH}} = (1-R) - \varepsilon_{\mathrm{MAP}}$.

Table 2: Thresholds for punctured spatially coupled turbo codes.

Ensemble	Rate	states	ρ_2	$\varepsilon_{\mathrm{BP}}$	$\varepsilon_{\mathrm{MAP}}$	$\varepsilon_{\mathrm{SC}}^{1}$	$\varepsilon_{\mathrm{SC}}^{3}$	$\varepsilon_{\mathrm{SC}}^{5}$	$m_{\min}$	δ_{SH}
$\mathcal{C}_{\mathrm{PCC}}/\mathcal{C}_{\mathrm{SC-PCC}}$	1/3	4	1.0	0.6428	0.6553	0.6553	0.6553	0.6553	1	0.0113
$\mathcal{C}_{\mathrm{SCC}}/\mathcal{C}_{\mathrm{SC-SCC}}$	1/3	4	1.0	0.5405	0.6654	0.6437	0.6650	0.6654	4	0.0012
$\mathcal{C}_{\mathrm{PCC}}/\mathcal{C}_{\mathrm{SC-PCC}}$	1/3	8	1.0	0.6368	0.6621	0.6617	0.6621	0.6621	2	0.0045
$\mathcal{C}_{\mathrm{SCC}}/\mathcal{C}_{\mathrm{SC-SCC}}$	1/3	8	1.0	0.5026	0.6663	0.6313	0.6647	0.6662	6	0.0003
$\mathcal{C}_{\mathrm{PCC}}/\mathcal{C}_{\mathrm{SC-PCC}}$	1/2	4	0.5	0.4606	0.4689	0.4689	0.4689	0.4689	1	0.0311
$\mathcal{C}_{\mathrm{SCC}}/\mathcal{C}_{\mathrm{SC-SCC}}$	1/2	4	0.5	0.3594	0.4981	0.4708	0.4975	0.4981	5	0.0019
$\mathcal{C}_{\mathrm{PCC}}/\mathcal{C}_{\mathrm{SC-PCC}}$	1/2	8	0.5	0.4651	0.4863	0.4862	0.4863	0.4863	2	0.0137
$\mathcal{C}_{\mathrm{SCC}}/\mathcal{C}_{\mathrm{SC-SCC}}$	1/2	8	0.5	0.3120	0.4993	0.4507	0.4970	0.4992	7	0.0007
$\mathcal{C}_{\mathrm{PCC}}/\mathcal{C}_{\mathrm{SC-PCC}}$	2/3	4	0.25	0.2732	0.2772	0.2772	0.2772	0.2772	1	0.0561
$\mathcal{C}_{\mathrm{SCC}}/\mathcal{C}_{\mathrm{SC-SCC}}$	2/3	4	0.25	0.2038	0.3316	0.3303	0.3305	0.3315	6	0.0018
$\mathcal{C}_{\mathrm{PCC}}/\mathcal{C}_{\mathrm{SC-PCC}}$	2/3	8	0.25	0.2945	0.3080	0.3080	0.3080	0.3080	1	0.0253
$\mathcal{C}_{\mathrm{SCC}}/\mathcal{C}_{\mathrm{SC-SCC}}$	2/3	8	0.25	0.1507	0.3326	0.2710	0.3278	0.3323	7	0.0007
$\mathcal{C}_{\mathrm{PCC}}/\mathcal{C}_{\mathrm{SC-PCC}}$	3/4	4	0.166	0.1854	0.1876	0.1876	0.1876	0.1876	1	0.0624
$\mathcal{C}_{\mathrm{SCC}}/\mathcal{C}_{\mathrm{SC-SCC}}$	3/4	4	0.166	0.1337	0.2486	0.2155	0.2471	0.2486	5	0.0014
$\mathcal{C}_{\mathrm{PCC}}/\mathcal{C}_{\mathrm{SC-PCC}}$	3/4	8	0.166	0.2103	0.2196	0.2196	0.2196	0.2196	1	0.0304
$\mathcal{C}_{\mathrm{SCC}}/\mathcal{C}_{\mathrm{SC-SCC}}$	3/4	8	0.166	0.0865	0.2495	0.1827	0.2416	0.2488	8	0.0005
$\mathcal{C}_{\mathrm{PCC}}/\mathcal{C}_{\mathrm{SC-PCC}}$	4/5	4	0.125	0.1376	0.1391	0.1391	0.1391	0.1391	1	0.0609
$\mathcal{C}_{\mathrm{SCC}}/\mathcal{C}_{\mathrm{SC-SCC}}$	4/5	4	0.125	0.0942	0.1990	0.1644	0.1968	0.1989	7	0.0011
$\mathcal{C}_{\mathrm{PCC}}/\mathcal{C}_{\mathrm{SC-PCC}}$	4/5	8	0.125	0.1628	0.1698	0.1698	0.1698	0.1698	1	0.0302
$\mathcal{C}_{\mathrm{SCC}}/\mathcal{C}_{\mathrm{SC-SCC}}$	4/5	8	0.125	0.0517	0.1996	0.1302	0.1885	0.1982	8	0.0004
$\mathcal{C}_{\mathrm{PCC}}/\mathcal{C}_{\mathrm{SC-PCC}}$	9/10	4	0.055	0.0578	0.0582	0.0582	0.0582	0.0582	1	0.0418
$\mathcal{C}_{\mathrm{SCC}}/\mathcal{C}_{\mathrm{SC-SCC}}$	9/10	4	0.055	0.0269	0.0996	0.0624	0.0930	0.0988	8	0.0012
$\mathcal{C}_{\mathrm{PCC}}/\mathcal{C}_{\mathrm{SC-PCC}}$	9/10	8	0.055	0.0732	0.0761	0.0761	0.0761	0.0761	1	0.0239
$\mathcal{C}_{\mathrm{SCC}}/\mathcal{C}_{\mathrm{SC-SCC}}$	9/10	8	0.055	0.0128	0.0999	0.0384	0.0765	0.0931	16	0.0001

For large enough coupling memory, we observe threshold saturation for

[14] We remark that nonsystematic codes, i.e., $\rho_0 < 1$, lead to better MAP thresholds. In this case, the optimum is to puncture last the parity bits of the inner encoder, i.e., for $R < 1/2$ $\rho_2 = 1$ and for $R \geq 1/2$ $\rho_0 = 0$, $\rho_1 = 0$ and $\rho_2 = 1/2R$.

[15] Alternatively, one may optimize ρ_1 and ρ_2 such that the BP threshold of the SC-SCC is optimized for a given coupling memory m.

Table 3: Thresholds for punctured Braided Convolutional Codes.

Ensemble	Rate	states	ρ_2	$\varepsilon_{\mathrm{BP}}$	$\varepsilon_{\mathrm{MAP}}$	$\varepsilon_{\mathrm{SC}}^{1}$	$\varepsilon_{\mathrm{SC}}^{3}$	$\varepsilon_{\mathrm{SC}}^{5}$	δ_{SH}
Type-I	1/3	4	1.0	0.5541	0.6653	0.6609	0.6644	0.6650	0.0013
Type-II	1/3	4	1.0	0.5541	0.6653	0.6651	0.6653	0.6653	0.0013
Type-I	1/2	4	0.5	0.3013	0.4993	0.4932	0.4980	0.4988	0.0007
Type-II	1/2	4	0.5	0.3013	0.4993	0.4988	0.4993	0.4993	0.0007
Type-I	2/3	4	0.25	–	0.3331	0.3257	0.3315	0.3325	0.0002
Type-II	2/3	4	0.25	–	0.3331	0.3323	0.3331	0.3331	0.0002
Type-I	3/4	4	0.166	–	0.2491	0.2411	0.2473	0.2484	0.0009
Type-II	3/4	4	0.166	–	0.2491	0.2481	0.2491	0.2491	0.0009
Type-I	4/5	4	0.125	–	0.1999	0.1915	0.1979	0.1991	0.0001
Type-II	4/5	4	0.125	–	0.1999	0.1986	0.1999	0.1999	0.0001
Type-I	9/10	4	0.055	–	0.0990	0.0893	0.0966	0.0980	0.0010
Type-II	9/10	4	0.055	–	0.0990	0.0954	0.0990	0.0990	0.0010

both SC-PCCs and SC-SCCs. The value $m_{\min}$ in Table 2 denotes the smallest coupling memory for which threshold saturation is observed numerically. Interestingly, thanks to the threshold saturation phenomenon, for large enough coupling memory SC-SCCs achieve better BP threshold than SC-PCCs. We remark that SCCs yield better minimum Hamming distance than PCCs [22].

Comparing ensembles with 8-state component encoders and ensembles with 4-state component encoders, we observe that the MAP threshold improves for all the considered cases, since the overall codes become stronger. For PCCs, the BP threshold also improves for 8-state component encoders, but only with puncturing, i.e., for $R > 1/3$. For SCCs, on the other hand, the BP threshold gets worse if higher memory component encoders are used. Due to this fact, a higher coupling memory $m_{\min}$ is needed for SC-SCCs with 8-state component encoders until threshold saturation is observed, and this effect becomes more pronounced for larger rates. However, the achievable BP thresholds of SC-SCCs are better than those of SC-PCCs for all rates.

In Table 3, we give BP thresholds for Type-I and Type-II SC-BCCs with different coupling memories and several rates.[16] As for PCCs, ρ_2 is the permeability rate of the parity bits of the upper encoder and the lower encoder. We also report the BP threshold and MAP threshold of the uncoupled ensembles. Almost in all rates, the BP decoder works poorly for uncoupled BCCs and the BP thresholds are worse than those of PCCs and SCCs (an exception are SCCs with $R = 1/3$). This is specially significant for rates $R \geq 2/3$, for which the BP thresholds of uncoupled BCCs are very close to zero. On the other hand, the

[16]The BP threshold of the Type-I BCC with $m = 1$ corresponds to the BP threshold of the original BCC.

MAP thresholds of BCCs are better than those of both PCCs and SCCs for all rates. As for SC-PCCs and SC-SCCs, the BP thresholds improve if coupling is applied. Type-II BCCs yield better thresholds than Type-I BCCs and achieve threshold saturation for small coupling memories. In contrast, for the coupling memories considered, threshold saturation is not observed for Type-I BCCs.

For comparison purposes, in Table 4 we report the $\varepsilon_{\mathrm{BP}}$, $\varepsilon_{\mathrm{MAP}}$, and $\varepsilon_{\mathrm{SC}}^{1}$ for three rate-1/2 LDPC code ensembles. As it is well known, by increasing the variable node degree, the MAP threshold improves, but the BP threshold decreases. Similarly to TCs, applying the coupling improves the BP threshold. Among all the ensembles shown in Table 4, the $(5, 10)$ LDPC ensemble has the best MAP threshold. However, for this ensemble the gap between the BP and MAP thresholds is larger than that of the other LDPC code ensembles and the coupling (with $m = 1$) is not able to completely close this gap, therefore $\varepsilon_{\mathrm{SC}}^{1}$ is worse than that of other two SC-LPDC code ensembles. Among all codes in Table 4, the best $\varepsilon_{\mathrm{BP}}$ is achieved by the Type II BCC ensemble. Similar to the $(5, 10)$ LDPC code ensemble, the gap between the BP and the MAP threshold is relatively large for BCCs. However, for BCCs the BP threshold increases significantly after applying coupling with $m = 1$. In addition, the only way to increase the MAP threshold of the LDPC codes is to increase their variable node degree, but in TCs the BP threshold can be improved by several different methods, e.g., increasing the component code memory, selecting a good ensemble, or increasing the variable node degree.

Table 4: Thresholds for rate-1/2 TCs, SC-TCs, LDPC and SC-LDPC codes.

Ensemble	states	$\varepsilon_{\mathrm{BP}}$	$\varepsilon_{\mathrm{MAP}}$	$\varepsilon_{\mathrm{SC}}^{1}$
LDPC $(3, 6)$	-	0.4294	0.4881	0.4880
LDPC $(4, 8)$	-	0.3834	0.4977	0.4944
LDPC $(5, 10)$	-	0.3415	0.4994	0.4826
$\mathcal{C}_{\mathrm{PCC}}/\mathcal{C}_{\mathrm{SC-PCC}}$	4	0.4606	0.4689	0.4689
$\mathcal{C}_{\mathrm{SCC}}/\mathcal{C}_{\mathrm{SC-SCC}}$	4	0.3594	0.4981	0.4708
$\mathcal{C}_{\mathrm{PCC}}/\mathcal{C}_{\mathrm{SC-PCC}}$	8	0.4651	0.4863	0.4862
$\mathcal{C}_{\mathrm{SCC}}/\mathcal{C}_{\mathrm{SC-SCC}}$	8	0.3120	0.4993	0.4507
Type-I $\mathcal{C}_{\mathrm{BCC}}$	4	0.3013	0.4993	0.4932
Type-II $\mathcal{C}_{\mathrm{BCC}}$	4	0.3013	0.4993	0.4988

Figure 8 shows the bit error rate (BER) for SC-SCCs with $L = 100$ and $m = 1$ on the binary erasure channel for two different rates, $R = 1/4$ (solid blue line) and $R = 1/3$ (solid red line). Here, we consider the coupling of SCCs with block length $K = 1024$, hence the information block length of the SC-SCC ensemble is $K = 101376$. In addition, we plot in the figure the BER curves

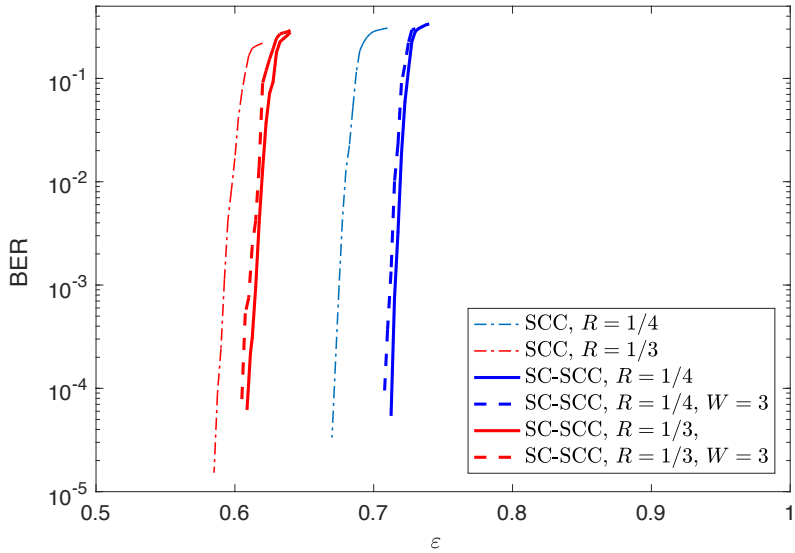


Figure 8: BER results for SC-SCCs with $L = 100$ and $m = 1$ on the binary erasure channel.

for the uncoupled ensemble (dotted lines) with $K = 3072$. For comparison, we also plot the BER using a sliding window decoder with window size $W = 3$ and $K = 1024$ (dashed lines) which has a decoding latency equal to that of the uncoupled ensemble. For both rates, the BER improves significantly applying coupling and the use of the window decoder entails only a slight performance degradation with respect to full decoder [17]. We remark that the comparison between SC-TCs and other types of codes is a new and ongoing field of research. In [38] the authors have compared BCC and SC-LDPC codes for rate 1/2 and under the assumption of similar latency for both. The results in [38] show that the considered BCC ensemble outperforms the SC-LDPC code ensemble.

8 Threshold Saturation

The numerical results in the previous section suggest that threshold saturation occurs for SC-TCs. In this section, for some relevant ensembles, we prove that,

[17] In this work, we are focusing on the BER of TC and SC-TC ensembles in the waterfall region. However, spatial coupling does also preserve, or even improve, the error floor performance. For example, the minimum distance of each SC-TC ensemble is lower bounded by the minimum distance of the corresponding uncoupled TC ensemble. This can be shown by extending the results for BCCs derived in [37].

indeed, threshold saturation occurs. To prove threshold saturation we use the proof technique based on potential functions introduced in [4], [7]. In the general case, the DE equations of TCs form a vector recursion. However, we show that, for some relevant TC ensembles, it is possible to rewrite the DE vector recursion in a form which corresponds to the recursion of a scalar admissible system. We can then prove threshold saturation using the framework in [4] for scalar recursions. Since the proof for scalar recursions is easier to describe, we first address this case, and we then highlight the proof for the general case of TCs with a vector recursion based on the framework in [7].

Definition 1 ([4], [5]) *A scalar admissible system (f, g), is defined by the recursion*

$$x^{(i)} = f\Big(g(x^{(i-1)}); \varepsilon\Big), \tag{59}$$

where $f : [0,1] \times [0,1] \to [0,1]$ and $g : [0,1] \to [0,1]$ satisfy the following conditions.

1. *f is increasing in both arguments $x, \varepsilon \in (0,1]$;*
2. *g is increasing in $x \in (0,1]$;*
3. *$f(0;\varepsilon) = f(x;0) = g(0) = 0$;*
4. *f and g have continuous second derivatives.*

In the following we show that the DE equations for some relevant TCs form a scalar admissible system.

8.1 Turbo-like codes as Scalar Admissible Systems

PCC

The DE equations (10)–(15) form a vector recursion. However, if the code is built from identical component encoders, i.e., $f_{\mathrm{U,s}} = f_{\mathrm{L,s}} \triangleq f_{\mathrm{s}}$, it follows

$$p_{\mathrm{U,s}}^{(i)} = p_{\mathrm{L,s}}^{(i)} \triangleq x^{(i)}.$$

Using this and substituting (12) into (10) and (15) into (13), the DE can then be written as

$$x^{(i)} = f_{\mathrm{s}}(\varepsilon x^{(i-1)}, \varepsilon), \tag{60}$$

with initialization $x^{(0)} = 1$.

Lemma 2 *The DE recursion of a PCC with identical component encoders, given in* (60)*, forms a scalar admissible system with $f(x;\varepsilon) = f_s(\varepsilon \cdot x, x)$ and $g(x) = x$.*

Proof 2 *It is easy to show that all conditions in Definition 1 are satisfied for* $g(x) = x$. *We now prove that* $f(x;\varepsilon)$ *satisfies Conditions 1, 3 and 4. Note that* $f(x;\varepsilon)$ *is the transfer function of a rate-*$1/2$ *convolutional encoder. According to equation* (1), *this function can be written as* $f(p_1, p_2)$, *where* $p_1 = \varepsilon \cdot x$ *and* $p_2 = \varepsilon$. *Using Lemma 1,* $f(p_1, p_2)$ *is increasing with* p_1 *and* p_2, *therefore* $f(x;\varepsilon)$ *is increasing with* x *and* ε *and Condition 1 is satisfied.*

To show that Condition 3 holds, it is enough to realize that for $\varepsilon = 0$ *the input sequence can be recovered perfectly from the received sequence, i.e.,* $f(x;0) = 0$, *as there is a one-to-one mapping between input sequences and coded sequences. Furthermore, when* $x = 0$, *the input sequence is fully known by a-priori information and the erasure probability at the output of the decoder is zero, i.e.,* $f(x;0) = 0$.

Finally, $f(x;\varepsilon)$ *is a rational function and its poles are outside the interval* $x, \varepsilon \in [0, 1]$ *(otherwise we may get infinite output erasure probability for a finite input erasure probability), hence it has continuous first and second derivatives inside this interval.*

SCC

Consider the DE equations of the SCC ensemble in (25)–(30), which form a vector recursion. For identical component encoders, $f_{\mathrm{I,s}} = f_{\mathrm{O,s}} \triangleq f_{\mathrm{s}}$ and $f_{\mathrm{I,p}} = f_{\mathrm{O,p}} \triangleq f_{\mathrm{p}}$. Using this and $q_{\mathrm{I}}^{(i)} \triangleq x^{(i)}$, by substituting (26)–(30) into (25), the DE recursion can be rewritten as

$$x^{(i)} = \varepsilon \cdot f_{\mathrm{s}}\Big(\varepsilon g(x^{(i-1)}), \varepsilon\Big), \tag{61}$$

where

$$g(x^{(i)}) = \frac{f_{\mathrm{s}}\Big(x^{(i)}, x^{(i)}\Big) + f_{\mathrm{p}}\Big(x^{(i)}, x^{(i)}\Big)}{2}, \tag{62}$$

and the initial condition is $x^{(0)} = 1$.

