UNIVERSITY OF CALIFORNIA, SAN DIEGO

On the Capacity of Finite State Channels and the Analysis of Convolutional Accumulate-m Codes

A dissertation submitted in partial satisfaction of the
requirements for the degree
Doctor of Philosophy

in

Electrical Engineering
(Communications Theory and Systems)

by

Henry D. Pfister

Committee in charge:

Professor Paul H. Siegel, Chair
Professor Edward Bender
Professor Alon Orlitsky
Professor Ruth Williams
Professor Jack Wolf

2003

The dissertation of Henry D. Pfister is approved, and it is acceptable in quality and form for publication on microfilm:

_____

_____

_____

_____

_____

Chair

University of California, San Diego

2003

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

It has been said many times that the journey is more important than the destination. I believe this saying fits the pursuit of a Ph.D. particularly well because neither the degree nor the dissertation captures the ethereal lessons learned through the triumphs and the failures. The following people helped me along the way and made the last five years both intellectually stimulating and personally rewarding.

First, I would like to thank my advisor, Professor Paul Siegel. So much of one's experience in graduate school is colored by the personality of their advisor, and his relaxed and open approach to research is the source of a warm and friendly atmosphere which permeates the group. Moreover, his dedication to teaching and to his students is without compare.

The other members of my dissertation committee were also invaluable along the way. Professor Jack Wolf's enthusiastic yet lighthearted approach to research was always refreshing, and his ability to explain complex problems in simple terms is truly astounding. Professor Alon Orlitsky provided me a firm foundation with his stellar first course on information theory and was always willing to provide a little more help along the way. Professor Edward Bender was also very helpful, and in particular I am grateful for his assistance with the proof of Theorem 3C.2. Professor Ruth Williams was always helpful as well, and I appreciate her assistance with some the mathematics of Chapter 4.

The administrative support provided by Cheryl Hacker, Karol Previte, and M'lissa Michelson has also been very valuable. Their efforts have allowed me to all but ignore the ceaseless paperwork that keeps the system running.

I would also like to thank my friends and colleagues in the STAR group and the Wolf Pack. Bruce Moision, who blazed the trail out of here and left some cool pottery in his wake. Mats Öberg for introducing me to "accumulate" codes and many valuable discussions along the way. Hugo Tullberg, whose quirky sense of humor and breadth of interests always made the lab fun. Jilei Hou for the fun we had in Washington D.C. and so many fruitful discussions in the lab. Marcus Marrow for a great trip to Dublin and keeping life properly focused on having fun. Mike Cheng, who arranged our tour of Texas and bore the burden of sysadmin. Joseph Soriaga for his patience with my ramblings and the collaboration we've had these past two years. Thanks also to Kai Tang, André des Rosiers, Brian Kurkoski, John Miller, Li Zhu, and Yan Zhang.

I would also like to acknowledge some of my other friends who have helped keep

me sane these past five years. Jay, Bill, Jeff, John, Arthur, and Kent for being there from the beginning. Kyle for all the good times in the Del Mar house and since then. Mike for our days of physics and motorcycles. Casey, Matt, Devon, and Amy for our legendary days on Pacifica. Chad for dragging me along on a crazy trip. Ashay for being a good roommate and a great friend. Shane, Marcus, Steve, and Brandon for keeping things lively around here.

I give special thanks to my family for their love and support through all of this. My mother, who always told me I could accomplish whatever goals I set, and my father, who gave me the tools and inspiration to study science. I'm also very lucky to have such a great sister and brother-in-law. I especially want to thank my wonderful girlfriend, Danielle, whose love and support has made the last two years very special for me.

The funding for this research was provided in part by the National Science Foundation (grant NCR-9612802), Marvell Semiconductor, Inc. (UC MICRO Grant 99-110), and the Center for Magnetic Recording Research (CMRR).

Chapter 2 is in part a reprint of the material in the papers: H. D. Pfister and P. H. Siegel, "The Serial Concatenation of Rate-1 Codes Through Uniform Random Interleavers," in *Proc. 37th Allerton Conference on Communication, Control and Computing*, Monticello, Illinois, pp. 260-269, Sep. 1999 and H. D. Pfister and P. H. Siegel, "The Serial Concatenation of Rate-1 Codes Through Uniform Random Interleavers," accepted *IEEE Trans. on Inform. Theory*, Oct. 2002. Chapter 3 is in part a reprint of the material in the paper: H. D. Pfister and P. H. Siegel, "Coding Theorems for Generalized Repeat Accumulate Codes," in *Proc. Int. Symp. on Inform. Theory and its Applications*, Honolulu, HI, pp. 21-25, Nov. 2000. Chapter 4 is in part a reprint of the material in the paper: H. D. Pfister, J. B. Soriaga and P. H. Siegel, "On the Achievable Information Rates of Finite State ISI Channels," in *Proc. IEEE Globecom*, San Antonio, TX, pp. 2992-2996, Nov. 2001. The dissertation author was the primary author of all these papers.

VITA

July 16, 1972	Born, Redondo Beach, California
1995	B.S., Physics, University of California, San Diego
2000	M.A., Electrical Engineering (Communications Theory and Systems), University of California, San Diego
2003	Ph.D., Electrical Engineering (Communications Theory and Systems), University of California, San Diego

PUBLICATIONS

H. D. Pfister and P. H. Siegel, "The Serial Concatenation of Rate-1 Codes Through Uniform Random Interleavers," in *Proc. 37th Allerton Conference on Communication, Control and Computing*, Monticello, Illinois, pp. 260-269, Sep. 1999.

Mats Öberg, H. D. Pfister, and P. H. Siegel, "Parity-accumulate Codes for Magnetic Recording," in *IEEE Intermag Digest of Technical Papers*, Toronto, Canada, pp. 517, Apr. 2000.

H. D. Pfister and P. H. Siegel, "Coding Theorems for Generalized Repeat Accumulate Codes," in *Proc. Int. Symp. on Inform. Theory and its Applications*, Honolulu, HI, pp. 21-25, Nov. 2000.

Jilei Hou, P. H. Siegel, L. B. Milstein, and H. D. Pfister, "Design of Low-Density Parity-Check Codes for Bandwidth Efficient Modulation," in *Proc. IEEE Inform. Theory Workshop*, Cairns, Australia, pp. 24-26, Sep. 2001.

H. D. Pfister, J. B. Soriaga and P. H. Siegel, "On the Achievable Information Rates of Finite State ISI Channels," in *Proc. IEEE Globecom*, San Antonio, TX, pp. 2992-2996, Nov. 2001.

Jilei Hou, P. H. Siegel, L. B. Milstein, and H. D. Pfister, "Multilevel Coding with Low-Density Parity-Check Component Codes," in *Proc. IEEE Globecom*, San Antonio, TX, pp. 1016-1020, Nov. 2001.

H. D. Pfister and P. H. Siegel, "The Serial Concatenation of Rate-1 Codes Through Uniform Random Interleavers," accepted *IEEE Trans. on Inform. Theory*, Oct. 2002.

J. B. Soriaga, H. D. Pfister, and P. H. Siegel, "On the Low Rate Shannon Limit for Binary Intersymbol Interference Channels," submitted *IEEE Trans. on Communications*, Oct. 2002.

ABSTRACT OF THE DISSERTATION

On the Capacity of Finite State Channels and the Analysis of Convolutional Accumulate-m Codes

by

Henry D. Pfister
Doctor of Philosophy in Electrical Engineering
(Communications Theory and Systems)

University of California San Diego, 2003

Professor Paul H. Siegel, Chair

What are the fundamental limits of communications channels and channel coding systems? In general, these limits manifest themselves as thresholds which separate what is possible from what is not. For example, the capacity of a communications channel is a coding rate threshold above which reliable communication is not possible. At any coding rate below capacity, however, reliable communication is possible. Likewise, all fixed rate coding schemes have channel noise thresholds above which the probability of decoding error cannot be made arbitrarily small. When the channel noise is below the threshold, many of the same coding systems can operate with very small error probability. In this dissertation, we consider the noise thresholds of Convolutional Accumulate-m (CA-m) codes, the capacity of finite state channels (FSCs), and the information rates achievable via joint iterative decoding of irregular low-density parity-check (LDPC) codes over channels with memory.

CA-m codes are a class of turbo-like codes formed by serially concatenating a terminated convolutional code with a cascade of m interleaved rate-1 "accumulate" codes. The first two chapters consider these codes from two different perspectives. First, the sequence of m encoders is analyzed as a Markov chain to show that these codes converge to random codes, which are nearly optimal, as m goes to infinity. Next, a detailed threshold analysis is performed for both maximum likelihood and iterative decoding of long CA-m codes with finite m.

A FSC is a discrete-time channel whose output depends on both the channel input and

the channel state. A simple Monte Carlo method is introduced which estimates the achievable information rate of any FSC driven by finite memory Markov inputs. Until recently, there has been no practical method of estimating the capacity of a FSC. This Monte Carlo method enables one to compute a non-decreasing sequence of lower bounds on the capacity.

The joint iterative decoding of irregular LDPC codes over channels with memory is also considered. For a class of erasure channels with memory, we derive a closed form recursion that can be used to verify necessary and sufficient conditions for successful decoding.

Chapter 1

Introduction

1.1 Layman's Summary

A great deal of human effort has been exerted throughout history to communicate messages quickly and reliably over long distances. For example, the Roman empire constructed a communications network based on smoke signals which ranged more than 4,500 kilometers. In 1794, the first mechanical optical telegraph used a network of signal flags mounted on towers to transmit messages across Europe. Commercial electrical telegraph service began in 1844 when Morse sent a message from Washington D.C. to Baltimore. Today, multiple digital and analog communications networks encircle the earth and provide telephone, internet, and television services.

In general, all communications systems such as these must make compromises between speed and reliability. For example, telegraph operators are more prone to error when they receive messages which are sent very rapidly. The noise level also increases with the length of the telegraph wire, degrading the reliability of the system. In 1948, Claude Shannon made this concept mathematically precise, and the field of *information theory* was born. One of his main results, the channel coding theorem, says that there is a fundamental limit on the rate that information can be transmitted reliably through a noisy medium [4].

To explain his result properly, we assume that some *sender* would like to transmit a *message* through a noisy medium (i.e., the *channel*) to a *receiver*. For simplicity, we assume the message is a sequence of *bits* (1's and 0's). While the sender may repeatedly transmit one message bit per channel use, uncertainty in the transmission due to noise may cause errors in

the received message. The reliability of the received message can, of course, be improved by sending the same bit many times and allowing the receiver to choose the most probable bit value. For example, if each message bit is transmitted q times, then the *rate* of transmission, which is the number of message bits transmitted per channel use, is said to be $1/q$. Shannon proved the remarkable result that, by using clever encoding and decoding, information can be transmitted reliably at any rate less than a fundamental limit known as the *channel capacity*. Furthermore, he proved that no system can transmit information reliably at rates above the channel capacity.

For example, consider a channel which erases every other bit. This means that the input sequence, 10110010, is mapped to the output sequence, $1?1?0?1?$. Clearly, one message bit can be transmitted reliably for every two channel uses by sending each message bit twice in a row. This establishes that reliable transmission is possible at rate $1/2$. Since the capacity of this channel also equals $1/2$, we know that reliable communication is impossible at any higher rate. The field of *channel coding* focuses on practical methods of encoding and decoding which achieve reliable communication at rates close to capacity. For most channels, the problem of designing effective codes is quite difficult.

For a long time, it was speculated that no relatively simple coding system could approach capacity. These speculations were based on the fact that the first 40 years of research had produced no such system. A major breakthrough occurred in 1993, when Berrou, Glavieux, and Thitimajshima introduced *turbo codes* [1]. Turbo codes shattered these myths by providing performance very close to capacity with only moderate complexity. The trick behind turbo codes was encoding the same data with two simple encoders and then decoding with two simple decoders working together cooperatively. In particular, the turbo decoder works by passing the output of one decoder to the input of the other decoder in a circular fashion. The name, turbo codes, is derived mainly from the similarity between the decoder structure and a turbo charged engine. This type of decoding, now referred to as *iterative decoding*, was actually invented in the 1960s by Gallager, but was ahead of its time and essentially forgotten [3].

The advent of turbo codes has also sparked new interest in the field of information theory. This has led, in turn, to some very exciting research in the past ten years. For one, the renewed interest in coding theory led to the rediscovery of Gallager codes. Variations of these codes have been shown to nearly achieve the channel capacity of channels whose outputs are independent of each other. Much of the work in this dissertation was directly or indirectly motivated by turbo codes. Chapters 2 and 3 deal with a variation of turbo codes known as

Convolutional Accumulate-m codes. These codes have some very interesting properties and are analyzed in detail. Chapter 4 focuses on a simple method of estimating the channel capacity for a class of time-varying channels, known as finite state channels. Before the work presented here, the capacity of these channels could not be estimated easily. Chapter 5 explains the application of a powerful linear code to a simple finite state channel. Using a combined iterative decoding strategy for the channel and the code, we provide a concise analysis of the decoder behavior. This analysis of the decoder allows us to algebraically construct codes whose iterative decoding performance is extremely close the theoretical limit.

The applications of this research include improving the efficiency of communication systems by either increasing data rates or noise tolerance. Either change brings the system closer to the fundamental limit defined by Shannon. For example, a hard disk drive can be thought of as a communications system which transmits data forward in time (i.e., data is written at one time and read at another). In this sense, techniques described in Chapter 4 may eventually be leveraged to increase the storage density of hard disk drives. This research can also be applied to reduce the complexity of encoding/decoding circuits while leaving the coding performance unchanged. In particular, the codes described in Chapters 2 and 3 operate fairly close to the fundamental limit and are very simple to encode and decode.

1.2 Outline of Dissertation

This dissertation consists of an introduction and four self-contained chapters. The chapters cover a fairly wide range of topics including Chapter 2's straightforward analysis of the serial concatenation of rate-1 codes, Chapter 3's thorough analysis of Convolutional Accumulate-m codes, Chapter 4's exposition on the capacity of finite state channels, and Chapter 5's sizable discussion of the joint iterative decoding of codes and channels with memory.

In Chapters 2 and 3, we consider a new class of simple codes with good performance near the Shannon limit. This inquiry was motivated initially by the Repeat Accumulate (RA) codes of Divsalar, Jin, and McEliece [2], which are perhaps the simplest turbo-like codes. These codes are encoded by repeating each message bit q times, randomly reordering all of these repeated message bits, and then computing the modulo-2 cumulative sum of the entire sequence. Although incredibly simple, their performance under iterative turbo-like decoding is very good.

From a coding perspective, an RA code consists of a repeat code followed by a inter-

leaver (which reorders the bits) and a rate-1 code (i.e., the modulo-2 cumulative sum). In Chapter 2, we investigate the performance of more general coding systems based on interleavers and rate-1 codes. In particular, we consider the serial concatenation, through random interleavers, of an arbitrary binary linear outer code and a cascade of m identical rate-1 binary linear inner codes. Our analysis shows that the maximum likelihood decoding performance of these codes, for large enough m, is extremely good. Simulation results are also provided for a practical configuration of these codes known as Convolutional Accumulate-m (CA^m) codes. CA^m codes are a novel generalization of RA codes formed by using a terminated convolutional code as the outer code and a cascade of m interleaved rate-1 "accumulate" codes as the inner code. The practical performance of CA^m codes using iterative decoding is similar, but generally slightly inferior, to that of turbo codes. In terms of complexity, these codes may still have some advantages.

In Chapter 3, we perform a rigorous analysis of asymptotically long CA^m codes for finite m. We prove a coding theorem for these codes which states that, if the outer code has minimum distance $d \geq 2$, and the Bhattacharyya channel parameter, z, is less than some threshold z^*, then the probability of word error is $O(n^\nu)$, where n is the block length and ν is determined solely by m and d. The minimum distance of these codes is also analyzed in some detail. Finally, the performance of asymptotically long CA^m codes under iterative decoding is analyzed via density evolution.

In Chapter 4, we consider the capacity of the set of time-varying channels known as finite state channels (FSCs). A FSC is a discrete time channel whose output distribution depends both on the channel input and the underlying channel state. This allows the channel output to depend implicitly on previous inputs and outputs via the channel state. We provide a simple Monte Carlo method of estimating the achievable information rates of any FSC, and focus on the problem of estimating the capacity of FSCs with intersymbol-interference. This approach is general enough to allow the mutual information rate to be maximized over Markov input distributions of increasing length, and thus estimate a sequence of non-decreasing lower bounds on capacity. Finally, exact information rates are derived analytically for a very simple channel known as the dicode erasure channel.

In Chapter 5, we consider the joint iterative decoding of irregular low-density parity-check (LDPC) codes and channels with memory. We start by introducing a new class of erasure channels with memory, known as generalized erasure channels (GECs). For these channels, we derive a single parameter recursion for density evolution of the joint iterative decoder. This al-

lows us to give necessary and sufficient conditions for decoder convergence and to algebraically construct sequences of LDPC degree distributions which appear to approach the symmetric information rate of the channel. Finally, we discuss the construction of arbitrary GECs and some implications for more general channels.

Bibliography

[1] C. Berrou, A. Glavieux, and P. Thitimajshima. Near Shannon limit error-correcting coding and decoding: Turbo-codes. In *Proc. IEEE Int. Conf. Commun.*, volume 2, pages 1064–1070, Geneva, Switzerland, May 1993. IEEE.

[2] D. Divsalar, H. Jin, and R. J. McEliece. Coding theorems for "turbo–like" codes. In *Proc. 36th Annual Allerton Conf. on Commun., Control, and Comp.*, pages 201–210, Monticello, IL, USA, Sept. 1998.

[3] R. G. Gallager. Low-density parity-check codes. Research Monograph 21, The M.I.T. Press, Cambridge, MA, USA, 1963.

[4] C. E. Shannon. A mathematical theory of communication. *The Bell Syst. Techn. J.*, 27:379–423, 623–656, July / Oct. 1948.

Chapter 2

The Serial Concatenation of Rate-1 Codes Through Uniform Interleavers

2.1 Introduction

Since the introduction of turbo codes by Berrou, Glavieux, and Thitimajshima [3], iterative decoding has made it practical to consider a myriad of different concatenated codes, and the use of "random" interleavers and recursive convolutional encoders has provided a good starting point for the design of new code structures. Many of these concatenated code structures fit into a class that Divsalar, Jin, and McEliece call "turbo-like" codes [4]. Perhaps the simplest codes in this class are Repeat Accumulate (RA) codes, which consist only of a repetition code, an interleaver, and an accumulator. Yet, Divsalar *et al.* prove that the maximum likelihood decoding (MLD) of RA codes has vanishing word error probability, for sufficiently low rates and any fixed signal to noise ratio (SNR) greater than a threshold, as the block length goes to infinity. This demonstrates that powerful error-correcting codes may be constructed from extremely simple components.

In this chapter we consider the serial concatenation of an arbitrary binary linear outer code of rate $r < 1$ with m identical rate-1 binary linear inner codes where, following the convention of the turbo-coding literature, we use the term serial concatenation to mean serial concatenation through a "random" interleaver. Any real system must, of course, choose a particular interleaver. Our analysis, however, will make use of the *uniform random interleaver* (URI) [2]

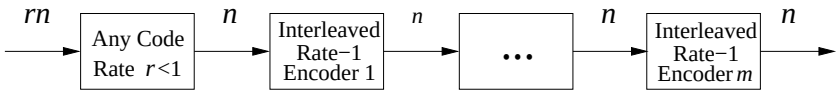


Figure 2.1.1: Our system consists of any rate $r < 1$ code followed by m rate-1 codes.

which is equivalent to averaging over all possible interleavers. We analyze this system using a probabilistic bound on the minimum distance and show that, for any fixed block length and large enough m, the ensemble contains some codes whose minimum distance achieves the Gilbert-Varshamov Bound (GVB) [6].

Our work is largely motivated by [4] and by the results of Öberg and Siegel [12]. Both papers consider the effect of a simple rate-1 "accumulate" code in a serially concatenated system. In [4] a coding theorem is proved for RA codes, while in [12] the "accumulate" code is analyzed as a precoder for the dicode magnetic recording channel. Benedetto *et al.* also investigated the design and performance of Double Serially Concatenated Codes in [1].

We also discuss a specific class of codes in this family, known as Convolutional Accumulate-m (CA^m) codes, which were introduced as Generalized Repeat Accumulate Codes in [15] and [16]. A CA^m code is a serially concatenated code where the outer code is a terminated convolutional code (CC) and the inner code is a cascade of m interleaved "accumulate" codes. These codes were studied in some depth for $m = 1$ by Jin in [10]. This chapter focuses on the case of $m > 1$, and gives a straightforward Markov chain based analysis of the distance properties and MLD performance.

The outline of the chapter is as follows. In Section 2.2, we review the *weight enumerator* (WE) of linear block codes and the union bound on the probability of error for maximum likelihood decoding. We also review the average weight enumerator for the serial concatenation of two linear block codes through a URI, and relate serial concatenation to matrix multiplication using a normalized form of each code's *input output weight enumerator* (IOWE). In Section 2.3, we introduce our system, shown in Fig. 2.1.1, compute its average output WE, and compare this WE to that of random codes. In Section 2.4, we consider some properties of rate-1 codes which affect the performance of our system. In Section 2.5, we discuss a probabilistic bound on the minimum distance of any code, taken from an ensemble, in terms of the ensemble averaged WE. Applying this bound to the WE from Section 2.3 gives an expression that is very similar to the Gilbert-Varshamov Bound (GVB) and that is asymptotically equal to the GVB for large

block lengths. We also evaluate this bound numerically for various CA^m codes and observe that 3 or 4 "accumulate" codes seem to be sufficient to achieve the bound derived for asymptotically large m. In Section 2.6, we evaluate the performance of those same CA^m codes using bounds on the MLD error probability and simulations for iterative decoding error probability. Finally, in Section 2.7, we share some conclusions and discuss the direction of our future work.

2.2 Weight Enumerators and Serial Concatenation

2.2.1 The Union Bound

In this section, we review the weight enumerator of a linear block code and the union bound on error probability for MLD. The IOWE $A_{w,h}$ of an (n, k) block encoder is the number of codewords with input Hamming weight w and output Hamming weight h, and the WE A_h is the number of codewords with any input weight and output weight h. Using these definitions, the MLD probability of word error is upper bounded by

$$P_W \leq \sum_{h=1}^{n} \sum_{w=1}^{k} A_{w,h} z^h,$$

and the MLD probability of bit error is upper bounded by

$$P_B \leq \sum_{h=1}^{n} \sum_{w=1}^{k} \frac{w}{k} A_{w,h} z^h.$$

The parameter z is known as the Bhattacharyya parameter, and z^h represents an upper bound on the pairwise error probability between any two codewords differing in h positions [21, p. 88]. It can be computed for any memoryless channel, and for the binary-input AWGN channel it is $z = e^{-E_s/N_0}$, where E_s/N_0 is the SNR of the decision statistic.

2.2.2 Serial Concatenation through a Uniform Interleaver

We now briefly review the serial concatenation of codes through a URI. The introduction of the URI in the analysis of turbo codes, by Benedetto and Montorsi [2], has made the analysis of complex concatenated coding systems relatively straightforward; using the URI for analysis is equivalent to averaging over all possible interleavers. The important property of the URI is that the output sequence distribution is a function only of the input weight distribution.

More precisely, given that the input to a URI has weight w, each output sequence of weight w will be observed with equal probability and all other output sequences will have zero probability.

Consider any (n, k) block encoder with IOWE $A_{w,h}$ preceded by a URI. We will refer to such a code as a *uniformly interleaved code* (UIC). The probability, $Pr\left(w \rightarrow h\right)$, of the combined system mapping an input sequence of weight w to an output sequence of weight h is

$$Pr\left(w \rightarrow h\right) \stackrel{def}{=} \frac{A_{w,h}}{\binom{k}{w}}. \tag{2.2.1}$$

Now we can consider the ensemble of (n, k) block codes formed by first encoding with an (n_1, k) outer code with IOWE $A^{(o)}_{w,h}$, permuting the output bits with a URI, and finally encoding again with an (n, n_1) inner code with IOWE $A^{(i)}_{w,h}$. The ensemble averaged IOWE $\overline{A}_{w,h}$ is given by

$$\begin{aligned} \overline{A}_{w,h} &= \sum_{h_1=0}^{n_1} A^{(o)}_{w,h_1} Pr\left(h_1 \rightarrow h\right) \\ &= \sum_{h_1=0}^{n_1} A^{(o)}_{w,h_1} \frac{A^{(i)}_{h_1,h}}{\binom{n_1}{h_1}}. \end{aligned} \tag{2.2.2}$$

The average IOWE for the serial concatenation of two codes may also be written as the matrix product of the IOWE for the outer code and a normalized version of the IOWE for the inner code. Let us define, for any code, the *input output weight transition probability* (IOWTP) $P_{w,h}$ as the probability that an input sequence of weight w is mapped to an output sequence of weight h. From (2.2.1), we can see that $P_{w,h} = Pr\left(w \rightarrow h\right)$. Substituting (2.2.1) into (2.2.2), we have

$$\overline{A}_{w,h} = \sum_{h_1=0}^{n_1} A^{(o)}_{w,h_1} P^{(i)}_{h_1,h} = \mathbf{A}^{(o)}\mathbf{P}^{(i)}.$$

where $\mathbf{A}^{(o)}\mathbf{P}^{(i)}$ is a matrix product and the matrix representations are defined by $\left[\mathbf{A}^{(o)}\right]_{w,h} = A^{(o)}_{w,h}$ and $\left[\mathbf{P}^{(i)}\right]_{w,h} = P^{(i)}_{w,h}$. Using induction, it is easy to verify that matrix multiplication by an arbitrary number of IOWTP matrices results in the average IOWE, $\overline{A}_{w,h}$, of the overall serial concatenation. It is also easy to verify, using (2.2.1), that all IOWTP matrices are stochastic.