Lemma 3 *The DE recursion of a SCC with identical component encoders, given in* (61) *and* (62), *form a scalar admissible system with* $f(x;\varepsilon) = \varepsilon \cdot f_s(\varepsilon \cdot x, \varepsilon)$ *and*

$$g(x) = \frac{f_s(x, x) + f_p(x, x)}{2}.$$

Proof 3 *The proof follows the same arguments as the proof of Lemma 2.*

BCC

Similarly to PCCs and SCCs, the DE equations of BCCs (see (42)–(44)) form a vector recursion. With identical component encoders, due to the symmetric

structure of the code, $f_{\mathrm{U},k} = f_{\mathrm{L},k} \triangleq f_k$ and $p^{(i)}_{\mathrm{U},k} = p^{(i)}_{\mathrm{U},k} \triangleq x^{(i)}_k$ for $k = 1, 2, 3$. Using this, (42)–(44) can be rewritten as

$$x^{(i)}_1 = f_1\Big(\varepsilon \cdot x^{(i-1)}_1, \varepsilon \cdot x^{(i-1)}_3, \varepsilon \cdot x^{(i-1)}_2\Big) \tag{63}$$

$$x^{(i)}_2 = f_2\Big(\varepsilon \cdot x^{(i-1)}_1, \varepsilon \cdot x^{(i-1)}_3, \varepsilon \cdot x^{(i-1)}_2\Big) \tag{64}$$

$$x^{(i)}_3 = f_3\Big(\varepsilon \cdot x^{(i-1)}_1, \varepsilon \cdot x^{(i-1)}_3, \varepsilon \cdot x^{(i-1)}_2\Big). \tag{65}$$

The above DE equations are still a vector recursion. To write the recursion in scalar form, it is necessary to have identical transfer functions for all the edges which are connected to factor nodes T^{U} and T^{L}. This is needed because all variable nodes in a BCC receive a-priori information. In order to achieve this property, we can apply some averaging over the different types of code symbols. In particular, we can randomly permute the order of the encoder outputs $v^{(l)}_\tau$, $l = 1, \ldots, n$. For each trellis section τ the order of these n symbols is chosen indepently according to a uniform distribution. Equivalently, instead of performing this permutation on the encoder outputs we can define a corresponding component encoder with a time-varying trellis in which the branch labels are permuted accordingly. Then, it results $x^{(i)}_1 = x^{(i)}_2 = x^{(i)}_3 \triangleq x^{(i)}$ and all transfer functions are equal to the average of the transfer functions f_1, f_2, f_3,

$$f_{\mathrm{ave}} = \frac{f_1 + f_2 + f_3}{3}.$$

Using this, the DE equations can be simplified as

$$x^{(i)} = f_{\mathrm{ave}}(\varepsilon \cdot x^{(i-1)}, \varepsilon \cdot x^{(i-1)}, \varepsilon \cdot x^{(i-1)}). \tag{66}$$

Lemma 4 *The DE recursion of a BCC with identical component encoders and time varying trellises, given in* (66)*, form a scalar admissible system with* $f(x;\varepsilon) = f_{ave}(\varepsilon \cdot x, \varepsilon \cdot x, \varepsilon \cdot x)$ *and* $g(x) = x$.

Proof 4 *The proof follows the same arguments as the proof of Lemma 2.*

8.2 Single System Potential

Definition 2 ([4], [5]) *For a scalar admissible system, defined in Definition 1, the potential function* $U(x;\varepsilon)$ *is*

$$\begin{aligned} U(x;\varepsilon) &= \int_0^x \big(z - f(g(x);\varepsilon)\big) g'(z) dz \\ &= x g(x) - G(x) - F(g(x);\varepsilon), \end{aligned} \tag{67}$$

where $F(x;\varepsilon) = \int_0^x f(z;\varepsilon) dz$ *and* $G(x) = \int_0^x g(z) dz$.

Proposition 1 ([4], [5]) *The potential function has the following properties.*

1. *$U(x;\varepsilon)$ is strictly decreasing in $\varepsilon \in (0,1]$;*
2. *An $x \in [0,1]$ is a fixed point of the recursion (59) if and only if it is a stationary point of the corresponding potential function.*

Definition 3 ([4], [5]) *If the DE recursion is the recursion of a BP decoder, the BP threshold is [4]*

$$\varepsilon^{BP} = \sup\left\{\varepsilon \in [0,1] : U'(x;\varepsilon) > 0,\ \forall x \in (0,1]\right\} .$$

According to Definition 3, for $\varepsilon < \varepsilon^{\mathrm{BP}}$, the derivative of the potential function is always larger than zero for $x \in (0,1]$, i.e., the potential function has no stationary point in $x \in (0,1]$.

Definition 4 ([4], [5]) *For $\varepsilon > \varepsilon^{BP}$, the minimum unstable fixed point is $u(\varepsilon) = \sup\left\{\tilde{x} \in [0,1] : f(g(x);\varepsilon) < x, x \in (0,\tilde{x})\right\}$. Then, the potential threshold is defined as [4]*

$$\varepsilon^{*} = \sup\left\{\varepsilon \in [0,1] : u(x) > 0, \min_{x\in[u(x),1]} U(x;\varepsilon) > 0\right\} .$$

The potential threshold depends on the functions $f(x;\varepsilon)$ and $g(x)$.

Example 4 *Consider rate-1/3 PCCs with identical 2-state component encoders with generator matrix $\boldsymbol{G} = (1, 1/3)$. For this code ensemble,*

$$f_s(\varepsilon \cdot x, \varepsilon) = \frac{x\varepsilon^2(2 - 2\varepsilon + x\varepsilon^2)}{(1 - \varepsilon + x\varepsilon^2)^2} .$$

Therefore,

$$F_s(x;\varepsilon) = \frac{x\varepsilon^2}{1 - \varepsilon + x\varepsilon^2} ,$$

and

$$U(x;\epsilon) = \frac{x\varepsilon^3 + (1 - \varepsilon - 2\varepsilon^2)x^2}{2(1 - \varepsilon + x\varepsilon^2)} .$$

△

Example 5 *Consider the PCC ensemble in Figure 2(b) with identical component encoders with generator matrix $\mathbf{G} = (1, 5/7)$. The DE recursion of this ensemble is given in (60), where f_s is the transfer function of the $(1, 5/7)$ component encoder. The corresponding potential function is*

$$U(x;\epsilon) = x^2 - G(x) - F_s(x;\epsilon) = \frac{x^2}{2} - F_s(x;\epsilon) , \tag{68}$$

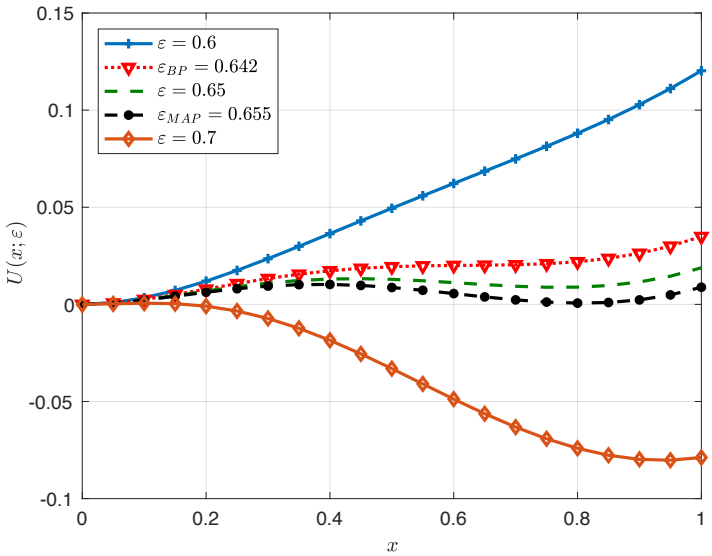


Figure 9: Potential function of a PCC ensemble.

where $F_s(x;\varepsilon) = \int_0^x f_s(\varepsilon \cdot z, \varepsilon)dz$ *and* $G(x) = \int_0^x g(z)dz = \frac{x^2}{2}$. *The potential function is shown in Figure 9 for several values of* ε. *As it is illustrated, for* $\varepsilon < 0.6428$ *the potential function has no stationary point. The BP threshold and the potential threshold are* $\varepsilon = 0.6428$ *and* $\varepsilon = 0.6553$, *respectively (see Definitions 3 and 4). These results match with the DE results in Table 2.* △

Example 6 *The potential function of the SCC ensemble in Figure 2(c) with identical component encoders with generator matrix* $\mathbf{G} = (1, 5/7)$ *is shown in Figure 10. The BP threshold and the potential threshold are* $\varepsilon = 0.689$ *and* $\varepsilon = 0.748$, *respectively, which match with the DE results in Table 2.* △

Example 7 *Consider the BCC ensemble in Figure 2(d) with identical component encoders with generator matrix given in* (58) *and time-varying trellises. The potential function of this code is depicted in Figure 11. The BP threshold and the potential threshold are* $\varepsilon = 0.5522$ *and* $\varepsilon = 0.6654$, *respectively. Note that these values are slightly different from the values in Table 3. This is due to the fact that we considered an ensemble with time-varying trellises, which can be modeled by means of a scalar recursion. The ensemble considered in Table 3 needs to be analyzed by means of a vector recursion.* △

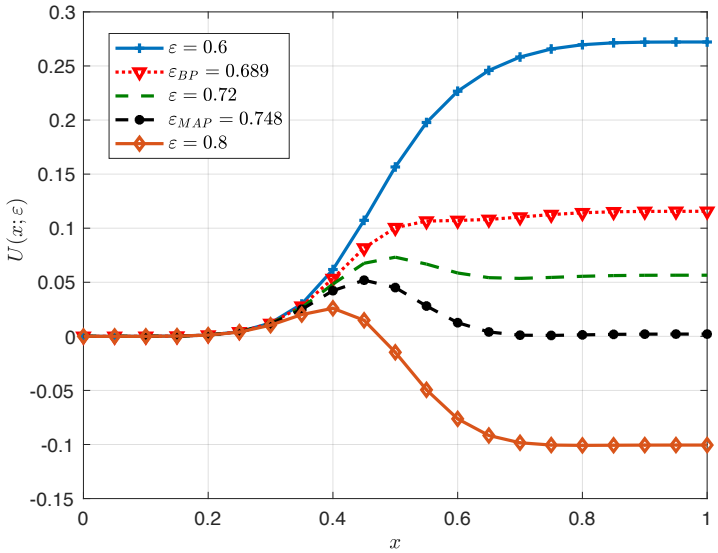


Figure 10: Potential function of a SCC ensemble.

8.3 Coupled System and Threshold Saturation

Theorem 1 *Consider a spatially coupled system defined by the following recursion at time t,*

$$x_t^{(i)} = \frac{1}{1+m}\sum_{j=0}^{m} f_{t+j}\Big(\frac{1}{1+m}\sum_{k=0}^{m} g(x_{t+j-k}^{(i-1)});\varepsilon\Big). \tag{69}$$

If $f(x;\varepsilon)$ and $g(x)$ form a scalar admissible system, for large enough coupling memory and $\varepsilon < \varepsilon^*$, the only fixed point of the recursion is $x = 0$.

Proof 5 *The proof follows from [4].*

In the following we show that the DE recursions of SC-TCs (with identical component encoders) can be written in the form (69). As a result, threshold saturation occurs for these ensembles.

PCCs

Consider the SC-PCC ensemble in Figure 5(a) with identical component encoders. Due to the symmetric coupling structure, it follows that (cf. (16) and (17))

$$\bar{q}_{\mathrm{U}}^{(i,t)} = \bar{q}_{\mathrm{L}}^{(i,t)} \triangleq x_t^{(i)}.$$

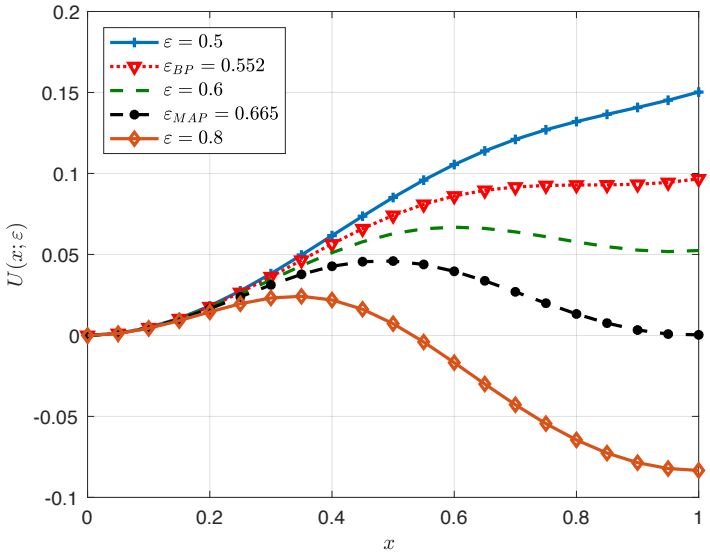


Figure 11: Potential function of a BCC ensemble.

Now, using $x_t^{(i)}$ in (18) and (21), we can write

$$q_{\mathrm{L}}^{(i,t)} = q_{\mathrm{U}}^{(i,t)} = \varepsilon \cdot \frac{1}{m+1} \sum_{k=0}^{m} x_{t-k}^{(i-1)}. \tag{70}$$

Finally, by substituting (70) into (19) and (20) and the results into (16) and (17), the recursion of SC-PCCs can be rewritten as

$$x_t^{(i)} = \frac{1}{1+m} \sum_{j=0}^{m} f_{\mathrm{s},t+j}\Big(\frac{\varepsilon}{m+1} \cdot \sum_{k=0}^{m} x_{t+j-k}^{(i-1)}, \varepsilon\Big). \tag{71}$$

Note that the recursion in (71) is identical to the recursion in (69).

SCCs

Consider the SC-SCC ensemble in Figure 5(b) with identical component encoders. Define $x_t^{(i)} \triangleq q_{\mathrm{I}}^{(i,t)}$ (see (32)) Now, use it in (33)–(36). Finally, by substituting the result in (32), the recursion of a SC-SCC an be rewritten as

$$x_t^{(i)} = \frac{1}{1+m} \sum_{j=0}^{m} \varepsilon \cdot f_{\mathrm{s},t+j}\Big(\frac{\varepsilon}{m+1} \cdot \sum_{k=0}^{m} g(x_{t+j-k}^{(i-1)}), \varepsilon\Big), \tag{72}$$

where $g(x)$ is shown in equation (62). The recursion in (72) is identical to the recursion in Theorem 1.

BCCs

Consider a coupling for BCCs slightly different from the one for Type-II BCCs. At time t, each of the parity sequences $\boldsymbol{v}_t^{\mathrm{U}}$ and $\boldsymbol{v}_t^{\mathrm{L}}$ is divided into $m+1$ sequences, $\boldsymbol{v}_{t,j}^{\mathrm{U}}$, $j=0,\ldots,m$, and $\boldsymbol{v}_{t,j}^{\mathrm{L}}$, $j=0,\ldots,m$, respectively (in Type-II BCCs they are divided into m sequences). The sequences $\boldsymbol{v}_{t-j,j}^{\mathrm{U}}$ and $\boldsymbol{v}_{t-j,j}^{\mathrm{L}}$ are multiplexed and reordered, and are used as the second input of the lower and upper encoder, respectively. Note that in this way of coupling, part of the parity bits at time t are used as input at the same time instant t. Now, similarly to uncoupled BCCs, consider identical time-varying trellises. Let $x_t^{(i)}$ denote the extrinsic erasure probability from T_t^{U} through all its edges in the ith iteration. The erasure probabilities to $\mathrm{T}_t^{\mathrm{U}}$ through all its incoming edges are equal and are given by the average of the erasure probabilities from variable nodes $\boldsymbol{v}_{t'}$, $t'=t-m,\ldots,t$,

$$q_t^{(i)}=\frac{\varepsilon}{1+m}\sum_{k=0}^{m}x_{t-k}^{(i-1)}.$$

Thus, the erasure probabilities from T_t^{U} and T_t^{L} are identical and equal to $f_{\mathrm{ave},t}(q_t^{(i)},q_t^{(i)},q_t^{(i)})$. Finally, the recursion at time slot t is

$$x_t^{(i)}=\frac{1}{1+m}\sum_{j=0}^{m}f_{\mathrm{ave},t+j}(q_{t+j}^{(i)},q_{t+j}^{(i)},q_{t+j}^{(i)}). \tag{73}$$

The recursion in (73) is identical to (69).

8.4 Random Puncturing and Scalar Admissible System

In the following, we show that the DE recursion of punctured TC ensembles can also be rewritten as a scalar admissible system for some particular cases. Then, threshold saturation follows from the discussion in the previous subsection.

PCC

Consider the PCC ensemble with identical component encoders and random puncturing of the parity bits with permeability rate ρ. The DE recursion can be rewritten as,

$$x^{(i)}=f_{\mathrm{s}}(\varepsilon x^{(i-1)},1-(1-\varepsilon)\rho).$$

The above equation is a recursion of a scalar admissible system and satisfies the conditions in Definition 1, where $g(x) = x$ and

$$f(x;\varepsilon) = f_s(\varepsilon \cdot x, 1-(1-\varepsilon)\rho) \ .$$

SCC

Consider random puncturing of the SCC ensemble with identical component encoders. Assuming $\rho_0 = \rho_1$ (i.e., we puncture also systematic bits of the outer code), we can rewrite the DE recursion as

$$x^{(i)} = \varepsilon_{\rho_1} \cdot f_{\mathrm{s}}(\varepsilon_{\rho_1} x^{(i-1)}, \varepsilon_{\rho_2}),$$

where $\varepsilon_{\rho_1} = 1-(1-\varepsilon)\rho_1$ and $\varepsilon_{\rho_2} = 1-(1-\varepsilon)\rho_2$. The above equation is the recursion of a scalar admissible system, where $f(x;\varepsilon) = \varepsilon_{\rho_1} f_s(\varepsilon_{\rho_1} \cdot x, \varepsilon_{\rho_2})$ and $g(x)$ is obtained by equation (62).

BCC

Consider random puncturing of the BCC ensemble with identical time-varying trellises. Assume that the systematic bits and the parity bits of the upper and lower encoders are punctured with the same permeability rate ρ. Then, the DE recursion can be rewritten as (66), where ε should be replaced by $\varepsilon_\rho = 1-(1-\varepsilon)\rho$.

8.5 Turbo-like Codes as Vector Admissible Systems

In general, the DE recursions of TCs are vector recursions. In this case, it is possible to prove threshold saturation using the technique proposed in [7] for vector recursions. The proof is similar to that of scalar recursions, albeit more involved. In the following, we show how to rewrite the recursion of punctured PCCs as a vector admissible system recursion. Then, following [7], we can prove threshold saturation. Using the same technique, it is possible to prove threshold saturation for SCCs and BCCs as well.

Consider the DE equations of the PCC ensemble in (10)–(15). To reduce the number of the equations, substitute (12) and (15) into (10) and (13), respectively. Consider random puncturing of information bits, upper encoder parity bits and lower encoder parity bits with permeability rates ρ_0, ρ_1 and ρ_2, respectively. By considering $x_1^{(i)} \triangleq p_{\mathrm{U,s}}$ and $x_2^{(i)} \triangleq p_{\mathrm{L,s}}$, the DE recursion can be simplified to

$$x_1^{(i)} = f_{\mathrm{U,s}}(\varepsilon_{\rho_0} \cdot x_2^{(i-1)}, \varepsilon_{\rho_1})$$
$$x_2^{(i)} = f_{\mathrm{L,s}}(\varepsilon_{\rho_0} \cdot x_1^{(i-1)}, \varepsilon_{\rho_2}).$$

The above equations can be written in vector format as

$$\boldsymbol{x}^{(i)} = \boldsymbol{f}(\boldsymbol{g}(\boldsymbol{x}^{(i-1)}); \varepsilon), \tag{74}$$

where, $\boldsymbol{x} = [x_1, x_2]$, $\boldsymbol{f}(\boldsymbol{x}; \varepsilon) = [f_{U,\mathrm{s}}(\varepsilon_{\rho_0} \cdot x_1, \varepsilon_{\rho_1}), f_{L,\mathrm{s}}(\varepsilon_{\rho_0} \cdot x_2, \varepsilon_{\rho_2})]$ and $\boldsymbol{g}(\boldsymbol{x}) = [x_2, x_1]$. Is it easy to verify that the recursion in (74) satisfies the conditions in [7, Def. 1], hence (74) is the recursion of a vector admissible system. For this vector admissible system, the line integral is path independent in [7, Eq. (2)] and the potential function is well defined. So, we can define (see [7]) $\boldsymbol{D} = I_{2\times 2}$, $G = x_1 \cdot x_2$ and

$$F = \int_0^{x_1} f_{\mathrm{U,s}}(\varepsilon_{\rho_0} \cdot z, \varepsilon_{\rho_1})\, dz + \int_0^{x_2} f_{\mathrm{L,s}}(\varepsilon_{\rho_0} \cdot z, \varepsilon_{\rho_2})\, dz.$$

It is possible to show that the DE recursion of SC-PCCs can be rewritten in the same form as [7, Eq. (5)] and by using [7, Th. 1], threshold saturation can be proven.

9 Conclusion

In this paper we investigated the impact of spatial coupling on the BP decoding threshold of turbo-like codes. We introduced the concept of spatial coupling for PCCs and SCCs, and generalized the concept of coupling for BCCs. Considering transmission over the BEC, we derived the exact DE equations for uncoupled and coupled ensembles. For all spatially coupled ensembles, the BP threshold improves and our numerical results suggest that threshold saturation occurs if the coupling memory is chosen sufficiently large. We therefore constructed rate-compatible families of SC-TCs that achieve close-to-capacity performance for a wide range of code rates.

We showed that the DE equations of SC-TC ensembles with identical component encoders can be properly rewritten as a scalar recursion. For SC-PCCs, SC-SCCs and BCCs we then proved threshold saturation analytically, using the proof technique based on potential functions proposed in [4], [5]. Finally, we demonstrated how vector recursions can be used to extend the proof to more general ensembles.

A generalization of our results to general binary-input memoryless channels is challenging, because the transfer functions of the component decoders can no longer be obtained in closed form. Even a numerical computation of the exact thresholds is difficult, but Monte Carlo methods and Gaussian approximation techniques could be helpful tools for finding approximate thresholds. EXIT charts, for example, have been widely used for analyzing uncoupled TCs and may be useful for estimating the thresholds of SC-TCs. A connection between EXIT functions and potential functions of spatially coupled systems is given

in [6]. An investigation of SC-TC ensembles along this line may be an interesting direction for future work. The simulation results for SC-BCCs over the AWGN channel in [23] and [38] clearly show that spatial coupling significantly improves the performance, suggesting that threshold saturation also occurs for this channel.

The invention of turbo codes and the rediscovery of LDPC codes, allowed to approach capacity with practical codes. Today, both turbo-like codes and LDPC codes are ubiquitous in communication standards. In the academic arena, however, the interest on turbo-like codes has been declining in the last years in favor of the (considered) more mathematically-appealing LDPC codes. The invention of spatially coupled LDPC codes has exacerbated this situation. Without spatial coupling, it is well known that PCCs yield good BP thresholds but poor error floors, while SCCs and BCCs show low error floors but poor BP thresholds. Our SC-TCs, however, demonstrate that turbo-like codes can also greatly benefit from spatial coupling. The concept of spatial coupling opens some new degrees of freedom in the design of codes on graphs: designing a concatenated coding scheme for achieving the best BP threshold in the uncoupled case may not necessarily lead to the best overall performance. Instead of optimizing the component encoder characteristics for BP decoding, it is possible to optimize the MAP decoding threshold and rely on the threshold saturation effect of spatial coupling. Powerful code ensembles with strong distance properties such as SCCs and BCCs can then perform close to capacity with low-complexity iterative decoding. We hope that our work on spatially coupled turbo-like codes will trigger some new interest in turbo-like coding structures.

References

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

[2] S. Kudekar, T. Richardson and R. 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] M. Lentmaier, A. Sridharan, D.J. Costello, Jr. and K.Sh. Zigangirov, "Iterative decoding threshold analysis for LDPC convolutional codes", *IEEE Trans. Inf. Theory*, vol. 56, no. 10, pp. 5274–5289, Oct. 2010.

[4] A. Yedla, Y. Y. Jian, P. S. Nguyen and H. D. Pfister, "A simple proof of threshold saturation for coupled scalar recursions", in *Proc. Int. Symp. Turbo Codes and Iterative Inf. Process. (ISTC)*, Gothenburg, Sweden, Aug. 2012.

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

[6] S. Kudekar, T. J. Richardson and R. L. Urbanke, "Wave-like solutions of general 1-d spatially coupled systems", *IEEE Trans. Inf. Theory*, vol. 61, no. 8, pp. 4117–4157, Aug. 2015.

[7] A. Yedla, Y. Y. Jian, P. S. Nguyen and H. D. Pfister, "A simple proof of threshold saturation for coupled vector recursions", in *Proc. IEEE Inf. Theory Work. (ITW)*, Lausanne, Switzerland, Sep. 2012.

[8] D. G. Mitchell, M. Lentmaier and D. J. Costello, "Spatially coupled LDPC codes constructed from protographs", *IEEE Trans. Inf. Theory*, vol. 61, no. 9, pp. 4866–4889, Sep. 2015.

[9] A. Ashikhmin, G. Kramer and S. ten Brink, "Extrinsic information transfer functions: Model and erasure channel properties", *IEEE Trans. Inf. Theory*, vol. 50, no. 11, pp. 2657–2673, Nov. 2004.

[10] M. Lentmaier, M. Tavares and G. Fettweis, "Exact erasure channel density evolution for protograph-based generalized LDPC codes", in *IEEE Int. Symp. on Inf. Theory (ISIT)*, Seoul, Korea, Jun. 2009.

[11] M. Lentmaier, B. Noethen and G. Fettweis, "Density evolution analysis of protograph-based braided block codes on the erasure channel", in *Proc. Int. ITG Conf. Source and Channel Coding (SCC)*, Siegen, Germany, Feb. 2010.

[12] M. Lentmaier and G. Fettweis, "On the thresholds of generalized LDPC convolutional codes based on protographs", in *IEEE Int. Symp. on Inf. Theory (ISIT)*, Austin, TX, USA, Jun. 2010.

[13] A. Feltström, D. Truhachev, M. Lentmaier and K. Zigangirov, "Braided block codes", *IEEE Trans. Inf. Theory*, vol. 55, no. 6, pp. 2640–2658, Jun. 2009.

[14] B. Smith, A. Farhood, A. Hunt, F. Kschischang and J. Lodge, "Staircase codes: FEC for 100 Gb/s OTN", *J. Lightw. Technol.*, vol. 30, no. 1, pp. 110–117, Jan. 2012.

[15] Y.-Y. Jian, H. Pfister and K. Narayanan, "Approaching capacity at high rates with iterative hard-decision decoding", in *IEEE Int. Symp. on Inf. Theory (ISIT)*, Cambridge, MA, USA, Jul. 2012.

[16] Y. Jian, H. Pfister, K. Narayanan, R. Rao and R. Mazahreh, "Iterative hard-decision decoding of braided BCH codes for high-speed optical communication", in *Proc. IEEE Global Telecommun. Conf. (GLOBECOM)*, Atlanta, GA, USA, Dec. 2013.

[17] L. Zhang, D. Truhachev and F. Kschischang, "Spatially-coupled split-component codes with bounded-distance component decoding", in *IEEE Int. Symp. on Inf. Theory (ISIT)*, Hong Kong, China, 2015.

[18] N. Wiberg, H. Loeliger and R. Köetter, "Codes and iterative decoding on general graphs", *European Trans. Telecommun.*, vol. 6, no. 5, pp. 513–525, Sep. 1995.

[19] N. Wiberg, "Codes and decoding on general graphs", PhD thesis, Linköping University, 1996.

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

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

[22] 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.

[23] 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.

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

[25] S. Moloudi and M. Lentmaier, "Density evolution analysis of braided convolutional codes on the erasure channel", in *Proc. IEEE Int. Symp. on Inf. Theory*, Honolulu, HI, USA, Jun. 2014.

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

[27] B. Kurkoski, P. Siegel and J. Wolf, "Exact probability of erasure and a decoding algorithm for convolutional codes on the binary erasure channel", in *Proc. IEEE Global Telecommun. Conf. (GLOBECOM)*, San Francisco, CA, USA, Dec. 2003.

[28] J. Shi and S. ten Brink, "Exact EXIT functions for convolutional codes over the binary erasure channel", in *Proc. Allerton Conf. Commun., Control, and Computing*, Monticello, IL, USA, 2006.

[29] A. Graell i Amat, S. Moloudi and M. Lentmaier, "Spatially coupled turbo codes: Principles and finite length performance", in *Proc. Int. Symp. Wir. Commun. Syst. (ISWCS)*, Barcelona, Spain, Agu. 2014.

[30] C. Measson, A. Montanari, T. Richardson and R. Urbanke, "The generalized area theorem and some of its consequences", *IEEE Trans. Inf. Theory*, vol. 55, no. 11, pp. 4793–4821, Nov. 2009.

[31] C. Méasson, "Conservation laws for coding", PhD thesis, École Polytechnique Fédérale de Lausanne, 2006.

[32] H. Pishro-Nik and F. Fekri, "Results on punctured LDPC codes", in *Proc. IEEE Inf. Theory Work. (ITW)*, San Antonio, Texas, USA, Oct. 2004.

[33] D. G. M. Mitchell, M. Lentmaier, A. E. Pusane and D. J. Costello, "Randomly punctured LDPC codes", *IEEE J. Sel. Areas Commun.*, vol. 34, no. 2, pp. 408–421, Feb. 2016.

[34] A. Graell i Amat, G. Montorsi and F. Vatta, "Design and performance analysis of a new class of rate compatible serially concatenated convolutional codes", *IEEE Trans. Commun.*, vol. 57, no. 8, pp. 2280–2289, Aug. 2009.

[35] C. Koller, A. Graell i Amat, J. Kliewer, F. Vatta, K. S. Zigangirov and D. J. Costello, "Analysis and design of tuned turbo codes", *IEEE Trans. Inf. Theory*, vol. 58, no. 7, pp. 4796–4813, Jul. 2012.

[36] A. Graell i Amat, L. Rasmussen and F. Brännström, "Unifying analysis and design of rate-compatible concatenated codes", *IEEE Trans. Commun.*, vol. 59, no. 2, pp. 343–351, Feb. 2011.

[37] S. Moloudi, M. Lentmaier and A. Graell i Amat, "Finite length weight enumerator analysis of braided convolutional codes", in *Proc. Int. Symp. Inf. Theory and Its Applicat. (ISITA)*, Oct. 2016.

[38] M. Zhu, D. G. M. Mitchell, M. Lentmaier, D. J. Costello and B. Bai, "Window decoding of braided convolutional codes", in *Proc. Inf. Theory Work. (ITW)*, Jeju, South Korea, Oct. 2015.

Paper IV

Spatially Coupled Hybrid Concatenated Codes

The main purpose of this paper is to make the study of spatially coupled turbo-like codes (SC-TCs) more complete by investigating the impact of spatial coupling on the thresholds of hybrid concatenated codes (HCCs). In our previous studies, we introduced some classes of SC-TCs and considered their density evolution (DE) analysis. The obtained results indicated that for a fixed coupling memory, braided convolutional codes (BCCs) have the best belief propagation (BP) thresholds among the considered classes. Besides having excellent BP thresholds, BCCs have good distance properties and their minimum distance grows linearly with block length. Similarities between BCCs and HCCs make HCCs good competitors for BCCs. This has motivated us to investigate the impact of spatial coupling on HCCs. In this paper, we introduce two spatially coupled ensembles of HCCs, referred to as Type-I SC-HCCs and Type-II SC-HCCs. Then, we derive the exact density evolution (DE) equations for the uncoupled and the coupled ensembles for the binary erasure channel (BEC). Finally, considering different component encoders, we compute the thresholds of the SC-HCC ensembles and compare them with the thresholds of BCCs for a range of different rates.

S. Moloudi, M. Lentmaier, and A. Graell i Amat,
"Spatially coupled hybrid concatenated codes,"
in *Proc. Int. ITG Conf. Systems, Commun. and Coding (SCC)*, Hamburg, Germany, 2017.

1 Introduction

In the last years, there has been a growing interest in low-density parity-check (LDPC) convolutional codes [1], also known as spatially coupled LDPC (SC-LDPC) codes [2]. These codes exhibit a remarkable behavior called threshold saturation; for them, the belief propagation (BP) decoder can achieve the threshold of the optimal maximum-a-posteriori (MAP) decoder.

However, spatial coupling is a general concept and is not limited to LDPC codes. Spatially coupled turbo-like codes (SC-TCs) were introduced in [3], [4]. In these articles, various ensembles of spatially coupled parallel and serially concatenated codes (SC-PCCs and SC-SCCs) were proposed. Moreover, two extensions of braided convolutional codes (BCCs) for higher coupling memory were introduced, referred to as Type-I BCCs and Type-II BCCs. For the binary erasure channel (BEC), the exact density evolution (DE) equations of the considered SC-TCs were computed and the BP thresholds of the coupled ensembles were obtained. The numerical results in [4] indicate improvements in the BP thresholds of the coupled ensembles and the occurrence of threshold saturation. Moreover, the occurrence of threshold saturation is proved analytically for SC-TCs over the BEC in [4], [5].

The DE analysis of SC-TCs shows that the Type-II BCC ensemble has the best BP threshold among the considered SC-TC ensembles. On the other hand, the finite block length analysis of BCCs in [6] indicates that the minimum distance of BCCs grows linearly with the permutation size. It is also shown that for BCCs very low error rates can be achieved by avoiding a small fraction of bad permutations. Having close-to-capacity thresholds and very low error floor, make BCCs a very promising class of codes.

Hybrid concatenated codes (HCCs) [7], [8] are a class of turbo-like codes which are closely related to BCCs. Similar to the BCC ensemble, the HCC ensemble is a mixture of parallel and serially concatenated code ensembles. Also for HCCs, the minimum distance grows linearly with the permutation size. In addition, they can achieve very low error rates in the floor region [7], [8]. The remarkable properties of HCCs and their similarities with BCCs, have motivated us to investigate the impact of spatial coupling on HCCs.

As a first step, we briefly review the SC-TCs. Then, we propose two ensembles of spatially coupled HCCs (SC-HCCs), referred to as Type-I SC-HCCs and Type-II SC-HCCs. We also derive the exact DE equations for the proposed ensembles and compute the thresholds of BP decoding for the BEC. Using the area theorem we compute the MAP threshold. We also consider different component encoders to investigate the impact of the component encoders on the decoding thresholds of SC-HCCs. By considering random puncturing, we perform a threshold analysis for a family of rate compatible SC-HCCs. Finally, we compare the obtained numerical results with the corresponding results for BCCs.

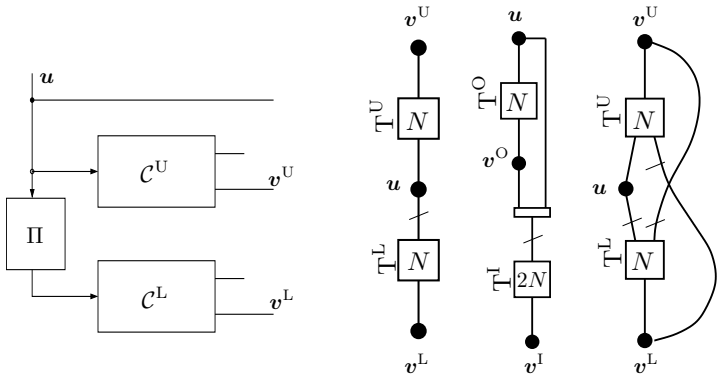


Figure 1: (a) Block diagram of PCCs, Compact graph representation of (b) PCCs, (c) SCCs and (d) BCCs.

2 Spatially Coupled Turbo-Like Codes

2.1 Compact Graph Representation

In our previous studies [4], we considered three main classes of TCs; including PCCs, SCCs, and BCCs. The compact graph representations of these codes are shown in Figure 1. This new representation makes the illustration of TCs and SC-TCs simpler, and makes the DE analysis of theses codes more convenient. In this graph representation, the variable nodes, corresponding to information and parity sequences, are shown by black circles, and the factor nodes corresponding to the component trellises are represented by squares. These factor nodes are also labeled by the length of the corresponding trellises.

As an example, the block diagram of the PCC encoder and the compact graph of PCCs are shown in Figure 1 (a) and (b), respectively. In the compact graph representation, the information sequence $\boldsymbol{u}$ is connected to the upper trellis T^{U} to produce the upper parity sequence $\boldsymbol{v}^{\mathrm{U}}$. Likewise, a reorded copy of $\boldsymbol{u}$ is connected to the lower trellis T^{L} to produce the lower parity sequence $\boldsymbol{v}^{\mathrm{L}}$. To illustrate that a reordered copy of $\boldsymbol{u}$ is connected to T^{L}, the corresponding permutation is shown by a cross line on the edge which connects $\boldsymbol{u}$ to T^{L}.

Consider the PCC ensemble at time t in Figure 2 (a). In order to obtain the coupled ensemble —as it is shown in Figure 2 (b)— the information sequence, $\boldsymbol{u}_t$, is divided into two sequences of equal size, $\boldsymbol{u}_{t,0}$ and $\boldsymbol{u}_{t,1}$, by a multiplexer (the multiplexer is illustrated by a rectangle in the graph). Then, the sequence $\boldsymbol{u}_{t,0}$ is used as a part of the input to the upper encoder at time t, and $\boldsymbol{u}_{t,1}$ is

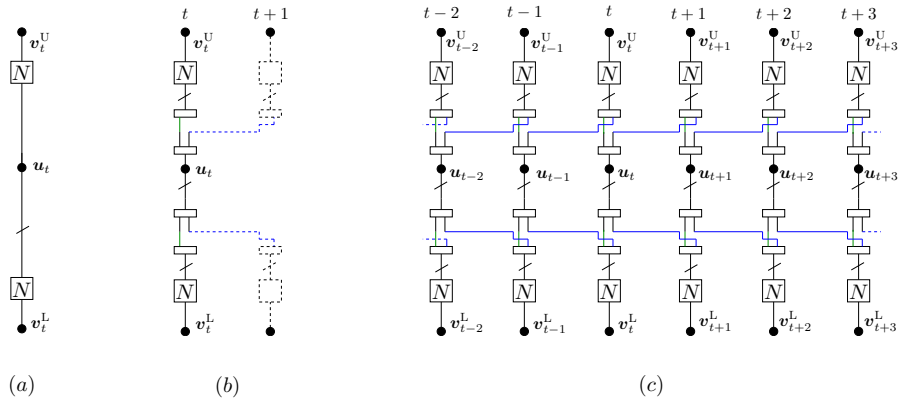


Figure 2: Compact graph representation (a) PCC (b) SC-PCC at time instant t (c) SC-PCC.

used as a part of the input to the upper encoder at time $t+1$. Likewise, a reordered copy of the information sequence, $\tilde{\boldsymbol{u}}_t$, is divided into two sequences $\tilde{\boldsymbol{u}}_{t,0}$ and $\tilde{\boldsymbol{u}}_{t,1}$. These sequences are connected to the lower encoders at time t and $t+1$, respectively.

Consider a collection of L PCCs at time instants $t = 1, ..., L$ (Figure 2 (c)), where L is called the coupling length. Similarly to Figure 2 (b), divide the information sequence $\boldsymbol{u}_t$, $t = 1, \ldots, L$ into two sequences $\boldsymbol{u}_{t,0}$ and $\boldsymbol{u}_{t,1}$. The input to the upper encoder at t is a reordered copy of $(\boldsymbol{u}_{t,0}, \boldsymbol{u}_{t-1,1})$. Likewise, the input to the lower encoder at time t is a reordered copy of $(\tilde{\boldsymbol{u}}_{t,0}, \tilde{\boldsymbol{u}}_{t-1,1})$.

In the SC-PCC ensemble in Figure 2 (c), the coupling memory is equal to $m = 1$ as $\boldsymbol{u}_t$ is used only at the time instants t and $t+1$. It is possible to obtain higher coupling memory m by dividing each of the sequences $\boldsymbol{u}_t$ and $\tilde{\boldsymbol{u}}_t$ into $m+1$ sequences of equal size and spread these sequences respectively to the input of the upper and the lower encoder at time slots t to $t+m$ [4].

Similarly to PCCs, it is possible to apply spatial coupling on SCCs and, increase the coupling memory for BCCs. The SC-TC enseble are described in detail and illustrated in [4].

2.2 Density Evolution Equations and Decoding Thresholds

Considering transmission over a BEC, we can analyze the asymptotic behavior of TCs and SC-TCs by tracking the evolution of the erasure probability in different iterations of the decoding procedure. This evolution can be shown as a set of equations called density evolution (DE) equations, and for the BEC, it is possible to derive an exact expression for them. By use of the DE equations,

Table 1: Thresholds of PCCs, SCCs and BCCs with $R = \frac{1}{3}$, $m = 1$.

Ensemble	$\varepsilon_{\mathrm{BP}}$	$\varepsilon_{\mathrm{MAP}}$	$\varepsilon_{\mathrm{SC}}$
PCC	0.6428	0.6553	0.6553
SCC	0.5405	0.6654	0.6437
Type-I BCC	0.5541	0.6653	0.6609
Type-II BCC	0.5541	0.6653	0.6651

we compute the threshold of BP decoding. The BP threshold is the largest channel erasure probability ε for that the erasure probability at the output of the BP decoder converges to zero as the block-length and number of iterations go to infinity. The BP thresholds, $\varepsilon_{\mathrm{BP}}$, of the considered TC ensembles are computed and summarized in Table 2 for rate $R = \frac{1}{3}$.