2.2.3 A Simple Example - The Accumulate Inner Code

We compute the IOWE and IOWTP of the rate-1 "accumulate" code [4]. The "accumulate" code is a block code formed by truncating, after n symbols, the recursive rate-1 CC with

Input Sequence	000	001	010	100	011	101	110	111
Input Weight	0	1	1	1	2	2	2	3
Output Sequence	000	001	011	111	010	110	100	101
Output Weight	0	1	2	3	1	2	1	2

Table 2.1: Input-output sequences and weight mappings for $n = 3$ "accumulate" code.

generator matrix $G(D) = 1/(1+D)$. The generator matrix for this block code is an $n \times n$ matrix with all ones in the upper triangle and all zeros elsewhere. For the case $n = 3$, the generator matrix is

$$\mathbf{C} = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}.$$

Using Table 2.1, we see that the uniformly interleaved "accumulate" code maps an input of weight 1 to an output of weight 1, 2, or 3, each with probability 1/3. So the $w = 1$ row of the IOWTP matrix is $\begin{bmatrix} 0 & 1/3 & 1/3 & 1/3 \end{bmatrix}$. The matrix representations of the IOWE and IOWTP are given by

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 2 & 1 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad \mathbf{P} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1/3 & 1/3 & 1/3 \\ 0 & 2/3 & 1/3 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

2.3 Multiple Rate-1 Serial Concatenations

2.3.1 The Input Output Weight Enumerator

Now, we consider the average IOWE, $\overline{A}_{w,h}$, of the (n, k) linear block encoder formed by first encoding with any (n, k) linear block encoder and then encoding with a cascade of m identical interleaved rate-1 block encoders. Let the outer encoder be defined by the $k \times n$ generator matrix $\mathbf{C}^{(o)}$ and the inner code be defined by the $n \times n$ generator matrix $\mathbf{C}^{(i)}$. The serial concatenation of linear block codes is achieved by multiplying their generator matrices, so

the generator matrix of any code in this ensemble can be written as

$$\mathbf{C} = \mathbf{C}^{(o)}\mathbf{\Pi}_1\mathbf{C}^{(i)}\mathbf{\Pi}_2\mathbf{C}^{(i)}\cdots\mathbf{\Pi}_m\mathbf{C}^{(i)}, \tag{2.3.1}$$

where each $\mathbf{\Pi}_i$ is an $n \times n$ permutation matrix. Our ensemble of encoders, denoted by $\Omega_m(\mathbf{C}^{(o)}, \mathbf{C}^{(i)})$, can be defined succinctly by a probability distribution over all $k \times n$ generator matrices. In theory, this distribution can be computed by counting the number of distinct ways each generator matrix can be written in the form of (2.3.1), but the large number of generator matrices makes this infeasible. Instead, we focus on computing the average IOWE of this ensemble. Let $A^{(o)}_{w,h}$ be the IOWE associated with the generator matrix $\mathbf{C}^{(o)}$ and let $A^{(i)}_{w,h}$ be the IOWE associated with the generator matrix $\mathbf{C}^{(i)}$. Let $\mathbf{P}$ be the IOWTP matrix associated with $A^{(i)}_{w,h}$, then the average IOWE $\overline{A}^{(m)}_{w,h}$ of this ensemble is

$$\overline{A}^{(m)}_{w,h} = \sum_{h_1=0}^{n} A^{(o)}_{w,h_1} \left[\mathbf{P}^m\right]_{h_1 h}. \tag{2.3.2}$$

The linearity of the code guarantees that inputs of weight zero will always be mapped to outputs of weight zero and inputs of weight greater than zero will always be mapped to outputs of weight greater than zero, so the matrix $\mathbf{P}$ will be block diagonal with two blocks. Let the first block be the 1×1 submatrix associated with $w = h = 0$ and the second block be the $n \times n$ submatrix formed by deleting the first row and column of $\mathbf{P}$. In general, we will refer to the second block as the $\mathbf{Q}$ submatrix of the IOWTP matrix, and we write

$$\mathbf{P} = \begin{bmatrix} 1 & 0 \\ 0 & \mathbf{Q} \end{bmatrix}.$$

Multiplication acts independently on the components of a block diagonal matrix, so we can also write

$$\mathbf{P}^m = \begin{bmatrix} 1 & 0 \\ 0 & \mathbf{Q}^m \end{bmatrix}.$$

If $\mathbf{P}$ is a finite dimensional stochastic matrix, then we can associate it with a finite-state Markov Chain (MC) with state transition matrix $\mathbf{P}$. In this case, both $\mathbf{P}$ and $\mathbf{Q}$ are finite dimensional stochastic matrices and the association matches states in the MC with input/output weights of the rate-1 UIC. Using some well-known definitions from the theory of MCs, we say

that $\boldsymbol{\pi} = [\pi_0, \ldots, \pi_n]$ is a stationary state distribution of the MC with transition probability matrix $\mathbf{P}$ if $\boldsymbol{\pi}\mathbf{P} = \boldsymbol{\pi}$ and $\sum \pi_i = 1$. This allows us to associate a stationary state distribution, $\boldsymbol{\pi}$, of the MC with a stationary weight distribution of the rate-1 UIC. If the average WE, $\overline{A}_h$, of a code ensemble is not changed by encoding every code in the ensemble with the same rate-1 UIC, then $\overline{A}_h$ is a stationary WE of that rate-1 UIC. Using (2.2.2), it is easy to verify that this occurs when

$$\left[\overline{A}_1, \ldots, \overline{A}_n\right] \mathbf{Q} = \left[\overline{A}_1, \ldots, \overline{A}_n\right],$$

which makes $\left[\overline{A}_1, \ldots, \overline{A}_n\right] / (2^k - 1)$ a stationary state distribution of the MC associated with state transition matrix $\mathbf{Q}$. Recall also that a MC, with state transition matrix $\mathbf{Q}$, is *irreducible* if and only if, for all i, j, there exists a positive $t_{i,j}$ such that $\left[\mathbf{Q}^{t_{i,j}}\right]_{i,j} > 0$ [19, p. 18].

Definition 2.3.1. A rate-1 UIC is *irreducible* if the $\mathbf{Q}$ submatrix of its IOWTP matrix, $\mathbf{P}$, can be associated with an irreducible MC.

We now draw upon some well-known theorems from the theory of non-negative matrices and MCs [19, p. 119].

Theorem 2.3.2 (Perron-Frobenius). *An irreducible Markov Chain has a unique positive stationary state distribution.* □

Proposition 2.3.3. *Let $\mathbf{P}$ be the IOWTP matrix of an irreducible rate-1 UIC with block length n. The infinite family of stationary state distributions, $\boldsymbol{\pi}(\alpha) = [\pi_0(\alpha), \ldots, \pi_n(\alpha)]$, of $\mathbf{P}$ is defined by*

$$\pi_h(\alpha) = \begin{cases} \alpha & h = 0 \\ (1-\alpha)\frac{\binom{n}{h}}{2^n - 1} & 1 \le h \le n \end{cases}.$$

Finally, the unique stationary distribution for inputs of non-zero weight is given by $\boldsymbol{\pi}(0)$.

Proof. The $(n+1) \times (n+1)$ matrix, $\mathbf{P}$, is block diagonal with the first block equal to the scalar, 1, and the second block equal to the $n \times n$ matrix $\mathbf{Q}$. It is easy to verify that $\mathbf{P}$ has exactly two irreducible components because a scalar is irreducible and $\mathbf{Q}$ is irreducible by Definition 2.3.1. The stationary distribution of the scalar component is the unit vector associated with inputs of weight zero because a linear code always maps the all zero input to the all zero output.

Now, we consider the stationary distribution of the $\mathbf{Q}$ irreducible component. The matrix $\mathbf{Q}$ represents the action of a rate-1 linear code on the set of all non-zero sequences, which is simply a permutation of these sequences. Therefore, a uniform distribution on the set of non-zero sequences will be stationary under this mapping. Now, we can simply calculate the weight distribution associated with a uniform distribution on the set of non-zero sequences. Simple combinatorics gives the answer

$$\pi_h = \frac{\binom{n}{h}}{2^n - 1}$$

for $1 \leq h \leq n$.

Any stationary distribution of $\mathbf{P}$ can be written as the convex combination of these two unique stationary distributions (one for each irreducible component). Restricting our attention to inputs of non-zero weight has the effect of making the stationary distribution unique and equal to the stationary distribution of the $\mathbf{Q}$ component. □

Example 2.3.4. The rate-1 code from Section 2.2.3 is irreducible, and applying Proposition 2.3.3 gives

$$\begin{bmatrix} 0 \\ 3/7 \\ 3/7 \\ 1/7 \end{bmatrix}^T \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1/3 & 1/3 & 1/3 \\ 0 & 2/3 & 1/3 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 3/7 \\ 3/7 \\ 1/7 \end{bmatrix}^T .$$

An MC with state transition matrix $\mathbf{Q}$ is *primitive* if and only if there exists a positive t such that $\left[\mathbf{Q}^t\right]_{i,j} > 0$ for all i, j. This is equivalent to the state transition matrix, $\mathbf{Q}$, having a unique eigenvalue of maximum modulus. The following theorem from the theory of MCs characterizes the asymptotic behavior of a primitive matrix taken to a large power [19, p.119].

Theorem 2.3.5 (Perron-Frobenius). *If $\mathbf{Q}$ is the state transition matrix of a primitive Markov Chain, with unique stationary distribution $\boldsymbol{\pi}$, then*

$$\lim_{m \to \infty} \mathbf{Q}^m = \begin{bmatrix} \boldsymbol{\pi} \\ \vdots \\ \boldsymbol{\pi} \end{bmatrix} .$$

Moreover, the convergence is uniform and geometric. Specifically, if we let λ_2 be the eigenvalue with second largest magnitude, then $\left|[\mathbf{Q}^m]_{ij} - \pi_j\right| = O(q^m)$, for any q satisfying $|\lambda_2| < q < 1$. □

Definition 2.3.6. An irreducible rate-1 UIC is *primitive* if the MC associated with the $\mathbf{Q}$ submatrix of its IOWTP matrix is primitive.

Corollary 2.3.7. *If $\mathbf{P}$ is the IOWTP matrix of a primitive rate-1 UIC with block length n, then*

$$\lim_{m\to\infty}[\mathbf{P}^m]_{ij} = \begin{cases} 1 & \text{if } i = j = 0 \\ \binom{n}{j}/(2^n - 1) & \text{if } i > 0 \text{ and } j > 0 \\ 0 & \text{otherwise} \end{cases} . \tag{2.3.3}$$

Example 2.3.8. The rate-1 code from Section 2.2.3 is also primitive, and applying Theorem 2.3.5 confirms that

$$\lim_{m\to\infty}\mathbf{P}^m = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 3/7 & 3/7 & 1/7 \\ 0 & 3/7 & 3/7 & 1/7 \\ 0 & 3/7 & 3/7 & 1/7 \end{bmatrix} .$$

2.3.2 A Large Number of Concatenations

We now use (2.3.2) and Theorem 2.3.5 to compute the average WE of any rate $r < 1$ outer code serially concatenated with m primitive rate-1 UICs, in the limit as m goes to infinity. The intriguing part of this result is that this average WE is independent of the particular outer encoder and inner encoder chosen. Using the notation from Section 2.3.1, we let $\mathbf{C}^{(o)}$ be the $k \times n$ generator matrix of the invertible outer code and $\mathbf{C}^{(i)}$ be the $n \times n$ generator matrix of the primitive rate-1 inner code, and we let $\Omega_m(\mathbf{C}^{(o)}, \mathbf{C}^{(i)})$ denote the ensemble of codes with m serial concatenations. Since this sequence of ensembles may not approach a well-defined limit as m goes to infinity, we avoid discussing properties of the infinite-m ensemble. Instead, we say that a property holds for $\Omega_*(\mathbf{C}^{(o)}, \mathbf{C}^{(i)})$ if there exists a finite m_0 such that the property holds for all $\Omega_m(\mathbf{C}^{(o)}, \mathbf{C}^{(i)})$, for $m \geq m_0$.

Remark 2.3.9. An interesting open question is whether the ensemble $\Omega_m(\mathbf{C}^{(o)}, \mathbf{C}^{(i)})$ contains all invertible linear codes, for sufficiently large m. Using the generator matrix definition, (2.3.1), it is possible to give a sufficient condition for this. Let S be the set of all $n \times n$ permutation

matrices and define $T = \left\{\mathbf{\Pi}\mathbf{C}^{(i)} | \mathbf{\Pi} \in S\right\}$. Since $\mathbf{C}^{(i)}$ is invertible by assumption and all permutation matrices are invertible, it is clear that T is a subset of the multiplicative group of $n \times n$ invertible binary matrices denoted $GL_n(\mathbb{F}_2)$. Let $T^m = \{\mathbf{V}_1\mathbf{V}_2 \dots \mathbf{V}_m | \mathbf{V}_i \in T\}$ and assume that there exists an m_0 such that $T^{m_0} = GL_n(\mathbb{F}_2)$. In this case, $\Omega_m(\mathbf{C}^{(o)}, \mathbf{C}^{(i)})$ will contain all invertible linear codes for all $m \geq m_0$. Furthermore, the limit, $\lim_{m\to\infty} \Omega_m(\mathbf{C}^{(o)}, \mathbf{C}^{(i)})$, exists and is equal to the ensemble of all invertible linear codes under the uniform distribution. For example, when $\mathbf{C}^{(i)}$ is the "accumulate" code, we have verified that this occurs for $n = 2, 3, 4$ with $m_0 = n + 1$.

Theorem 2.3.10. *Let $\overline{A}_h^{(m)}(n,k)$ be the average output WE of the ensemble, $\Omega_m(\mathbf{C}^{(o)}, \mathbf{C}^{(i)})$, where $\mathbf{C}^{(o)}$ is the $k \times n$ generator matrix of the outer code and $\mathbf{C}^{(i)}$ is the $n \times n$ generator matrix of the primitive rate-1 inner code. If we define $\overline{A}_h^{(\infty)}(n,k)$ to be $\lim_{m\to\infty} \overline{A}_h^{(m)}(n,k)$, then we have*

$$\overline{A}_h^{(\infty)}(n,k) = \begin{cases} \left(2^k - 1\right) \frac{\binom{n}{h}}{2^n - 1} & \text{if } h \geq 1 \\ 1 & \text{if } h = 0 \end{cases} . \tag{2.3.4}$$

Furthermore, for any $\gamma > 0$, there exists an m_0 such that $\left|\overline{A}_h^{(\infty)}(n,k) - \overline{A}_h^{(m)}(n,k)\right| < \gamma$ for all $m \geq m_0$.

Proof. Starting with (2.3.2) gives

$$\overline{A}_h^{(\infty)}(n,k) = \lim_{m\to\infty} \sum_{w=1}^{k} \sum_{h_1=1}^{n} A_{w,h_1}^{(o)} \left[\mathbf{P}^m\right]_{h_1 h} .$$

Applying (2.3.3) gives

$$\overline{A}_h^{(\infty)}(n,k) = \left(\sum_{w=1}^{k} \sum_{h_1=1}^{n} A_{w,h_1}^{(o)} \right) \frac{\binom{n}{h}}{2^n - 1},$$

and the double sum is independent of the outer code and equal to the number of codewords (excluding the all zeros codeword), so

$$\overline{A}_h^{(\infty)}(n,k) = \left(2^k - 1\right) \frac{\binom{n}{h}}{2^n - 1}.$$

For the second statement, we start with

$$\left|\overline{A}_h^{(\infty)} - \overline{A}_h^{(m)}\right| = \left|\sum_{w=1}^{rn}\sum_{h_1=1}^{n} A_{w,h_1}^{(o)}\left(\pi_h - [\mathbf{P}^m]_{h_1 h}\right)\right|$$

and then we separate the terms and apply Theorem 2.3.5 to get

$$\begin{aligned}\left|\overline{A}_h^{(\infty)} - \overline{A}_h^{(m)}\right| &\leq \left(\sum_{w=1}^{rn}\sum_{h_1=1}^{n} A_{w,h_1}^{(o)}\right)\left|\pi_h - [\mathbf{P}^m]_{h_1 h}\right| \\ &= (2^{rn}-1)O(q^m).\end{aligned}$$

Although the $(2^k - 1)$ term is possibly quite large, it is a constant with respect to m, so this expression is still $O(q^m)$. Since $q < 1$, it follows that, for any $\gamma > 0$, there exists an m_0 such that, for all $m \geq m_0$, the inequality $\left|\overline{A}_h^{(\infty)}(n,k) - \overline{A}_h^{(m)}(n,k)\right| < \gamma$ holds. □

Let us define the uniform ensemble of linear codes as the ensemble generated by the set of all $k \times n$ generator binary matrices. This is equivalent to the ensemble formed by letting each entry of a random generator matrix be chosen independently and equiprobably from the set $\{0, 1\}$. For non-zero input weights, the average WE is computed by simply noting there are $2^k - 1$ input sequences, each of which will be mapped to a weight-h codeword with probability $\binom{n}{h}/2^n$. Of course, the all zero input is always mapped to the all zero output. Therefore, the average WE of the uniform ensemble is given by

$$\overline{A}_h^U(n,k) = \begin{cases} (2^k - 1)\frac{\binom{n}{h}}{2^n} & \text{for } 1 \geq h \geq n \\ 1 + \frac{2^k - 1}{2^n} & \text{for } h = 0 \end{cases}. \tag{2.3.5}$$

Since the average number of weight zero codewords is larger than one, there will always be some codes in this ensemble which are not invertible.

It turns out that the WE, $\overline{A}_h^{(\infty)}(n,k)$, is almost identical to the average WE of the uniform ensemble of random linear codes. The main difference between these two ensembles is that all of the codes in $\Omega_m(\mathbf{C}^{(o)}, \mathbf{C}^{(i)})$ are invertible, while the uniform ensemble contains a small percentage of non-invertible codes. The following Corollary of Theorem 2.3.10 explicitly compares the average WE of the ensemble, $\Omega_m(\mathbf{C}^{(o)}, \mathbf{C}^{(i)})$, with the average WE of the uniform ensemble of random codes.

Corollary 2.3.11. *Let $\overline{A}_h^{(m)}(n,k)$ be the average WE of the ensemble $\Omega_m(\mathbf{C}^{(o)},\mathbf{C}^{(i)})$, as defined in Theorem 2.3.10. For any $0<r<1$ and $\epsilon>0$, there exist integers n_0 and m_0 such that*

$$\left|\overline{A}_h^{U}(n_0,\lceil rn_0\rceil)-\overline{A}_h^{(m)}(n_0,\lceil rn_0\rceil)\right|\leq\epsilon$$

for all $m\geq m_0$.

Proof. Using the fact that $r<1$, it is easy to verify that, for any $\epsilon>0$, there exists an n_0 such that

$$\left|\overline{A}_h^{U}(n,\lceil rn\rceil)-\overline{A}_h^{(\infty)}(n,\lceil rn\rceil)\right|\leq\frac{\epsilon}{2},$$

for all $n\geq n_0$. Using Theorem 2.3.10, it is also easy to verify that, for any $\epsilon>0$, there exists an m_0 such that

$$\left|\overline{A}_h^{(\infty)}(n_0,\lceil rn_0\rceil)-\overline{A}_h^{(m)}(n_0,\lceil rn_0\rceil)\right|\leq\frac{\epsilon}{2},$$

for all $m\geq m_0$. Combining these two bounds completes the proof. □

2.4 Properties of Rate-1 Codes

2.4.1 Conditions for Primitivity

In this section, we consider the conditions under which a rate-1 linear code is primitive. Theorem 2.4.1 gives a sufficient condition by showing that the rate-1 block code formed by truncating any rate-1/1 CC is primitive. Surprisingly, this also includes non-recursive CCs, which are seldom considered in practical turbo coding systems.

Theorem 2.4.1. *Let $\mathbf{h}=h_0,h_1,h_2,\ldots$ be the semi-infinite impulse response of a nontrivial, causal, rate-1/1 convolutional code. To avoid degenerate cases, assume that $h_0=1$. Define l to be the smallest positive integer such that $h_l=1$. Then, the rate-1 block code formed by truncating this convolutional code, to any length $n\geq l+1$, is primitive.*

Proof. This proof is given in the Appendix. □

Proposition 2.4.2 establishes a simple necessary condition for primitivity. In fact, we conjecture that this condition is also sufficient.

Proposition 2.4.2. *A primitive rate-1 linear code must have at least one row of even weight in its generator matrix.*

Proof. Assume that all rows of the generator matrix have odd weight. It is easy to see that any linear combination of an even (odd) number of rows will have even (odd) weight. So even (odd) weight inputs will map only to even (odd) weight outputs and there will be no weight paths from odd weights to even weights and vice-versa. Therefore, the MC associated with this code is reducible into at least two components and the rate-1 code is not primitive. □

Now, we discuss two exceptional classes of rate-1 codes which are not primitive. Remember that a rate-1 code cannot be primitive if its associated MC is reducible. First, consider any rate-1 code whose generator matrix is an $n \times n$ permutation matrix. All of these codes map inputs of weight h to outputs of weight h and therefore their associated MCs are reducible into $n + 1$ components. Next, for even n, consider any rate-1 code whose generator matrix is the complement of an $n \times n$ permutation matrix. For inputs of even weight, this maps inputs of weight h to outputs of weight h. For inputs of odd weight, this maps inputs of weight h to outputs of weight $n - h$. Therefore, the MC associated with any of these codes is reducible into roughly $3n/4$ components.

In fact, we have been unable to construct a rate-1 code that is not primitive and that still has at least one row of even weight. This leads us to conjecture that the necessary condition implied by Proposition 2.4.2 is also sufficient.

Remark 2.4.3. Suppose the MC associated with a rate-1 code breaks into exactly two components based on parity (cf. the proof of Proposition 2.4.2). In this case, a variant of Theorem 2.3.10 will still apply. This is because the code will preserve the odd or even parity of its inputs. Since the outer code is linear, either none of the codewords will have odd weight or exactly half of the codewords will have odd weight. If exactly half have odd weight, then the average WE will be identical to (2.3.4). If none have odd weight, then the even weight terms of the overall code will be roughly doubled while the odd weight terms will be exactly zero. For this reason, this type of reducibility based on parity is essentially irrelevant in terms of minimum distance and performance.

2.4.2 Recursive vs. Non-Recursive Rate-1/1 CCs

If we consider the average WE of the ensemble $\Omega_m(\mathbf{C}^{(o)}, \mathbf{C}^{(i)})$, for finite m, then there is a distinct difference between using a generator matrix, $\mathbf{C}^{(i)}$, derived from a recursive rate-1/1 CC and one derived from a non-recursive rate-1/1 CC. This difference manifests itself in the convergence rate of the matrix product $\mathbf{P}^m$ to its limiting value for large m. This is very much related to the convergence rate, in m, of the average WE of the ensemble $\Omega_m(\mathbf{C}^{(o)}, \mathbf{C}^{(i)})$ to the value predicted by Theorem 2.3.10. Since the WE predicted by Theorem 2.3.10 has almost no codewords of small output weight, we compare these two ensembles by considering the number of cascaded rate-1 UICs required to map an input of small weight to an output whose weight grows linearly with the block length.

Consider the non-recursive CC with generator $G(D) = 1 + D$. It is easy to verify that the output weight of this code will be at most twice the input weight. If the desired output weight is ρn and the input weight is 1, then the minimum number of encodings required is $\log_2 \rho n$. More generally, for any non-recursive CC with an impulse response of weight d, the minimum number of encodings is $\log_d \rho n$. Therefore, for fixed m and asymptotically large n, there will be no mappings from input weight 1 to output weight ρn. So, for any finite m, we expect this ensemble to have low weight codewords.

Now consider the recursive CC with generator $G(D) = 1/(1+D)$. It is easy to verify that this encoder maps an input of weight 1 at position $i = 1, \ldots, n$ to an output weight of $n - i + 1$. Moreover, most inputs of small weight are mapped to outputs of large weight. If we view this code simply as the inverse of the previous code, then it is clear that if one code maps $A_{w,h}$ input sequences from weight w to weight h then the other code maps the same number of sequences from weight h to weight w. So, for fixed m and asymptotically large n, the interleaved cascade of m recursive rate-1/1 CCs has no paths from weight ρn to weight 1. In practice, recursive CCs are preferred because this is a much more desirable property for error correcting codes. In fact, the results of Section 2.5.2 imply that many codes with relatively small m still have large minimum distance.

Remark 2.4.4. Another way to see the difference between recursive and non-recursive rate-1 CCs is in the second largest eigenvalue, λ_2, of the $\mathbf{Q}$ submatrix of the IOWTP matrix. Numerical observations suggest that the magnitude of this eigenvalue for the $G(D) = 1/(1 + D)$ code is $|\lambda_2| = O(n^{-1})$ while for the $G(D) = 1 + D$ code, it is $|\lambda_2| = O(1)$. It is well known that the

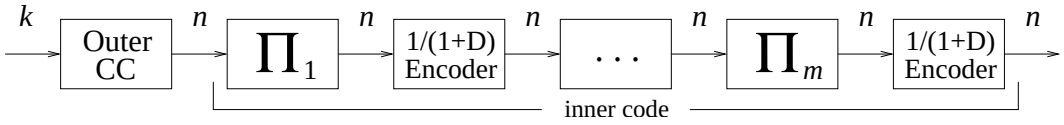


Figure 2.5.1: Encoder for a CA^m Code with the block size indicated at each stage.

convergence of the matrix product $\mathbf{P}^m$ to its limiting value is very sensitive to the magnitude of λ_2 (cf. Theorem 2.3.5). Moreover, we believe this behavior may be characteristic of all recursive and non-recursive codes, and if this is true, then it is another factor which favors recursive CCs over non-recursive CCs.

2.5 Bounds on the Minimum Distance

2.5.1 The Minimum Distance Distribution

In this section, we examine minimum distance properties of the ensemble $\Omega_m(\mathbf{C}^{(o)}, \mathbf{C}^{(i)})$. We make use of a general upper bound on the probability that any code in some ensemble has minimum distance, d_{min}, less than d. The key property of this bound is that it can be computed using only the average WE of the ensemble. The bound, a simple corollary of the Markov inequality [13, p. 114], has been used previously by Gallager [7]and by Kahale and Urbanke [11]. For convenience and completeness, we now explicitly state and prove this bound.

Lemma 2.5.1. *The probability that a code, randomly chosen from an ensemble of linear codes with average WE $\overline{A}_h$, has $d_{min} < d$ is bounded by*

$$Pr(d_{min} < d) \leq (\overline{A}_0 - 1) + \sum_{h=1}^{d-1} \overline{A}_h. \tag{2.5.1}$$

Proof. Let A_h be a random variable equal to the number of codewords with weight h in a code randomly chosen from an ensemble of codes with average WE $\overline{A}_h$. We can bound the probability that a code in the ensemble has minimum distance less than d with

$$Pr(d_{min} < d) = Pr\left((A_0 > 1) \cup \bigcup_{i=1}^{d-1} (A_i > 0) \right).$$

Since A_h takes only positive integer values, we can apply the union bound and then the Markov inequality to get

$$\begin{aligned} Pr(d_{min} < d) &\leq Pr(A_0 - 1 \geq 1) + \sum_{h=1}^{d-1} Pr(A_h \geq 1) \\ &\leq (\overline{A}_0 - 1) + \sum_{h=1}^{d-1} \overline{A}_h. \end{aligned}$$

□

Now, we use Lemma 2.5.1 to compare the minimum distance distribution of the uniform ensemble with that of $\Omega_m(\mathbf{C}^{(o)}, \mathbf{C}^{(i)})$. Both of these are also compared to the well-known Gilbert-Varshamov Bound (GVB).