We also computed the MAP thresholds of the ensembles, $\varepsilon_{\mathrm{MAP}}$, by use of the area theorem [9]. According to the area theorem, the MAP threshold[18] can be obtained from the following equation:

$$\int_{\varepsilon_{\mathrm{MAP}}}^{1} \bar{p}_{\mathrm{extr}}(\varepsilon) d\varepsilon = R \ ,$$

where R is the rate of the code and $\bar{p}_{\mathrm{extr}}(\varepsilon)$ is the average extrinsic probability of erasure for all transmitted bits.

According to the values shown for $\varepsilon_{\mathrm{BP}}$ and $\varepsilon_{\mathrm{MAP}}$, while the uncoupled BCC ensembles have the worst BP thresholds, they have very good MAP thresholds. The last column of the table shows the BP thresholds of coupled ensembles with coupling memory $m = 1$. The BP threshold of the Type-II BCC ensemble improves significantly and this coupled ensemble has the best BP threshold for $m = 1$.

3 Hybrid Concatenated Codes

In this paper, we consider a rate $R = \frac{1}{5}$ HCC ensemble consisting of a PCC encoder as an outer encoder which is serially concatenated with an inner encoder. The block diagram representation of the HCC encoder is shown in Figure 5. The outer encoder is built of two rate-1 recursive systematic convolutional (RSC) encoders with N trellis sections, referred to as upper and lower encoders, respectively. The inner encoder is an RSC encoder with $2N$ trellis sections.

[18] The obtained threshold from the area theorem is an upper bound on the MAP threshold. However, the numerical results show that the threshold of the coupled ensemble converges to this upper bound. This indicates that the upper bound on the MAP threshold is a tight bound.

The information sequence $\boldsymbol{u}$ is connected to $\mathcal{C}^{\mathrm{U}}$ to produce the upper parity sequence $\boldsymbol{v}^{\mathrm{U}}$. Likewise, a reordered copy of $\boldsymbol{u}$ is connected to $\mathcal{C}^{\mathrm{L}}$, to produce the lower parity sequence $\boldsymbol{v}^{\mathrm{L}}$. Then, the sequences $\boldsymbol{v}^{\mathrm{U}}$ and $\boldsymbol{v}^{\mathrm{L}}$ are multiplexed and properly reordered by permutation Π^{I}. The resulting sequence is used as the input sequence for the inner encoder $\mathcal{C}^{\mathrm{I}}$ to produce the parity sequence $\boldsymbol{v}^{\mathrm{I}}$. Finally, the encoded sequence is $\boldsymbol{v} = (\boldsymbol{u}, \boldsymbol{v}^{\mathrm{U}}, \boldsymbol{v}^{\mathrm{L}}, \boldsymbol{v}^{\mathrm{I}})$.

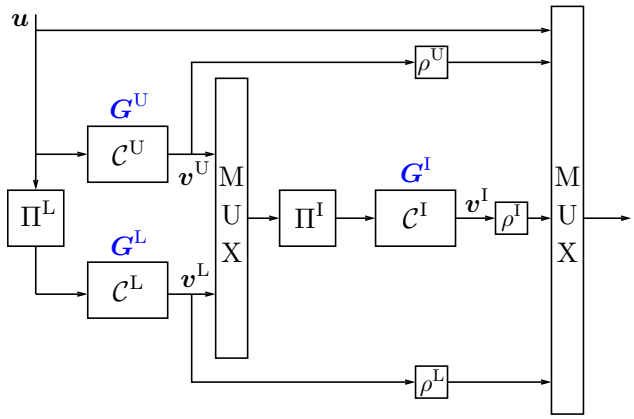


Figure 3: Block diagram representation of an HCC encoder.

A family of rate-compatible SC-HCCs can be obtained by applying puncturing. We denote by $\rho \in [0, 1]$ the fraction of surviving bits in a sequence after puncturing, referred to as permeability rate. Consider random puncturing with the permeability rates ρ^{U}, ρ^{L} and ρ^{I} for the sequences $\boldsymbol{v}_t^{\mathrm{U}}$, $\boldsymbol{v}_t^{\mathrm{L}}$, and $\boldsymbol{v}_t^{\mathrm{I}}$, respectively. The overall rate of the code is

$$R = \frac{1}{1 + \rho^{\mathrm{U}} + \rho^{\mathrm{L}} + 2\rho^{\mathrm{I}}}.$$

Figure 2(a) shows the compact graph representation of the considered HCC ensemble. The factor nodes corresponding to upper, lower, and inner trellises are represented by squares and denoted by T^{U}, T^{L} and T^{I}, respectively.

The information sequence $\boldsymbol{u}$ is connected to T^{U} to produce the upper parity sequence $\boldsymbol{v}^{\mathrm{U}}$. Likewise, a reordered copy of $\boldsymbol{u}$ is connected to T^{L}, to produce $\boldsymbol{v}^{\mathrm{L}}$. Note that in the graph, the permutation Π^{L} is illustrated by the line which crosses the edge between $\boldsymbol{u}$ and T^{L}. The sequences $\boldsymbol{v}^{\mathrm{U}}$ and $\boldsymbol{v}^{\mathrm{L}}$ are multiplexed and properly reordered. The resulting sequence is connected to T^{I} to produce $\boldsymbol{v}^{\mathrm{I}}$.

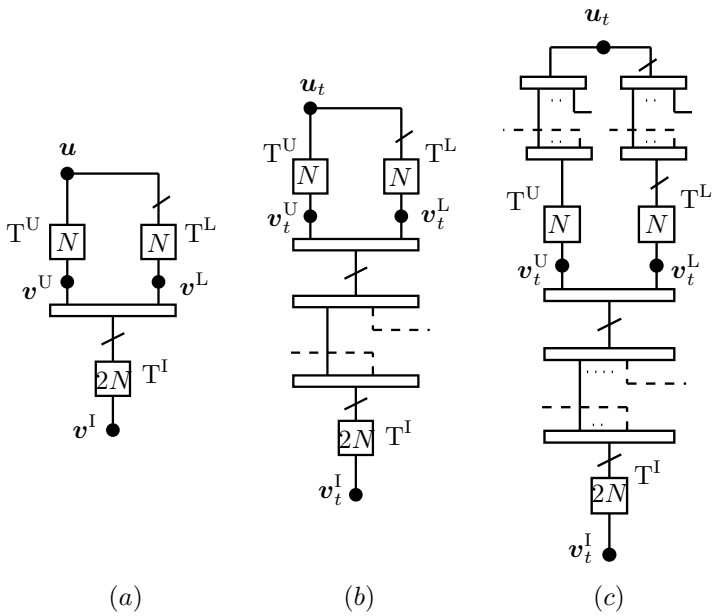


Figure 4: (a) Compact graph representation of (a) HCC (b) Type-I SC-HCC (c) Type-II SC-HCC.

4 Spatially Coupled Hybrid Concatenated Codes

4.1 Type-I Spatially Coupled HCCs

The compact graph representation of the Type-I SC-HCC ensemble with coupling memory m is shown in Figure 2(b) for time instant t. Consider a collection of L blocks of HCCs at time instants $t = 1, \ldots, L$. The information sequence at time t is denoted by $\boldsymbol{u}_t$. Similarly to uncoupled HCCs, $\boldsymbol{u}_t$ and a reordered copy of $\boldsymbol{u}_t$ are connected to $\mathrm{T}_t^{\mathrm{U}}$ and $\mathrm{T}_t^{\mathrm{L}}$ to produce the current parity sequences $\boldsymbol{v}_t^{\mathrm{U}}$ and $\boldsymbol{v}_t^{\mathrm{L}}$, respectively. Then, $\boldsymbol{v}_t^{\mathrm{U}}$ and $\boldsymbol{v}_t^{\mathrm{L}}$ are multiplexed and reordered. The resulting sequence is denoted by $\tilde{\boldsymbol{v}}_t^{\mathrm{O}}$. In order to obtain a coupled ensemble with memory m, $\tilde{\boldsymbol{v}}_t^{\mathrm{O}}$ is divided into $m+1$ equal-sized sequences, denoted by $\tilde{\boldsymbol{v}}_{t,j}^{\mathrm{O}}$, $j = 0, \ldots, m$. At time t, the input of the inner encoder is a reordered version of $(\tilde{\boldsymbol{v}}_{t,0}^{\mathrm{O}}, \tilde{\boldsymbol{v}}_{t-1,1}^{\mathrm{O}}, \ldots, \tilde{\boldsymbol{v}}_{t-m,m}^{\mathrm{O}})$. The corresponding parity sequence is denoted by $\boldsymbol{v}_t^{\mathrm{I}}$. Finally, the unpunctured code sequence is $\boldsymbol{v}_t = (\boldsymbol{u}_t, \boldsymbol{v}_t^{\mathrm{U}}, \boldsymbol{v}_t^{\mathrm{L}}, \boldsymbol{v}_t^{\mathrm{I}})$.

4.2 Type-II Spatially Coupled HCCs

Figure 2(c) depicts the compact graph representation of the Type-II SC-HCC ensemble. This ensemble is equivalent to the Type-I SC-HCC ensemble in most of the parts. For Type-II SC-HCCs, in addition to the coupling of the parity sequences $\boldsymbol{v}_t^{\mathrm{U}}$ and $\boldsymbol{v}_t^{\mathrm{L}}$, we consider the coupling of the information sequence $\boldsymbol{u}_t$. At time t, $\boldsymbol{u}_t$ is divided into $m+1$ equal-sized sequences $\boldsymbol{u}_{t,j}$, $j = 0, \ldots, m$. Likewise, a reordered copy of the information sequence, $\tilde{\boldsymbol{u}}_t$, is divided into $m+1$ equal-sized sequences $\tilde{\boldsymbol{u}}_{t,j}$, $j = 0, \ldots, m$. At time t, the sequence $(\boldsymbol{u}_{t-0,0}, \boldsymbol{u}_{t-1,1}, \ldots, \boldsymbol{u}_{t-m,m})$ and a reordered copy of the sequence $(\tilde{\boldsymbol{u}}_{t-0,0}, \tilde{\boldsymbol{u}}_{t-1,1}, \ldots, \tilde{\boldsymbol{u}}_{t-m,m})$ are the input sequences for the upper and the lower encoder, respectively.

5 Density Evolution Analysis on the BEC

In this section, we assume transmission over the BEC with erasure probability ε. Then, we derive the exact DE equations for the SC-HCC ensembles with the coupling memory m. Note that the DE equations for the uncoupled HCC ensemble can be obtained by considering $m = 0$ and removing the time index in the DE equations of the SC-HCC ensembles. Using the obtained DE equations, we analyze the asymptotic behaviors of the ensembles in the next section.

5.1 Type-I Spatially Coupled HCCs

Consider the Type-I SC-HCC ensemble with coupling memory m, in Figure 2(b). The factor node T^{U} is connected to the variable nodes $\boldsymbol{u}_t$ and $\boldsymbol{v}_t^{\mathrm{U}}$. In the ith iteration, the average extrinsic erasure probabilities from T^{U} to $\boldsymbol{u}_t$ and $\boldsymbol{v}_t^{\mathrm{U}}$ are denoted by $p_{\mathrm{U,s}}^{(i,t)}$ and $p_{\mathrm{U,p}}^{(i,t)}$, respectively. Likewise, $p_{\mathrm{L,s}}^{(i,t)}$ and $p_{\mathrm{L,p}}^{(i,t)}$ denote the average extrinsic erasure probabilities from T^{L} to $\boldsymbol{u}_t$ and $\boldsymbol{v}_t^{\mathrm{L}}$, respectively. Then, the DE updates for T^{U} are

$$p_{\mathrm{U,s}}^{(i,t)} = f_{\mathrm{U,s}}\Big(q_{\mathrm{L}}^{(i-1,t)}, q_{\mathrm{I}}^{(i-1,t)}\Big), \tag{1}$$

$$p_{\mathrm{U,p}}^{(i,t)} = f_{\mathrm{U,p}}\Big(q_{\mathrm{L}}^{(i-1,t)}, q_{\mathrm{I}}^{(i-1,t)}\Big), \tag{2}$$

where

$$q_{\mathrm{L}}^{(i,t)} = \varepsilon \cdot p_{\mathrm{L,s}}^{(i,t)}, \tag{3}$$

$$q_{\mathrm{I}}^{(i,t)} = \varepsilon \cdot \frac{\sum_{j=0}^{m} p_{\mathrm{I,s}}^{(i,t+j)}}{m+1}, \tag{4}$$

and $f_{\mathrm{U,s}}$ and $f_{\mathrm{U,p}}$ are the transfer functions of T^{U} for the systematic and parity bits, respectively. Note that $p_{\mathrm{I,s}}^{(i,t)}$ in equation (4), denotes the average extrinsic

erasure probabilities from T^{I} to the set of $\boldsymbol{v}_{t'}^{\mathrm{U}}$ and $\boldsymbol{v}_{t'}^{\mathrm{L}}$, $t' = t-m, \ldots, t$ which are connected to it. The method proposed in [10] is used to obtain the exact transfer functions of the component decoders.

The DE updates of the lower decoder are identical to those of the upper decoder if the indexes U and L are interchanged.

Similarly the DE updates of T^{I} can be written as

$$p_{\mathrm{I,s}}^{(i,t)} = f_{\mathrm{I,s}}\Big(q_{\mathrm{UL}}^{(i-1,t)}, \varepsilon\Big), \tag{5}$$

$$p_{\mathrm{I,p}}^{(i,t)} = f_{\mathrm{I,p}}\Big(q_{\mathrm{UL}}^{(i-1,t)}, \varepsilon\Big), \tag{6}$$

where

$$q_{\mathrm{UL}}^{(i,t)} = \varepsilon \cdot \frac{\sum_{k=0}^{m} p_{\mathrm{U,s}}^{(i,t-k)} + p_{\mathrm{L,s}}^{(i,t-k)}}{2(m+1)}, \tag{7}$$

and $f_{\mathrm{I,s}}$ and $f_{\mathrm{I,p}}$ are the transfer functions of T^{I} for the systematic and parity bits, respectively.

Finally, the a-posteriori erasure probability on u_t at time t and iteration i is

$$p_a^{(i,t)} = \varepsilon \cdot p_{\mathrm{U,s}}^{(i,t)} \cdot p_{\mathrm{L,s}}^{(i,t)}. \tag{8}$$

5.2 Type-II Spatially Coupled HCCs

Consider the Type-II SC-HCC ensemble with coupling memory m in Figure 2(c). As we discussed in the previous section, this ensemble is identical to the Type-I SC-HCC ensemble except that in the Type-II SC-HCC ensemble the information bits are also coupled. Therefore, the DE updates of the Type-II SC-HCC ensemble are identical to the DE updates of the Type-I SC-HCC ensemble except for the equations (3). According to the compact graph representation in Figure 2(c), the information variable node $\boldsymbol{u}_t$ is connected to the set of $T_{t''}^{\mathrm{U}}$s at time instants $t'' = t, \ldots, t+m$. The reordered copy of $\boldsymbol{u}_t$ is also connected to the set of $T_{t''}^{\mathrm{L}}$s, $t'' = t, \ldots, t+m$. Thus, the equations (3) is rewritten as

$$q_{\mathrm{L}}^{(i,t)} = \varepsilon \cdot \frac{1}{(m+1)^2} \sum_{k=0}^{m} \sum_{j=0}^{m} p_{\mathrm{L,s}}^{(i,t+j-k)}, \tag{9}$$

Finally, the a-posteriori erasure probability on u_t at time t and iteration i is

$$p_a^{(i,t)} = \frac{q_{\mathrm{U}}^{(i,t)} \cdot q_{\mathrm{L}}^{(i,t)}}{\varepsilon}. \tag{10}$$

5.3 Random Puncturing

Assume transmission over a BEC with erasure probability ε. Puncturing a sequence with permeability rate ρ is equivalent to transmitting the sequence over a BEC with erasure probability $\varepsilon_\rho = 1-(1-\varepsilon)\rho$. Thus, we can modify the DE equations of SC-HCCs to account for the random puncturing by considering the corresponding ε_ρs for the transmitted sequences.

As we discussed in the previous section, we denote the permeability rates for the upper, lower, and inner sequence by ρ^{U}, ρ^{L} and ρ^{I}, respectively. The DE updates for the punctured Type-I SC-HCCs are obtained by substituting $\varepsilon_{\rho^{\mathrm{U}}} \leftarrow \varepsilon$ in equation (4) ($\varepsilon_{\rho^{\mathrm{L}}} \leftarrow \varepsilon$ in the corresponding equation for the lower decoder) and $\varepsilon_{\rho^{\mathrm{I}}} \leftarrow \varepsilon$ in equations (5) and (6). Moreover, the equation (7) is modified to

$$q_{\mathrm{UL}}^{(i,t)} = \frac{\sum_{k=0}^{m} \varepsilon_{\rho^{\mathrm{U}}} \cdot p_{\mathrm{U,s}}^{(i,t-k)} + \varepsilon_{\rho^{\mathrm{L}}} \cdot p_{\mathrm{L,s}}^{(i,t-k)}}{2(m+1)}. \tag{11}$$

The DE updates for the punctured Type-II SC-HCC ensemble are identical to those of the punctured Type-I SC-HCC ensemble, except of the modified versions of the equation (3) and its corresponding equation for the lower decoder. For the punctured Type-II SC-HCCs, $q_{\mathrm{L}}^{(i,t)}$ is obtained by substituting $\varepsilon_{\rho^{\mathrm{U}}} \leftarrow \varepsilon$ in equation (9). Likewise, $q_{\mathrm{U}}^{(i,t)}$ is obtained by substituting $\varepsilon_{\rho^{\mathrm{L}}} \leftarrow \varepsilon$ in the corresponding equation for the lower decoder.

6 Results and Discussion

In this chapter, we compute the BP thresholds of HCCs and SC-HCCs by use of the DE equations derived in Section IV. In order to investigate the impact of the component encoders on the thresholds of HCCs, we consider three different cases, referred to as HCC-I, HCC-II and HCC-III. In all cases, we assume identical upper and lower component encoders. The generator matrices of the component encoders are shown in Table 2, in octal notation. In this table, the generator matrix of the upper, lower, and inner encoder are denoted by $\boldsymbol{G}^{\mathrm{U}}$, $\boldsymbol{G}^{\mathrm{L}}$ and $\boldsymbol{G}^{\mathrm{I}}$, respectively.

Table 2: Generator matrices of the component encoders

Ensemble	$\boldsymbol{G}^{\mathrm{U}} = \boldsymbol{G}^{\mathrm{L}}$	$\boldsymbol{G}^{\mathrm{I}}$.
SC-HCC-I	$(1, 1/3)$	$(1, 5/7)$
SC-HCC-II	$(1, 5/7)$	$(1, 5/7)$
SC-HCC-III	$(1, 5/7)$	$(1, 1/3)$

In HCC-I, the upper and lower component encoders are considered to be a simple 2-state RSC encoder with generator matrix $\boldsymbol{G} = (1, 1/3)$. The inner component encoder is a 4-state RSC encoder with generator matrix $\boldsymbol{G} = (1, 5/7)$. In HCC-II, we consider three identical RSC encoders for the upper, lower and inner components. These encoders have generator matrix $\boldsymbol{G} = (1, 5/7)$. Finally, in HCC-III, we considered similar component encoders as in HCC-I but with a different order. The upper and lower components are the 2-state RSC encoders, while the inner component is the 4-state RSC encoder.

The corresponding thresholds to Table 2 are computed for HCC-I, HCC-II and HCC-III. These results are summarized in Table 3. In order to obtain a code with rate $R = 1/3$, random puncturing is considered with $\rho^{\mathrm{U}} = 0$, $\rho^{\mathrm{L}} = 0$ and $\rho^{\mathrm{I}} = 1$.

According to our numerical results, in general, all three considered HCC ensembles suffer from relatively bad BP thresholds and the HCC-II ensemble has the weakest BP threshold. The MAP thresholds, $\varepsilon_{\mathrm{MAP}}$, are almost identical, but that of the HCC-III ensemble is slightly smaller; however, the MAP thresholds of all three cases are excellent, even better than the MAP thresholds of BCCs and SCCs. In other words, for the HCC ensembles, the gap to the Shannon limit is smaller than that for the BCC and SCC ensembles. Applying the coupling results in improved BP thresholds. Similarly to BCCs, Type-II SC-HCC ensembles have better BP thresholds than Type-I SC-HCC ensembles.

As the HCC-II ensemble has the smallest BP threshold, the gap between BP and MAP threshold is big for this ensemble. Although its BP threshold improves significantly after applying spatial coupling with $m = 1$, the coupled threshold $\varepsilon_{\mathrm{SC}}^{1}$ is still much smaller than those of the other cases. The SC-HCC-I ensemble has the best $\varepsilon_{\mathrm{SC}}^{1}$ between the considered SC-HCCs ensembles. Overall, however, the Type-II BCC ensemble still has the best $\varepsilon_{\mathrm{SC}}^{1}$ according to the results in Table 3.

To make the comparison more complete, we consider the SC-HCC ensembles with some higher rates and higher coupling memories. In order to obtain higher rate R, we consider random puncturing with $\rho^{\mathrm{U}} = 0$, $\rho^{\mathrm{L}} = 0$ and $\rho_2 = \rho^{\mathrm{I}} = \frac{1-R}{2R}$.[19] The obtained BP and MAP thresholds are summarized in Table 4. The corresponding BP thresholds for Type-II BCCs in [4] are also given in this table. As we discussed, Type-II BCCs have better thresholds than Type-I BCCs. Therefore, only the thresholds of Type-II BCCs are reported in Table 4.

According to the results in the table, for all rates, the HCC ensembles suffer from small BP thresholds and among them, the HCC-II ensemble has the smallest BP threshold. The MAP thresholds of the HCC ensembles are almost identical and very close to the Shannon limit for all rates. However, for

[19] To have consistent notation with [4], we replace ρ^{I} with ρ_2 in the table.

Table 3: Thresholds for PCCs, SCCs, BCCs and HCCs with $R = \frac{1}{3}$.

Ensemble	Type	$\varepsilon_{\mathrm{BP}}$	$\varepsilon_{\mathrm{MAP}}$	$\varepsilon_{\mathrm{SC}}^{1}$
PCC	-	0.6428	0.6553	0.6553
SCC	-	0.5405	0.6654	0.6437
BCC	Type-I	0.5541	0.6653	0.6609
BCC	Type-II	0.5541	0.6653	0.6651
HCC-I	Type-I	0.4961	0.6666	0.6398
HCC-I	Type-II	0.4961	0.6666	0.6611
HCC-II	Type-I	0.3480	0.6666	0.5667
HCC-II	Type-II	0.3480	0.6666	0.6181
HCC-III	Type-I	0.5456	0.6665	0.5943
HCC-III	Type-II	0.5456	0.6665	0.6382

some rates, the HCC-III ensemble has smaller MAP threshold than those of the two other HCC ensembles. But this threshold is still slightly better than the MAP threshold of the BCC ensemble.

The BP thresholds of the spatially coupled ensembles with coupling memory $m = 1, 3, 5$ are presented in the columns corresponding to $\varepsilon_{\mathrm{SC}}^{1}$, $\varepsilon_{\mathrm{SC}}^{3}$ and $\varepsilon_{\mathrm{SC}}^{5}$, respectively. In all considered cases of SC-HCCs, the BP thresholds improve by increasing the coupling memory. For a large enough coupling memory, the BP thresholds achieve the threshold of the MAP decoder. It can be seen that, for a fixed coupling memory, the Type-II SC-HCC ensembles have better BP thresholds than the corresponding Type-I SC-HCC ensembles and for them, saturation occurs for smaller m. Although the Type-II BCC ensemble has the best BP threshold for $m = 1$ for all rates, by increasing m, the BP thresholds of the SC-HCC ensembles get better those of BCCs.

7 Conclusions

In this paper, we have investigated the impact of spatial coupling on the BP thresholds of HCCs. Similarly to BCCs, these codes are a powerful class of turbo-like codes and their MAP thresholds are even better than those of BCCs. We have shown that the BP thresholds of the HCC ensembles increase significantly by applying spatial coupling and threshold saturation occurs. By selecting the component encoders properly, we can optimize the HCC ensemble for higher BP or MAP thresholds. However, optimizing the HCC ensemble for higher BP or MAP threshold does not guarantee a high BP threshold for SC-HCC for a fixed coupling memory.

Table 4: Thresholds for punctured BCCs and HCCs.

Ensemble	Type	Rate	ρ_2	$\varepsilon_{\mathrm{BP}}$	$\varepsilon_{\mathrm{MAP}}$	$\varepsilon_{\mathrm{SC}}^{1}$	$\varepsilon_{\mathrm{SC}}^{3}$	$\varepsilon_{\mathrm{SC}}^{5}$
BCC	Type-II	1/3	1.0	0.5541	0.6653	0.6651	0.6653	0.6653
HCC-I	Type-I	1/3	1.0	0.4961	0.6666	0.6398	0.6621	0.6651
HCC-I	Type-II	1/3	1.0	0.4961	0.6666	0.6611	0.6666	0.6666
HCC-II	Type-I	1/3	1.0	0.3480	0.6666	0.5667	0.6166	0.6312
HCC-II	Type-II	1/3	1.0	0.3480	0.6666	0.6181	0.6652	0.6666
HCC-III	Type-I	1/3	1.0	0.5456	0.6665	0.5943	0.6243	0.6352
HCC-III	Type-II	1/3	1.0	0.5456	0.6665	0.6382	0.6655	0.6663
BCC	Type-II	1/2	0.5	0.3013	0.4993	0.4988	0.4993	0.4993
HCC-I	Type-I	1/2	0.5	0.2486	0.4999	0.4601	0.4947	0.4982
HCC-I	Type-II	1/2	0.5	0.2486	0.4999	0.4846	0.4999	0.4999
HCC-II	Type-I	1/2	0.5	0.1502	0.4999	0.3766	0.4472	0.4659
HCC-II	Type-II	1/2	0.5	0.1502	0.4999	0.4272	0.4970	0.4999
HCC-III	Type-I	1/2	0.5	0.3501	0.4999	0.4135	0.4540	0.4685
HCC-III	Type-II	1/2	0.5	0.3501	0.4999	0.4597	0.4979	0.4994
BCC	Type-II	2/3	0.25	–	0.3331	0.3323	0.3331	0.3331
HCC-I	Type-I	2/3	0.25	0.0622	0.3333	0.2671	0.3274	0.3314
HCC-I	Type-II	2/3	0.25	0.0622	0.3333	0.2952	0.3327	0.3333
HCC-II	Type-I	2/3	0.25	0.0331	0.3333	0.1972	0.2787	0.3024
HCC-II	Type-II	2/3	0.25	0.0331	0.3333	0.2355	0.3252	0.3328
HCC-III	Type-I	2/3	0.25	0.1820	0.3332	0.2434	0.2876	0.3044
HCC-III	Type-II	2/3	0.25	0.1820	0.3332	0.2821	0.3295	0.3327
BCC	Type-II	3/4	0.166	–	0.2491	0.2481	0.2491	0.2491
HCC-I	Type-I	3/4	0.166	0.0199	0.2499	0.1662	0.2398	0.2481
HCC-I	Type-II	3/4	0.166	0.0199	0.2499	0.1930	0.2479	0.2499
HCC-II	Type-I	3/4	0.166	0.0102	0.2492	0.1161	0.1919	0.2184
HCC-II	Type-II	3/4	0.166	0.0102	0.2492	0.1431	0.2348	0.2477
HCC-III	Type-I	3/4	0.166	0.1106	0.2491	0.1624	0.2043	0.2215
HCC-III	Type-II	3/4	0.166	0.1106	0.2491	0.1933	0.2433	0.2485
BCC	Type-II	4/5	0.125	–	0.1999	0.1986	0.1999	0.1999
HCC-I	Type-I	4/5	0.125	0.0085	0.1999	0.1091	0.1821	0.1982
HCC-I	Type-II	4/5	0.125	0.0085	0.1999	0.1315	0.1956	0.1997
HCC-II	Type-I	4/5	0.125	0.0043	0.1999	0.0747	0.1406	0.1677
HCC-II	Type-II	4/5	0.125	0.0043	0.1999	0.0940	0.1795	0.1970
HCC-III	Type-I	4/5	0.125	0.0743	0.1999	0.1173	0.1557	0.1726
HCC-III	Type-II	4/5	0.125	0.0743	0.1999	0.1422	0.1917	0.1990
BCC	Type-II	9/10	0.055	–	0.0990	0.0954	0.0990	0.0990
HCC-I	Type-I	9/10	0.055	0.0006	0.0999	0.0245	0.0603	0.0822
HCC-I	Type-II	9/10	0.055	0.0006	0.0999	0.0317	0.0798	0.0960
HCC-II	Type-I	9/10	0.055	0.0003	0.0990	0.0159	0.0427	0.0610
HCC-II	Type-II	9/10	0.055	0.0003	0.0990	0.0208	0.0617	0.0850
HCC-III	Type-I	9/10	0.055	0.0190	0.0990	0.0367	0.0587	0.0714
HCC-III	Type-II	9/10	0.055	0.0190	0.0990	0.0463	0.0805	0.0941

References

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

[2] S. Kudekar, T. Richardson and R. 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] S. Moloudi, M. Lentmaier and A. Graell i Amat, "Spatially coupled turbo codes", in *Proc. Int. Symp. Turbo Codes and Iterative Inf. Process. (ISTC)*, Bremen, Germany, Agu. 2014.

[4] ——, "Spatially coupled turbo-like codes", *IEEE Trans. Inf. Theory*, vol. 63, no. 10, pp. 6199–6215, Oct. 2017.

[5] ——, "Threshold saturation for spatially coupled turbo-like codes over the binary erasure channel", in *Proc. IEEE Inf. Theory Work.- Fall (ITW)*, Jeju Island, South Korea, Oct. 2015.

[6] ——, "Finite length weight enumerator analysis of braided convolutional codes", in *Proc. Int. Symp. Inf. Theory and Its Applicat. (ISITA)*, Oct. 2016.

[7] E. Rosnes and A. Graell i Amat, "Performance analysis of 3-D turbo codes", *IEEE Trans. Inf. Theory*, vol. 57, no. 6, pp. 3707–3720, Jun. 2011.

[8] C. Koller, A. Graell i Amat, J. Kliewer, F. Vatta and D. J. Costello, "Hybrid concatenated codes with asymptotically good distance growth", in *Proc. Int. Symp. Turbo Codes and Related Topics (ISTC)*, Sep. 2008.

[9] C. Méasson, "Conservation laws for coding", PhD thesis, École Polytechnique Fédérale de Lausanne, 2006.

[10] B. Kurkoski, P. Siegel and J. Wolf, "Exact probability of erasure and a decoding algorithm for convolutional codes on the binary erasure channel", in *Proc. IEEE Global Telecommun. Conf. (GLOBECOM)*, San Francisco, CA, USA, Dec. 2003.

Paper V

Spatially Coupled Turbo-Like Codes: A New Trade-off Between Waterfall and Error Floor

Spatially coupled turbo-like codes (SC-TCs) have been shown to have excellent decoding thresholds due to the threshold saturation effect. Furthermore, even for moderate block lengths, simulation results demonstrate very good bit error rate performance (BER) in the waterfall region. In this paper, we discuss the effect of spatial coupling on the performance of TCs in the finite block-length regime. We investigate the effect of coupling on the error-floor performance of SC-TCs by establishing conditions under which spatial coupling either preserves or improves the minimum distance of TCs. This allows us to investigate the error-floor performance of SC-TCs by performing a weight enumerator function (WEF) analysis of the corresponding uncoupled ensembles. While uncoupled TC ensembles with close-to-capacity performance exhibit a high error floor, our results show that SC-TCs can simultaneously approach capacity and achieve very low error floor.

---

based on
S. Moloudi, M. Lentmaier, and A. Graell i Amat,
"Spatially coupled turbo-like codes: a new trade-off between waterfall and error floor,"
submitted to *IEEE Trans. commun.*

1 Introduction

Turbo-like codes (TCs) [1] and low-density parity check (LDPC) codes [2] are adopted in many communication standards because they can practically approach the Shannon limit. Recently, it has been proved that LDPC convolutional codes [3], [4] —also known as spatially coupled LDPC (SC-LDPC) codes—exhibit the remarkable threshold saturation phenomenon [5], [6], [7], [8], i.e., for an SC-LDPC ensemble, the belief propagation (BP) decoder can achieve the threshold of the optimal maximum-a-posteriori (MAP) decoder of the underlying uncoupled ensemble. It then follows that regular SC-LDPC codes achieve capacity as their variable node degrees tend to infinity. Spatially coupled TCs (SC-TCs) were introduced in [9], [10], [11], and it was proved that threshold saturation also occurs for this class of codes. A density evolution analysis shows that, by having stronger component codes, SC-TCs can achieve excellent decoding thresholds with variable nodes of degree one and two only.

In this paper, motivated by the excellent asymptotic behavior of SC-TCs, we investigate the performance of these codes in the finite block-length regime. We consider the same TC ensembles as those in [9], [10], [11], namely parallel concatenated codes (PCCs) [1], serially concatenated codes (SCCs) [12], [13], braided convolutional codes (BCCs) [14], [15], and hybrid concatenated codes (HCCs) [16], [17]. As the first step of our investigation, using the decoding thresholds of the binary erasure channel (BEC) obtained in [10], [11] and the method described in [18], [19], we predict the decoding thresholds over the additive white Gaussian noise (AWGN) channel. Using these thresholds together with the provided simulation results, we discuss the effect of spatial coupling on the performance of TCs in the waterfall region over the AWGN channel. Then, we investigate the effect of coupling on the error-floor performance of TCs. We prove that under certain conditions the minimum distance of a coupled SC-TC ensemble cannot get smaller than that of the corresponding TC ensemble. This means that the error-floor performance of the TCs is not degraded by spatial coupling. These conditions can be seen as a guideline for unwrapping the TC ensembles. This connection between the minimum distance of TC and SC-TC ensembles allows us to avoid the complexity of computing the weight enumerator functions (WEFs) of the coupled ensembles. Instead, we simply perform a WEF analysis for the uncoupled TC ensemble to investigate and discuss the distance properties of SC-TCs. Thus, we compute the WEFs of TC ensembles [12], [20], [21], [22] to obtain bounds on their bit error rate (BER) performance and a bound on the minimum distance. Finally, in the last step of our investigation, we use the obtained results to discuss the overall performance of SC-TCs for the finite block-length regime.

The remainder of the paper is organized as follows. In Section 2, we briefly describe several TC and SC-TC ensembles by use of the compact graph rep-

resentation introduced in [10]. We discuss the decoding thresholds of these ensembles in Section 3. In the same section, we provide some simulation results to discuss the waterfall region performance of SC-TCs. In Section 4, we prove that the minimum distance of SC-TC ensembles is either better or equal than that of the corresponding uncoupled ensemble. In Section 5, we compute the average WEF of TC ensembles to obtain bounds on their BER performance and minimum distance. Finally, in Section 6, we discuss the trade-off between waterfall and error floor performance of SC-TCs, and we conclude the paper in the same section.

2 Spatially Coupled Turbo-Like Codes

In this section, we briefly describe four major classes of TCs— namely, PCCs, SCCs, BCCs, and HCCs— and their coupled counterparts. In particular, we discuss PCCs and SC-PCCs with coupling memory $m = 1$, and refer the interested reader to [10] for details on the other SC-TC ensembles and higher coupling memories, $m > 1$.

Figure 1(a) shows the block diagram of a rate $R = 1/3$ PCC encoder built of two recursive systematic convolutional encoders, referred to as upper and lower encoder. As shown in the figure, the information sequence $\boldsymbol{u}$ is encoded by the upper encoder $\mathcal{C}^{\mathrm{U}}$ to produce the upper parity sequence $\boldsymbol{v}^{\mathrm{U}}$. Likewise, a reordered copy of $\boldsymbol{u}$ is encoded by the lower encoder $\mathcal{C}^{\mathrm{L}}$ to produce the lower parity sequence $\boldsymbol{v}^{\mathrm{L}}$. The corresponding permutation is denoted by Π^{Un}. Finally, the output of the PCC encoder is the sequence $\boldsymbol{v} = (\boldsymbol{u}, \boldsymbol{v}^{\mathrm{U}}, \boldsymbol{v}^{\mathrm{L}})$.

The compact graph representation [10] of the PCC ensemble is depicted in Figure 1(b). Each of the sequences $\boldsymbol{u}$, $\boldsymbol{v}^{\mathrm{U}}$, and $\boldsymbol{v}^{\mathrm{L}}$ is represented by a black circle, referred to as variable node. The trellises corresponding to the component encoders are shown by squares, called factor nodes, and they are labeled by the length of the trellises. The sequences $\boldsymbol{u}$ and $\boldsymbol{v}^{\mathrm{U}}$ are connected to the upper trellis T^{U}. Likewise, a reordered copy of $\boldsymbol{u}$ and $\boldsymbol{v}^{\mathrm{L}}$ are connected to the lower trellis T^{L}. In order to emphasize that a reordered copy of $\boldsymbol{u}$ is connected to T^{L}, the corresponding permutation is represented by a line that crosses the edge which connects $\boldsymbol{u}$ to T^{L}.

Figure 1(c) shows the compact graph representation of the spatially coupled PCC (SC-PCC) ensemble with coupling memory $m = 1$ at time t. Consider a collection of PCC ensembles at time slots $t = 1, \ldots, L$, where L is the coupling length. The SC ensemble can be obtained by dividing the information sequence at time t, $\boldsymbol{u}_t$, and its reordered copy, $\tilde{\boldsymbol{u}}_t$, into two subsequences, denoted by $\boldsymbol{u}_{t,j}$ and $\tilde{\boldsymbol{u}}_{t,j}$, $j = 0, 1$, respectively. Then these subsequences are spread over time t and $t + 1$. The input sequence to the upper encoder at time t is the

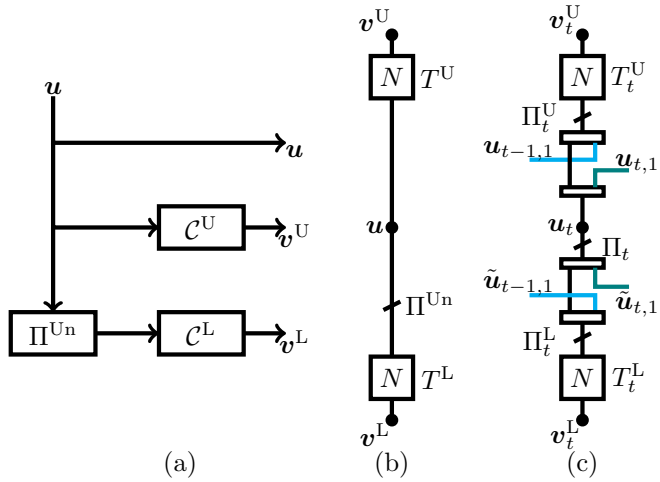


Figure 1: (a) Encoder block diagram of PCC. Compact graph representation of (b) PCC, (c) SC-PCC.

Table 1: Predicted AWGN channel thresholds for rate-1/3 TCs and SC-TCs.

Ensemble	Type	$\mathrm{E_b/N_0}$ [dB]$_{\mathrm{BP}}$	$\mathrm{E_b/N_0}$ [dB]$_{\mathrm{MAP}}$	$\mathrm{E_b/N_0}$[dB]$^{1}_{\mathrm{SC}}$	$\mathrm{E_b/N_0}$ [dB]$^{3}_{\mathrm{SC}}$	$\mathrm{E_b/N_0}$[dB]$^{5}_{\mathrm{SC}}$
$\mathcal{C}_{\mathrm{PCC}}/\mathcal{C}_{\mathrm{SC-PCC}}$	-	-0.1053	-0.3070	-0.3070	-0.3070	-0.3070
$\mathcal{C}_{\mathrm{SCC}}/\mathcal{C}_{\mathrm{SC-SCC}}$	-	1.4024	-0.4740	-0.1196	-0.4673	-0.4740
$\mathcal{C}_{\mathrm{BCC}}$	Type-I	1.2139	-0.4723	-0.3992	-0.4573	-0.4673
$\mathcal{C}_{\mathrm{BCC}}$	Type-II	1.2139	-0.4723	-0.4690	-0.4723	-0.4723
$\mathcal{C}_{\mathrm{HCC}}/\mathcal{C}_{\mathrm{SC-HCC}}$	Type-I	3.8846	-0.4941	1.0366	0.3038	0.0780
$\mathcal{C}_{\mathrm{HCC}}/\mathcal{C}_{\mathrm{SC-HCC}}$	Type-II	3.8846	-0.4941	0.2809	-0.4706	-0.4941

sequence $(\boldsymbol{u}_{t,0}, \boldsymbol{u}_{t-1,1})$, reordered by permutation Π_t^{U}.[20] Likewise, the input sequence to the lower encoder at time t is the sequence $(\tilde{\boldsymbol{u}}_{t,0}, \tilde{\boldsymbol{u}}_{t-1,1})$, reordered by permutation Π_t^{L}. The information bits at time slots $t \leq 0$ are initialized by zero and the information bits at $t = L$ are chosen in such a way that $\boldsymbol{u}_{L,1} = 0$ and $\tilde{\boldsymbol{u}}_{L,1} = 0$ (i.e., we consider the termination of the coupled chain).

Figure 2 shows the compact graph representation of the SCC, BCC, and HCC ensembles, and their corresponding spatially coupled ensembles. In this paper, we restrict ourselves to PCC, SCC and HCC ensembles with identical 4-state component trellises and generator matrix $\boldsymbol{G} = (1, 5/7)$, in octal notation. For the BCC ensemble, we consider two identical 4-state component trellises with generator matrix

$$\boldsymbol{G}(D) = \begin{pmatrix} 1 & 0 & 1/7 \\ 0 & 1 & 5/7 \end{pmatrix} . \tag{1}$$

[20]The multiplexer is represented by a rectangular in the compact graph representation.