Using the counting argument of Gilbert [9], it is easy show that there exists at least one code with n code bits, k information bits, and minimum distance d if

$$(2^k - 1) \sum_{h=0}^{d-1} \binom{n}{h} < 2^n. \tag{2.5.2}$$

Varshamov derives a slightly better bound by considering only linear codes, and the similarity between the two permits one to refer to them jointly as the GVB [6]. Let $d_{GVB}(n,k)$ be the largest d which satisfies (2.5.2) for a particular n and k. This is the largest minimum distance which is guaranteed to be achievable by the GVB.

Consider the bound which results from applying Lemma 2.5.1 to the average WE of the uniform ensemble of linear codes, given in (2.3.5). For this ensemble, we find

$$Pr(d_{min} < d) \leq S(n,k,d),$$

where

$$\begin{aligned} S(n,k,d) &= \left(\overline{A}_0^U(n,k) - 1\right) + \sum_{h=1}^{d-1} \overline{A}_h^U(n,k) \\ &= \frac{2^k - 1}{2^n} \sum_{h=0}^{d-1} \binom{n}{h}. \end{aligned}$$

Let $d_U(n,k,\epsilon)$ be the largest d such that $S(n,k,d) < \epsilon$. Notice that the inequality, (2.5.2), is actually equivalent to the inequality, $S(n,k,d) < 1$. Therefore, this bound contains the GVB as a special case and $d_U(n,k,1) = d_{GVB}(n,k)$.

Now, we apply Lemma 2.5.1 to the average WE $\overline{A}_h^{(\infty)}(n,k)$, given in (2.3.4). In this case, we get

$$Pr(d_{min} < d) \leq T(n,k,d),$$

where

$$\begin{aligned} T(n,k,d) &= \left(\overline{A}_0^{(\infty)}(n,k) - 1\right) + \sum_{h=1}^{d-1} \overline{A}_h^{(\infty)}(n,k) \\ &= \frac{2^k - 1}{2^n - 1} \sum_{h=1}^{d-1} \binom{n}{h}. \end{aligned}$$

Proposition 2.5.2. *The inequality $T(n,k,d) < S(n,k,d)$ holds for all $n \geq 2$, $0 < k < n$, and $0 \leq d \leq n$.*

Proof. Notice that the difference, $T(n,k,d) - S(n,k,d)$, is given by the expression,

$$\frac{2^k - 1}{2^n - 1} \sum_{h=1}^{d-1} \binom{n}{h} - \frac{2^k - 1}{2^n} \sum_{h=0}^{d-1} \binom{n}{h},$$

which can be simplified to

$$\frac{2^k - 1}{2^n(2^n - 1)} \left(\sum_{h=1}^{d-1} \binom{n}{h} \right) - \frac{2^k - 1}{2^n}.$$

Notice that this expression is negative for $d = 0$, strictly increasing with d, and equal to zero for $d = n+1$. Therefore, this expression is negative for $0 \leq d \leq n$ and $T(n,k,d) < S(n,k,d)$. □

Let $d_\Omega(n,k,\epsilon)$ be the largest d such that $T(n,k,d) < \epsilon$ and notice that Proposition 2.5.2 implies that $d_\Omega(n,k,\epsilon) \geq d_U(n,k,\epsilon)$. Recall that the WE of the ensemble $\Omega_m(\mathbf{C}^{(o)}, \mathbf{C}^{(i)})$ can be made arbitrarily close to $\overline{A}_h^{(\infty)}(n,k)$ by increasing m. Since $T(n,k,d) < S(n,k,d)$, this shows that there exists an m_0 such that, for all $m \geq m_0$, the minimum distance guaranteed by Lemma 2.5.1 for $\Omega_m(\mathbf{C}^{(o)}, \mathbf{C}^{(i)})$ is greater than or equal to $d_U(n,k,\epsilon)$. Qualitatively, it is interesting to note that this proves (independently of the GVB) that there exists at least one code satisfying $d_{min} \geq d_{GVB}(n,k)$.

The asymptotic form of the GVB says that, in the limit as n goes to infinity, there exists a code with rate $r = k/n$ and normalized minimum distance $\delta = d_{min}/n$ if

$$H(\delta) \leq 1 - r \tag{2.5.3}$$

where $H(x) = -x\log_2 x - (1-x)\log_2(1-x)$ is the binary entropy function [6]. Let $\delta^*_{GVB}(r)$ be the largest $\delta \leq 1/2$ which satisfies (2.5.3) for a particular block length and rate. This is the largest normalized minimum distance which is guaranteed to be achievable by the GVB.

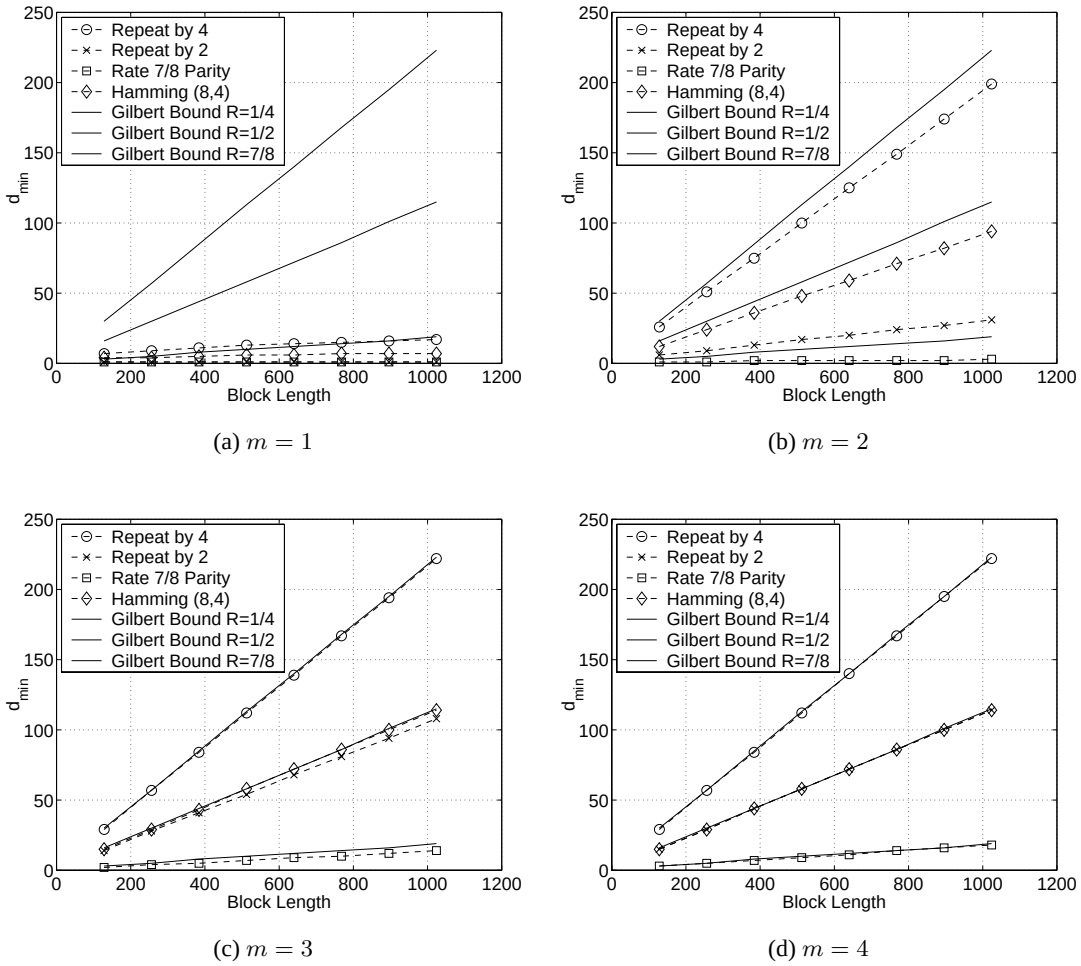
Now, we can define similar normalized distance bounds for the uniform ensemble and for the ensemble, $\Omega_*(\mathbf{C}^{(o)}, \mathbf{C}^{(i)})$. Let $\delta_U(r,\epsilon)$ be the largest δ such that $\lim_{n\to\infty} S(n, \lceil rn \rceil, \delta n) < \epsilon$ and let $\delta_\Omega(r,\epsilon)$ be the largest δ such that $\lim_{n\to\infty} T(n, \lceil rn \rceil, \delta n) < \epsilon$. Following the approach taken by Pierce in [17], it is easy to verify that $\delta_\Omega(r,\epsilon) = \delta_U(r,\epsilon) = \delta_{GVB}(r)$ for any $\epsilon < 1$.

2.5.2 Convolutional Accumulate-m (CA^m) Codes

Now, we apply Lemma 2.5.1 to get some numerical results for the minimum distance of specific CA^m codes. Recall that CA^m codes are the serial concatenation of a terminated CC and m interleaved rate-1 "accumulate" codes. The encoder for CA^m codes is shown in Fig. 2.5.1. We note that the MLD performance of RA codes and some other CA^m codes with $m = 1$ was reported in [10]. Generalizations to $m > 1$ were introduced in [15] and a coding theorem for these codes was given in [16]. Now, we give results pertaining to the minimum distance of CA^m codes using a few examples. For simplicity, our examples use CCs with memory 0, which may also be viewed as repeated block codes [15].

In order to apply Lemma 2.5.1 to a specific ensemble, we must compute the ensemble averaged WE and choose an ϵ. Let C_i be a sequence of code ensembles with k_i information bits and n_i code bits such that the rate, $r = k_i/n_i$, is fixed. We define $d^*_C(n_i,\epsilon)$ as the largest minimum distance guaranteed, with probability $1-\epsilon$, by applying Corollary 2.5.1 to the ensemble averaged WE of C_i. In the following results, we look at the sequence $d^*_C(n_i, 1/2)$ using numerically averaged WEs for various code ensembles. This means that at least half of the codes in each ensemble have a minimum distance of at least $d^*_C(n_i,1/2)$. We consider 16 ensembles formed by choosing one of four outer codes and the number of "accumulate" codes $m = 1, \ldots, 4$. Each outer code is referred to in shorthand: repeat by 2 (R2), repeat by 4 (R4), rate 8/9 single parity check (P9), and the $(8,4)$ extended Hamming code (H8). The results, over a range of codeword lengths, are shown in Fig. 2.5.2.

We compare these code ensembles to the uniform ensemble by focusing on the rate at which the minimum distance grows with the block length. It is important to note that, at a



(a) $m = 1$ (b) $m = 2$

(c) $m = 3$ (d) $m = 4$

Figure 2.5.2: Probabilistic bound on the minimum distance of various CA^m codes.

fixed rate, a "good" code is defined by a minimum distance which grows linearly with the block length. When examining these results, we will focus on whether or not the minimum distance appears to be growing linearly with block length and on how close the $d^*_C(n_i, 1/2)$ is to the GVB. For ensembles of CA^m codes with $m = 1$, it is known that the minimum distance of almost all of the codes grows like $O(n^{(d^o-2)/d^o})$, where d^o is the free distance of the outer terminated CC [11]. Examining Fig. 2.5.2 for $m = 1$, we see that the minimum distance grows slowly for R4 and H8 (which have $d^o \geq 3$) and not at all for R2 and P9 (which have $d^o = 2$). For $m = 2$, the growth rate of the minimum distance for R4, H8, and R2 appears distinctly linear. It is difficult

to determine the growth rate of P9 with $m = 2$ from these results. With $m = 3$, all of the codes appear to have a minimum distance growing linearly with the block length. In fact, the apparent growth rates are very close to $\delta^*_{GVB}(r)$. Finally, with $m = 4$, the bounds on minimum distance and $d^*_{GVB}(n, r)$ are almost indistinguishable. These results are very encouraging and suggest that, over a range of rates, even a few "accumulate" codes are sufficient to approach the behavior of an asymptotically large number.

2.5.3 Expurgated Ensembles

One of the problems with average WEs is that some terms may be dominated by the probability of choosing very bad codes. For example, at large enough SNR, the ensemble averaged probability of error will always be dominated by the code with smallest minimum distance even if the probability of choosing that code is extremely small. Now, suppose we could remove all of the codes with minimum distance $d_{min} < d$ from a particular ensemble. Then, every code in the new *expurgated ensemble* must have minimum distance $d_{min} \geq d$. Note that we must choose d carefully, otherwise there may be no codes left in the new ensemble. Suppose that we choose d and ϵ together such that the total probability of picking a code with $d_{min} \geq d$ from the original ensemble is exactly $1 - \epsilon$.

We can bound the ensemble averaged IOWE of the expurgated ensemble, which only contains codes with $d_{min} \geq d$, by dividing the original ensemble into two disjoint sets. Let $\overline{B}_{w,h}$ be the average IOWE for the ensemble with $d_{min} < d$, which has probability ϵ, and let $\overline{C}_{w,h}$ be the average IOWE for the ensemble with $d_{min} \geq d$, which has probability $1 - \epsilon$. We can write the original average IOWE as

$$\overline{A}_{w,h} = \epsilon \overline{B}_{w,h} + (1 - \epsilon)\overline{C}_{w,h},$$

and solving for $\overline{C}_{w,h}$ gives

$$\overline{C}_{w,h} = \frac{\overline{A}_{w,h} - \epsilon \overline{B}_{w,h}}{1 - \epsilon}.$$

Dropping the $\epsilon \overline{B}_{w,h}$ term gives the upper bound

$$\overline{C}_{w,h} \leq \frac{1}{1 - \epsilon} \overline{A}_{w,h}.$$

Up to this point, we have assumed that ϵ is known exactly. It is sufficient, however, to have an upper bound on ϵ which is less than 1. Applying Lemma 2.5.1 to this end gives

$$\overline{C}_{w,h} \leq \frac{1}{1 - \sum_{i=1}^{d-1} \sum_{j=1}^{k} \overline{A}_{j,i}} \overline{A}_{w,h}.$$

It is also clear, from the definition of $\overline{C}_{w,h}$, that $\overline{C}_{w,h} = 0$ for all $h < d$ and $w > 0$.

It is important to note that this result allows one to derive performance bounds which can be much tighter for typical codes in the ensemble. For example, suppose that all of the codes in the ensemble with small minimum distance have a small total probability, ϵ, so that the rest of the codes, which have very good minimum distance, have a large total probability. Performance bounds based on the average WE will always have an error floor based upon ϵ and the small minimum distance, while bounds based on the expurgated ensemble will represent the performance of the typical codes, which have large minimum distance.

2.6 Performance

2.6.1 The Error Exponent

In this section, we draw on a generalization of Gallager's derivation of the error exponent [8] due to Shulman and Feder [20]. This generalization allows one to upper bound the probability of MLD error, using Gallager's random coding error exponent, for any binary linear code. Applying Theorem 1 from [20] to (2.3.4) shows that, for any symmetric memoryless channel with binary inputs and discrete outputs, the ensemble $\Omega_*(\mathbf{C}^{(o)}, \mathbf{C}^{(i)})$ has nearly the same error exponent as the Shannon ensemble of random codes. A Shannon random code is generated by picking the 2^{rn} codewords uniformly from the 2^n possible binary sequences with replacement, and the Shannon ensemble is the set of possible codes chosen in this manner with their associated probabilities. Since the Shannon ensemble achieves the capacity of any symmetric discrete memoryless channel, this proves that the ensemble $\Omega_*(\mathbf{C}^{(o)}, \mathbf{C}^{(i)})$ can operate at rates arbitrarily close to capacity.

Theorem 2.6.1 (Shulman-Feder). *The probability of word error P_W for a family of $(n, \lceil rn \rceil)$ linear codes, transmitted over a symmetric memoryless channel with binary inputs and discrete outputs, is upper bounded by*

$$P_W \leq 2^{-nE\left(r + \frac{\log_2 \alpha}{n}\right)} \tag{2.6.1}$$

where $E(\cdot)$ is the error exponent of the channel and

$$\alpha = \max_{1 \le h \le n} \frac{\overline{A}_h}{2^{\lceil rn \rceil} - 1} \frac{2^n}{\binom{n}{h}}.$$

□

Corollary 2.6.2. *Consider the ensemble $\Omega_m(\mathbf{C}^{(o)}, \mathbf{C}^{(i)})$ with n code bits and $\lceil rn \rceil$ information for $0 < r < 1$. Let P_W be the average probability of word error when a code is randomly chosen from the ensemble and used on some channel with MLD. There exists an m_0 such that, for all $m \ge m_0$, we have*

$$P_W \le 2^{-nE(r+O(1/n))},$$

where $E(\cdot)$ is the error exponent of the channel.

Proof. Using Theorem 2.6.1, we must simply show that the constant α for the ensemble $\Omega_m(\mathbf{C}^{(o)}, \mathbf{C}^{(i)})$ remains essentially constant as n increases. Using the formula for α and Theorem 2.3.10, we find that

$$\begin{aligned} \alpha &= \max_{1 \le h \le n} \frac{\overline{A}_h^{(m)}}{2^{\lceil rn \rceil} - 1} \frac{2^n}{\binom{n}{h}} \\ &\le \frac{1}{1 + 2^{-n}} + \max_{1 \le h \le n} \frac{\gamma\, 2^n}{(2^{\lceil rn \rceil} - 1)\binom{n}{h}}. \end{aligned}$$

Since γ can be made arbitrarily small by increasing m (from Theorem 2.3.10), we choose m_0 such that

$$\max_{1 \le h \le n} \frac{\gamma\, 2^n}{(2^{\lceil rn \rceil} - 1)\binom{n}{h}} \le \frac{1}{(1 - 2^{-n})}$$

for all $m \ge m_0$. This gives the upper bound $\alpha \le 2/(1 - 2^{-n})$, and now we can estimate $(\log_2 \alpha)/n$ using

$$\frac{1}{n} \log_2 \alpha \le \frac{1}{n} - \frac{1}{n} \log_2(1 - 2^{-n}) = O\left(\frac{1}{n}\right).$$

This completes the proof. □

Remark 2.6.3. Since the constant γ is proportional to q^m for some $q < 1$, this proof actually requires the value of m_0 to grow linearly with n. This is because the probability that a poor code is chosen from the ensemble decays very slowly. Nonetheless, we believe that almost all of the codes in the ensemble will achieve the error exponent as long as m_0 grows faster than logarithmically in n.

2.6.2 Maximum Likelihood Decoding Performance

In Section 2.5.2, we applied (2.3.2) to compute the averaged WEs for several CA^m code ensembles. Using the WEs and the Viterbi-Viterbi Bound [22], we calculated upper bounds on the probability of MLD error for some of these ensembles. The results for the R2, R4, and P9 ensembles are given in Figs. 2.7.1, 2.7.2, and 2.7.3, respectively. These figures also show the results of iterative decoding simulations which will be discussed in the next section. At high SNR, these bounds are dominated by the probability of picking a code with small minimum distance, as reflected in the pronounced error floors of the non-expurgated ensembles in Figs. 2.7.1, 2.7.2, and 2.7.3. For this reason, we also considered the expurgated ensembles, as described in Section 2.5.3, with $\epsilon = 1/2$.

The results of applying the Viterbi-Viterbi bound to these ensembles have some characteristics worth mentioning. In all cases, increasing m, the number of "accumulate" codes, seems to improve the performance both by shifting the cliff region to the left and by lowering the error floor. We also see that, in some cases, the effect of expurgation is negligible, which implies that almost all of the codes in the ensemble have small minimum distance. As we saw in Section 2.5.2, the minimum distance of the expurgated ensemble depends on the outer code and the number of "accumulate" codes. The minimum distance of the outer code does not seem to completely explain the behavior though, because the P9 ensemble requires one more "accumulate" code than the R2 ensemble in order for expurgation to make a significant difference. Of course, at longer block lengths this may change.

The axes of the figures were chosen to show details of the performance curves, but in many cases the error floor of the expurgated ensemble is too low to be shown. Consider the R2 ensemble with $m = 2$; the WER of the expurgated ensemble remains steep until around 10^{-28} where it flattens somewhat. The expurgated R2 ensemble with $m = 3$ has a WER which remains steep until well below the numerical accuracy of our computations. The expurgated P9 ensemble with $m = 3$ also shows no error-floor region, but the curve loses some steepness at a WER of 10^{-18}. These error floors are interesting because understanding the performance of these codes at high SNR, where simulation is infeasible, is important for applications where very low error rates are required.

2.6.3 Iterative Decoding Performance

In this section, we use computer simulation to evaluate the performance of iterative decoding for these codes. A single decoding iteration corresponds to $2m - 1$ APP decoding operations of an "accumulate" code (a backward/forward pass through all "accumulate" encoders) and a single APP decoding operation of the outer code. It is worth noting that the complexity of iterative decoding is linear in both m and n, making it quite feasible to implement. All simulation results were obtained using between 20 and 50 decoding iterations, depending on the particular code, and modest gains are observed (but not shown) when the number of iterations is increased to 200. These results are compared with analytical bounds in Figs. 2.7.1, 2.7.2, and 2.7.3 and shown by themselves in Figs. 2.7.4 and 2.7.5.

The discrepancies between the simulation results and MLD bounds in Figs. 2.7.1, 2.7.2, and 2.7.3 are very pronounced. While the MLD bounds predict uniformly improving performance with increasing m, it is clear that the performance of iterative decoding does not behave in this manner. The optimum m depends on the desired error rate and the minimum distance of the outer code. In general, it appears that increasing m moves the the cliff region of the error curve to the right and makes the floor region steeper. This seems reasonable because more rate-1 decoders (which have no coding gain) are applied before the outer code (with all of the coding gain) is decoded. This results in a phenomenon where the iterative decoder often does not converge, but rarely makes a mistake when it does converge.

The expurgated WE can also be used to detect the presence of bad codes which are chosen with low probability. If the MLD expurgated bound is better than the non-expurgated bound, then the effect of these bad codes has been reduced. The MLD expurgated bound is not shown when it coincides with the non-expurgated bound. In some cases, iterative decoding is performing better than the MLD expurgated bound (e.g., the R4 ensemble with $m = 1$). This may occur because the use of well designed (e.g., S-random [5]) interleavers can provide a minimum distance which is better than that guaranteed by Lemma 2.5.1.

The Interleaver Gain Exponent (IGE) conjecture is based on the observations of Benedetto and Montorsi [2] and is stated rigorously in [4]. It states that the probability of MLD decoding error for turbo-like codes will decay as $O(n^{-\nu})$, where ν depends on the details of the coding system. If the IGE conjecture predicts that the BER (resp. WER) will decay with the block length, then we say that the system has BER (resp. WER) interleaver gain. It is easy to

verify that WER interleaver gain implies BER interleaver gain. The IGE and the MLD expurgated bound are quite closely connected. If a system has WER interleaver gain, the probability of picking a code with codewords of fixed weight must decay to zero as the block length increases. Therefore, one would expect the MLD expurgated bound to beat the non-expurgated bound. On the other hand, if a system has only BER interleaver gain, then it is likely that the MLD expurgated bound will equal the non-expurgated bound.

Finally, the IGE Conjecture predicts that the R2 code will have no WER interleaver gain (i.e. $P_W = O(n^{-1})$) for $m = 1$, but that it will have WER interleaver gain (i.e. $P_W = O(n^{-1})$) for $m = 2$. In Fig. 2.7.4, the WER of the R2 code with $m = 1$ does indeed appear to be independent of block length and the WER of the R2 code with $m = 2$ is clearly decreasing with block length. In Fig. 2.7.5, we see similar behavior for the interleaver gain of the P9 codes.

2.7 Conclusions and Future Work

In this chapter, we introduce a new ensemble of binary linear codes consisting of any rate $r < 1$ outer code followed by a large number of uniformly interleaved rate-1 codes. We show that this ensemble is very similar to the ensemble of uniform random linear codes in terms of minimum distance and error exponent characteristics. A key tool in the analysis of these codes is a correspondence between input output weight transition probability matrices and Markov Chains (MC), which allows us to draw on some well-known limit theorems from MC theory. We derive a probabilistic bound on the minimum distance of codes from this ensemble, and show it to be almost identical to the Gilbert-Varshamov Bound (GVB). In particular, our analysis implies that almost all long codes in the ensemble have a normalized minimum distance meeting the GVB.

Next, we consider a particular class of these codes, which we refer to as Convolutional Accumulate-m (CA^m) codes. These codes consist of an outer terminated convolutional code followed by m uniformly interleaved "accumulate" codes. We evaluate the minimum distance bound for a few specific CA^m codes for $m = 1, \ldots, 4$ and observe that these relatively small m values may be sufficient to approach the GVB. Finally, we use computer simulation to evaluate the bit error rate and word error rate performance of these CA^m codes with iterative decoding and compare this to the performance predicted by union bounds for maximum likelihood decoding (MLD).

Remark 2.7.1. An MLD coding theorem for CA^m codes can be found in [16], with numerical estimates of the corresponding noise thresholds. Also given there are the thresholds which result from applying density evolution to the iterative decoding of these codes [18]. Finally, a comprehensive treatment of both of these subjects can be found in [14].

Acknowledgement. We are very grateful to M. Öberg for posing the initial question that led to this research and for many helpful discussions along the way.

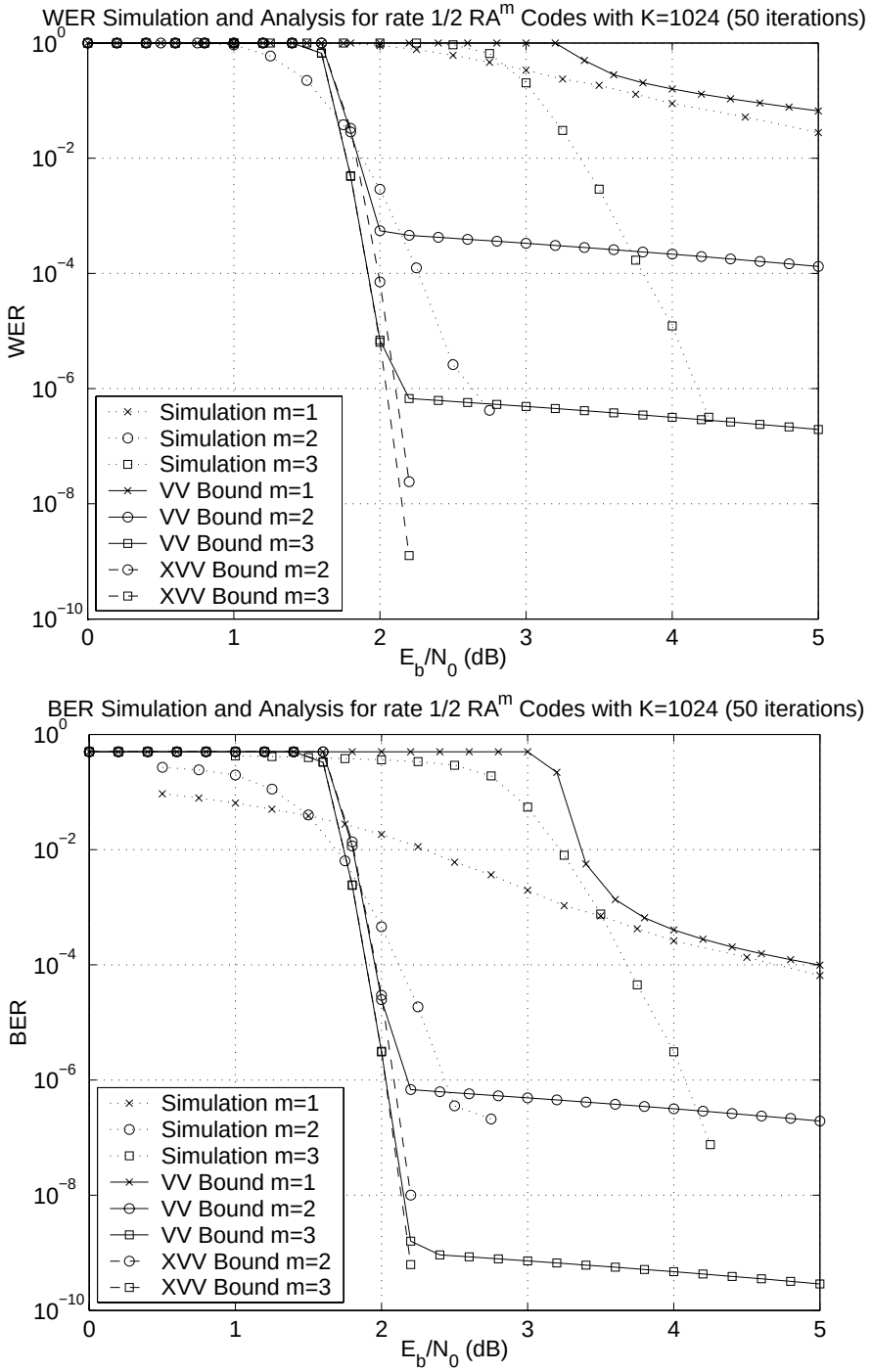


Figure 2.7.1: Analytical and simulation results for a rate 1/2 RA^m code with $k = 1024$ and $m = 1, 2, 3$. Simulations are completed using 50 decoding iterations and the top plot shows the word error rate (WER) while the bottom plot shows the bit error rate (BER). The label XVV signifies the Viterbi-Viterbi (VV) Bound applied to the expurgated ensembles.

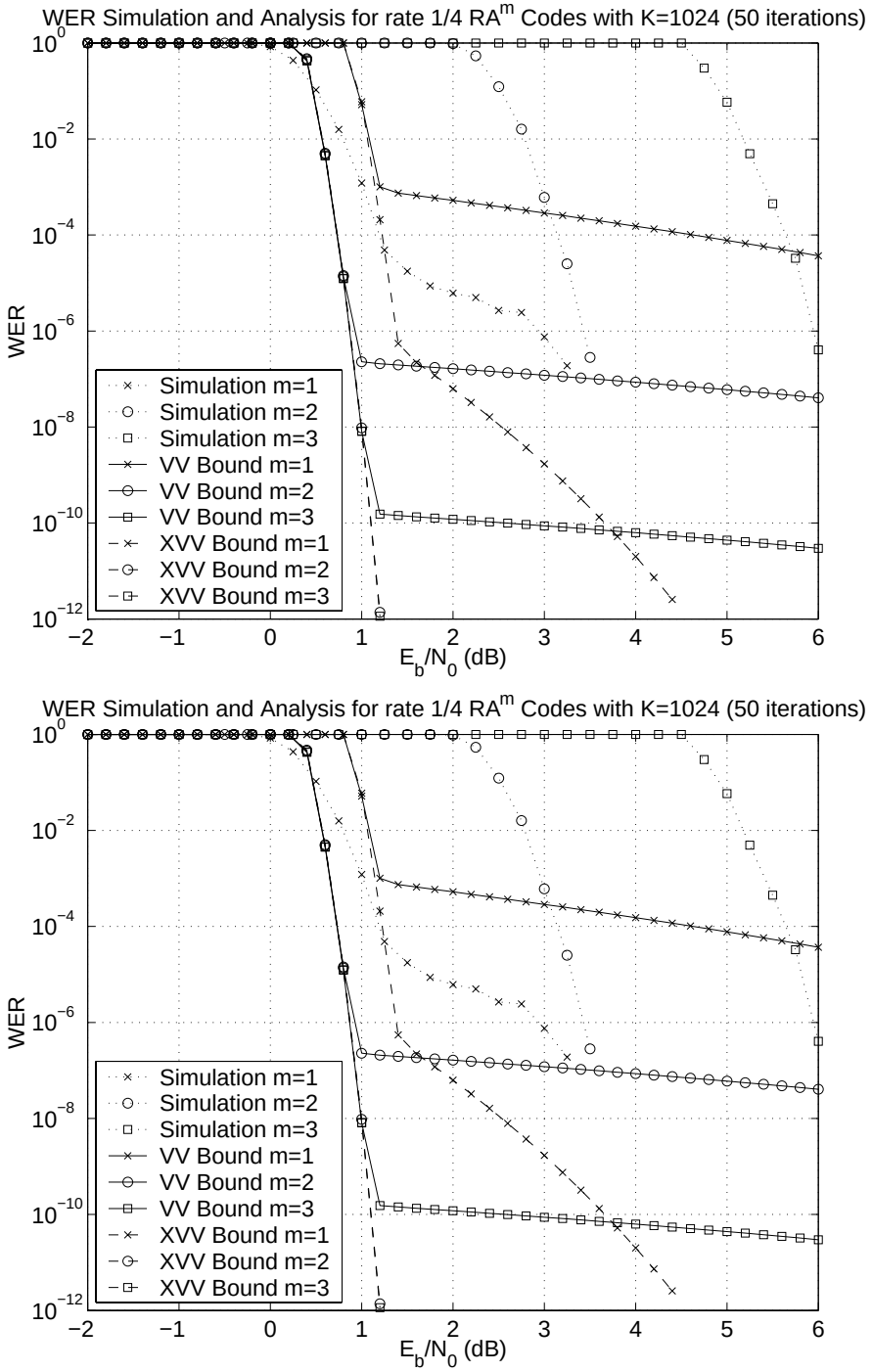


Figure 2.7.2: Analytical and simulation results for a rate 1/4 RA^m code with $k = 1024$ and $m = 1, 2, 3$. Simulations are completed using 50 decoding iterations and the top plot shows the word error rate (WER) while the bottom plot shows the bit error rate (BER). The label XVV signifies the Viterbi-Viterbi (VV) Bound applied to the expurgated ensembles.

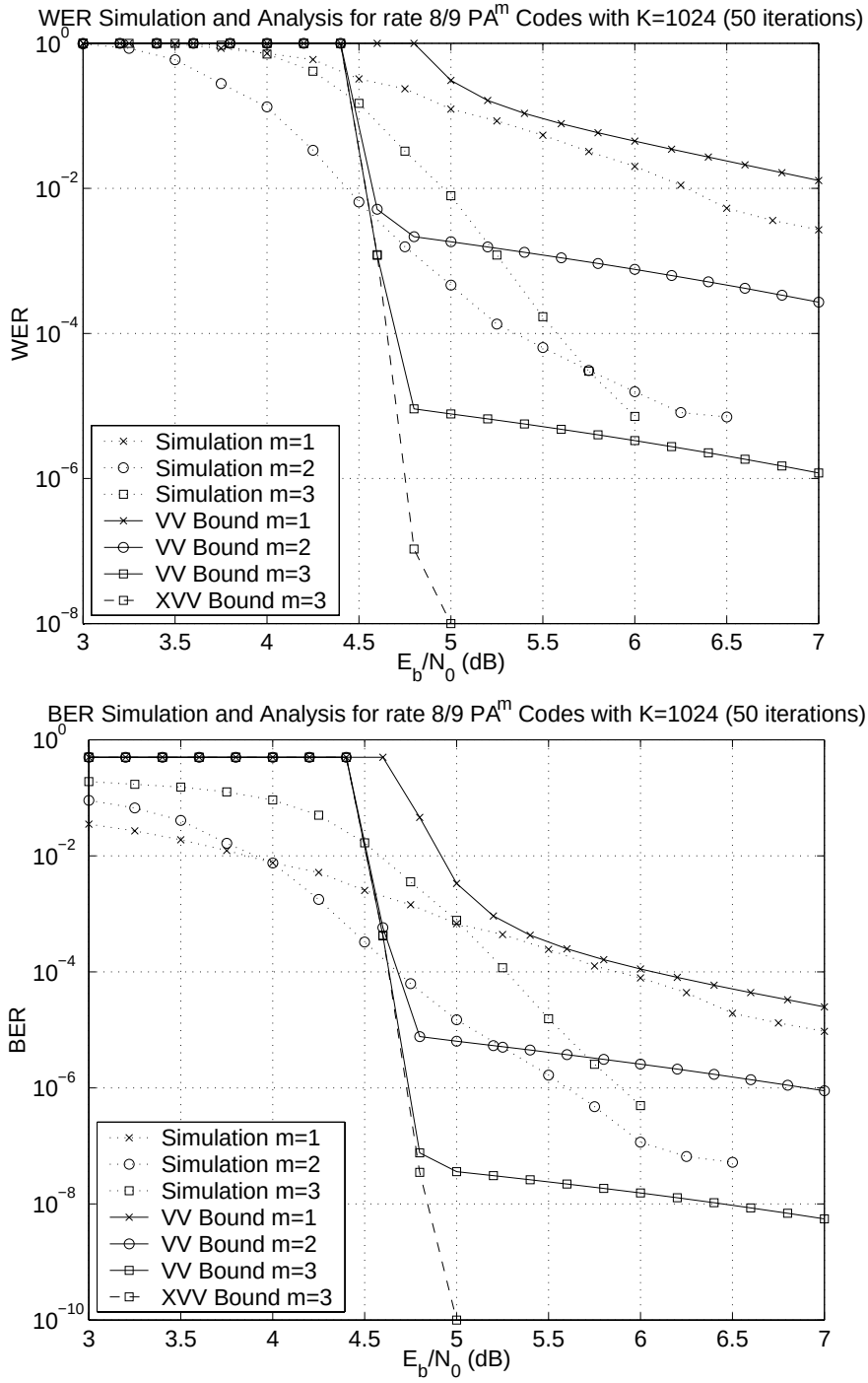


Figure 2.7.3: Analytical and simulation results for a rate 8/9 PA^m with $k = 1024$ and $m = 1, 2, 3$. Simulations are completed using 50 decoding iterations and the top plot shows the word error rate (WER) while the bottom plot shows the bit error rate (BER). The label XVV signifies the Viterbi-Viterbi (VV) Bound applied to the expurgated ensembles.

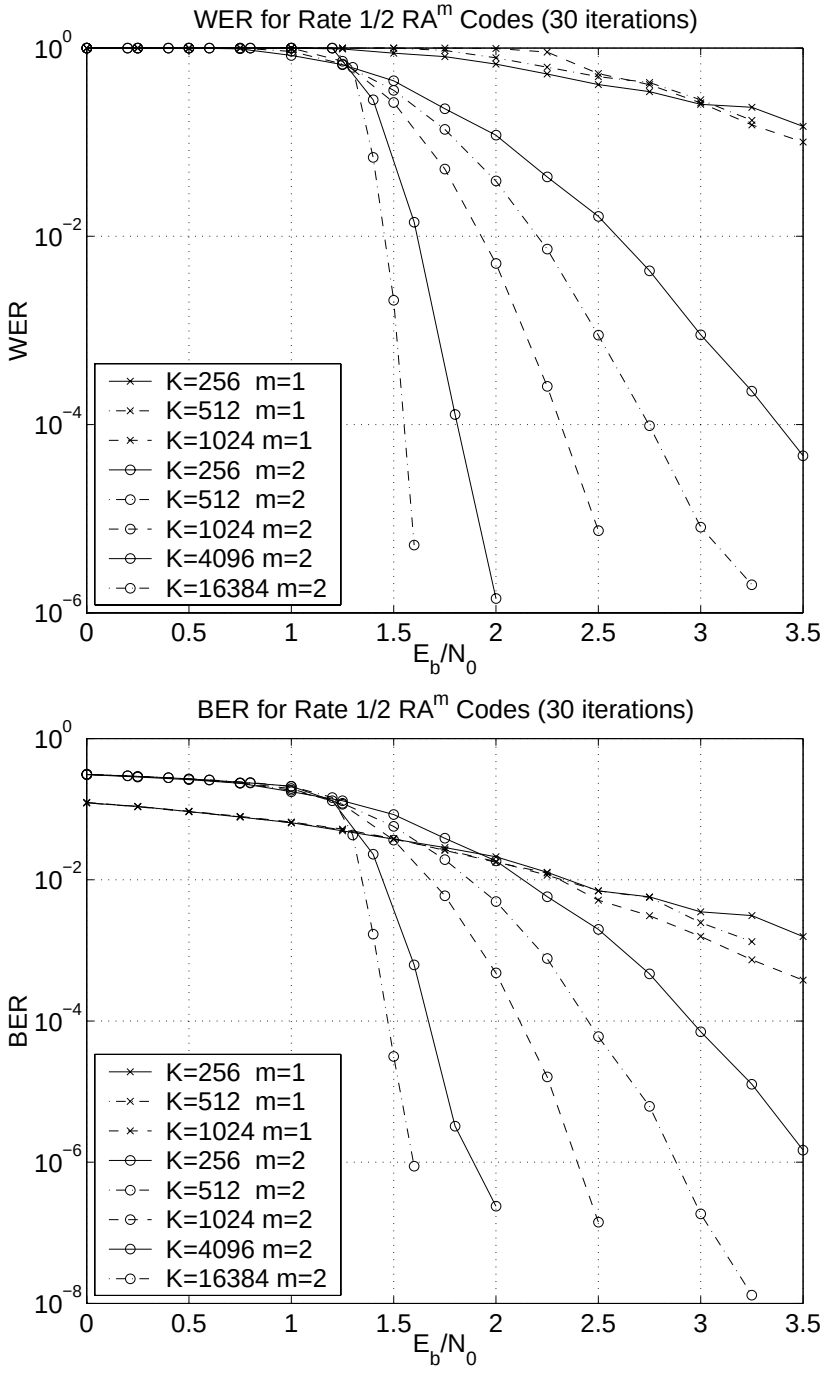


Figure 2.7.4: Simulation results for rate 1/2 RA^m codes for 30 decoding iterations with $m = 1, 2$ and $k = 1024, 2048, 4096, 8192, 16384$. The top plot shows the word error rate (WER) while the bottom plot shows the bit error rate (BER).

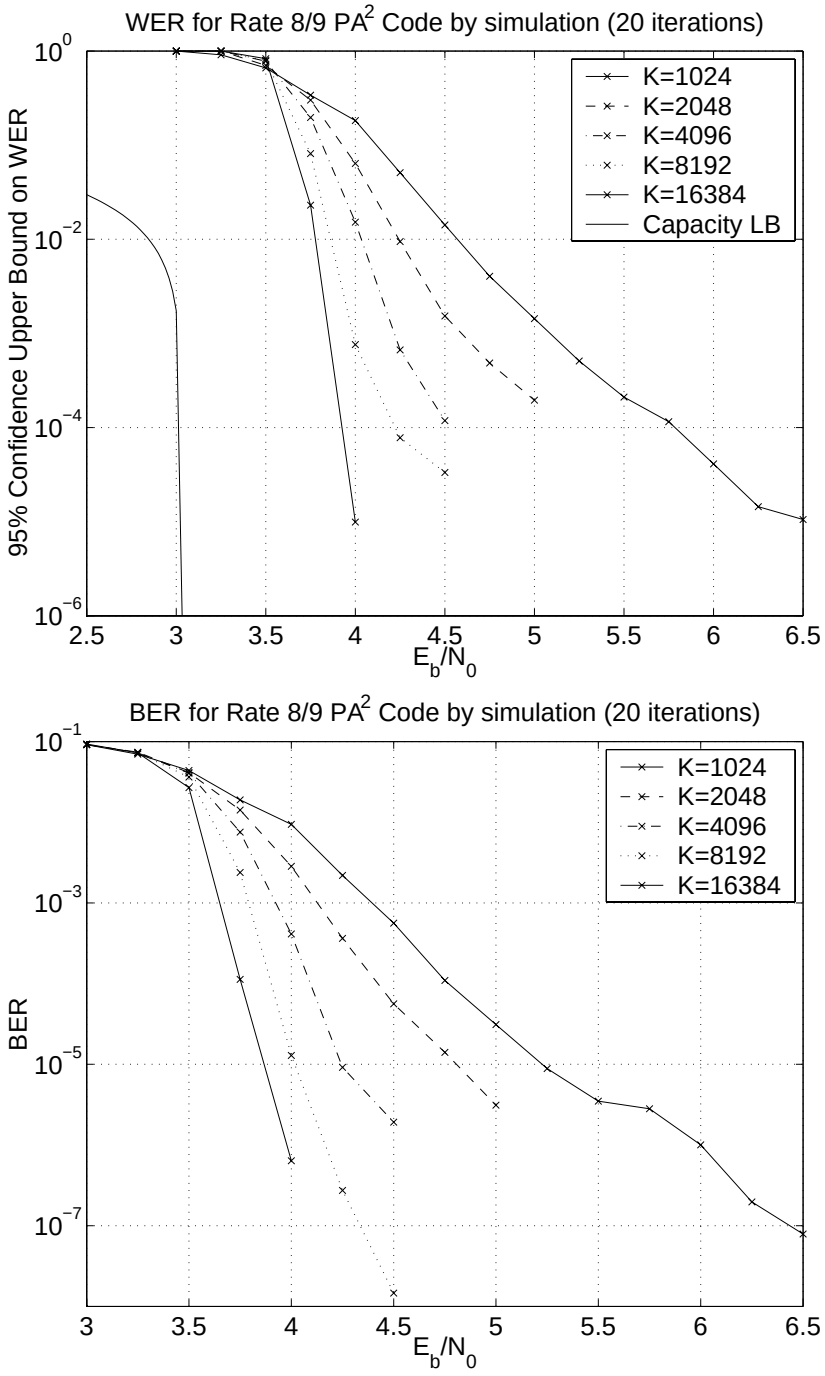


Figure 2.7.5: Simulation results for rate 8/9 PA^2 codes with $k = 1024, 2048, 4096, 8192, 16384$ and 20 decoding iterations. The top plot shows the word error rate (WER) while the bottom plot shows the bit error rate (BER).

2A Proof of Theorem 2.4.1

The generator matrix, $\mathbf{T}_n$, of the length n block code is

$$\mathbf{T}_n = \begin{bmatrix} h_0 & h_1 & h_2 & \dots & h_{n-1} \\ & h_0 & h_2 & \dots & h_{n-2} \\ & & h_0 & \ddots & \vdots \\ & & & h_0 & h_1 \\ & & & & h_0 \end{bmatrix}.$$

For simplicity of notation, we define $m = l + 1$. By hypothesis, the generator matrix of this code will be the identity matrix for any $n < m$, making the code trivial. We will show that the block code of length $n \geq m$ is primitive by first establishing that the length-m block code is primitive, then showing that the length $n + 1$ block code is primitive if the length n code is primitive, and finally using induction to extend the proof to arbitrarily large n.

Recall that a rate-1 block code is primitive if the MC associated with the $\mathbf{Q}$ submatrix of the code's IOWTP matrix is primitive. Let $\mathbf{Q}_n$ be the $\mathbf{Q}$ submatrix of the length-n block code's IOWTP matrix. It is easy to verify that $[\mathbf{Q}_n]_{i,j}$ is greater than zero iff the corresponding component of the IOWE of the length-n block code, $A_{i,j}^{(n)}$, is greater than zero. Thinking of the latter as an adjacency matrix, we associate to the length-n block code a directed graph G_n, which we call the *weight-mapping graph*. The vertices of G_n, which are labeled $1, 2, \dots, n$, correspond to the Hamming weights of input and output sequences of the code. Denote the Hamming weight of a binary vector $\mathbf{v}$ by $|\mathbf{v}|$. For each binary input to the code, $\mathbf{b} = b_1, b_2, \dots, b_n$, there is a directed edge from the vertex labeled $|\mathbf{b}|$ to the vertex labeled $|\mathbf{c}|$ if the input vector $\mathbf{b}$ produces the output vector $\mathbf{c}$. This implies that the graph G_n will have a directed edge from vertex i to vertex j iff $A_{i,j}^{(n)} > 0$. Therefore, the graph G_n has the same connectivity as the MC associated with $\mathbf{Q}_n$, and we have reduced the problem to showing that each G_n, for $n \geq m$, is primitive.

We will prove that each G_n is primitive by establishing that it is both irreducible and aperiodic. By definition, a graph is irreducible if there is a directed path from each vertex to every other vertex. A graph is aperiodic if the greatest common divisor of the lengths of all its cycles (i.e., paths which start and end in the same state) is one. Therefore, for aperiodicity, it is sufficient to exhibit a single vertex with a self-loop (i.e., a directed edge from a vertex back to itself). The verification of these properties for G_n will be simplified by the fact, proved below, that G_n is a subgraph of G_{n+1}.

For the primary case, corresponding to length $n = m$, the generator matrix of the code is

$$\mathbf{T}_m = \begin{bmatrix} 1 & 0 & \dots & 0 & 1 \\ & 1 & 0 & 0 & 0 \\ & & \ddots & 0 & \vdots \\ & & & 1 & 0 \\ & & & & 1 \end{bmatrix}.$$

Consider strings of the form $[1^s, 0^{n-s}]$ and $[1^{s-1}, 0^{n-s}, 1]$, where a^j refers to a string of j repeated symbols a. For each $s = 1, \dots, n-1$, the input $\mathbf{b} = [1^s, 0^{n-s}]$ has weight s and produces an output $\mathbf{c} = [1^s, 0^{n-s-1}, 1]$ which has weight $s+1$. Likewise, for each $s = 2, \dots, n$, the input $\mathbf{b} = [1^{s-1}, 0^{n-s}, 1]$ has weight s and produces an output $\mathbf{c} = [1^{s-1}, 0^{n-s+1}]$ which has weight $s-1$. Now consider any vertex, labeled i, in the graph G_m. These input-output pairs establish that there is a directed edge from the vertex labeled i to the vertex labeled $i+1$ and to the vertex labeled $i-1$, if those vertices exist. So there is a directed path from any vertex to any other vertex, and G_m is irreducible. The input $\mathbf{b} = [0^{n-1}, 1]$ produces the output $\mathbf{c} = [0^{n-1}, 1]$ which establishes that the vertex labeled 1 has a self-loop. So the graph G_m is also aperiodic, and therefore primitive.

Now we assume that G_n is primitive for some $n \geq m$, and use this to prove that G_{n+1} is primitive. We start by proving the result mentioned above: G_n is a subgraph of G_{n+1}. Consider any input, $\mathbf{b}$, to the rate-1 block code with generator matrix $\mathbf{T}_n$. The output will be $\mathbf{b}\mathbf{T}_n$ and the weight mapping graph, G_n, will have an edge from the vertex labeled $|\mathbf{b}|$ to the vertex labeled $|\mathbf{b}\mathbf{T}_n|$. In fact, all edges of G_n are enumerated by considering all possible inputs. Notice that the generator matrix, T_{n+1}, can be written as

$$\mathbf{T}_{n+1} = \left[\begin{array}{c:ccc} h_0 & h_1 & \dots & h_n \\ \hdashline 0 & & & \\ \vdots & & \mathbf{T}_n & \\ 0 & & & \end{array}\right].$$

This implies that $\begin{bmatrix} 0 & \mathbf{b} \end{bmatrix} \mathbf{T}_{n+1} = \begin{bmatrix} 0 & \mathbf{b}\mathbf{T}_n \end{bmatrix}$ and proves, for each $\mathbf{b}$, that the weight mapping graph G_{n+1} also has a directed edge from the vertex labeled $|\mathbf{b}|$ to the vertex labeled

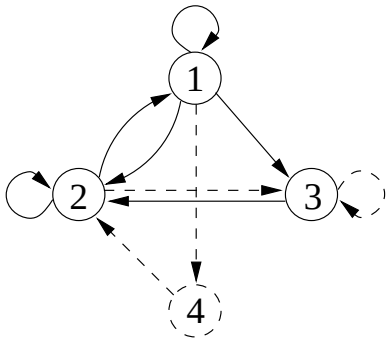


Figure 2A.1: The weight mapping graph, G_4, with the G_3 subgraph drawn in solid lines.

$|\mathbf{b}\mathbf{T}_n|$. So, for every directed edge in G_n connecting two labeled vertices, there is a directed edge in G_{n+1} connecting two vertices with the same labels. The vertices of G_n are also a subset of the vertices of G_{n+1}, so G_n is a subgraph of G_{n+1}.

To prove that the graph G_{n+1} is irreducible, it now suffices to show that G_{n+1} has a directed edge from the vertex labeled $n+1$ to some vertex with label $i \neq n+1$, as well as a directed edge from some such vertex to vertex $n+1$. Consider $\mathbf{b} = [1^{n+1}]$, the only input of weight $n+1$, and notice that the mth column of T_{n+1} has exactly two ones. Therefore the mth element of $\mathbf{b}\mathbf{T}_{n+1}$ must be zero and $\mathbf{b}\mathbf{T}_{n+1} \neq \mathbf{b}$. This implies that an input of weight $n+1$ produces an output of weight $i < n+1$. Therefore, G_{n+1} has a directed edge from the vertex labeled $n+1$ to a vertex labeled i where $i < n+1$. Next, we notice that $\mathbf{T}_{n+1}$ is upper triangular and has all ones on the main diagonal, which makes it invertible. This means that there must be a unique input, $\mathbf{b}'$, which is mapped to the output $\mathbf{b} = [1^{n+1}]$. We know that this input must obey the equation $\mathbf{b}'\mathbf{T}_{n+1} = \mathbf{b}$, and since $\mathbf{b}\mathbf{T}_{n+1} \neq \mathbf{b}$**,** we also know that $\mathbf{b}' \neq \mathbf{b}$. Since $\mathbf{b}$ is the only length-$(n+1)$ sequence of weight $n+1$, we conclude that $|\mathbf{b}'| < n+1$. This implies that there is an input of weight $i = |\mathbf{b}'| < n+1$ which produces an output of weight $n+1$. Therefore, G_{n+1} has a directed edge from a vertex labeled i, for some $i < n+1$, to the vertex labeled $n+1$. We conclude that G_{n+1} is irreducible.