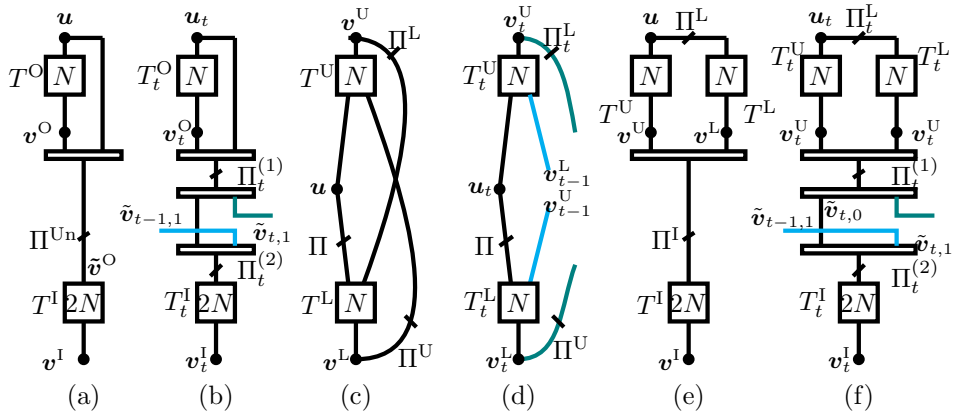


Figure 2: Compact graph representation of (a) SCC, (b) SC-SCC, (c) BCC, (d) SC-BCC, (e) HCC, and (f) SC-HCC.

We also restrict ourselves to systematic TCs and SC-TCs with rate $R = 1/3$. Therefore, for the SCC and HCC ensembles, we consider full puncturing of the parity sequences of the outer encoders [10], [11].

3 Spatial Coupling: Waterfall Region Performance

3.1 Asymptotic Performance

Using the decoding threshold of an ensemble computed for the BEC, it is possible to predict its decoding threshold over the AWGN channel [18], [19]. This allows us to use the decoding thresholds of the TC and SC-TCs from [10], [11] to predict the corresponding thresholds over the AWGN channel. The results are shown in Table 1. Similar to the BEC, among all the uncoupled TC ensembles, the PCC ensemble has the best BP threshold but the worst MAP threshold. Conversely, the HCC ensemble has the worst BP threshold but the best MAP threshold, which is very close to the Shannon limit. It can also be seen that for all coupled ensembles, threshold saturation occurs. In general, as the numerical results in Table 1 suggest, SC-TC ensembles can achieve close-to-capacity BP thresholds.

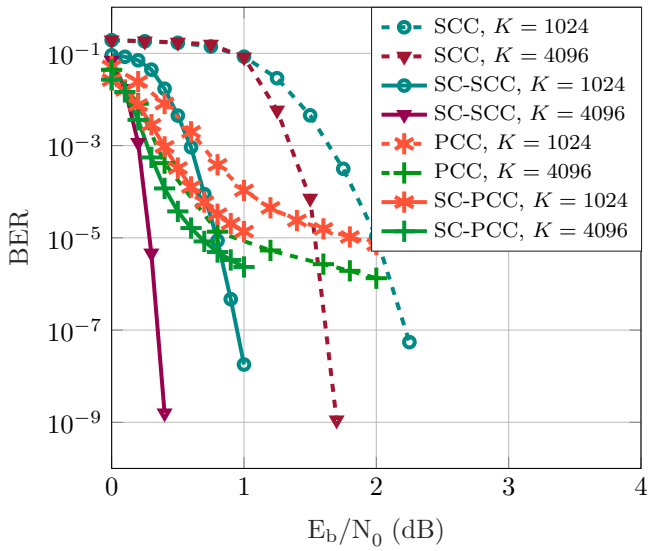


Figure 3: Simulation results for PCC, SC-PCC vs. SCC, SC-SCC, $R = 1/3$.

3.2 Finite Block-Length Performance

Figure 3 shows BER simulation results for PCCs, SCCs, SC-PCCs, and spatially coupled SCC (SC-SCCs) with $R = 1/3$ and input block length $K = 1024$ and $K = 4096$. For the coupled ensembles, we consider a coupling length $L = 100$ and a sliding window decoder with window size $W = 4$ [15]. The decoding latency is $W \cdot K$. It is well known that the PCC ensemble yields better performance than the SCC ensemble in the waterfall region [12]; however, the SCC ensemble has a much lower error floor than the PCC ensemble. By applying spatial coupling, the performance of the PCC and SCC ensembles improves significantly for both input block lengths. This improvement is more substantial for the SCC ensemble than for the PCC ensemble. For instance, the performance of the SCC ensemble with $K = 1024$ at BER$= 10^{-5}$ improves more than 1 dB with coupling. The coupling gains are in agreement with the decoding thresholds in Table 1. As it can be seen, the gap between the BP and MAP threshold of the SCC ensemble is larger than that of the PCC ensemble, hence the expected gain from coupling is bigger for the SCC ensemble.

In Figure 3, the uncoupled ensemble with $K = 4096$ and the coupled ensemble with $K = 1024$ have equal latency, i.e., both ensembles have a decoding latency of 4096 bits. For this latency, the SC-SCC ensemble performs better than the SCC ensemble. However, in the case of PCCs, for a latency of 4096 bits, the uncoupled ensemble performs slightly better than the corresponding

coupled ensemble. Interestingly, for equal latency, the SC-SCC ensemble outperforms the SC-PCC ensemble in the waterfall region. Thus, the SC-SCC ensemble yields better performance in both the waterfall and error floor regions.

In the following section, we investigate the impact of spatial coupling on the error floor performance of TCs.

4 Spatial Coupling: Error Floor Region Performance

Similar to uncoupled TC ensembles, to analyze the performance of SC-TC ensembles in the error floor region, one could derive bounds based on the WEFs of the ensembles. Unfortunately, deriving the WEF for SC-TCs is cumbersome. In this section, we establish a connection between the WEF of SC-TC ensembles and that of the corresponding uncoupled ensembles. In particular, we prove that, under certain conditions, spatial coupling does not decrease the minimum distance of TCs. This allows us to use the WEF analysis of TCs to estimate the error floor performance of SC-TCs. A similar connection between LDPC and SC-LDPC codes is proved in [23], [24], [25]. Here, we restrict ourselves to SC-TCs with coupling memory $m = 1$, but the proof can be generalized to higher coupling memories.

Theorem 2 *Consider an uncoupled PCC, $\tilde{\mathcal{C}}$, (see Figure 1(b)) with permutation Π^{Un} and parity-check matrices $\boldsymbol{H}_U$ and $\boldsymbol{H}_L$ corresponding to the upper and lower component encoders. It is possible to unwrap the PCC to form an SC-PCC, $\mathcal{C}$ (Figure 1(c)). For the SC-PCC, we assume a length-L coupled chain with termination or tailbiting, and time-invariant permutations. Let us denote the permutations by $\Pi_t^U = \Pi^U$, $\Pi_t^L = \Pi^L$, and $\Pi_t = \Pi$, and assume that they satisfy*

$$\Pi^{Un} = (\Pi^U)^{-1} \cdot \Pi \cdot \Pi^L.$$

Then, for any codeword $\boldsymbol{v} \in \mathcal{C}$, $\boldsymbol{v} = (\boldsymbol{v}_1, \ldots, \boldsymbol{v}_L)$, $\boldsymbol{v}_t = (\boldsymbol{u}_t, \boldsymbol{v}_t^U, \boldsymbol{v}_t^L)$, with Hamming weight $w_H(\boldsymbol{v})$, there exists a codeword $\tilde{\boldsymbol{v}} \in \tilde{\mathcal{C}}$ such that

$$w_H(\tilde{\boldsymbol{v}}) \leq w_H(\boldsymbol{v}) \, .$$

Proof 6 *We prove this theorem for tailbiting of the coupled chain, which contains termination as a special case. The result is thus valid for both cases. Any codeword $\boldsymbol{v} \in \mathcal{C}$ satisfies the local constraints for $t = 1, \ldots, L$. Therefore, at time t,*

$$\left((\boldsymbol{u}_{t,0}, \boldsymbol{u}_{t-1,1}) \cdot \Pi^U \quad \boldsymbol{v}_t^U\right) \cdot \boldsymbol{H}_U^T = \boldsymbol{0}, \tag{2}$$

$$\left((\boldsymbol{u}'_{t,0}, \boldsymbol{u}'_{t-1,1}) \cdot \Pi^L \quad \boldsymbol{v}_t^L\right) \cdot \boldsymbol{H}_L^T = \boldsymbol{0}, \tag{3}$$

where $\boldsymbol{u}'_t = \boldsymbol{u}_t \cdot \Pi$. *The constraints are linear and time-invariant. Thus, for* $t = 1, \ldots, L$, *any superposition of the vectors* $\left((\boldsymbol{u}_{t-1,1}, \boldsymbol{u}_{t,2}) \cdot \Pi_1 \quad \boldsymbol{v}_t^U\right)$ *and* $\left((\boldsymbol{u}'_{t-1,1}, \boldsymbol{u}'_{t,2}) \cdot \Pi_1 \quad \boldsymbol{v}_t^L\right)$ *satisfies* (2) *and* (3), *respectively. In particular, consider*

$$\begin{aligned}
&\sum_{t=1}^{L} \left((\boldsymbol{u}_{t,0}, \boldsymbol{u}_{t-1,1}) \cdot \Pi^U \quad \boldsymbol{v}_t^U\right) \\
&= \Big(\sum_{t=1}^{L} (\boldsymbol{u}_{t,0}, \boldsymbol{u}_{t-1,1}) \cdot \Pi^U \quad \sum_{t=1}^{L} \boldsymbol{v}_t^U\Big) \\
&= \Big(\sum_{t=1}^{L} \boldsymbol{u}_t \cdot \Pi^U \quad \sum_{t=1}^{L} \boldsymbol{v}_t^U\Big)\,,
\end{aligned} \tag{4}$$

and

$$\begin{aligned}
&\sum_{t=1}^{L} \left((\boldsymbol{u}'_{t,0}, \boldsymbol{u}'_{t-1,1}) \cdot \Pi^L \quad \boldsymbol{v}_t^L\right) \\
&= \Big(\sum_{t=1}^{L} (\boldsymbol{u}'_{t,0}, \boldsymbol{u}'_{t-1,1}) \cdot \Pi^L \quad \sum_{t=1}^{L} \boldsymbol{v}_t^L\Big) \\
&= \Big(\sum_{t=1}^{L} \boldsymbol{u}'_t \cdot \Pi^L \quad \sum_{t=1}^{L} \boldsymbol{v}_t^L\Big) \\
&= \Big(\sum_{t=1}^{L} \boldsymbol{u}_t \cdot \Pi \cdot \Pi^L \quad \sum_{t=1}^{L} \boldsymbol{v}_t^L\Big)\,.
\end{aligned} \tag{5}$$

Let

$$\tilde{\boldsymbol{u}} = \sum_{t=1}^{L} \boldsymbol{u}_t \cdot \Pi^U, \quad \tilde{\boldsymbol{v}}^U = \sum_{t=1}^{L} \boldsymbol{v}_t^U, \quad \tilde{\boldsymbol{v}}^L = \sum_{t=1}^{L} \boldsymbol{v}_t^L\,.$$

Then, the vectors obtained from (4) *and* (5) *can be rewritten as* $(\tilde{\boldsymbol{u}} \quad \tilde{\boldsymbol{v}}^U)$ *and* $(\tilde{\boldsymbol{u}} \cdot \Pi^{Un} \quad \tilde{\boldsymbol{v}}^L)$, *respectively.*

The vectors from (4) *and* (5) *satisfy* (2) *and* (3), *respectively. Thus,*

$$(\tilde{\boldsymbol{u}} \quad \tilde{\boldsymbol{v}}^U) \cdot \boldsymbol{H}_U^T = \boldsymbol{0}, \tag{6}$$

$$(\tilde{\boldsymbol{u}} \cdot \Pi^{Un} \quad \tilde{\boldsymbol{v}}^L) \cdot \boldsymbol{H}_L^T = \boldsymbol{0}. \tag{7}$$

Therefore, $\tilde{\boldsymbol{v}} = (\tilde{\boldsymbol{u}}, \tilde{\boldsymbol{v}}^U, \tilde{\boldsymbol{v}}^L)$ is a codeword of the uncoupled ensemble.

If all nonzero elements of $\boldsymbol{v}_t$, $t = 1, \ldots, L$, occur at different positions, then $w_H(\tilde{\boldsymbol{v}}) = w_H(\boldsymbol{v})$. Otherwise, the overlap of the nonzero elements reduces the weight of $\tilde{\boldsymbol{v}}$ and $w_H(\tilde{\boldsymbol{v}}) < w_H(\boldsymbol{v})$.

This theorem can be extended to the other TC ensembles.

Theorem 3 *Consider an uncoupled SCC (BCC/HCC), $\tilde{\mathcal{C}}$, (Figure 2). It is possible to unwrap the SCC (BCC/HCC) to form an SC-SCC (BCC/SC-HCC), $\mathcal{C}$ (Figure 2). For the coupled code, we assume a length-L coupled chain with termination or tailbiting, and time-invariant permutations which satisfy certain conditions. Then, for any codeword $\boldsymbol{v} \in \mathcal{C}$, $\boldsymbol{v} = (\boldsymbol{v}_1, \ldots, \boldsymbol{v}_t, \ldots, \boldsymbol{v}_L)$, $\boldsymbol{v}_t = (\boldsymbol{u}_t, \boldsymbol{v}_t^U, \boldsymbol{v}_t^L)$, there exists a codeword $\tilde{\boldsymbol{v}} \in \tilde{\mathcal{C}}$ such that*

$$w_H(\tilde{\boldsymbol{v}}) \leq w_H(\boldsymbol{v}) \, .$$

Proof 7 *See Appendix.*

Corollary 1 *The minimum distance of an SC-TC ensemble $\mathcal{C}$ is larger than or equal to the minimum distance of the underlying uncoupled TC ensemble $\tilde{\mathcal{C}}$,*

$$d_{min}(\mathcal{C}) \geq d_{min}(\tilde{\mathcal{C}}) \, .$$

□

By the above theorems, we establish conditions on the permutations under which SC-TCs have equal or better minimum distance than their corresponding TCs. These conditions can be considered as guidelines for selecting proper permutations for SC-TCs. According to the theorems and the corollary above, the WEF analysis of uncoupled TC ensembles can be used to investigate the error floor and the minimum distance of SC-TC ensembles.

5 Weight Enumerator Analysis

In this section, we describe how to derive upper bounds on the error rate performance of TC ensembles and bounds on the minimum distance of these ensembles based on their WEFs [12], [20]. Then, we compare these bounds for different classes of TCs. For that, we first derive the average input-parity WEF (IP-WEF) of the component encoders. In particular, we describe the steps for a rate-2/3 recursive systematic convolutional encoder. A similar method can be used to derive the IP-WEF of any convolutional encoder with arbitrary rate R. Then, we use the obtained IP-WEFs to compute the average IP-WEFs of the TC ensembles.

5.1 Input-Parity Weight Enumerator

Let $A(I_1, I_2, P)$ denote the IP-WEF of a rate-2/3 recursive systematic convolutional encoder,

$$A(I_1, I_2, P) = \sum_{i_1} \sum_{i_2} \sum_{p} A_{i_1,i_2,p}\, I^{i_1} I^{i_2} P^p,$$

where the coefficient $A_{i_1,i_2,p}$ denotes the number of codewords with weight i_1, i_2, and p for the first input, the second input, and the parity sequence, respectively.

$A(I_1, I_2, P)$ can be computed as follows. For a trellis with s states, transitions within a trellis section can be described by an $s \times s$ matrix $\boldsymbol{M}$. The element of $\boldsymbol{M}$ in the rth row and the cth column, $[\boldsymbol{M}]_{r,c}$, corresponds to the trellis branch from the rth state to the cth state. More precisely, $[\boldsymbol{M}]_{r,c}$ is a monomial $I_1^{i_1} I_2^{i_2} P^p$, where i_1, i_2, and p are the weights corresponding to the first, second, and third outputs of the transition from the rth state to the cth state. For a trellis with N sections, the overall transition matrix is $\boldsymbol{M}^N$. Considering that the trellis is initialized and terminated to the all-zero state, the IP-WEF is given by the element $[\boldsymbol{M}^N]_{1,1}$.

Example 8 *Assume a terminated, rate-*2/3 *convolutional encoder with three trellis sections and generator matrix in* (1)*.*

The transition matrix can be written as

$$\boldsymbol{M}(I_1, I_2, P) = \begin{pmatrix} 1 & I_2P & I_1I_2 & I_1P \\ I_1 & I_1I_2P & I_2 & P \\ I_2P & 1 & I_1P & I_1I_2 \\ I_1I_2P & I_1 & P & I_2 \end{pmatrix},$$

and the IP-WEF becomes

$$\begin{aligned} &A(I_1, I_2, P) = [\boldsymbol{M}^3]_{1,1} = \\ &1 + I_2^3P^2 + 2I_1I_2P + I_1I_2P^3 + 2I_1I_2^2P + \\ &I_1I_2^2P^3 + I_1^2I_2 + 2I_1^2I_2P^2 + 3I_1^2I_2^2P^2 + I_1^3P + I_1^3I_2^3P\ . \end{aligned}$$

△

For a rate-1/2 convolutional encoder, we can obtain the transition matrix $\boldsymbol{M}$ in a similar way. Then, the IP-WEF of the encoder is given by $[\boldsymbol{M}^N]_{1,1}$ and can be written as

$$A(I, P) = \sum_{i} \sum_{p} A_{i,p}\, I^i P^p,$$

where $A_{i,p}$ is the number of codewords of input weight i and parity weight p.

Consider the PCC ensemble shown in Figure 2(b). Let $A^{\mathrm{T_U}}(I,P)$ and $A^{\mathrm{T_L}}(I,P)$ denote the IP-WEFs corresponding to the upper and lower component encoder, respectively. The overall IP-WEF depends on the IP-WEF of the component encoders and the permutation used. Averaging over all possible permutations, the coefficients of the average IP-WEF of the PCC ensemble, $\bar{A}^{\mathrm{PCC}}_{i,p}$, can be obtained as [20]

$$\bar{A}^{\mathrm{PCC}}_{i,p} = \frac{\sum_{p_1} A^{\mathrm{T_U}}_{i,p_1} \cdot A^{\mathrm{T_L}}_{i,p-p_1}}{\binom{N}{i}} \,. \tag{8}$$

For the SCC ensemble shown in Figure 2(b), we denote the IP-WEFs of the outer and inner encoder by $A^{\mathrm{T_O}}(I,P)$ and $A^{\mathrm{T_I}}(I,P)$, respectively. Similar to PCCs, the average IP-WEF of the SCC ensemble, $\bar{A}^{\mathrm{SCC}}_{i,p}$, can be computed by averaging over all possible permutations [12]. The coefficients $\bar{A}^{\mathrm{SCC}}_{i,p}$ can be written as

$$\bar{A}^{\mathrm{SCC}}_{i,p} = \sum_{p_1} \frac{A^{\mathrm{T_O}}_{i,p_1} \cdot A^{\mathrm{T_I}}_{i+p_1,p-p_1}}{\binom{2N}{i+p_1}} \,. \tag{9}$$

We denote the IP-WEFs corresponding to the upper and lower component encoders of the BCC ensemble (Figure 2(c)) by $A^{\mathrm{T_U}}(I,P)$ and $A^{\mathrm{T_L}}(I,P)$, respectively. The coefficients of the average IP-WEF, $\bar{A}^{\mathrm{BCC}}_{i,p}$, can be computed as

$$\bar{A}^{\mathrm{BCC}}_{i,p} = \sum_{p_1} \frac{A^{\mathrm{T_U}}_{i,p_1,p-p_1} \cdot A^{\mathrm{T_L}}_{i,p-p_1,p_1}}{\binom{N}{i}\binom{N}{p_1}\binom{N}{p-p_1}} \,. \tag{10}$$

To compute the average IP-WEF of the HCC ensemble, $\bar{A}^{\mathrm{HCC}}_{i,p}$, it is possible to combine the methods that we used for PCCs and SCCs. First, the average IP-WEF of the parallel component is computed. Then, $\bar{A}^{\mathrm{HCC}}_{i,p}$ can be obtained by substituting $A^{\mathrm{T_O}}(I,P)$ in (9) by the computed average IP-WEF of the parallel component [16],

$$\bar{A}^{\mathrm{HCC}}_{i,p} = \sum_{p_1} \sum_{p_2} \frac{A^{\mathrm{T_U}}_{i,p_1} \cdot A^{\mathrm{T_L}}_{i,p_2} \cdot A^{\mathrm{T_I}}_{p_1+p_2,p-p_1-p_2}}{\binom{N}{i}\binom{2N}{p_1+p_2}} \,. \tag{11}$$

It is worth mentioning that by the use of the compact graph representation, TCs can be seen as a class of protograph-based generalized LDPC (GLDPC) codes. Therefore, equivalently, it is possible to compute the average IP-WEF of TCs by the method developed for GLDPC codes in [21], [22].