The aperiodicity of G_{n+1} follows immediately from the fact that the subgraph $G_m \subset G_{n+1}$ contains a self-loop at vertex 1. This completes the proof that G_{n+1} is primitive, and, therefore, the proof that the rate-1 block code of length $n+1$ is primitive, as desired. □

We illustrate the proof technique using the “accumulate” code example from Section

2.2.3. The impulse response, $\mathbf{h}$, of the "accumulate" code is the infinite sequence of ones, $h_i = 1$, for $i \geq 0$. The weight mapping graph, G_4, is shown in Fig. 2A.1 and the G_3 subgraph is drawn with solid lines. It is easy to see that G_3 is both irreducible and aperiodic; in particular, note the self-loop at the vertex labeled 1. There is an edge G_4 from vertex 4 to vertex 2, corresponding to the weight-4 input vector $\mathbf{b} = [1^4]$, and a directed edge from vertex 1 to vertex 4, corresponding to the weight-4 input vector $\mathbf{b}' = [1, 0^3]$. Together with the irreducibility of G_3, this implies that G_4 is irreducible. The self-loop at vertex 1 ensures the aperiodicity and, therefore, the primitivity of G_4.

Bibliography

[1] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara. Analysis, design, and iterative decoding of double serially concatenated codes with interleavers. *IEEE J. Select. Areas Commun.*, 16(2):231–244, Feb. 1998.

[2] S. Benedetto and G. Montorsi. Unveiling turbo codes: Some results on parallel concatenated coding schemes. *IEEE Trans. Inform. Theory*, 42(2):409–428, March 1996.

[3] C. Berrou, A. Glavieux, and P. Thitimajshima. Near Shannon limit error-correcting coding and decoding: Turbo-codes. In *Proc. IEEE Int. Conf. Commun.*, volume 2, pages 1064–1070, Geneva, Switzerland, May 1993. IEEE.

[4] D. Divsalar, H. Jin, and R. J. McEliece. Coding theorems for "turbo–like" codes. In *Proc. 36th Annual Allerton Conf. on Commun., Control, and Comp.*, pages 201–210, Monticello, IL, USA, Sept. 1998.

[5] D. Divsalar and F. Pollara. Turbo codes for PCS applications. In *Proc. IEEE Int. Conf. Commun.*, pages 54–59, Seattle, WA, USA, June 1995. IEEE.

[6] T. Ericson. *Bounds on the Size of a Code*, pages 45–69. Number 128 in Lecture Notes in Control and Information Sciences. Springer-Verlag Berlin, Heidelberg, Germany, 1989.

[7] R. G. Gallager. *Low-Density Parity-Check Codes*. The M.I.T. Press, Cambridge, MA, USA, 1963.

[8] R. G. Gallager. A simple derivation of the coding theorem and some applications. *IEEE Trans. Inform. Theory*, 11(1):3–17, Jan. 1965.

[9] E. N. Gilbert. A comparison of signalling alphabets. *The Bell Syst. Techn. J.*, 31:504–522, May 1952.

[10] H. Jin. *Analysis and Design of Turbo-like Codes*. PhD thesis, Caltech, May 2001.

[11] N. Kahale and R. Urbanke. On the minimum distance of parallel and serially concatenated codes. In *Proc. IEEE Int. Symp. Information Theory*, page 31, Cambridge, MA, USA, Aug. 1998. IEEE.

[12] M. Öberg and P. H. Siegel. Performance analysis of turbo-equalized dicode partial-response channel. In *Proc. 36th Annual Allerton Conf. on Commun., Control, and Comp.*, pages 230–239, Monticello, IL, USA, Sept. 1998.

[13] A. Papoulis. *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill, New York, NY, USA, 3rd edition, 1991. ISBN 0-07-048477-5.

[14] H. D. Pfister. *On the Capacity of Finite State Channels and the Analysis of Convolutional Accumulate-m Codes*. PhD thesis, University of California, San Diego, La Jolla, CA, USA, March 2003.

[15] H. D. Pfister and P. H. Siegel. The serial concatenation of rate-1 codes through uniform random interleavers. In *Proc. 37th Annual Allerton Conf. on Commun., Control, and Comp.*, pages 260–269, Monticello, IL, USA, Sept. 1999.

[16] H. D. Pfister and P. H. Siegel. Coding theorems for generalized repeat accumulate codes. In *Int. Symp. Inform. Theory and its Appl.*, volume 1, pages 21–25, Honolulu, HI, USA, Nov. 2000. IEEE.

[17] J. N. Pierce. Limit distribution of the minimum distance of random linear codes. *IEEE Trans. Inform. Theory*, 13:595–599, Oct. 1967.

[18] T. J. Richardson and R. L. Urbanke. The capacity of low-density parity check codes under message-passing decoding. *IEEE Trans. Inform. Theory*, 47(2):599–618, Feb. 2001.

[19] E. Seneta. *Non-Negative Matrices: An Introduction to Theory and Applications*. Wiley, New York, NY, USA, 2nd edition, 1981.

[20] N. Shulman and M. Feder. Random coding techniques for nonrandom codes. *IEEE Trans. Inform. Theory*, 45(6):2101–2104, Sept. 1999.

[21] A. J. Viterbi and J. K. Omura. *Principles of Digital Communication and Coding*. McGraw-Hill, New York, NY, USA, 1979.

[22] A. M. Viterbi and A. J. Viterbi. Improved union bound on linear codes for the input-binary AWGN channel, with applications to turbo codes. In *Proc. IEEE Int. Symp. Information Theory*, volume 1, page 29, Cambridge, MA, USA, Sept. 1998. IEEE.

Chapter 3

Coding Theorems for Convolutional Accumulate-m Codes

3.1 Introduction

It is well-known that long random codes achieve reliable communication at noise levels up to the Shannon limit, but they provide no structure for efficient decoding. The introduction and analysis of Repeat Accumulate (RA) codes by Divsalar, Jin, and McEliece [10] shows that the concatenation of a repetition code and a rate-1 code, through a random interleaver, can also achieve reliable communication at noise levels near the Shannon limit. A more general analysis of serially concatenated rate-1 codes also implies that using more than one interleaved rate-1 code may yield further improvement [23].

The coding theorem for the ensemble of RA codes under maximum likelihood decoding, given in [10], states that, for all E_b/N_0 greater than a threshold which depends only on the repeat order $q \geq 3$, the serial concatenation of a repetition code and a rate-1 "accumulate" code will have vanishing word error probability as the block length goes to infinity. In [14], this theorem was extended to serial turbo codes, for outer codes with minimum distance $d \geq 3$.

In this chapter, we combine two different generalizations of RA codes. The first involves using either a single parity check (SPC) or a terminated convolutional code (TCC) as the outer code, and we refer to these codes as Parity Accumulate (PA) and Convolutional Accumulate (CA) codes respectively. The second involves using a cascade of m interleaved rate-1

"accumulate" codes as the outer code [23], and we refer to these codes as either RA^m, PA^m, or CA^m codes respectively. Of these classes, CA^m codes are the most general and both RA^m and PA^m can also be viewed as CA^m codes by choosing the TCC appropriately. He also discusses repeat accumulate accumulate (RAA) codes in [13], perhaps overlooking their previous work in [24].

Following the approach pioneered in [10], we then prove a coding theorem for ensembles of CA^m codes on a memoryless channel with maximum likelihood decoding. The theorem states that if the outer code has minimum distance $d \geq 2$ and the channel parameter z is less than some threshold z^*, then the probability of word error is $O(n^\nu)$, where n is the block length and ν is determined solely by m and d. The proof, based on the union bound, also gives loose lower bounds on the threshold z^*. A new tighter bound by Jin and McEliece [16] allows us to compute very accurate E_b/N_0 thresholds for the additive white Gaussian noise (AWGN) channel. For $m = 3$, many of these thresholds are virtually identical to the Shannon limit.

The chapter is organized as follows. In Section 3.2, we review key results relating to turbo-like codes which will be required for later sections. In Section 3.3, we discuss new and existing bounds on the weight enumerators of TCCs. In Section 3.4, we consider bounds on the input output weight transition probabilities of the rate-1 "accumulate" code. In Section 3.5, we apply the bounds of the two previous sections to RA and CA codes with a single rate-1 "accumulate" code. In Section 3.6, we state and prove our coding theorem for CA^m codes and follow up by considering the minimum distance of these codes. In Section 3.7, we discuss the iterative decoding and density evolution for CA^m codes. In Section 3.8, we present E_b/N_0 and minimum distance thresholds for CA^m codes and discuss the numerical methods used to compute them. Finally, in Section 3.9, we offer some concluding remarks.

3.2 Preliminaries

3.2.1 Weight Enumerators and the Union Bound

In this section, we review the weight enumerator of a linear block code and the union bound on error probability for maximum likelihood decoding. The *input output weight enumerator* (IOWE), $A_{w,h}$, of an (n, k) linear block code is the number of codewords with input weight w and output weight h, and the *weight enumerator* (WE), A_h, is the number of codewords with

output weight h and any input weight. Using these definitions, the probability of word error for maximum likelihood (ML) decoder is upper bounded by

$$P_W \leq \sum_{h=1}^{n} \sum_{w=1}^{k} A_{w,h} z^h = \sum_{h=1}^{n} A_h z^h \tag{3.2.1}$$

because the pairwise error probability between any two codewords differing in h positions is upper bounded by z^h.

The parameter z is known as the Bhattacharyya parameter and can be computed for any memoryless channel [30, p. 88]. For a binary-input discrete output channel with M outputs, it is defined as

$$z = \sum_{j=0}^{M-1} \sqrt{p(j|0)p(j|1)},$$

where $p(j|i)$ is the probability of output j given input i. For channels with continuous outputs, the parameter z is given by the integral

$$z = \int_Y \sqrt{p(y|0)p(y|1)}dy,$$

where $p(y|i)$ is the output p.d.f. of y given input i and Y is the set of possible outputs. For the BSC this gives $z_{BSC}(p) = \sqrt{4p(1-p)}$, and for the AWGN channel this gives $z_{AWGN}(\sigma^2) = e^{-1/(2\sigma^2)}$, where $E_s/N_0 = (k/n)E_b/N_0 = 1/(2\sigma^2)$.

Finally, the bit error probability is upper bounded by

$$P_B \leq \sum_{h=1}^{n} B_h z^h, \tag{3.2.2}$$

where the *bit normalized weight enumerator*, B_h , is given by

$$B_h = \sum_{w=1}^{k} \frac{w}{k} A_{w,h}. \tag{3.2.3}$$

3.2.2 Serial Concatenation through a Uniform Interleaver

We now briefly review the serial concatenation of codes through a uniform random interleaver (URI). Using a URI is equivalent to averaging over all possible interleavers and was introduced for the analysis of turbo codes by Benedetto and Montorsi [4].

Consider the serial concatenation of an (n_1, k_1) outer code and an (n_2, k_2) inner code. Let the IOWEs of the two codes be $A^{(1)}_{w,h}$ and $A^{(2)}_{w,h}$, respectively. The average IOWE of the serial concatenation, $\overline{A}_{w,h}$ is given by

$$\overline{A}_{w,h} = \sum_{v=0}^{n_1} A^{(1)}_{w,v} \frac{A^{(2)}_{v,h}}{\binom{k_1}{v}} = \sum_{v=0}^{n_1} A^{(1)}_{w,v} P^{(2)}_{v,h}, \tag{3.2.4}$$

where

$$P^{(i)}_{w,h} = \frac{A^{(i)}_{w,h}}{\binom{k_i}{w}} \tag{3.2.5}$$

is known as the *input output weight transition probability* (IOWTP). This definition reflects the fact that $P^{(i)}_{w,h}$ is equal to the probability that this code will map a randomly chosen input sequence of weight w to an output of weight h.

Since the form of (3.2.4) with $P^{(2)}_{w,h}$ is essentially a matrix multiplication, the definition of the IOWTP makes a connection between linear algebra and serial concatenation. This was first observed in [23], where it was used to show that the WE of CA^m codes approaches that of a random code for large m .

3.2.3 Code Ensembles and Spectral Shape

In this section, we review code ensembles and spectral shape as defined in [1]. Let a *code ensemble* be a set, C, of (n, k) linear codes, each chosen with probability $1/|C|$. For any particular code, $\mathcal{C} \in C$, we group the codewords by weight and define $A_h(\mathcal{C})$ to be the number of codewords with output weight h and $A_{w,h}(\mathcal{C})$ to be the number of codewords of input weight w and output weight h. This allows the *average weight enumerator* to be defined as

$$\overline{A}_h(C) = \frac{1}{|C|} \sum_{\mathcal{C} \in C} A_h(\mathcal{C}),$$

the *average input-output weight enumerator* to be defined as

$$\overline{A}_{w,h}(C) = \frac{1}{|C|} \sum_{\mathcal{C} \in C} A_{w,h}(\mathcal{C}),$$

and the *average bit normalized weight enumerator* to be defined as

$$\overline{B}_h(C) = \frac{1}{|C|} \sum_{\mathcal{C} \in C} \sum_{w=1}^{k} \frac{w}{k} A_{w,h}(\mathcal{C}).$$

Finally, the *spectral shape* of an ensemble is defined to be

$$r(\delta; C) = \frac{1}{n} \ln \overline{A}_{\lfloor \delta n \rfloor}(C),$$

for $0 \leq \delta \leq 1$.

We also consider sequences, $\{C_{n_i}\}_{i \geq 0}$, of code ensembles, where each C_{n_i} is an ensemble of (n_i, k_i) codes. We assume that the sequences, $\{n_i\}_{i \geq 0}$ and $\{k_i\}_{i \geq 0}$, are unbounded and lead to a well-defined rate, $R = \lim_{i \to \infty}(k_i/n_i)$. This leads us to define the *spectral shape sequence*,

$$r_{n_i}(\delta; C) = \frac{1}{n_i} \ln \overline{A}_{\lfloor \delta n_i \rfloor}(C_{n_i}), \tag{3.2.6}$$

and the *asymptotic spectral shape*,

$$r(\delta; C) = \limsup_{i \to \infty} r_{n_i}(\delta; C). \tag{3.2.7}$$

In general, we will abuse our notation slightly by writing $\overline{A}_h(n)$ and $r_n(\delta)$ when it is clear which sequence of code ensembles in being considered. Furthermore, all limits taken as n goes to infinity are assumed to be along the subsequence $\{n_i\}_{i \geq 0}$.

Remark 3.2.1. It is worth considering the validity of the limit, (3.2.7). Suppose, we have a code ensemble where n_i is odd for all i and $\overline{A}_h(C_{n_i})$ is zero for odd h. It is eays to construct an ensemble sequence of regular low-density parity-check (LDPC) codes, with odd row weight, which has these properties. Choosing $\delta = 1/2$, we find that $\overline{A}_{\lfloor n_i/2 \rfloor}(C_{n_i}) = 0$ for all i, which means that $r(1/2, C) = -\infty$. In general, this is not a problem because one typically deals with a sequence of continuous functions, $f_{n_i}(h)$, which upper bound $\overline{A}_h(C_{n_i})$ at integer h. To avoid technical problems with the limit, however, one could define $f_{n_i}(h)$ to be the linear interpolation of the non-zero terms of $\overline{A}_h(C_{n_i})$. Let $h_{min}(n_i)$ be the smallest $h \geq 1$ such that $\overline{A}_h(C_{n_i}) > 0$ and let $h_{max}(n_i)$ be the largest $h \leq n_i$ such that $\overline{A}_h(C_{n_i}) > 0$. This allows the spectral shape to be defined as

$$r(\delta; C) = \limsup_{i \to \infty} \frac{1}{n_i} \ln f_{n_i}(\delta n_i)$$

for any $\delta_{min} \leq \delta \leq \delta_{max}$ where $\delta_{min} = \lim_{i \to \infty} h_{min}(n_i)$ and $\delta_{max} = \lim_{i \to \infty} h_{max}(n_i)$. For many codes, including turbo and LDPC codes, we believe that this $r(\delta; C)$ will also be continuous and differentiable for $\delta_{min} \leq \delta \leq \delta_{max}$.

Remark 3.2.2. Another problem with the definition of asymptotic spectral shape is that subsets of codes with exponentially vanishing probability may still affect the value of $r(\delta)$. We believe that

$$\tilde{r}(\delta; C) = \limsup_{i \to \infty} \frac{1}{|C_i|} \sum_{\mathcal{C} \in C_i} \frac{1}{n_i} \ln A_{\lfloor \delta n_i \rfloor}(\mathcal{C}).$$

may be a better definition of spectral shape because it does not have this problem. This is because $\frac{1}{n_i} \ln A_{\lfloor \delta n_i \rfloor}(\mathcal{C})$ is upper bounded by $(k_i/n_i) \ln 2$, so that subsets of codes with vanishing probability will contribute nothing to $\tilde{r}(\delta; C)$.

For many sparse graph codes, including turbo-like and LDPC codes, we also believe that $\tilde{r}(\delta; C)$ is the mean of a tightly concentrated probability distribution. Consider the probability,

$$P_i(\delta) = Pr\left(\left| \frac{1}{n_i} \ln A_{\lfloor \delta n_i \rfloor}(\mathcal{C}) - \tilde{r}(\delta, C) \right| > \epsilon \right),$$

when the code, $\mathcal{C}$, is chosen randomly from the ensemble, C_i. For any $0 \leq \delta \leq 1$ and any $\epsilon > 0$, we believe that $\lim_{i \to \infty} P_i(\delta) = 0$.

These observations are purely academic, however, because we know of no general method of computing $\tilde{r}(\delta; C)$. All may not be lost, however, because some physicists have started approximating this quantity using something known as the replica method [29]. Ironically, we note that the most straightforward approach to analyzing $\tilde{r}(\delta; C)$ is probably upper bounding it by $r(\delta; C)$, since the concavity of the logarithm implies that $\tilde{r}(\delta; C) \leq r(\delta; C)$.

3.2.4 Asymptotic Order of Functions

This chapter makes frequent use of the standard asymptotic notation, as defined in [19]. Specifically, the notation $O(\cdot)$, $\Omega(\cdot)$, $\Theta(\cdot)$, $o(\cdot)$, and $\omega(\cdot)$ is defined in the following manner. The expression $g(n) = O(f(n))$ means that there exist positive constants c and n_0, such that $g(n) \leq cf(n)$ for all $n \geq n_0$. Similarly, the expression $g(n) = \Omega(f(n))$ means that there exist positive constants c and n_0, such that $g(n) \geq cf(n)$ for all $n \geq n_0$. The term $g(n) = \Theta(f(n))$ combines these two and implies that that $g(n) = O(f(n))$ and $g(n) = \Omega(f(n))$. For strict bounds, we have the expressions $g(n) = o(f(n))$ and $g(n) = \omega(f(n))$ which mean that $\limsup_{n \to \infty} |g(n)/f(n)| = 0$ and $\limsup_{n \to \infty} |f(n)/g(n)| = 0$, respectively.

3.2.5 The IGE Conjecture

The Interleaver Gain Exponent (IGE) conjecture is based on the observations of Benedetto and Montorsi [4] and is stated rigorously in [10]. It was also considered for double serially concatenated codes in [3]. The conjecture considers the growth rate of $\overline{A}_h(n)$, for fixed h, for an ensemble sequence as i goes to infinity. Following [10], we define

$$\alpha(h) = \limsup_{n\to\infty} \log_n \overline{A}_h(n) \tag{3.2.8}$$

and

$$\beta_M = \max_{h\geq 1} \alpha(h). \tag{3.2.9}$$

Essentially, the IGE Conjecture [10] predicts that there exists a threshold channel parameter z^* such that, for any $z < z^*$, the probability of word error is $P_W = O\left(n^{\beta_M}\right)$. Another commonly cited variation of the IGE Conjecture also predicts that, under the same conditions, the probability of bit error is $P_B = O\left(n^{\beta_M - 1}\right)$.

This conjecture was first proven for repeat accumulate (RA) codes in [10], then extended to a range of more general turbo codes [9]. In this paper, the IGE conjecture for GRA^m codes is resolved in the affirmative by Theorem 3.6.4.

3.2.6 Noise Thresholds

Many modern coding systems exhibit a threshold behavior, whereby on one side of the threshold, the probability of decoding error is bounded away from zero, and on the other side of the threshold the probability of error approaches zero rapidly as the block length increases. In particular, most derivatives of turbo and LDPC codes, including CA^m codes, exhibit this behavior. In this section, we provide a framework for discussing this phenomenon, and the corresponding noise thresholds. We note that, in general, the threshold depends both on the code and the decoder.

Definition 3.2.3. Suppose we have a binary-input channel with parameter α, and a sequence of code ensembles, $\{C_i\}_{i\geq 0}$. Let $P_\bullet(C;\alpha)$ be the probability of a particular error type for a particular decoder. For example, one might write $P_{MLW}(C;\alpha)$ to represent the word error rate under ML decoding. The $P_\bullet$ *noise threshold*, $\alpha_\bullet$, of this ensemble sequence is the largest $\alpha.$ such

that

$$\limsup_{i\to\infty} P_{\bullet}(C_i;\alpha) = 0$$

for all $0 \leq \alpha \leq \alpha_{\bullet}$. Although $\alpha_{\bullet}$ is well-defined as long as $P_{\bullet}(C_i;0) = 0$, we will generally be dealing with $P_{\bullet}(C_i;\alpha)$ functions which are strictly increasing in α. Furthermore, we say that the ensemble has a $P_{\bullet}$ *decay rate* of at least $f(n)$ if we have $P_{\bullet}(C_i;\alpha) = O\left(f(n_i)\right)$ for all $0 \leq \alpha \leq \alpha_{\bullet}$. We also note that upper bounds on the probability of error can be used to provide lower bounds on the threshold, $\alpha_{\bullet}$.

The Bhattacharyya union bound, (3.2.1), can be used to derive lower bounds on the maximum likelihood word error noise threshold, c_{UB}. This approach was first used for turbo codes in [10]. While thresholds based on the union bound are generally quite pessimistic, the simplicity of the union bound enables one to analytically show the existence of noise thresholds for all channels simultaneously. The Bhattacharyya parameter threshold is given by $z^* = e^{-c_{UB}}$, where c_{UB} is

$$c_{UB} = \sup_{0\leq\delta\leq 1} \left(r(\delta;C)/\delta\right). \tag{3.2.10}$$

For the AWGN channel, the Viterbi-Viterbi Bound [31] is always tighter. In fact, it can be used to prove that the ensemble sequence achieves capacity as the rate approaches zero. The Viterbi-Viterbi E_s/N_0 threshold is given by

$$c_{VV} = \sup_{0\leq\delta\leq 1} \left((1-\delta)r(\delta;C)/\delta\right). \tag{3.2.11}$$

There are quite a number of other bounds for the AWGN channel, and [8][27] give nice overviews of the subject. In the next section, we discuss typical set decoding bounds which can be used on any memoryless symmetric channel and give quite good results.

3.2.7 Typical Set Decoding Bound

The typical set decoding bound on word error probability is very tight because it breaks the problem into two parts. First, it considers the probability that the noise is atypical. Second, it considers the probability of error given that the noise is typical. The probability of a memoryless channel having atypical noise decays rapidly with the block length, so we can essentially ignore

this probability. It turns out that the probability of error given typical noise lends itself to a very nice combinatorial analysis [1][16].

Consider a discrete memoryless symmetric channel with M outputs where p_i is the probability of the ith output given a zero input. Let the input to the channel be a sequence of n zeros, and assume that output statistics are collected by letting m_i be the number of times the ith output is observed.

Definition 3.2.4. For any $\epsilon > 0$, we say that the noise sequence is *typical* if $|m_i/n - p_i| \leq n^{-1/2+\epsilon}$ for $i = 1, \ldots, M$. We also say that any other output sequence is *jointly typical* with the all-zero sequence if its frequency statistics satisfy the same condition.

Definition 3.2.5. Consider the probability, $P_h(T_n; \alpha)$, that a codeword of weight h and length n is jointly typical with the all-zero codeword after being transmitted through a memoryless symmetric channel with parameter α. The *typical set decoding exponent*, $K(\delta, \alpha)$, is defined by

$$K(\delta, \alpha) = -\lim_{n\to\infty} \frac{1}{n} \ln P_{\lfloor \delta n \rfloor}(T_n; \alpha).$$

Lemma 3.2.6. *For any $\epsilon < 1/4$, there exists an n_0 such that for all $n \geq n_0$, the probability that the noise sequence is atypical is upper bounded by e^{-n^ϵ}.*

Proof. First, we notice that the distribution of each m_i is binomial with mean $p_i n$ and variance $np_i(1-p_i)$. Since the test for typicality allows variations in the frequency statistics of $O(n^{1/2+\epsilon})$ and the central limit theorem holds for variations of $o(n^{3/4})$, we can use Gaussian tail bounds for $\epsilon < 1/4$. Using the standard exponential bound for the Gaussian tail ($Q(x) \leq e^{-x^2/2}$), we see that the probability that any m_i fails the test is upper bounded by $2e^{-O(n^{2\epsilon})}$. Since all M bins must pass the test, the probability that a sequence is not typical is upper bounded by $2Me^{-O(n^{2\epsilon})}$. For large enough n, this can be upper bounded by e^{-n^ϵ}. □

Consider a sequence of code ensembles with average WE, $\overline{A}_h(n)$, spectral shape, $r_n(\delta)$, and asymptotic spectral shape, $r(\delta)$. The following conditions characterize the code ensemble well enough to give a fairly general coding theorem. We note that these results are taken mainly from [1].

Condition 3.2.7. There exists a sequence of integers, $\{L_n\}_{n\geq 1}$, and a function, $f(n)$, which satisfy $L_n = \omega(\ln n)$ and

$$\sum_{h=1}^{L_n - 1} \overline{A}_h(n) z^h = O\left(f(n)\right),$$

for any $z < 1$.

Condition 3.2.8. The spectral shape converges to the asymptotic spectral shape fast enough that

$$r_n(\delta; C) \leq r(\delta; C) + o\left(\frac{L_n}{n}\right)$$

and the behavior of $r(\delta)$ near zero is such that

$$\lim_{\delta \to 0^+} \frac{r(\delta; C)}{\delta} < \infty.$$

Now, consider any memoryless symmetric channel, with parameter α, whose Bhattacharyya parameter is $z(\alpha)$ and whose typical set decoding exponent is $K(\delta, \alpha)$. We define the *typical set decoding threshold* to be

$$\alpha_{TS} = \inf_{0<\lambda\leq 1} \alpha_{mix}(\lambda), \tag{3.2.12}$$

where

$$\alpha_{mix}(\lambda) = \sup\left\{\alpha \in \mathfrak{R}^+ | r(\delta; C)/\delta < -\ln z(\alpha), \delta \in [0, \lambda] \text{ and } r(\delta; C) < K(\delta, \alpha), \delta \in [\lambda, 1]\right\}.$$

Theorem 3.2.9 ([1]). *Suppose Conditions 3.2.7 and 3.2.8 hold. Let λ any real number in $(0, 1]$ and suppose also that the channel parameter α is greater than the threshold, $\alpha_{mix}(\lambda)$. In this case, there exists an $\epsilon > 0$ such that the probability of word error for the ensemble sequence, P_W, is given by*

$$P_W = O\left(f(n)\right) + O\left(ne^{-\epsilon L_n}\right) + O\left(e^{-n^\epsilon}\right). \tag{3.2.13}$$

In general, the first term will dominate but this also depends on the particular choice of L_n and $f(n)$.

Sketch of Proof. We start by breaking up the probability of word error with

$$P_W = P_W^{(UB)} + P_W^{(TS)},$$

where $P_W^{(UB)}$ is the contribution of the small output weights handled by the union bound and $P_W^{(TS)}$ is the contribution of the large output weights handled by the typical set bound. Using (3.2.1), we can write

$$P_W^{(UB)} \leq \sum_{h=1}^{L_n - 1} \overline{A}_h(n) z^h + \sum_{h=L_n}^{\lambda n} e^{h[r_n(h/n;C)/(h/n) + \ln z(\alpha)]},$$

for any $z < 1$. Condition 3.2.7 shows that first term is $O(f(n))$. Combining Condition 3.2.8 with the fact that $\alpha > \alpha_{mix}(\lambda)$, shows that there exists an n_0 and $\epsilon > 0$ such that $\sup_{0\leq\delta\leq\lambda} r_n(\delta;C)/\delta + \ln z(\alpha) \leq -\epsilon$ for $0 < \delta \leq \lambda_0$ and all $n \geq n_0$. Since the terms of the second sum are decreasing, we can upper bound the value by n times the first term or $O\left(ne^{-L_n\epsilon}\right)$.

Next, we write

$$P_W^{(TS)} \leq Pr(\text{noise atypical}) + n \max_{\lambda\leq\delta\leq 1} e^{n[r(\delta;C)-K(\delta,\alpha)+o(1)]},$$

and use Lemma 3.2.6 to show that $Pr(\text{noise atypical}) \leq O\left(e^{-n^\epsilon}\right)$ for some $\epsilon > 0$. If $\alpha > \alpha_{mix}(\lambda)$, then there also exists an n_0 and $\epsilon > 0$ such that $\sup_{\lambda\leq\delta\leq 1} r(\delta;C) - K(\delta,\alpha) \leq -\epsilon$ for all $n \geq n_0$. This means that the second term decays like $O\left(e^{-n\epsilon}\right)$ and can be ignored. Combining $P_W^{(UB)}$ and $P_W^{(TS)}$ completes the proof. □

Corollary 3.2.10. *Suppose the conditions of Theorem 3.2.9 hold, and that there also exists a $g(n) \leq f(n)$ such that*

$$\sum_{h=1}^{L_n-1} \overline{B}_h(n) z^h = O(g(n)),$$

for any $z < 1$, where $\overline{B}_h(n)$ is the bit normalized WE defined in (3.2.3). In this case, there exists an $\epsilon > 0$ such that the probability of bit error, P_B, is given by

$$P_B = O\left(g(n)\right) + O\left(ne^{-\epsilon L_n}\right) + O\left(e^{-n^\epsilon}\right).$$

Proof. The proof is identical to that of Theorem 3.2.9, except that (3.2.2) is used for the union bound portion of the bound. □

Remark 3.2.11. Since Theorem 3.2.9 essentially applies the union bound for $0 \leq \delta \leq \lambda$ and the typical set decoding bound for $\lambda \leq \delta \leq 1$, it is easy to see that separate spectral shapes could be used for each bound. For example, a simple upper bound on the spectral shape could be used for $0 \leq \delta \leq \lambda$, while numerical evaluation of the exact spectral shape and typical set decoding bound could be used for $\lambda \leq \delta \leq 1$. This would allow the typical set decoding threshold to be treated rigorously without considering Condition 3.2.8 for the exact spectral shape.

Remark 3.2.12. It is also worth noting that the quantity $\lim_{\delta\to 0^+}(r(\delta;C)/\delta)$, which equals $r'(0;C)$ by l'Hôpital's rule, seems to play an important role in noise thresholds. If $r'(0;C) < \infty$,

then a bit error rate noise threshold usually exists, while ensembles with $r'(0;C) = 0$ usually admit a word error rate threshold. Furthermore, if $r'(0;C) = 0$, then the noise threshold is usually determined by the typical set decoding bound (i.e., there exists a $\lambda_0 > 0$ such that $\alpha_{TS} = \sup_{\lambda_0 \leq \lambda \leq 1} \{\alpha | r(\delta;C) < K(\delta,\alpha), \lambda \leq \delta \leq 1\}$).

3.3 Terminated Convolutional Codes

In this section, we consider the WEs of terminated convolutional codes. In particular, we focus both on useful analytical bounds on the WE and exact numerical methods for computing the spectral shape of a CC. The analytical bound is a generalization of [18, Lemma 3], while the formula for the spectral shape can be seen as a generalization of Gallager's Chernov bounding technique [12, Eqn. 2.12] or as an application of [21].

3.3.1 Analytical Bounds

Now, we consider a useful bound on the weight enumerator of the block code formed by terminating a CC. This bound is essentially identical to [18, Lemma 3], which was proven for any rate-1/2 recursive systematic TCC. The major contribution of our result is that all constants are computable from the derivation. All previous derivations prove only the existence of bounds of this form. We also provide a proof which is valid for any TCC.

Theorem 3.3.1. *Let τ be the numbers of bits output by a CC per trellis step and consider the (n,k) block code formed by terminating a CC to a length of n/τ trellis steps. We denote the free distance of the CC by d, the transfer function of the CC by $T(D)$, and the smallest real positive root of the equation $T(D) = 1$ by D^*. The number of weight h codewords in the block code, $A_h^{(o)}(n)$, is upper bounded by*

$$A_h^{(o)}(n) \leq \sum_{t=1}^{\lfloor h/d \rfloor} \binom{n/\tau}{t} g^h, \tag{3.3.1}$$

where $g = 1/D^*$.

Furthermore, if a non-catastrophic convolutional encoder is used, then there exists a constant $\rho > 0$ such that the input weight, w, can be upper bounded with $w \leq \rho h$. In this case,

the bit normalized weight enumerator, $B_h^{(o)}$, can be upper bounded by

$$B_h^{(o)}(n) \leq \frac{\rho h}{n} \sum_{t=1}^{\lfloor h/d \rfloor} \binom{n/\tau}{t} g^h. \tag{3.3.2}$$

Proof. Proof of this theorem is provided in Appendix 3B.1. □

Various upper bounds can also be applied to the binomial sum in (3.3.1) to make this bound more useful. The next corollary bounds $A_h^{(o)}$ in a manner which makes it easy to upper bound $\sum A_h^{(o)} x^h$ by an exponential.

Corollary 3.3.2. *The binomial sum in (3.3.1) can be upper bounded with (3A.7) to get*

$$A_h^{(o)}(n) \leq \frac{(n/\tau + 1)^{\lfloor h/d \rfloor}}{\lfloor h/d \rfloor !} g^h, \tag{3.3.3}$$

where $g = 1/D^*$. If $\tau > d$, then this result also requires that $2^{1/\tau} g^{1.72d/\tau} \geq 2^R$ and $(de/\tau)^{1/d} \left(\sqrt{2\pi n}\right)^{-1/n} g \geq 2^R$, where R is the code rate.

If a non-catastrophic encoder is used, then the bit normalized weight enumerator, $B_h^{(o)}$, can also be upper bounded by

$$B_h^{(o)}(n) \leq C \frac{(n/\tau + 1)^{\lfloor h/d \rfloor - 1}}{(\lfloor h/d \rfloor - 1)!} g^h, \tag{3.3.4}$$

where $C = \frac{2\rho d}{\tau R} \frac{n+\tau}{n}$ and $g = 1/D^*$.

Proof. Proof of this corollary is provided in Appendix 3B.2. □

The bound presented in the next corollary was originally stated in [24] without proof. We present it here mainly because of this and because it follows easily from Theorem 3.3.1 and Corollary 3.3.2.

Corollary 3.3.3. *Using (3A.6) to upper bound the binomial sum instead, gives*

$$A_h^{(o)}(n) \leq C \left(\frac{n}{h}\right)^{\lfloor h/d \rfloor} g^h, \tag{3.3.5}$$

where $C = \left(\frac{\tau}{d}\right)^{(d-1)/d}$ and $g = \left(\frac{1}{D^*}\right) \left(\frac{de}{\tau}\right)^{1/d}$. If $\tau > d$, then this result also requires that $2^{1/\tau} g^{1.88d/\tau} \geq 2^R$ and $(de/\tau)^{1/d} g \geq 2^R$, where R is the code rate.

If a non-catastrophic encoder is used, then the bit normalized weight enumerator, $B_h^{(o)}$, can also be upper bounded by

$$B_h^{(o)}(n) \leq \frac{\rho}{R} \left(\frac{n}{h}\right)^{\lfloor h/d \rfloor - 1} g^h. \tag{3.3.6}$$

Proof. Proof of this corollary is provided in Appendix 3B.3. □

Remark 3.3.4. The basic ideas behind this theorem were introduced by Kahale and Urbanke in [18]. Their treatment, however, focused solely on rate-1/2 recursive systematic CCs. The generalization to arbitrary convolutional codes, (3.3.5), was given in [24] without proof. Recently, a bound similar to (3.3.1) was given without proof by Jin and McEliece in [17]. Using our notation, their result can be written as: there exists a g such that

$$A_h^{(o)} \leq \binom{n/\tau}{\lfloor h/d_{free}^{(o)} \rfloor} g^h.$$

Unfortunately, this bound does not hold for general convolutional codes. Consider, as a counterexample, the memory 0 CC formed by using a $(8, 4)$ Hamming code for each trellis step (i.e., $\tau = 8$ and $d_{free}^{(o)} = 4$). Choosing $h^* = n/2 + 4$ forces the binomial coefficient to 0 and results in the mistaken conclusion that $A_{h^*}^{(o)} \leq 0$, when in fact A_{h^*} is growing exponentially with n.

Remark 3.3.5. Consider the additional conditions required by Corollaries 3.3.2 and 3.3.3 for $\tau > d$. First, it is worth noting that we have not found any CCs which do not satisfy these conditions. Second, if a CC is found which does not satisfy these conditions, the parameter, g, can always be artificially inflated so that the conditions are satisfied. This results is a weaker, but provably accurate, bound of the same form. Furthermore, the constant, C, can also be removed by inflating g.

3.3.2 Analytical Bound Examples

Now, we consider three different TCCs and compare the true WE of each with (3.3.1) and (3.3.3), which are referred to as upper bound 1 and 2 respectively. In general, we see that (3.3.1) is tighter than (3.3.3) and that both bounds are reasonably tight for small output weights.

The (7,3) Hamming Code

This code can be thought of as a TCC with $\tau = 7$, $d = 3$, and $T(D) = 7D^3 + 7D^4 + D^7$. Solving the equation $T(D) = 1$ with Mathematica gives the result $D^* \approx 0.46012$. Figure 3.3.1 shows the WE of this code for $n = 1400$ and the corresponding bounds.

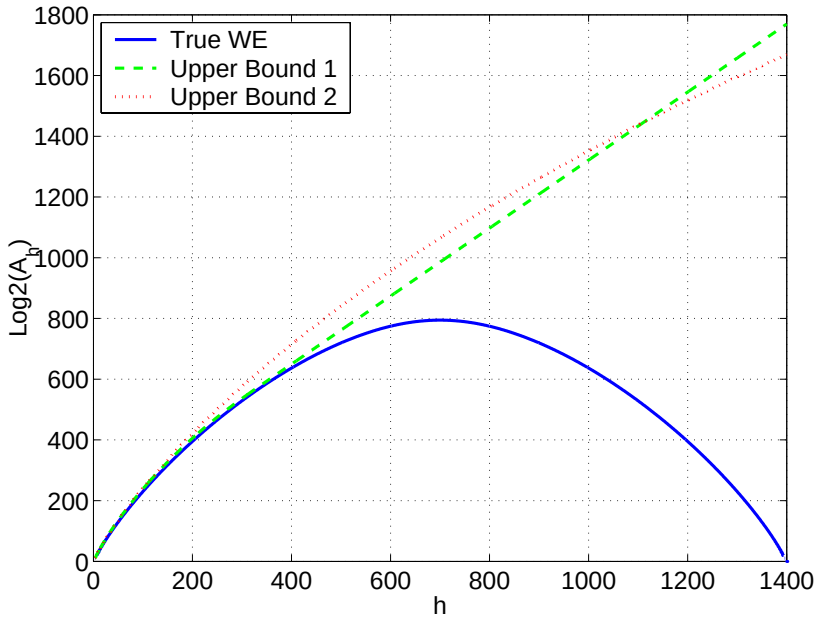


Figure 3.3.1: The true WE and upper bounds for the Hamming (7,3) code.

The (9,8) Single Parity Check Code

This code can be thought of as a TCC with $\tau = 9$, $d = 2$, and $T(D) = 36D^2 + 126D^4 + 84D^6 + 9D^8$. Solving the equation $T(D) = 1$ with Mathematica gives the result $D^* \approx 0.15959$. Figure 3.3.2 shows the WE of this code for $n = 1080$ and the corresponding bounds.

The Convolutional Code with Generator $G(D) = [1, 1 + D]$

This is really the only non-trivial memory-1 rate-1/2 CC, and it has parameters $\tau = 2$, $d = 3$, and $T(D) = D^3/(1 - D)$. Solving the equation $T(D) = 1$ with Mathematica gives the result $D^* \approx 0.68233$. Figure 3.3.3 shows the WE of this code for $n = 1400$ and the corresponding bounds. We note that this bound can also be computed by taking k trellis steps at a time (e.g., $\tau = 2k$). This has the effect of decreasing D^*, however, and the combination improves the bound only marginally.

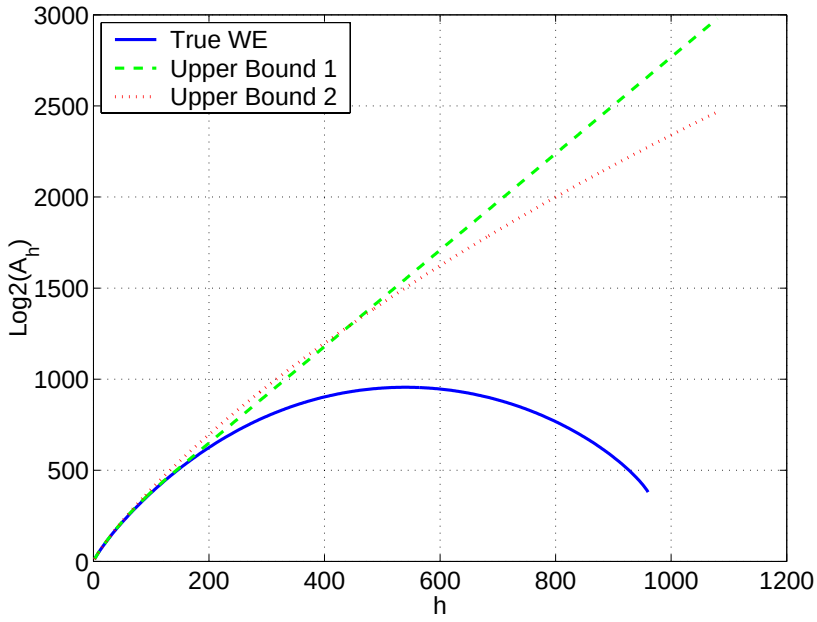


Figure 3.3.2: The true WE and upper bounds for the single parity check (9,8) CC.

3.3.3 The Exact Spectral Shape

In this section, we generalize the Chernov type WE bound of [12, Eqn. 2.12] to convolutional codes (CCs). A more general treatment of the underlying math problem was completed by Miller in [21]. Since the bound is exponentially tight, it enables the exact numerical computation of the spectral shape of block codes constructed from CCs. Furthermore, the spectral shape does not depend on the method of construction (e.g., truncation, termination, or tailbiting) used.

Theorem 3.3.6. *Let $\mathbf{G}(x)$ be the $M \times M$ state transition matrix of a CC which outputs τ symbols per trellis step. For example, we have*

$$\mathbf{G}(x) = \begin{bmatrix} 1 & x^2 \\ x & x \end{bmatrix}$$

for the two-state CC with generator matrix $[1, 1/(1+D)]$. If the the state diagram of the CC is irreducible and aperiodic, then we find that, for $x > 0$, the matrix $\mathbf{G}(x)$ has a unique eigenvalue, $\lambda_1(x)$, of maximum modulus. In this case, the spectral shape, $r(\delta; TCC)$, of the block code

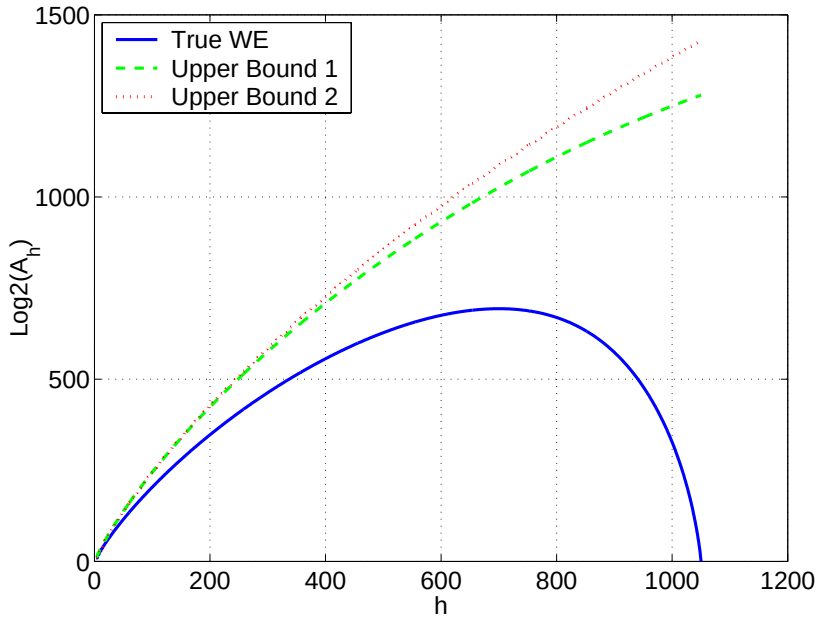


Figure 3.3.3: The true WE and upper bounds for the $G(D) = [1, 1 + D]$ CC.

formed by terminating the CC is given parametrically by $\delta(x) = x\lambda_1'(x)/(\tau\lambda_1(x))$ *and*

$$r(\delta(x); TCC) = \frac{1}{\tau} \ln [\lambda_1(x)] - \delta(x) \ln x. \tag{3.3.7}$$

Furthermore, both the function $r(\delta(x); TCC)$ *and the parametric curve are strictly convex.*

Proof. Proof of this theorem is provided in Appendix 3B.4. □

Remark 3.3.7. It also turns out that this formula can be evaluated numerically without resorting to numerical estimation of $\lambda_1'(x)$. Let the characteristic polynomial of $\mathbf{G}(x)$ be

$$f(\lambda, x) = \det(\lambda I - \mathbf{G}(x)) = \sum f_{ij} \lambda^i x^j,$$

and recall that the eigenvalues, for a particular x, are the roots of the equation, $f(\lambda, x) = 0$. Now, we can use implicit differentiation to solve for $d\lambda/dx$. We start by computing the differential form of $f(\lambda, x) = 0$, which is given by

$$\sum f_{ij}(i\lambda^{i-1}x^j d\lambda + j\lambda^i x^{j-1} dx) = 0.$$

Next, we solve for $d\lambda/dx$ as a function of λ and x to get

$$\frac{d\lambda}{dx} = \frac{-\sum_{ij} f_{ij} j\lambda^i x^{j-1}}{\sum_{ij} f_{ij} i\lambda^{i-1} x^j d\lambda}.$$

This allows a point on the $r(\delta;\text{TCC})$ curve to be computed by choosing any $x > 0$ and numerically computing the eigenvalue, $\lambda_1(x)$. Next, we compute the derivative, $d\lambda/dx$, for the (λ, x) pair and use (3.3.7) to compute $\delta(x)$ and $r(\delta(x);\text{TCC})$.

3.4 The Accumulate Code

In this section, we consider the "accumulate" code which is generated by a $1/(1+D)$ differential encoder.

3.4.1 A Simple Bound on the IOWTP

In this section, we consider the IOWTP of the "accumulate" code. The exact IOWE of the "accumulate" code was published first in [10] and [22], and this allows the IOWTP to be written as

$$P_{w,h}(n) = \begin{cases} \frac{\binom{n-h}{\lfloor w/2 \rfloor}\binom{h-1}{\lceil w/2 \rceil - 1}}{\binom{n}{w}} 1 & w \geq 1 \text{ and } h \geq 1 \\ 1 & w = h = 0 \\ 0 & \text{otherwise} \end{cases} . \tag{3.4.1}$$

It is also worth noting that the "accumulate" code never maps an input word of weight w to an output word of weight $h < \lceil w/2 \rceil$. This property is quite useful, so we summarize it in the following condition.

Fact 3.4.1. *Consider the IOWTP of the "accumulate" code, $P_{w,h}(n)$, for $w \geq 1$ and $h \geq 1$. In this case, $P_{w,h}(n)$ is non-zero if and only if $h \geq \lceil w/2 \rceil$ and $n - h \geq \lfloor w/2 \rfloor$. This can be seen easily by noticing that one of the binomial coefficients in the numerator of (3.4.1) will be zero if either condition is not met.*

Now, we derive a new upper bound on the IOWTP of the "accumulate" code. This bound is quite useful in analysis because of its simplicity, yet it is also tight enough to reproduce various qualitative results for RA codes. The result is presented as a corollary of Theorem 3C.2, which is stated and proven in Appendix 3C.

Corollary 3.4.2. *The IOWTP of the "accumulate" code, $P_{w,h}(n)$, is upper bounded by*

$$P_{w,h}(n) \leq \frac{\lceil w/2 \rceil}{h} 2^w \left(\frac{h}{n}\right)^{\lceil w/2 \rceil} \left(\frac{n-h}{n}\right)^{\lfloor w/2 \rfloor} \tag{3.4.2}$$

and

$$P_{w,h}(n) \leq 2^w \left(\frac{h}{n}\right)^{\lceil w/2 \rceil} \left(\frac{n-h}{n}\right)^{\lfloor w/2 \rfloor}. \tag{3.4.3}$$

While some care should be taken when applying this bound with $w = 0$, $h = 0$, or $h = n$, we note that using the definition $0^0 = 1$ makes the bound valid for $0 \leq w \leq n$ and $0 \leq h \leq n$.

Proof. Proof of this corollary is provided in Appendix 3C.2. □

3.4.2 An Exponentially Tight Bound on the IOWTP

The exact exponential form of $P_{w,h}(n)$ is very useful for computing tight numerical bounds on the WE of codes based on the "accumulate" code. It is defined by

$$\begin{aligned} p(x,y) &= \lim_{n\to\infty} \frac{1}{n} \log P_{\lfloor xn \rfloor, \lfloor yn \rfloor}(n) \\ &= yH\left(\frac{x}{2y}\right) + (1-y)H\left(\frac{x}{2(1-y)}\right) - H(x), \end{aligned} \tag{3.4.4}$$

and the limit can be evaluated by using the upper and lower bounds given by (3A.2). When the argument of any entropy function is greater than one, the true value of $p(x,y)$ is negative infinity. This can be seen by applying Fact 3.4.1 to see $\lim_{n\to\infty} P_{\lfloor xn \rfloor, \lfloor yn \rfloor}(n) = 0$ if $y < x/2$ or $y > 1 - x/2$.

Remark 3.4.3. It turns out that there is a remarkable similarity between (3.4.3) and the Bhattacharyya bound on pairwise error probability for the BSC, which is given by $(4p(1-p))^{h/2}$. This might seem accidental at first, but we believe that there is something deeper to this connection. In fact, the exponential form of the IOWTP of the "accumulate" code, (3.4.4), and the typical set decoding exponent for the BSC, [1, Eqn. 2.8], are actually identical.

The fact that these two quantities are mathematically identical has at least one very interesting consequence. Suppose that we have any ensemble sequence whose noise threshold for typical set decoding on the BSC is p^*. If we serially concatenate this code with an interleaved "accumulate" code, then the typical minimum distance of the new ensemble will be p^*n. This observation is based on the fact that the BSC typical set decoding threshold and this typical minimum distance are both given by the same expression. Namely, they are both given by the smallest $\delta > 0$ which satisfies $\max_x r(x) + p(x,\delta) = 0$, where $r(\delta)$ is the spectral shape of the ensemble sequence and $p(x,y)$ is given by (3.4.4).

3.4.3 A Simple Bound on the Cumulative IOWTP

Now, we derive a new upper bound on the cumulative IOWTP (CIOWTP) of the "accumulate" code. This bound is quite useful in analysis because of its simplicity, yet it is also tight enough to reproduce various qualitative results for RA codes. The result is presented as a corollary of Theorem 3C.2, which is stated and proven in Appendix 3C.

Corollary 3.4.4. *The CIOWTP of the "accumulate" code, $P_{w,\leq h}(n)$, is defined by*

$$P_{w,\leq h}(n) = \sum_{i=0}^{h} P_{w,i}(n) = \begin{cases} \frac{\sum_{i=1}^{h} \binom{n-h}{\lfloor w/2 \rfloor} \binom{h-1}{\lceil w/2 \rceil - 1}}{\binom{n}{w}} 1 & w \geq 1 \text{ and } h \geq 1 \\ 1 & h \geq w = 0 \\ 0 & w > h = 0 \end{cases} .$$

This quantity can be upper bounded with

$$P_{w,\leq h}(n) \leq 2^w \left(\frac{h}{n} \right)^{\lceil w/2 \rceil} . \tag{3.4.5}$$

Using the definition $0^0 = 1$ makes the bound valid for $0 \leq w \leq n$ and $0 \leq h \leq n$.

Proof. Proof of this theorem is provided in Appendix 3C.3. □

Corollary 3.4.5. *The CIOWTP of the cascade of m "accumulate" codes, $P^{(m)}_{w,\leq h}(n)$, is upper bounded by*

$$P^{(m)}_{w,\leq h}(n) \leq \frac{2^{m-1} \left(\frac{2^{m+1} h}{n} \right)^{\sum_{i=1}^{m} \lceil w/2^i \rceil}}{\left(1 - \frac{2^{m+1} h}{n} \right)^{m-1}}, \tag{3.4.6}$$

for $h < n/2^{m+1}$.

Proof. Proof of this corollary is provided in Appendix 3C.4. □

Remark 3.4.6. The upper bound provided by Corollary 3.4.5 is actually quite loose, but it suffices for our purposes. We believe the weakness is mainly due to the fixed upper bound $h_i \leq 2^m h_{m+1}$ for $i = 1, \ldots, m$ used to derive it.

3.5 Single Accumulate Codes

3.5.1 Repeat Accumulate Codes

A Repeat Accumulate (RA) code is the serial concatenation of a repeat code and an interleaved rate-1 "accumulate" code. The elegant simplicity of these codes allowed their inventors, Divsalar, Jin and McEliece, to rigorously prove a coding theorem in [10]. In this section, we derive a new closed form bound on the WE of an RA code with repeat order q. The quality and simplicity of this new bound is mainly due to the new bound on the IOWTP of the "accumulate" code given by (3.4.3).