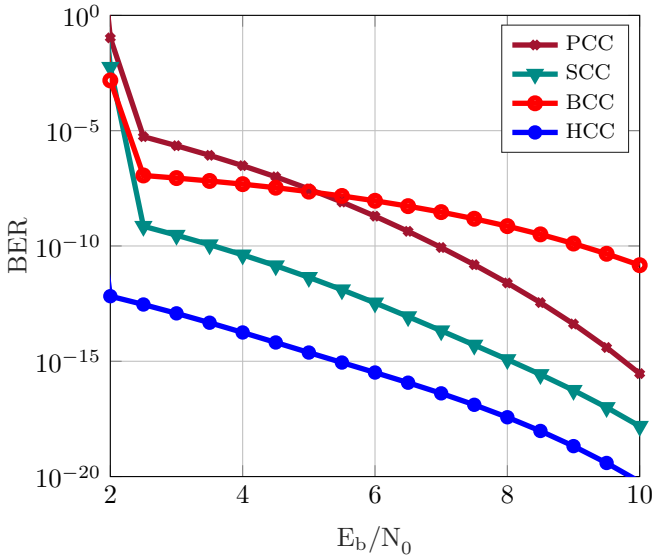


Figure 4: Union bound on performance of the TCs, $K = 512$, $R = 1/3$.

5.2 Bounds on the Error Probability

Consider transmission of codewords of a rate-R TC ensemble over the AWGN channel. For a maximum likelihood (ML) decoder, the BER is upper bounded by

$$P_b \leq \sum_{i=1}^{N} \sum_{p=1}^{N(1/R-1)} \frac{i}{N} \bar{A}_{i,p} \, \mathrm{Q}\left(\sqrt{2(i+p)R\frac{E_b}{N_0}}\right). \tag{12}$$

Likewise, the frame error rate (FER) is upper bounded by

$$P_F \leq \sum_{i=1}^{N} \sum_{p=1}^{N(1/R-1)} \bar{A}_{i,p} \, \mathrm{Q}\left(\sqrt{2(i+p)R\frac{E_b}{N_0}}\right), \tag{13}$$

where $Q(.)$ is the Q-function and E_b/N_0 is the signal-to-noise ratio.

Figure 4 shows the bounds on the BER performance of the different classes of TCs for $R = 1/3$ and $K = 512$. The bounds are truncated at weight $w = 320$, which is larger than the corresponding Gilbert-Varshamov limit. The HCC ensemble has the lowest error floor, while the BCC and PCC ensembles have the highest error floors. Surprisingly, the error floor of the BCC ensemble is not only high but also has the worst slope among all TC ensembles. On the other hand, the excellent MAP thresholds of the BCC ensemble suggest

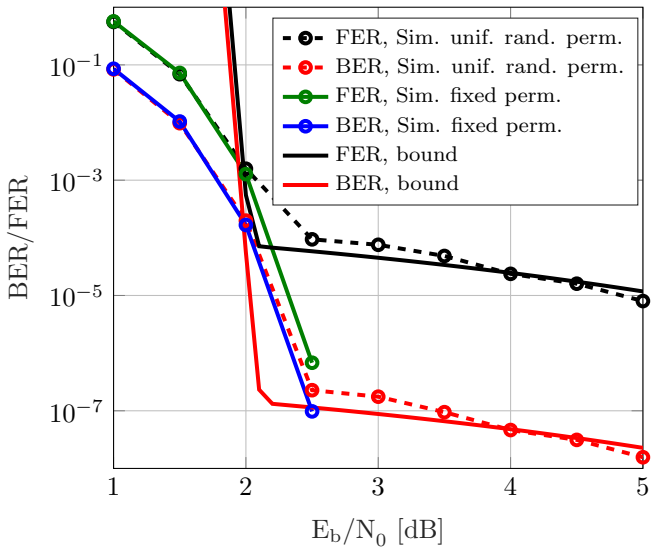


Figure 5: Bounds on performance of the BCCs and simulation results for uniformly random and fixed permutations, $K = 512$, $R = 1/3$.

a good performance for this ensemble under MAP decoding. The contradiction between the excellent MAP threshold of the BCC ensemble and its poor bound suggests that the performance is dominated by few bad permutations. To verify this, we simulated the BCCs for two scenarios; first, a randomly selected but fixed set of permutations; second randomly chosen permutations for each simulated block. The results are shown in Figure 5, together with the corresponding bounds. The figure shows that the bounds are in agreement with the simulation results for uniformly random permutations. However, it indicates a significant improvement in FER for the fixed set of permutations. For example, at $E_b/N_0 = 2.5$ dB, the FER improves from $9.5 \cdot 10^{-5}$ to $6.8 \cdot 10^{-7}$. This significant improvement caused by fixing the permutations, supports that the high floor of the BCC ensemble is caused by the poor performance of a small fraction of codes. Thus, in the next section, we compute expurgated union bounds.

5.3 Bound on the Minimum Distance and Expurgated Bounds

Consider a TC ensemble consisting of Ω codes in total. The value Ω follows from the different possible combinations of permutations and depends on the type of the ensemble. Assume that all codes in the ensemble are selected with equal

probability. Then, the number of codewords with weight w over all possible codes in the ensemble is $\Omega\bar{A}_w$, where $\bar{A}_w$ is the average WEF of the ensemble. Therefore, given an integer value $\tilde{d}$, the total number of codewords with weight $w < \tilde{d}$ can be computed as

$$\Omega_{w<\tilde{d}} = \Omega \sum_{w=1}^{\tilde{d}-1} \bar{A}_w \ .$$

By considering that these codewords are spread over different possible codes, we can obtain an upper bound on the number of codes with minimum distance $d_{\min} \geq \tilde{d}$,

$$\Omega_{w\geq\tilde{d}} < \Omega - \Omega \sum_{w=1}^{\tilde{d}-1} \bar{A}_w \ .$$

Let α denote the fraction of codes with $d_{\min} \geq \tilde{d}$. Then, α is upperbounded by

$$\alpha < 1 - \sum_{w=1}^{\tilde{d}-1} \bar{A}_w \ . \tag{14}$$

For a given α and $\bar{A}_w$, an analytical bound on the minimum distance of an ensemble can be obtained by computing the largest $\tilde{d}$ which satisfies (14). In fact, this bound shows that a fraction α of all possible codes has minimum distance $d_{\min} \geq \tilde{d}$. In Figure 6, considering different classes of TCs with $R = 1/3$, this bound is computed for $\alpha = 0.5$ and different input block lengths. As it can be seen, the HCC ensemble has the best minimum distance, and the PCC ensemble the worst. As an example, for $K = 300$ the values computed for HCCs, BCCs, SCCs, PCCs are $\tilde{d} = 129$, 99, 37, and 10, respectively. Comparing the results in Figure 6 and the thresholds in Table 1, we can observe that the TC ensembles with good MAP threshold also have good minimum distance. According to Figure 6, for both the BCC and HCC ensembles, the minimum distance grows linearly with the input block length [16], [26]. However, the bound on the minimum distance of the HCC ensemble has a higher slope and grows faster than that of the BCC ensemble.

Consider excluding a fraction $(1 - \alpha)$ of codes with $d_{\min} < \tilde{d}$ from a TC ensemble. Then, it is possible to compute the upper bound on the performance of this expurgated ensemble. The average BER of the expurgated ensemble is upperbounded by

$$P_b \leq \frac{1}{\alpha} \sum_{i=1}^{kN} \sum_{w=\tilde{d}}^{nN} \frac{i}{N} \bar{A}_{i,w} \, \mathsf{Q}\left(\sqrt{2wR\frac{E_\mathrm{b}}{N_0}}\right). \tag{15}$$

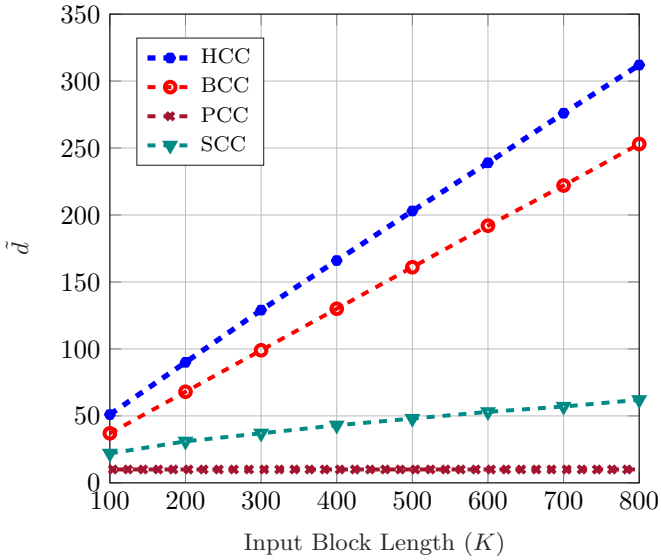


Figure 6: Bound on the minimum distance, fraction $\alpha = 0.5$ of codes have $d_{\min} > \tilde{d}$.

The bounds for the expurgated TC ensembles are shown in Figure 7 for $\alpha = 0.5$, which means that half of the codes with $d_{\min} < \tilde{d}(\alpha)$ are excluded. For comparison, we also provide the corresponding union bounds in the same figure. It can be seen that for all TC ensembles except the PCC ensemble, the error floor estimated by the expurgated bound is much steeper and lower than that resulting from the union bound. In other words, expurgation improves the performance of the SCC, BCC, and HCC ensembles significantly.

For the BCC and HCC ensembles, the gap between the expurgated bound and the union bound is very large and notable. To investigate the influence of expurgation on the performance of these ensembles, in Figure 8 we provide the expurgated bound on the BER of the BCC and HCC ensembles for $\alpha = 0.5$ and $\alpha = 0.99$. Note that for $\alpha = 0.99$, the expurgated bounds are computed by excluding only 1% of the possible codes, and these bounds are still significantly lower and steeper than the corresponding union bounds. For the BCC ensemble, the gap between the expurgated bounds for $\alpha = 0.5$ and $\alpha = 0.99$ is much smaller than that of the HCC ensemble. Therefore, for $\alpha = 0.99$, the BCC ensemble has slightly steeper and lower error floor than the HCC ensemble. The fact that changing α has a little impact on the expurgation of the BCC ensemble suggests that only a small fraction of the codes have poor distance properties. This means that for a BCC with randomly selected but fixed permutations, with high probability the error floor is as steep and low as

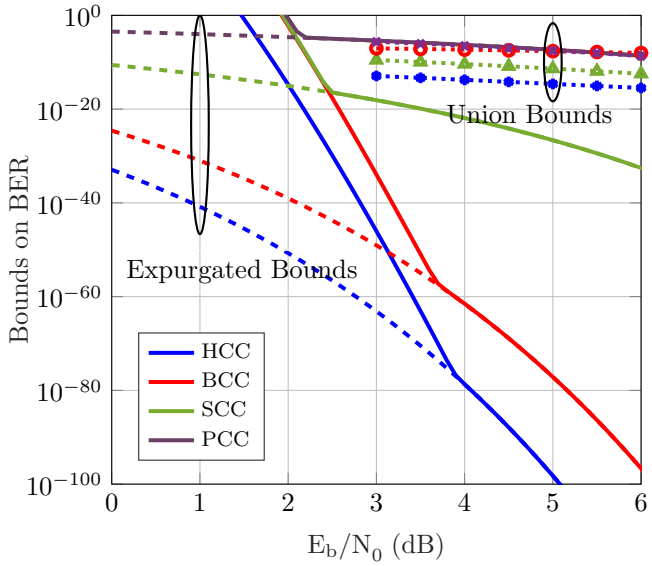


Figure 7: Expurgated union bound on performance of TCs, $\alpha = 0.5$, $K = 512$, and $R = 1/3$.

the corresponding expurgated bound for an ML decoder.

6 Discussion and Conclusion

We investigated the performance of SC-TC ensembles with finite block length in both waterfall and error floor regions. The two primary results can be summarized as follows. First, spatial coupling not only improves the asymptotic decoding threshold of an ensemble but also, for finite length and given latency, it improves the performance of the ensemble in the waterfall region. Second, considering certain conditions, spatial coupling either improves or preserves the minimum distance of the ensemble. Therefore, the coupled ensembles cannot have worse error floor than the corresponding uncoupled TC ensembles. Using this fact, instead of performing the cumbersome WEF analysis for the coupled ensemble, we derived the WEF for uncoupled ensembles. Then, based on the WEF, we computed bounds on BER performance and the minimum distance of TCs. As the coupled ensembles have equal or larger minimum distance than the uncoupled ensembles, the computed bounds for TCs can be used to approximate the error floor of SC-TCs. The results from the WEF analysis of TCs demonstrate very low error floors for SCC, BCC, and HCC esnembles. Moreover, for the BCC and HCC ensembles, the minimum distance

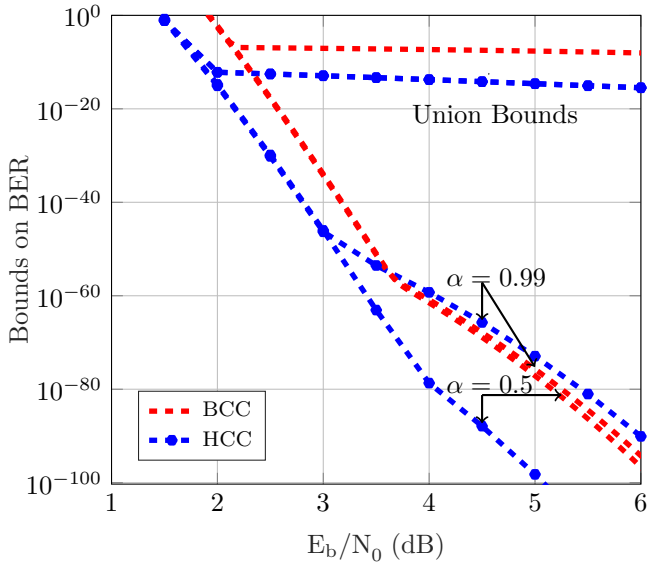


Figure 8: Expurgated union bound of HCCs and BCCs for $\alpha = 0.5$ and $\alpha = 0.99$, $K = 512$, $R = 1/3$.

grows linearly with block length.

Comparing the thresholds of SC-TC ensembles (see Table 1) and the results from the WEF analysis, we observe that the ensembles with better MAP thresholds also have larger minimum distance and lower error floor. However, so far, only PCCs have been widely used in various standards—such as UMTS and LTE— because of their good BP thresholds and good performance in the waterfall region. Other TC ensembles have received much less attention for commercial use, although they have better MAP threshold and distance properties than PCCs. Our results confirm that the BP thresholds of these ensembles can be significantly improved by applying coupling. Also, regarding the finite length regime, while their error floor stays at very low error probabilities, their waterfall performance gets much closer to capacity. This brings us to the conclusion that by coupling a given ensemble with close to capacity MAP threshold and low error floor, such as SCCs, BCCs, and HCCs, the resulting ensemble is very promising and can perform close-to-capacity, yet achieving low error floor, with a low complexity iterative decoder.

Finally, we should remark that the considered bounds estimate the error floor of an ML decoder. To obtain bounds on the performance of the BP decoder, more investigations on the corresponding absorbing sets [27] and pseudo-codewords [28] need to be done.

Appendix

Proof of *Theorem 2*

We prove the theorem for the general case of tailbiting.

Serially Concatenated Codes

Consider the SCC and SC-SCC ensembles in Figure 2(a) and (b), and assume that

$$\Pi^{\mathrm{Un}} = \Pi^{(1)} \cdot \Pi^{(2)}.$$

Any codeword $\boldsymbol{v} \in \mathcal{C}$ satisfies the local constraints for $t = 1, \ldots, L$. Therefore, at time t,

$$\begin{pmatrix} \boldsymbol{u}_t & \boldsymbol{v}_t^{\mathrm{O}} \end{pmatrix} \cdot \boldsymbol{H}_{\mathrm{O}}^T = \boldsymbol{0}\,, \tag{1}$$

$$\begin{pmatrix} (\tilde{\boldsymbol{v}}'_{t,0} & \tilde{\boldsymbol{v}}'_{t-1,1}) \cdot \Pi^{(2)} & \boldsymbol{v}_t^{\mathrm{I}} \end{pmatrix} \cdot \boldsymbol{H}_{\mathrm{I}}^T = \boldsymbol{0}\,, \tag{2}$$

where $\tilde{\boldsymbol{v}}'_t = (\boldsymbol{u}_t, \boldsymbol{v}_t^{\mathrm{O}}) \cdot \Pi^{(1)}$. The constraints are linear and time-invariant. Therefore, any superposition of the vectors $\begin{pmatrix} \boldsymbol{u}_t & \boldsymbol{v}_t^{\mathrm{O}} \end{pmatrix}$ and $\begin{pmatrix} (\tilde{\boldsymbol{v}}'_{t-1,1} & \tilde{\boldsymbol{v}}'_{t,2}) \cdot \Pi^{(2)} & \boldsymbol{v}_t^{\mathrm{I}} \end{pmatrix}$ from different time slots $t = 1, \ldots, L$, satisfy (1) and (2), respectively. In particular, we can consider

$$\sum_{t=1}^{L} \begin{pmatrix} \boldsymbol{u}_t & \boldsymbol{v}_t^{\mathrm{O}} \end{pmatrix} = \Big(\sum_{t=1}^{L} \boldsymbol{u}_t \quad \sum_{t=1}^{L} \boldsymbol{v}_t^{\mathrm{O}} \Big), \tag{3}$$

and

$$\begin{aligned}
&\sum_{t=1}^{L} \big((\tilde{\boldsymbol{v}}'_{t,0} \quad \tilde{\boldsymbol{v}}'_{t-1,1}) \cdot \Pi^{(2)} \quad \boldsymbol{v}_t^{\mathrm{I}}\big) \\
&= \big(\sum_{t=1}^{L} (\tilde{\boldsymbol{v}}'_{t,0} \quad \tilde{\boldsymbol{v}}'_{t-1,1}) \cdot \Pi^{(2)} \quad \sum_{t=1}^{L} \boldsymbol{v}_t^{\mathrm{I}}\big) \\
&= \big(\sum_{t=1}^{L} \tilde{\boldsymbol{v}}'_t \cdot \Pi^{(2)} \quad \sum_{t=1}^{L} \boldsymbol{v}_t^{\mathrm{I}}\big) \\
&= \big(\sum_{t=1}^{L} (\boldsymbol{u}_t \quad \boldsymbol{v}_t^{\mathrm{O}}) \cdot \Pi^{(1)} \cdot \Pi^{(2)} \quad \sum_{t=1}^{L} \boldsymbol{v}_t^{\mathrm{I}}\big). \qquad (4)
\end{aligned}$$

Let

$$\tilde{\boldsymbol{u}} = \sum_{t=1}^{L} \boldsymbol{u}, \quad \tilde{\boldsymbol{v}}^{\mathrm{O}} = \sum_{t=1}^{L} \boldsymbol{v}^{\mathrm{O}}, \quad \tilde{\boldsymbol{v}}^{\mathrm{I}} = \sum_{t=1}^{L} \boldsymbol{v}^{\mathrm{I}},$$

and substitute (3) and (4) into (1) and (2), respectively. Then

$$\big(\tilde{\boldsymbol{u}} \quad \tilde{\boldsymbol{v}}^{\mathrm{O}}\big) \cdot \boldsymbol{H}_{\mathrm{O}}^{T} = \boldsymbol{0}\,, \qquad (5)$$

$$\big((\tilde{\boldsymbol{u}} \quad \tilde{\boldsymbol{v}}^{\mathrm{O}}) \cdot \Pi^{\mathrm{Un}} \quad \tilde{\boldsymbol{v}}^{\mathrm{I}}\big) \cdot \boldsymbol{H}_{\mathrm{I}}^{T} = \boldsymbol{0}\,. \qquad (6)$$

Therefore, $\tilde{\boldsymbol{v}} = (\tilde{\boldsymbol{u}}, \tilde{\boldsymbol{v}}^{\mathrm{O}}, \tilde{\boldsymbol{v}}^{\mathrm{I}})$ is a codeword of the uncoupled code. If all nonzero elements of $\boldsymbol{v}_t$, $t = 1, \ldots, L$, occur at different positions, then $w_{\mathrm{H}}(\tilde{\boldsymbol{v}}) = w_{\mathrm{H}}(\boldsymbol{v})$. Otherwise, the overlap of the non zero elements reduces the weight of $\tilde{\boldsymbol{v}}$ and $w_{\mathrm{H}}(\tilde{\boldsymbol{v}}) < w_{\mathrm{H}}(\boldsymbol{v})$.