Starting with the general formula for serial concatenation,

$$\overline{A}_h^{\text{RA}}(n) = \sum_{w=1}^{n} A_w^{(o)}(n) P_{w,h}(n),$$

we can substitute the WE of the repeat code,

$$A_h^{(o)}(n) = \begin{cases} \binom{n/q}{h/q} & \text{if } h/q \text{ integer} \\ 0 & \text{otherwise} \end{cases},$$

and apply (3.4.3) to get

$$\overline{A}_h^{\text{RA}}(n) \leq \sum_{i=1}^{n/q} \binom{n/q}{i} 2^{qi} (h/n)^{\lceil qi/2 \rceil} (1-h/n)^{\lfloor qi/2 \rfloor}.$$

Next we define $\delta = h/n$ to normalize the output weight and simplify the notation. For q even, the binomial theorem can be used to simplify this sum to

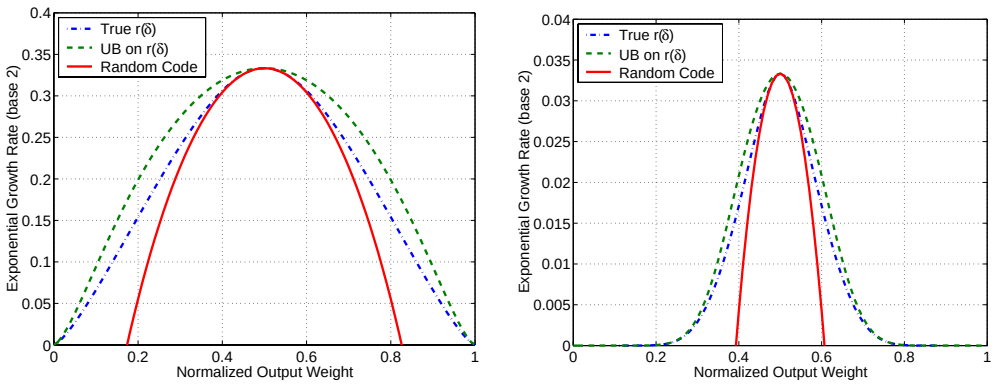
$$\begin{aligned} \overline{A}_{\delta n}^{\text{RA}}(n) &\leq \sum_{i=1}^{n/q} \binom{n/q}{i} \left(2^q \delta^{q/2} (1-\delta)^{q/2}\right)^i \\ &= \left(1 + (4\delta(1-\delta))^{q/2}\right)^{n/q} - 1. \end{aligned} \tag{3.5.1}$$

For q odd, we can sum the odd and even terms separately by defining the function

$$Z^{\pm}(x,k) = \frac{(1+x)^k \pm (1-x)^k}{2},$$

since $Z^+(x,k)$ gives even terms in a binomial sum and $Z^-(x,k)$ gives the odd terms in a binomial sum. Using this, we write

$$\overline{A}_{\delta n}^{\text{RA}}(n) \leq Z^+\left((4\delta(1-\delta)^{q/2}, n/q\right) - 1 + \frac{\delta}{1-\delta} Z^-\left((4\delta(1-\delta)^{q/2}, n/q\right). \tag{3.5.2}$$



(a) $q = 3$ (b) $q = 30$

Figure 3.5.1: An upper bound on the spectral shape of RA codes.

Applying (3.2.7) to (3.5.1) and (3.5.2), it is easy to verify that the asymptotic spectral shape of an RA code is upper bounded by

$$r^{(q)}(\delta; \mathrm{RA}) \leq \frac{1}{q} \ln \left(1 + (4\delta(1-\delta))^{q/2} \right) \tag{3.5.3}$$

for $q \geq 2$ and $0 \leq \delta \leq 1$. Figure 3.5.1 compares the actual spectral shape of two RA codes with the upper bounds. For $q = 30$, one can see that the upper bound matches the true spectral shape very well for $\delta < 0.3$. While, for $q = 3$, the bound matches only for very small δ.

3.5.2 Convolutional Accumulate Codes

A Convolutional Accumulate (CA) code is the serial concatenation of a terminated convolutional code with an interleaved rate-1 "accumulate" code. These codes generally perform well with iterative decoding and have very good ML decoding thresholds. Their discovery in [11] actually predates RA codes as well. In this section, we derive a general upper bound on the WE of a CA which captures some of the important properties of CA codes.

Starting with the general formula for serial concatenation,

$$\overline{A}_h^{\mathrm{CA}}(n) = \sum_{i=d}^{n} A_i^{(o)}(n) P_{i,h}^{(acc)}(n),$$

we can derive an upper bound on the WE of a CA code. Using (3.3.3) to upper bound the WE

of the CC and (3.4.3) to upper bound the IOWTP of the "accumulate" code gives

$$\overline{A}_h^{\text{CA}}(n) \leq \sum_{i=d}^{\infty} \frac{(n/\tau+1)^{\lfloor i/d \rfloor}}{\lfloor i/d \rfloor!} g^i \frac{\lceil i/2 \rceil}{h} 2^i (h/n)^{\lceil i/2 \rceil} (1-h/n)^{\lfloor i/2 \rfloor}.$$

Using the normalized output weight, $\delta = h/n$, and the upper bound,

$$\delta^{\lceil i/2 \rceil}(1-\delta)^{\lfloor i/2 \rfloor} \leq \delta^{i/2}(1-\delta)^{i/2}/(1-\delta),$$

gives

$$\overline{A}_{\delta n}^{\text{CA}}(n) \leq \frac{1}{1-\delta} \sum_{i=d}^{\infty} \frac{(n/\tau+1)^{\lfloor i/d \rfloor}}{\lfloor i/d \rfloor!} \left(g\sqrt{4\delta(1-\delta)} \right)^i.$$

Now, we define $\gamma = g\sqrt{4\delta(1-\delta)}$ and simplify the expression to

$$\overline{A}_{\delta n}^{\text{CA}}(n) \leq \frac{1}{1-\delta} \left(1 + \gamma + \ldots + \gamma^{d-1}\right) \sum_{j=1}^{\infty} \frac{(n/\tau+1)^j}{j!} \gamma^{dj}.$$

Finally, we can write the infinite sum in closed form and use the fact that $\left(1 + \gamma + \ldots + \gamma^{d-1}\right) = (\gamma^d - 1)/(\gamma - 1)$ to get

$$\overline{A}_{\delta n}^{\text{CA}}(n) \leq \frac{1}{1-\delta} \frac{\gamma^d - 1}{\gamma - 1} \left(e^{\gamma^d (n+\tau)/\tau} - 1 \right). \tag{3.5.4}$$

We can also upper bound the spectral shape using (3.2.7) and (3.5.4) to get

$$r(\delta; \text{CA}) \leq \frac{1}{\tau} \left(g\sqrt{4\delta(1-\delta)} \right)^d.$$

3.5.3 Properties of the Bounds

Although the upper bounds, (3.5.1), (3.5.2), and (3.5.4), computed in this section are quite loose in some cases, they do capture some important characteristics of the underlying WEs. For example, we will show that they correctly characterize the α in the growth rate of the minimum distance, $d_{min} \sim n^{\alpha}$. This fact is a straightforward generalization of the well-known result given in [18]. We will also show that (3.5.3) is tight enough to prove that the ML AWGN threshold of an RA code approaches -1.59 dB as q goes to infinity. This fact was originally proven in [15].

Since the only difference between (3.4.5) and (3.4.3) is the factor of $(1-h/n)^{\lfloor w/2 \rfloor}$, it is straightforward to repeat the derivation using (3.4.3) and one finds that the upper bound on the

WE is converted to an upper bound on the cumulative WE simply by dropping the $(1-h/n)^{\lfloor w/2 \rfloor}$ term. Applying this technique to (3.5.4) and substituting h/n for δ gives

$$\overline{A}^{\text{CA}}_{\leq h}(n) \leq \frac{1}{1-h/n} \frac{(2g\sqrt{h/n})^d - 1}{(2g\sqrt{h/n}) - 1} \left(e^{(2g\sqrt{h/n})^d (n+\tau)/\tau} - 1 \right). \tag{3.5.5}$$

The probability that a randomly chosen code from this ensemble will have a minimum distance less than t is upper bounded by $\overline{A}^{\text{CA}}_{\leq t}(n)$ [23, Theorem 4]. Let E be the event that a very long code from this ensemble has a minimum distance less than $t(n) = an^{(d-2)/d}$, for some constant a. We can upper bound the probability of E by considering $\overline{A}^{\text{CA}}_{\leq t(n)}(n)$ as n goes to infinity, which gives

$$\begin{aligned} Pr(E) &\leq \lim_{n\to\infty} \frac{1}{1-an^{(d-2)/d}/n} \frac{(2g\sqrt{an^{(d-2)/d}/n})^d - 1}{(2g\sqrt{an^{(d-2)/d}/n}) - 1} \left(e^{(2g\sqrt{an^{(d-2)/d}/n})^d (n+\tau)/\tau} - 1 \right) \\ &= e^{(4g\sqrt{a})^d/\tau} - 1. \end{aligned}$$

It is easy to see that this upper bound can be made arbitrarily close to zero by decreasing a. Therefore, almost all of the codes in the ensemble will have a minimum distance which grows like $n^{(d-2)/d}$.

Now, let us consider the ML decoding threshold of an RA code in AWGN by applying Viterbi-Viterbi bound. It was shown in [15], using a great deal of analysis, that this threshold approaches -1.59 dB (i.e., the low-rate Shannon limit) as q goes to infinity. It turns out that (3.5.3) is tight enough to reproduce the same result almost trivially. Substituting (3.5.3) into (3.2.11) and normalizing for the rate (i.e., multiplying by q) shows that the Viterbi-Viterbi E_b/N_0 threshold of a rate-$1/q$ RA code is given by

$$T_q = \max_{0 \leq \delta \leq 1} f_q(\delta),$$

where

$$f_q(\delta) = \frac{(1-\delta)}{\delta} q r^{(q)}(\delta; \text{RA}) = \frac{(1-\delta)}{\delta} \ln \left(1 + (4\delta(1-\delta))^{q/2} \right).$$

Since we are interested in the limit of T_q as q goes to infinity, we start by noting that, for $\delta \in [0, 1/2) \bigcup (1/2, 1]$, $f_q(\delta)$ decreases strictly to 0 as q increases (i.e., $f_q(\delta) > 0$ implies that $f_{q+1}(\delta) < f_q(\delta)$ for all $\delta \in [0, 1/2) \bigcup (1/2, 1]$). This implies that $\lim_{q\to\infty} T_q \leq \lim_{q\to\infty} f_q(1/2)$. Furthermore, it is easy to see that $\lim_{q\to\infty} T_q \geq \lim_{q\to\infty} f_q(1/2)$ because we can lower bound the maximum over an interval by choosing any point inside. Combining these two results shows that $T_\infty = \lim_{q\to\infty} f_q(1/2) = \ln 2 = -1.59\, dB$.

3.6 Convolutional Accumulate-m Codes

3.6.1 Description

A CA^m code is the multiple serial concatenation of a TCC and m interleaved rate-1 "accumulate" codes [24]. Any CA^m code is completely defined by its outer TCC, and its m interleavers. Therefore, a random ensemble of CA^m codes is formed, for a particular outer TCC, by choosing each interleaver randomly from the set of all permutations. This type of ensemble lends itself nicely to the average analysis introduced by [4] for turbo codes. Let $\overline{A}_h^{(i+1)}(n)$ be the ensemble averaged WE after the ith "accumulate" code, then we have

$$\overline{A}_{h_{m+1}}^{(m+1)}(n) = \sum_{h_1,\ldots,h_m} A_{h_1}^{(1)}(n) \prod_{i=1}^{m} P_{h_i,h_{i+1}}(n), \tag{3.6.1}$$

where $P_{w,h}(n)$ is given by (3.4.1) and $\overline{A}_h^{(1)}$ equals the WE of the outer TCC, $A_h^{(o)}$. This WE can also be written in an incremental form,

$$\overline{A}_{h_{i+1}}^{(i+1)}(n) = \sum_{h_i=1}^{n} \overline{A}_{h_i}^{(i)}(n) P_{h_i,h_{i+1}}(n), \tag{3.6.2}$$

which highlights the Markov nature of the serial concatenation.

Definition 3.6.1. The tuple, $h_1, \ldots, h_{m+1}$, corresponds to the codeword weight at each stage through the $m+1$ encoders. We refer to this tuple as a *weight path* through the encoders. Using this definition, one can think of (3.6.1) as a sum over all weight paths. Furthermore, we say that a weight path is valid if it does not violate basic conditions such as Fact 3.4.1. For example, the weight path, $h_1, \ldots, h_{m+1}$, is valid if $h_1 \geq d$ and $h_{i+1} \geq \lceil h_i/2 \rceil$ for $i = 1, \ldots, m-1$. All weight paths which are not valid provide no contribution to the sum.

3.6.2 The IGE Conjecture for CA^m Codes

Now, we can apply the IGE conjecture to (3.6.1) by defining

$$\alpha(h_{m+1}) = \limsup_{n\to\infty} \left(\log_n \sum_{h_1,\ldots,h_m} A_{h_1}^{(1)}(n) \prod_{i=1}^{m} P_{h_i,h_{i+1}}(n) \right). \tag{3.6.3}$$

Of course, the sum in (3.6.3) is lower bounded by its largest term. Using Definition 3.6.1, it is easy to verify that all valid weight paths ending at h_{m+1} obey $h_i \leq 2^m h_{m+1}$ for $i = 1, \ldots m$.

This means that the number of non-zero terms in the sum is upper bounded by $(2^m h_{m+1})^m$, and that

$$\sum_{h_1,\dots,h_m} A_{h_1}^{(1)}(n) \prod_{i=1}^{m} P_{h_i,h_{i+1}}(n) \leq (2^m h_{m+1})^m \max_{h_1,\dots,h_m} A_{h_1}^{(1)}(n) \prod_{i=1}^{m} P_{h_i,h_{i+1}}(n).$$

These upper and lower bounds, along with fact that $\lim_{n\to\infty} \log_n (2^m h_{m+1})^m = 0$, for fixed h_{m+1}, allow us to replace the sum over weight paths in (3.6.3) by a maximum over weight paths. The results of Appendix 3D.1 show that

$$\lim_{n\to\infty} \left(\log_n A_{h_1}^{(1)}(n) \prod_{i=1}^{m} P_{h_i,h_{i+1}}(n) \right) \leq \alpha(h_1, \dots, h_{m+1})$$

where $\alpha(h_1, \dots, h_{m+1}) = \lfloor h_1/d \rfloor - \sum_{i=1}^{m} \lceil h_i/2 \rceil$. We also note that the bound holds with eqaulity if h_1 is an integer multiple of d. This implies only that $\alpha(h_{m+1})$ will be upper bounded by the maximum of $\alpha(h_1, \dots, h_{m+1})$ over all valid weight paths. In fact, we will find that $\alpha(h_{m+1})$ is equal to this quantity because the maximum occurs when h_1 is an integer multiple of d.

The following Lemma provides a few results on the maximization of $\alpha(h_1, \dots, h_{m+1})$.

Lemma 3.6.2. *Let the set of valid paths starting at h_1, $V(h_1)$, be the set of all tuples, $h_1, \dots, h_{m+1}$, where $h_i > 0$ for $i = 1, \dots m+1$ and $h_{i+1} \geq \lceil h_i/2 \rceil$ for $i = 1, \dots, m-1$. Let the function, $\alpha(h_1, \dots, h_{m+1})$, be defined by*

$$\alpha(h_1, \dots, h_{m+1}) = \lfloor h_1/d \rfloor - \sum_{i=1}^{m} \lceil h_i/2 \rceil .$$

The maximum of $\alpha(h_1, \dots, h_{m+1})$ over the set $V(h_1)$ with $h_1 \geq 2$ is equal to

$$\nu(h_1) = \lfloor h_1/d \rfloor - \sum_{i=1}^{m} \lceil h_1/2^i \rceil . \tag{3.6.4}$$

Also, the maximum of $\nu(h_1)$ for $h_1 \geq d \geq 2$ is equal to $\nu(d)$. Finally, for $d \geq 3$ or $m \geq 2$, we also show that $\nu(h) \leq \nu(d) - 1$ for all $h \geq 4d$.

Proof. Proof of this lemma is given in Appendix 3D.2. □

Since $\alpha(h_1, \dots, h_{m+1})$ does not depend on h_{m+1}, we can apply Lemma 3.6.2 to show that $\alpha(h_{m+1}) = \nu(d)$. Furthermore, it is clear that $\beta_M = \max_{h_{m+1} \geq 1} \alpha(h_{m+1}) = \nu(d)$, so the maximum exponent, ν, is given by $\nu = \nu(d)$ or

$$\nu = 1 - \sum_{i=1}^{m} \lceil d/2^i \rceil . \tag{3.6.5}$$

3.6.3 The Worst Case Minimum Distance

Using Fact 3.4.1, we can compute the minimum possible output weight, d_{min}, of a GRA^m code. This worst case minimum distance is found by minimizing h_{m+1} subject to the constraints that $h_{i+1} \geq \lceil h_i/2 \rceil$ and $h_1 \geq d$. It is easy to see that picking h_1 as small as possible allows us to pick h_2 as small as possible, and so on. Therefore, the weight path which minimizes h_{m+1} is given by $h_1 = d$ and $h_{i+1} = \lceil h_i/2 \rceil$. One might notice from the previous section that this weight path also maximizes the exponent of the IGE conjecture. Simplifying the expression for h_{m+1} gives

$$d_{min} = \lceil d/2^m \rceil \,. \tag{3.6.6}$$

3.6.4 Weight Enumerator Bound

In this section, we derive an upper bound on the cumulative WE of a CA^m which will be used to prove the main theorem of the chapter, Theorem 3.6.4. The cumulative WE of a CA^m code can be written in terms of the WE of the outer TCC and the CIOWTP of m cascaded "accumulate" codes with

$$\overline{A}_{\leq h}^{(m+1)}(n) = \sum_{w=1}^{n} A_w(n) P_{w,\leq h}^{(m)}.$$

For $h \leq n/2^{m+1}$, this can be upper bounded by using (3.3.5) and (3.4.6) to get

$$\overline{A}_{\leq h}^{(m+1)}(n) \leq \frac{2^{m-1}}{\left(1 - \frac{2^{m+1}h}{n}\right)^{m-1}} \sum_{w=d}^{2^m h} \frac{(n/\tau + 1)^{\lfloor w/d \rfloor}}{\lfloor w/d \rfloor !} g^w \left(\frac{2^{m+1}h}{n}\right)^{\sum_{i=1}^{m} \lceil w/2^i \rceil} . \tag{3.6.7}$$

We note that the upper limit, $2^m h$, of the sum is due to the fact that $P_{w,\leq h}^{(m)} = 0$ for $w \geq 2^m h$.

For the next step, we need the bound $\sum_{i=1}^{m} \lceil w/2^i \rceil \geq d(1 - 2^{-m}) \lfloor w/d \rfloor$, which is easily verified by noticing that

$$\sum_{i=1}^{m} \lceil w/2^i \rceil \geq w \sum_{i=1}^{m} 2^{-i} = w(1 - 2^{-m})$$

and $w \geq d \lfloor w/d \rfloor$. Using this bound, we can write the cumulative WE as

$$\overline{A}_{\leq h}^{(m+1)}(n) \leq \frac{2^{m-1}}{\left(1 - \frac{2^{m+1}h}{n}\right)^{m-1}} \sum_{w=d}^{2^m h} \frac{(n/\tau + 1)^{\lfloor w/d \rfloor}}{\lfloor w/d \rfloor !} g^w \left(\frac{2^{m+1}h}{n}\right)^{c\lfloor w/d \rfloor} ,$$

where $c = d(1-2^{-m})$. Now, we can change the index of summation from w to $i = \lfloor w/d \rfloor$ and extend the upper limit of the sum to get

$$\overline{A}_{\leq h}^{(m+1)}(n) \leq \frac{2^{m-1}}{\left(1-\frac{2^{m+1}h}{n}\right)^{m-1}} \left(1+g+\ldots+g^{d-1}\right) \sum_{i=1}^{\infty} \frac{(n/\tau+1)^i}{i!} g^{di} \left(\frac{2^{m+1}h}{n}\right)^{ci}.$$

Evaluating the sum and applying the identity, $\sum_{w=0}^{d-1} g^w = (g^d-1)/(g-1)$, gives

$$\overline{A}_{\leq h}^{(m+1)}(n) \leq \frac{2^{m-1}}{\left(1-\frac{2^{m+1}h}{n}\right)^{m-1}} \frac{g^d-1}{g-1} \left(e^{g^d(2^{m+1}h/n)^c(n+\tau)/\tau}-1\right), \tag{3.6.8}$$

for $h < n/2^{m+1}$. Writing the logarithm of the cumulative WE as

$$\ln \overline{A}_{\leq h}^{(m+1)}(n) \leq O(1) + \frac{n}{\tau} g^d (2^{m+1}h/n)^{d(1-2^{-m})}, \tag{3.6.9}$$

for $h < n/2^{m+1}$, makes it easy to see that the spectral shape is given by

$$r^{(m+1)}(\delta; \mathrm{CA}^m) \leq \frac{1}{\tau} g^d (2^{m+1}\delta)^{d(1-2^{-m})}, \tag{3.6.10}$$

for $\delta < 1/2^{m+1}$.

3.6.5 The Main Theorem

Almost all of the pieces are now in place to consider the main theorem of the chapter. Before continuing, however, with the statement of the main theorem, we state the following lemma, which will be used in its proof.

Lemma 3.6.3. *Consider the serial concatenation of a TCC, with free distance d, and an "accumulate" code. The probability that the resulting code has a codeword of minimum weight (i.e., $h = \lceil d/2 \rceil$) is $P_M(n) = \Theta(n^{1-\lceil d/2 \rceil})$ where n is the block length.*

Proof. Proof of this lemma is given in Appendix 3D.3. □

The following theorem is the main theorem of the chapter and essentially extends the results of [10][14] to CA^m codes.

Theorem 3.6.4. *Consider the average performance of a sequence of CA^m code ensembles, based on a particular outer TCC with minimum distance $d \geq 2$, transmitted over a memoryless channel with Bhattacharyya channel parameter z. There exists a positive threshold z^* such that, for any $z < z^*$, the probability of word error under maximum likelihood decoding is $P_W = \Theta(n^\nu)$, where $\nu = 1 - \sum_{i=1}^{m} \lceil d/2^i \rceil$. Furthermore, if a non-catastrophic encoder is used for the CC, then the probability of bit error is $P_B = \Theta(n^{\nu-1})$.*

Proof. The proof can broken into four main parts. The first part uses (3.6.7) to verify that the WE of a CA^m code satisfies Condition 3.2.7. This also includes finding the error decay rates, which are $P_W = O(n^\nu)$ and $P_B = O(n^{\nu-1})$. The second part uses the upper bound, (3.6.9), to verify that the WE of a CA^m code satisfies Condition 3.2.8. The third part uses Theorem 3.2.9 and Corollary 3.2.10 to establish the basic coding theorem. The final part uses Lemma 3.6.3 to lower bound the probability of error and establish that $P_W = \Omega(n^\nu)$ and $P_B = \Omega(n^{\nu-1})$.

First, we choose $L_n = (\ln n)^2$ and verify that Condition 3.2.7 holds. To do this, we consider an upper bound on cumulative WE, (3.6.7), for small output weights ($h = L_n$). In this case, we can upper bound (3.6.7) by $2^m h$ times the largest term to get

$$\overline{A}_{\leq h}^{(m+1)}(n) \leq \frac{2^{2m-1}h}{\left(1 - \frac{2^{m+1}h}{n}\right)^{1-m}} \max_{d \leq w \leq 2^m h} \frac{(n/\tau + 1)^{\lfloor w/d \rfloor}}{\lfloor w/d \rfloor !} g^w \left(\frac{2^{m+1}h}{n}\right)^{\sum_{i=1}^{m} \lceil w/2^i \rceil} . \tag{3.6.11}$$

It should be clear that the exponent of n in this expression plays the crucial role for large n and $h = O\left((\ln n)^2\right)$. This exponent is the same as that given in the IGE conjecture with the help of Lemma 3.6.2. For simplicity, we restate it as

$$\nu(w) = \lfloor w/d \rfloor - \sum_{i=1}^{m} \lceil w/2^i \rceil .$$

For large enough n, the maximum in (3.6.11) will be determined first by the set of w's which give the maximum exponent of n. If this set has more than member, then the term which also maximizes the exponent of h will be chosen because $h = O\left((\ln n)^2\right)$. So we apply Lemma 3.6.2 to show that the maximum exponent of n, which we denote by ν, is given by $\nu = \max_{w \geq d} \nu(w) = \nu(d)$. Now, we can consider all weight paths which achieve the maximum exponent of n, and find the path in this set with the maximum exponent of h. Once again, we

apply Lemma 3.6.2 to show that $\nu(w) \leq \nu - 1$ for all $w \geq 4d$. It is easy to verify that the exponent of h in (3.6.11) is given by $1 + \sum_{i=1}^{m} \lceil w/2^i \rceil$. Since this value is non-decreasing with w, we find that the maximum exponent of h is upper bounded by $1 + \sum_{i=1}^{m} \lceil 4d/2^i \rceil \leq 1 + 4d + m$. This means that

$$\overline{A}_{\leq h}^{(m+1)}(n) = O\left(n^{\nu} h^{4d+m+1}\right), \tag{3.6.12}$$

for $h = O\left((\ln n)^2\right)$. We note that the second part of Lemma 3.6.2 does not hold for the case of $d = 2$ and $m = 1$, and this case will be dealt with separately.

Now, for $d \geq 3$ or $m \geq 2$, we can upper bound the probability of error associated with small output weights. Combining (3.2.1) and (3.6.12) allows us to upper bound the probability of word error associated with small output weights by

$$\sum_{h=1}^{L_n} O\left(n^{\nu} h^{4d+m+1}\right) z^h = O(n^{\nu}),$$

for any $z < 1$. We note that the sum can be evaluated by taking derivatives of the geometric sum formula. This proves that the WE of any CA^m code with $d \geq 3$ or $m \geq 2$ satisfies Condition 3.2.7 with $L_n = (\ln n)^2$ and $f(n) = n^{\nu}$. The probability of bit error can also be upper bounded by revisiting the entire derivation of (3.6.7), and starting with $B_h^{(o)}$ instead of $A_h^{(o)}$. If the encoder of the outer code is non-catastrophic, then we find that the result is scaled by a constant and the exponent is reduced by one. Therefore, the bit error rate condition of Corollary 3.2.10 is satisfied with $g(n) = n^{\nu-1}$.