Braided Convolutional Codes

Consider the SCC and SC-SCC ensembles in Figure 2(c) and (d), and assume that $\Pi_t = \Pi$, $\Pi_t^{\mathrm{U}} = \Pi^{\mathrm{U}}$ and $\Pi_t^{\mathrm{L}} = \Pi^{\mathrm{L}}$. A valid code sequence of $\mathcal{C}$ has to satisfy the local constraints

$$\big(\boldsymbol{u}_t \quad \boldsymbol{v}_{t-1}^{\mathrm{L}} \cdot \Pi_t^{\mathrm{U}} \quad \boldsymbol{v}_t^{\mathrm{U}}\big) \cdot \boldsymbol{H}_{\mathrm{U}}^{T} = \boldsymbol{0}\,, \qquad (7)$$

$$\big(\boldsymbol{u}_t \cdot \Pi_t \quad \boldsymbol{v}_{t-1}^{\mathrm{U}} \cdot \Pi_t^{\mathrm{L}} \quad \boldsymbol{v}_t^{\mathrm{L}}\big) \cdot \boldsymbol{H}_{\mathrm{L}}^{T} = \boldsymbol{0} \qquad (8)$$

for all $t = 1, \ldots, L$, where $\boldsymbol{H}_{\mathrm{U}}$ and $\boldsymbol{H}_{\mathrm{L}}$ are the parity-check matrices that represent the constraints imposed by the trellises of the upper and lower component encoders, respectively. Since these constraints are linear and time-invariant, it follows that any superposition of vectors $\boldsymbol{v}_t = (\boldsymbol{u}_t, \boldsymbol{v}_t^{\mathrm{U}}, \boldsymbol{v}_t^{\mathrm{U}})$ from different time instants $t \in \{1, \ldots, L\}$ will also satisfy (7) and (8). In particular, if we let

$$\tilde{\boldsymbol{u}} = \sum_{t=1}^{L} \boldsymbol{u}_t\,, \quad \tilde{\boldsymbol{v}}^{\mathrm{L}} = \sum_{t=1}^{L} \boldsymbol{v}_t^{\mathrm{L}}\,, \quad \tilde{\boldsymbol{v}}^{\mathrm{U}} = \sum_{t=1}^{L} \boldsymbol{v}_t^{\mathrm{U}}$$

then

$$\begin{pmatrix} \tilde{\boldsymbol{u}} & \tilde{\boldsymbol{v}}^{\mathrm{L}} \cdot \Pi^{\mathrm{U}} & \tilde{\boldsymbol{v}}^{\mathrm{U}} \end{pmatrix} \cdot \boldsymbol{H}_{\mathrm{U}}^{T} = \mathbf{0} \,, \tag{9}$$

$$\begin{pmatrix} \tilde{\boldsymbol{u}} \cdot \Pi & \tilde{\boldsymbol{v}}^{\mathrm{U}} \cdot \Pi^{\mathrm{L}} & \tilde{\boldsymbol{v}}^{\mathrm{L}} \end{pmatrix} \cdot \boldsymbol{H}_{\mathrm{L}}^{T} = \mathbf{0} \,. \tag{10}$$

Here we have implicitly made use of the fact that $\boldsymbol{v}_t = \mathbf{0}$ for $t < 1$ and $t > L$. But now it follows from (9) and (10) that $\tilde{\boldsymbol{v}} = (\tilde{\boldsymbol{u}}, \tilde{\boldsymbol{v}}^{\mathrm{U}}, \tilde{\boldsymbol{v}}^{\mathrm{L}}) \in \tilde{\mathcal{C}}$, i.e., we obtain a codeword of the uncoupled code. If all nonzero symbols within $\boldsymbol{v}_t$ occur at different positions for $t = 1, \ldots, L$, then $w_{\mathrm{H}}(\tilde{\boldsymbol{v}}) = w_{\mathrm{H}}(\boldsymbol{v})$. If, on the other hand, the support of nonzero symbols overlaps, the weight of $\tilde{\boldsymbol{v}}$ is reduced accordingly and $w_{\mathrm{H}}(\tilde{\boldsymbol{v}}) < w_{\mathrm{H}}(\boldsymbol{v})$.

The same result can be proved for HCCs by combining the proofs for PCCs and SCCs.

References

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

[2] R. Gallager, *Low-density parity-check codes*. Cambridge, MA: MIT Press, 1963.

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

[4] M. Lentmaier, A. Sridharan, D. J. Costello and K. S. Zigangirov, "Iterative decoding threshold analysis for LDPC convolutional codes", *IEEE Trans. Inf. Theory*, vol. 56, no. 10, pp. 5274–5289, Oct. 2010.

[5] S. Kudekar, T. Richardson and R. 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.

[6] A. Yedla, Y. Y. Jian, P. S. Nguyen and H. D. Pfister, "A simple proof of threshold saturation for coupled scalar recursions", in *Proc. Int. Symp. Turbo Codes and Iterative Inf. Process. (ISTC)*, Gothenburg, Sweden, Aug. 2012.

[7] A. Yedla, Y. Y. Jian, P. S. Nguyen and H. D. Pfister, "A simple proof of threshold saturation for coupled vector recursions", in *Proc. IEEE Inf. Theory Work. (ITW)*, Lausanne, Switzerland, Sep. 2012.

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

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

[10] ——, "Spatially coupled turbo-like codes", *IEEE Trans. Inf. Theory*, vol. 63, no. 10, pp. 6199–6215, Oct. 2017.

[11] ——, "Spatially coupled hybrid concatenated codes", in *Proc. 11th Int. ITG Conf. Systems, Commun. and Coding (SCC)*, Hamburg, Germany, 2017.

[12] 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.

[13] A. Graell i Amat, G. Montorsi and F. Vatta, "Design and performance analysis of a new class of rate compatible serially concatenated convolutional codes", *IEEE Trans. Commun.*, vol. 57, no. 8, pp. 2280–2289, Aug. 2009.

[14] 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.

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

[16] E. Rosnes and A. Graell i Amat, "Performance analysis of 3-D turbo codes", *IEEE Trans. Inf. Theory*, vol. 57, no. 6, pp. 3707–3720, Jun. 2011.

[17] C. Koller, A. Graell i Amat, J. Kliewer, F. Vatta and D. J. Costello, "Hybrid concatenated codes with asymptotically good distance growth", in *Proc. Int. Symp. Turbo Codes and Related Topics (ISTC)*, Sep. 2008.

[18] D. G. M. Mitchell, M. Lentmaier, A. E. Pusane and D. J. Costello, "Randomly punctured LDPC codes", *IEEE J. Sel. Areas Commun.*, vol. 34, no. 2, pp. 408–421, Feb. 2016.

[19] M. U. Farooq, S. Moloudi and M. Lentmaier, *Thresholds of Braided Convolutional Codes on the AWGN Channel*, submitted to *IEEE Int. Symp. Inf. Theory (ISIT)*, available online at `https://arxiv.org/abs/1802.10540`, 2018.

[20] S. Benedetto and G. Montorsi, "Unveiling turbo codes: Some results on parallel concatenated coding schemes", *IEEE Trans. Inf. Theory*, vol. 42, no. 2, pp. 409–428, Mar. 1996.

[21] S. Abu-Surra, W. Ryan and D. Divsalar, "Ensemble enumerators for protograph-based generalized LDPC codes", *Ieee global telecommun. conf. (globecom)*, pp. 1492–1497, 2007.

[22] S. Abu-Surra, D. Divsalar and W. E. Ryan, "Enumerators for protograph-based ensembles of LDPC and generalized LDPC codes", *Ieee trans. inf. theory*, vol. 57, no. 2, pp. 858–886, Feb. 2011.

[23] D. Truhachev, K. S. Zigangirov and D. J. Costello, "Distance bounds for periodically time-varying and tail-biting LDPC convolutional codes", *Ieee trans. inf. theory*, vol. 56, no. 9, pp. 4301–4308, Sep. 2010.

[24] D. G. M. Mitchell, A. E. Pusane and D. J. Costello, "Minimum distance and trapping set analysis of protograph-based LDPC convolutional codes", *Ieee trans. inf. theory*, vol. 59, no. 1, pp. 254–281, Jan. 2013.

[25] R. Smarandache, A. E. Pusane, P. O. Vontobel and D. J. Costello, "Pseudo-codeword performance analysis for LDPC convolutional codes", *Ieee trans. inf. theory*, vol. 55, no. 6, pp. 2577–2598, Jun. 2009.

[26] S. Moloudi, M. Lentmaier and A. Graell i Amat, "Finite length weight enumerator analysis of braided convolutional codes", in *Proc. Int. Symp. Inf. Theory and Its Applicat. (ISITA)*, Oct. 2016.

[27] D. G. M. Mitchell, L. Dolecek and D. J. Costello, "Absorbing set characterization of array-based spatially coupled LDPC codes", in *IEEE Int. Symp. Inf. Theory (ISIT)*, Honolulu, HI, USA, Jun. 2014.

[28] E. Rosnes, M. Helmling and A. Graell i Amat, "Minimum pseudoweight analysis of 3-dimensional turbo codes", *IEEE Trans. Commun.*, vol. 62, no. 7, pp. 2170–2182, Jul. 2014.

Paper VI

A Unified Ensemble of Concatenated Convolutional Codes

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.

1 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 [21]. In other words, the different trellises are combined to a single self-concatenated trellis. Probably, the most famous class of

[21] 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.

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.

2 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.

2.1 Parallel Concatenated Codes

Figure 3.4(a) and (b) show the encoder block diagram and compact graph representation [6] of a PCC, respectively. The 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 Figure 3.4(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 Figure 3.4(b) by a line that crosses the edge between $\boldsymbol{u}$ and $\boldsymbol{v}^{\mathrm{L}}$.

Figure 3.4(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 Figure 3.4(c), an RA code can

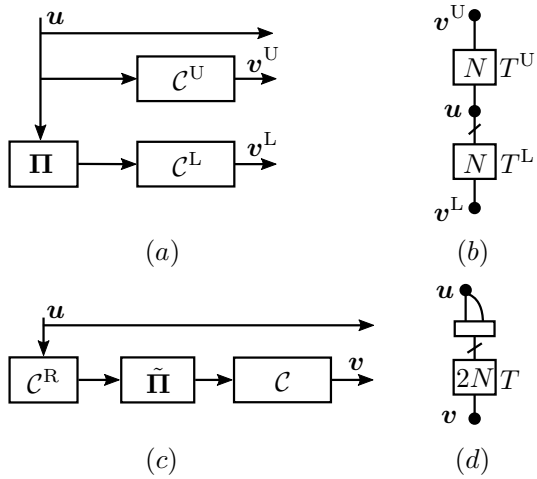


Figure 1: (a) Block diagram of a PCC, (b) Compact graph of a PCC, (c) Block diagram of a self-concatenated PCC, and (d) Compact graph of a self-concatenated PCC.

be obtained. However, the ensemble in Figure 3.4(c) is more general as $\mathcal{C}$ can be any convolutional encoder.

In the compact graph representation (see Figure 3.4(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

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

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

2.2 Serially Concatenated Codes

The encoder block diagram and the compact graph representation of the SCC ensemble are shown in Figure 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

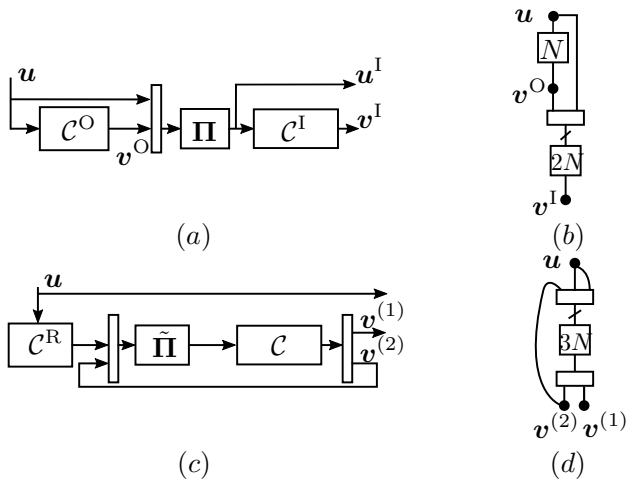


Figure 2: (a) Block diagram of a SCC, (b) Compact graph of a SCC, (c) Block diagram of a self-concatenated SCC, and (d) Compact graph of a self-concatenated SCC.

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 Figure 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 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

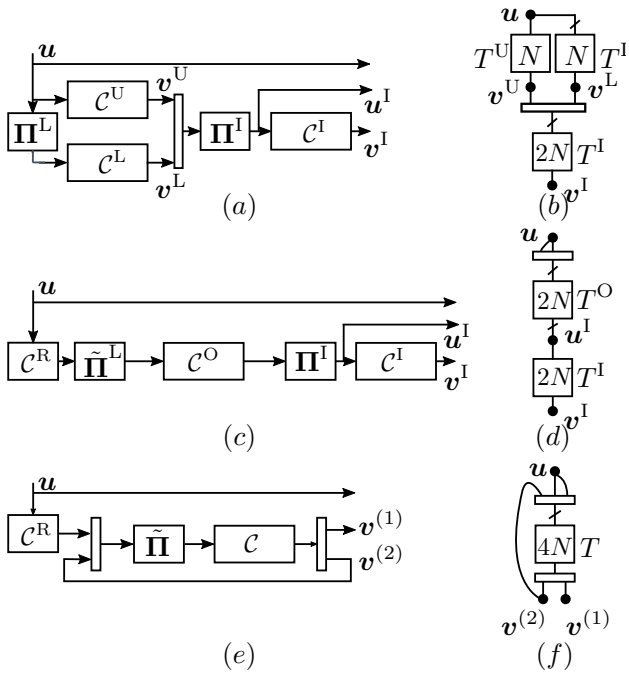


Figure 3: (a) Block diagram of a HCC, (b) Compact graph a HCC, (c) Block diagram of a self-concatenated HCC (step 1), (d) Compact graph of a self-concatenated HCC (step 1), (e) Block diagram of a self-concatenated HCC (step 2), and (f) Compact graph of a self-concatenated HCC (step 2).

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.

2.3 Hybrid Concatenated Codes

Figure 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 Figure 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 Figure 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 Figure 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-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.

2.4 Braided Convolutional Codes

Figure 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 [22] 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 Figure 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. Figure 4(c) and (d) show the encoder block diagram and compact graph representation of the corresponding self-concatenated ensemble. 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

[22] The component encoders are rate-2/3 convolutional encoders with all systematic bits punctured.

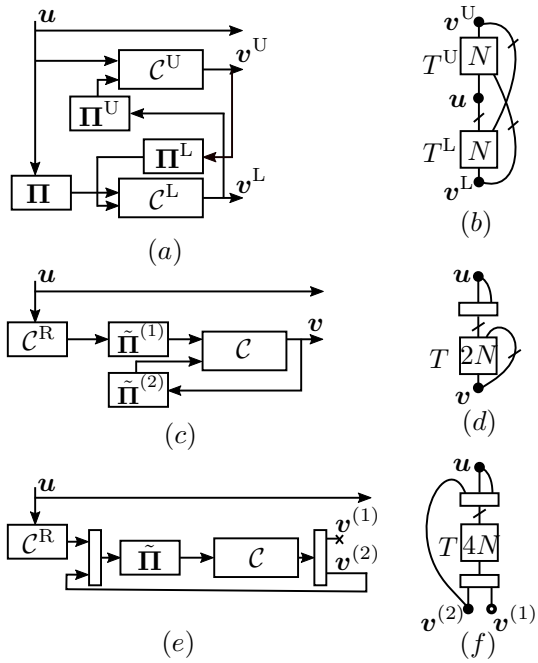


Figure 4: (a) Block diagram of a BCC, (b) Compact graph a BCC, (c) Block diagram of a self-concatenated BCC (step 1), (d) Compact graph of a self-concatenated BCC (step 1), (e) Block diagram of a self-concatenated BCC (step 2), and (f) Compact graph of a self-concatenated BCC (step 2).

$\tilde{\boldsymbol{\Pi}}^{(2)}$. 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 Figure 4(a) and (c) are equivalent.

The encoder ensembles in Figure 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

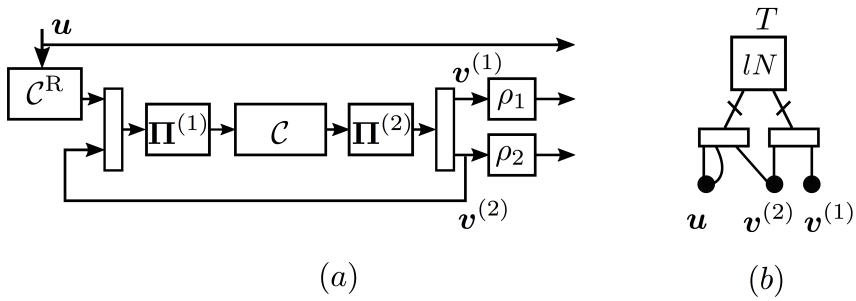


Figure 5: (a) Block diagram of encoder (b) Compact graph representation of the unified ensemble.

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 Figure 4(e) and its corresponding compact graph representation is depicted in Figure 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 $4N$.

3 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 Figure 3.4-Figure 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 Figure 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 1.

The sequence $\boldsymbol{v}$ is divided into two sequences $\boldsymbol{v}^{(1)}$ and $\boldsymbol{v}^{(2)}$ of length $l_1 N$ and $l_2 N$, 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

Table 1: 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

in Table 1. Note that $l = l_1 + l_2$.

Parameters ρ_1 and ρ_2 in Table 1 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 Figure 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 1, 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.

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

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

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

Table 2: Thresholds of PCCs, SCCs, BCCs and HCCs.

Ensemble	R	$\varepsilon_{\mathrm{BP}}$	$\varepsilon_{\mathrm{BP}}^{\mathrm{U}}$	$\varepsilon_{\mathrm{MAP}}$	$\varepsilon_{\mathrm{MAP}}^{\mathrm{U}}$
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

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 2. The table shows the BP threshold $\varepsilon_{\mathrm{BP}}$ and the MAP threshold $\varepsilon_{\mathrm{MAP}}$ of the original ensembles. To obtain $\varepsilon_{\mathrm{BP}}$ and $\varepsilon_{\mathrm{MAP}}$, we used the corresponding DE equations and the area theorem, respectively. We also report in the table the BP threshold $\varepsilon_{\mathrm{BP}}^{\mathrm{U}}$ and the MAP threshold $\varepsilon_{\mathrm{MAP}}^{\mathrm{U}}$ 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, $\varepsilon_{\mathrm{BP}}$ and $\varepsilon_{\mathrm{BP}}^{\mathrm{U}}$ 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, $\varepsilon_{\mathrm{BP}}$ is larger than $\varepsilon_{\mathrm{BP}}^{\mathrm{U}}$. For BCCs, replacing the rate-2 component encoder by a rate-1 component encoder with puncturing, is another reason for observing the gap between $\varepsilon_{\mathrm{BP}}$ and $\varepsilon_{\mathrm{BP}}^{\mathrm{U}}$. We also computed the thresholds for the self-concatenated ensemble in Figure 4(c). The obtained BP and MAP thresholds for this ensemble are identical to those of the original BCC ensemble.

The results in Table 2 demonstrate that thresholds similar to those of the original TC classes can be obtained by changing the design parameters in the unified ensemble.

4 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, "Time-varying periodic convolutional codes with low-density parity-check matrix", *IEEE Trans. Inf. Theory*, vol. 45, no. 5, pp. 2181–2190, Sep. 1999.

[2] S. Kudekar, T. Richardson and R. 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, Y.-Y. Jian, P. Nguyen and H. 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. Int. Symp. Turbo Codes and Iterative Inf. Process. (ISTC)*, Bremen, Germany, Agu. 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. Conf. Signal Process. and Commun. (SPCOM)*, Bangalore, India, Jul. 2014.

[6] S. Moloudi, M. Lentmaier and A. Graell i Amat, "Spatially coupled turbo-like codes", *IEEE Trans. Inf. Theory*, vol. 63, no. 10, pp. 6199–6215, Oct. 2017.

[7] C. Berrou, A. Glavieux and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes", in *Proc. IEEE Int. Conf. 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 *Proc. IEEE Int. Symp. on Inf. Theory*, 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. Int. Symp. Turbo Codes and Related Topics (ISTC)*, Sep. 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, Feb. 2011.

[13] D. Divsalar, H. Jin and R. J. McEliece, "Coding theorems for turbo-like codes", in *Proc. of the Annual Allerton Conf. on Commun. Control and Computing*, Illinois, USA, 1998.