For $d = 2$ and $m = 1$, we can bound the probability of error more directly. The WE bound, (3.5.4), can be simplified for the case of $d = 2$ and $h = O\left((\ln n)^2\right)$, and it is easy to verify that

$$\overline{A}_h^{\text{CA}}(n) \leq O(1) e^{4g^2 h/\tau}.$$

Using this, the probability of word error, (3.2.1), can be upper bounded by

$$\sum_{h=1}^{L_n} O(1) e^{4g^2 h/\tau} z^h = O(1),$$

as long as $z < e^{-4g^2/\tau}$. It is worth noting that this is exactly the same threshold that will be predicted by the bound of large output weights. This proves that the WE of any CA^m code with

$d = 2$ or $m = 2$ satisfies Condition 3.2.7 with $L_n = (\ln n)^2$ and $f(n) = 1$. As before, the probability of bit error, (3.2.2), can be upper bounded by revisiting the derivation of (3.5.4) and starting with $B_h^{(o)}$ instead of $A_h^{(o)}$. If the encoder of the outer code is non-catastrophic, then we find that the the exponent is reduced by one. Therefore, the bit error rate condition of Corollary 3.2.10 is satisfied with $g(n) = n^{-1}$. Since the exponent, ν, is zero for $d = 2$ and $m = 1$, both of these decay rates satisfy the theorem.

Next, we can verify that Condition 3.2.8 holds by first using (3.2.6) and (3.6.9) to show that

$$r_n^{(m+1)}(\delta; \text{CA}^m) = \frac{1}{n} \ln \overline{A}_{\leq h}^{(m+1)}(n) = \frac{1}{\tau} g^d (2^{m+1} h/n)^{d(1-2^{-m})} + O\left(\frac{1}{n}\right).$$

Combining this with the fact that $L_n = (\ln n)^2$ shows that the first part of Condition 3.2.8 holds because $\frac{1}{n} = o\left(\frac{(\ln n)^2}{n}\right)$. Now, we can use (3.6.10) to verify that $\lim_{\delta \to 0^+} \left(r^{(m+1)}(\delta; \text{CA}^m)/\delta\right) < \infty$. It is easy to verify that the limit is given by

$$\lim_{\delta \to 0^+} \frac{r^{(m+1)}(\delta; \text{CA}^m)}{\delta} \leq \begin{cases} 4g^2/\tau & \text{if } d = 2 \text{ and } m = 1 \\ 0 & \text{if } d \geq 3 \text{ or } m \geq 2 \end{cases}.$$

This proves that the WE of any CA^m code with $d \geq 2$ satisfies Condition 3.2.8.

Now that we have established the validity of Conditions 3.2.7 and 3.2.8, we can apply Theorem 3.2.9 and Corollary 3.2.10. Using only the union bound, rather than the tighter typical set bound, corresponds to choosing $\lambda = 1$ and makes the noise threshold equal to $\alpha_T(1)$. Using the definition, (3.2.10), gives the same threshold in terms of the Bhattacharyya parameter, namely that $z^* = e^{-c_{UB}}$. Since $r(\delta) < \infty$ and $\lim_{\delta \to 0^+} \left(r^{(m+1)}(\delta; \text{CA}^m)/\delta\right) < \infty$, it is clear that $c_{UB} < \infty$ and this proves that there exists a positive threshold such that, for any $z < z^*$, the probability of word error under ML decoding is $P_W = \Theta(n^\nu)$. The corollary extends this result to the probability of bit error with a decay rate of $P_B = \Theta(n^{\nu-1})$.

Finally, we consider a lower bound on the probability of error associated with small output weights. Consider the weight path of the worst case minimum distance, which is given by $h_{i+1} = \lceil d/2^i \rceil$ for $i = 0, \ldots, m$. The probability of picking a code, from the ensemble, which has a codeword of this distance is lower bounded by

$$P_M(n) \prod_{i=2}^{m} P_{h_i, h_{i+1}}(n),$$

where $P_M(n)$ is the probability that there is a codeword of weight $\lceil d/2 \rceil$ after the first interleaver. We note that this is a lower bound because it does not take into account the effect of multiple codewords of minimum weight at each stage. Now, we can combine the fact that $P_{h_i,h_{i+1}}(n) = \Theta(n^{-\lceil h_i/2 \rceil})$ with the result of Lemma 3.6.3 (i.e., $P_M = \Omega(n^{1-\lceil d/2 \rceil})$) to show that the probability of picking a code with worst case minimum distance is

$$\Omega\left(n^{1-\sum_{i=1}^{m}\lceil d/2^i \rceil}\right) = \Omega\left(n^{\nu}\right).$$

Since the probability of word error is a constant for codewords of fixed output weight, this means that the probability of word error is $\Omega(n^{\nu})$. Furthermore, the number of bit errors generated by such a word error is a constant, so the probability of bit error is $\Omega(n^{\nu-1})$. Combining these lower bounds with the previously discussed upper bounds completes the proof that $P_W = \Theta(n^{\nu})$ and $P_B = \Theta(n^{\nu-1})$. □

3.6.6 The Exact Spectral Shape

Let $r^{(i+1)}(x)$ be the spectral shape of the WE after the ith "accumulate" encoder. It turns out that we can compute $r^{(i+1)}(x)$ exactly by noting that (3.6.1) can be upper and lower bounded with

$$\max_{h_1,\ldots,h_m} A_{h_1}^{(1)}(n) \prod_{i=1}^{m} P_{h_i,h_{i+1}}(n) \leq \overline{A}_{h_{m+1}}^{(m+1)}(n) \leq n^m \max_{h_1,\ldots,h_m} A_{h_1}^{(1)}(n) \prod_{i=1}^{m} P_{h_i,h_{i+1}}(n).$$

Using these bounds, it is easy to verify that the asymptotic spectral shape is given by

$$r^{(m+1)}(x_{m+1}; \mathrm{CA}^m) = \max_{x_1,\ldots,x_m} \left[r^{(1)}(x) + \sum_{i=1}^{m} p(x_i, x_{i+1}) \right],$$

where $p(x,y)$ is given by (3.4.4). This can also be computed using the incremental form,

$$r^{(i+1)}(x_{i+1}; \mathrm{CA}^m) = \max_{0<x_i<1} \left[r^{(i)}(x_i) + p(x_i, x_{i+1}) \right]. \tag{3.6.13}$$

The functional form of (3.6.13) makes it quite amenable to analysis. It turns out that (3.6.13) is simply a linear transform in the max-plus semiring [5]. We start by showing that the function, $H(x) + C$, is a left eigenvector of $p(x,y)$, which essentially means that $\max_{0\leq x\leq 1}\left[H(x) + C + p(x,y)\right] = H(y) + C$. Using (3.4.4) to expand the $p(x,y)$ on the LHS of this expression gives

$$\max_{0\leq x\leq 1}\left[H(x) + C + p(x,y)\right] = \max_{0\leq x\leq 1}\left[C + yH\left(\frac{x}{2y}\right) + (1-y)H\left(\frac{x}{2(1-y)}\right)\right].$$

It is easy to verify that $x = 2y(1-y)$ maximizes the RHS, and that the maximum is given by $H(y) + C$. This is really not that surprising, however, because this analysis is quite similar to the Markov chain approach taken in [23] and gives the same result. On the other hand, we believe that a more detailed analysis of this operation may also allow one to bound the rate of convergence. In fact, we make the following conjecture.

Conjecture 3.6.5. *Let $r^{(m+1)}(x; CA^m)$ be the spectral shape of any CA^m code of rate R, and let $r^{(\infty)}(x; CA^m)$ be the stationary spectral shape as m goes to infinity. We conjecture that $r^{(\infty)}(x; CA^m) = [H(x) + 1 - R]^+$, where $[x]^+ = x$ for $x \geq 0$ and zero otherwise, and that*

$$\left| r^{(m+1)}(x; CA^m) - r^{(\infty)}(x; CA^m) \right| = O\left(\frac{1}{m}\right).$$

Remark 3.6.6. It is worth noting that the floor of the spectral shape at zero is basically due to the fact that $p(0, y) = 0$. This means that inputs of small output weight are mapped by the accumulate code to outputs of arbitrary weight with a probability that does not decay exponentially in the block length. This essentially sets up the lower bound $r^{(i+1)}(y; \mathrm{CA}^m) \geq r^{(i+1)}(0; \mathrm{CA}^m) + p(0, y) = 0$. Also, this result implicitly assumes that m grows independently of the block length because of the order in which limits are taken.

3.6.7 The Typical Minimum Distance

Now, we prove that the typical minimum distance of GRA^m codes grows linearly with the block length for $m \geq 2$. We do this by first proving this result for $m = 2$, and then showing that it must also hold for any finite $m > 2$. The basic method involves bounding the cumulative WE of the code and then using the fact that

$$Pr(d_{min} \leq h) \leq \overline{A}_{\leq h}.$$

First, we simplify the WE for CA codes. Starting with (3.5.4), we can drop the -1 and separate the exponential to get

$$\overline{A}^{\mathrm{CA}}_{\delta n}(n) \leq \frac{1}{1-\delta} \frac{\gamma^d - 1}{\gamma - 1} \left(e^{\gamma^d (n+\tau)/\tau} - 1 \right) \leq \frac{1}{1-\delta} \frac{\gamma^d - 1}{\gamma - 1} e^{\gamma^d} e^{\gamma^d n/\tau}.$$

Since $\gamma = g\sqrt{4\delta(1-\delta)} \leq g$ and $g \geq 1$, we can simplify the constant using the fact that

$$\frac{\gamma^d - 1}{\gamma - 1} e^{\gamma^d} \leq \frac{g^d - 1}{g - 1} e^{g^d} \leq g^d e^{g^d}.$$

For $d \geq 2$, we can also bound the $\gamma^d n/\tau$ term in the exponential using

$$\gamma^d n/\tau = g^d \left(4\delta(1-\delta)\right)^{d/2} n/\tau \leq g^d \left(4\delta(1-\delta)\right) n/\tau \leq 4g^d h/\tau.$$

Combining these bounds together gives

$$\overline{A}_h^{\text{CA}}(n) \leq \frac{g^d e^{g^d}}{1-h/n} e^{4g^d h/\tau}. \tag{3.6.14}$$

The remainder of the derivation must be handled separately for codes with $d = 2$ and codes with $d \geq 3$.

Convolutional Accumulate-2 Codes with $d = 2$

Now, we derive an upper bound on the cumulative WE of CA^2 codes with $d = 2$ by combining (3.6.14) and (3.4.5) to get

$$\overline{A}_{\leq h}^{\text{CA}^2}(n) \leq g^2 e^{g^2} \sum_{w=1}^{2h} \frac{1}{1-w/n} e^{4g^2 w/\tau} (4h/n)^{\lceil w/2 \rceil}.$$

Using the fact that $1/(1-w/n) \leq 1/(1-2h/n)$ for $1 \leq w \leq 2h$, we can rewrite this sum with $w = 2i$ to get

$$\overline{A}_{\leq h}^{\text{CA}^2}(n) \leq (e^{-4g^2/\tau} + 1) \sum_{i=1}^{h} e^{8g^2 i/\tau} (4h/n)^i, \tag{3.6.15}$$

for $h < n/2$. Upper bounding this sum by the infinite sum and letting $h = \delta n$ gives

$$\overline{A}_{\leq \delta n}^{\text{CA}^2}(n) \leq \frac{2g^2 e^{g^2}}{1-2\delta} \frac{\left(4\delta e^{8g^2/\tau}\right)}{1-4\delta e^{8g^2/\tau}},$$

for $\delta < 1/\left(4e^{8g^2/\tau}\right)$. Now, we point out that for any $\epsilon > 0$ there exists a $\delta > 0$ such that $\overline{A}_{\leq \delta n}^{\text{CA}^2}(n) \leq \epsilon$. Therefore, almost all of the codes in the ensemble will have a minimum distance growing linearly with the block length. Since the geometric sum in (3.6.15) also grows exponentially in n for $\delta > 1/\left(4e^{8g^2/\tau}\right)$, one might conjecture that the minimum distance is almost always equal to $1/\left(4e^{8g^2/\tau}\right)$. Numerical evidence suggests otherwise, however.

Remark 3.6.7. Let δ^* be the smallest δ such that $\overline{A}_{\leq \delta n}^{\text{CA}^2}(n)$ grows exponentially in n. Numerical evidence suggests that $\lim_{n\to\infty} \overline{A}_{\leq \delta n}^{\text{CA}^2}(n) = f(\delta)$ is a well-defined function of δ for $0 \leq \delta < \delta^*$.

This function can be used as an upper bound on the cumulative distribution function of minimum distance ratio for the code ensemble. Simple analytical arguments show that $f(\delta)$ starts at $f(0) = 0$ and is strictly increasing towards $f(\delta^*) = \infty$. Finally, the largest minimum distance ratio provable via the average WE is given by the δ which solves $f(\delta) = 1$. Unfortunately, while the numerical methods of Section 3.8 may be used to estimate δ^*, we are not aware of any simple method for computing $f(\delta)$.

Convolutional Accumulate-2 Codes with $d \geq 3$

For $d \geq 3$, we can bound $\overline{A}^{\text{CA}}_{\leq h}(n)$ differently for small and large output weights. Using (3.6.12) for small output weights and (3.6.14) for large output weights gives

$$\overline{A}^{\text{CA}}_{\leq h}(n) \leq \begin{cases} O\left(n^{1-\lceil d/2 \rceil} h^{4d+3}\right) & h \leq (\ln n)^2 \\ \frac{g^d e^{g^d}}{1-h/n} e^{4g^d h/\tau} & \text{otherwise} \end{cases}.$$

Now, we can upper bound the cumulative WE of CA^2 codes with $d \geq 3$ by combining this with (3.4.5) to get

$$\overline{A}^{\text{CA}^2}_{\leq h}(n) \leq \sum_{w=1}^{(\ln n)^2} O\left(n^{1-\lceil d/2 \rceil} w^{4d+3}\right) (4h/n)^{\lceil w/2 \rceil} + g^d e^{g^d} \sum_{w=(\ln n)^2}^{2h} \frac{e^{4g^d w/\tau}}{1-w/n} (4h/n)^{\lceil w/2 \rceil}.$$

It is easy to verify that the first sum is $O\left(n^{1-\lceil d/2 \rceil}\right)$, for $h/n < 1/4$, by taking derivatives of the geometric sum formula. The second sum can be rewritten with $w = 2i$ by using the fact that $1/(1-w/n) \leq 1/(1-2h/n)$ for $1 \leq w \leq 2h$. This gives

$$\overline{A}^{\text{CA}^2}_{\leq h}(n) \leq O\left(n^{1-\lceil d/2 \rceil}\right) + \frac{2g^d e^{g^d}}{1-2h/n} \sum_{i=(\ln n)^2/2}^{h} e^{8g^d i/\tau} (4h/n)^i.$$

Upper bounding this sum by the infinite sum and letting $h = \delta n$ gives

$$\overline{A}^{\text{CA}^2}_{\leq \delta n}(n) \leq O\left(n^{1-\lceil d/2 \rceil}\right) + \frac{2g^d e^{g^d}}{1-2\delta} \frac{\left(4\delta e^{8g^d/\tau}\right)^{(\ln n)^2/2}}{1-4\delta e^{8g^d/\tau}}.$$

Since this expression is $O\left(n^{1-\lceil d/2 \rceil}\right)$ for $\delta < 1/\left(4e^{8g^2/\tau}\right)$, almost all of the codes in the ensemble will have a minimum distance ratio of $1/\left(4e^{8g^2/\tau}\right)$ or larger.

Remark 3.6.8. Again, we let δ^* be the smallest δ such that the true $\overline{A}^{\text{CA}^2}_{\leq \delta n}(n)$ grows exponentially in n. In this case, we conjecture that almost all codes in the ensemble have a minimum

distance ratio of δ^*. Assuming this is true, we can calculate the minimum distance ratio using the numerical methods of Section 3.8.

Convolutional Accumulate-m Codes

Suppose we serially concatenate any code, whose minimum distance grows like δn, with an interleaved "accumulate" code. Using Fact 3.4.1, it is clear that that the minimum distance of the new code is greater than $\delta n/2$. This means that if the minimum distance is $\Omega(n)$ for any m_0 then it is $\Omega(n)$ for any finite $m \geq m_0$. This concludes the proof that the minimum distance of any CA^m code, with $m \geq 2$ (and $m < \infty$), grows linearly with the block length. Although the minimum distance growth rate guaranteed by this argument decreases with m, this does not imply that the actual growth rate decreases with m. In fact, analytical evidence strongly suggests the growth rate increases monotonically to the limit implied by the Gilbert-Varshamov bound.

3.7 Iterative Decoding of CA^m Codes

3.7.1 Decoding Graphs

The iterative decoding of CA^m codes is based on a message passing decoder which operates on a graph representing the code constraints. This approach was introduced by Gallager in [12], and then generalized by Tanner in [28] and Wiberg in [32]. We refer to the resulting graphical representation of code constraints as a Gallager-Tanner-Wiberg (GTW) graph. The GTW graph of a code is not unique, however, and different graphs representing the same constraints may have very different decoding performances.

Belief propagation (BP) is a general algorithm for distributing information on a graph representing local constraints. Most message passing decoders described in the literature implement some form of BP on a code's GTW graph [20]. If the graph has no cycles, then BP is equivalent to the optimal soft output decoding, known as *a posteriori* probability (APP) decoding. This is sometimes cited as the reason why these decoders work quite well if the GTW graph does not have too many short cycles.

The GTW graph of the rate-1 "accumulate" code is shown in Figure 3.7.1. The nodes drawn as circles represent equality constraints (e.g., all edges attached to these nodes represent

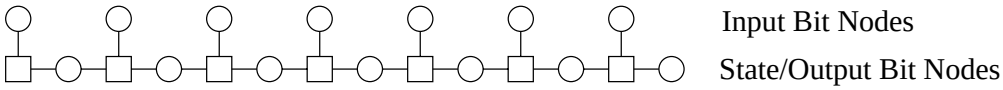


Figure 3.7.1: A GTW graph for the rate-1 “accumulate” code.

the same bit), and the nodes drawn as squares represent parity constraints (e.g., all edges attached to these nodes must sum to zero modulo-2). Let $u_1, \ldots, u_n$ be the input bits from left to right and let $x_1, \ldots, x_n$ be the output (state) sequence. We note that all addition between bits is assumed to be modulo-2. The outputs of the “accumulate” code can be computed using the recursive formula, $x_{i+1} = x_i + u_i$, with the initial condition $x_0 = 0$. This recursive formula can also be seen in the structure of the graph. Assuming all of input bits are known, an encoder can step from left to right on the graph computing the next output bit each time. The recursive update equation can also be rewritten as $u_i + x_i + x_{i+1} = 0$, and the graph reflects this in that each parity check involves an input bit and two adjacent output bits. It is also worth noting that the output sequence is equal to the encoder state sequence.

A GTW graph for general CA^2 codes, shown in Figure 3.7.2, is the concatenation of the outer code constraints with two “accumulate” GTW graphs mapped through permutations. From an encoding point of view, the outer code generates the input bits at the top of the graph and they are encoded by each “accumulate” GTW graph as they travel downward. When they reach the bottom, they are transmitted through the channel. From a decoding point of view, the channel starts the process with noisy estimates of the transmitted codeword at the bottom of the graph. Belief propagation can then used to propagate messages through the graph until all of the messages satisfy the constraints or some maximum iteration number is reached.

3.7.2 Message Passing Rules

The message passed along any edge in Figure 3.7.2 is the probability distribution of the edge’s true value given the subgraph below that edge. If the true edge values are binary, then the log-likelihood ratio (LLR) can be used to represent the distribution. Similar to the notion of a probability, we define the LLR function of a binary random variable to be

$$LLR(X|Y) = \log \frac{Pr(X = 1|Y)}{Pr(X = 0|Y)}.$$

The message passing decoder propagates LLRs around the graph by assuming that all input messages arriving at a constraint are independent. Using the input messages from all but one edge, the constraint can be combined with Bayes' rule to calculate an output message for the edge left out. This rule is used to calculate all of the output messages for that constraint node, and generally all of these messages will be different.

Consider an equality constraint with j edges. In this case, the true value of each edge must be the same and we will have j LLRs for a single random bit. It is clear that the true bit, which we refer to as X, must either be a one or a zero. The output passed to each edge is a function only of the other $j-1$ edges, so computing the output message involves combining $j-1$ independent LLR messages. Let $M_1, \ldots, M_j$ be the LLR input messages, and let $\hat{M}_1, \ldots, \hat{M}_j$ be the output messages. This means that $\hat{M}_i = LLR(X|M_1, \ldots, M_{i-1}, M_{i+1}, \ldots, M_j)$, and using the product rule for independent observations gives

$$\hat{M}_i = \log \prod_{k \neq i} \frac{Pr(X=0|M_k)}{Pr(X=1|M_k)} = \sum_{k \neq i} M_k. \tag{3.7.1}$$

Consider a parity constraint with j edges. In this case, the modulo-2 sum of true bits must be zero. Let the true bits associated with edge be $X_1, \ldots, X_j$. It is clear that the modulo-2 sum of any $j-1$ of these bits must equal the bit which was left out. The same idea can be applied to LLRs using a soft-XOR operation. Given two independent binary random variables, A and B, we define their soft-XOR to be $LLR(A+B)$. It is easy to verify that this function is given by

$$LLR(A+B) = 2\tanh^{-1}\left(\tanh\left(\frac{LLR(A)}{2}\right)\tanh\left(\frac{LLR(B)}{2}\right)\right),$$

and this can be found in [26]. Let $M_1, \ldots, M_j$ be the LLR input messages, and let $\hat{M}_1, \ldots, \hat{M}_j$ be the output messages. If we let Z be the modulo-2 sum, $\sum_{k \neq i} X_i$, then this means that

$$\hat{M}_i = LLR\left(Z|M_1, \ldots, M_{i-1}, M_{i+1}, \ldots, M_j\right).$$

Writing $\hat{M}_i$ in terms of the soft-XOR function gives

$$\hat{M}_i = 2\tanh^{-1}\left(\prod_{k \neq i} \tanh\frac{M_k}{2}\right). \tag{3.7.2}$$

Now, we consider the constraints imposed by the outer code. If the outer code is a repeat or single parity check code, then these constraints are easily represented using the equality

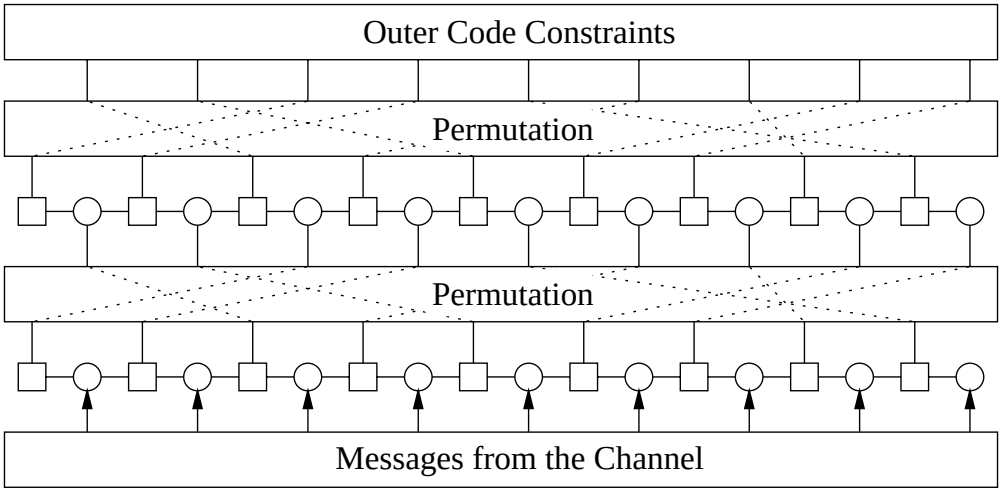


Figure 3.7.2: A Tanner graph for an arbitrary CA^2 code.

and parity constraints discussed above. If the outer code is a more general TCC, then the GTW graph for the code will include state variables and belief propagation is very similar to the BCJR algorithm [2]. We refer to soft-output variations of the BCJR algorithm as APP algorithms. A thorough discussion of this can be found in [20].

3.7.3 Message Passing Schedule

The message passing schedule is the order in which the messages are updated on the graph. While there are almost an unlimited number of message passing schedules, there are two in particular worth mentioning. We will refer to them as as turbo style decoding and LDPC style decoding.

In turbo style decoding, each horizontal slice of the GTW graph, shown in 3.7.2, is treated as an independent APP algorithm. So starting at the bottom with subgraph representing the “accumulate” code, messages are passed left and right until the APPs are computed for that slice. Since this subgraph is cycle free, the message passing algorithm computes the exact APPs. Next, the output messages are passed upwards to the next stage, where another APP decoding is done. Finally, the process reaches the outer code at the top and reverses itself by stepping back down the graph. This is identical to the standard turbo decoding of serially concatenated turbo codes.

In LDPC style decoding, the messages for all edges are computed at the same time.

This implicitly results in a two step process where bit nodes first pass messages to the check nodes, and then the check nodes pass messages back to the bits nodes. There appears to be no significant performance difference between these two message passing schedules if a large number of iterations are performed. Also, while the LDPC style decoder requires more operations per iteration, all of these operations can be done in parallel.

3.7.4 Density Evolution

Density evolution (DE) is a very useful technique that can be used to analyze the expected performance of a message passing decoder. The basic idea is that, by assuming that all messages arriving at a constraint node are independent, one can easily track the probability density functions of the LLR messages being passed around the graph. The independence assumption is theoretically justified for large sparse graphs and small iteration numbers. This type of analysis was first performed by Gallager for LDPC codes [12], and later generalized (and put on firm theoretical ground) by Richardson and Urbanke [26].

Since LLRs are simply summed up at equality constraint nodes, the density of the output message is simply the convolution of the density of the input messages. So, if the input messages are all drawn i.i.d. from a LLR density function, then the output messages will also be i.i.d. but with a different distribution. Let $P(x)$ be the density function of X and $Q(y)$ be the density function of Y, then we write the density function of $Z = X + Y$ as $(P \otimes Q)(z)$. The effect of the parity constraint on message densities is much more complicated, so we write the density function of

$$Z = 2\tanh^{-1}\left(\tanh\left(\frac{X}{2}\right)\tanh\left(\frac{Y}{2}\right)\right)$$

as $(P \oplus Q)(z)$. It is easy to verify that both of these operators are commutative, associative, and distributive over the addition of densities. Furthermore, the identity of $\otimes$ is the delta function at zero, Δ_0, and the identity of $\oplus$ is the delta function at infinity, Δ_∞.

Now, we consider a general CA code and focus on the message density on the edges out of the equality constraint for the "accumulate" code. Let the message density of these edges after l decoding iterations be P_l, where P_0 is the initial LLR density of the channel. Let the output of the APP decoder for the outer code have LLR density $f(Q)$ when the inputs have LLR

density Q. Tracking one cycle of the P message around the graph gives the density evolution,

$$P_{l+1} = (f(P_l \oplus P_l) \oplus P_l) \otimes P_0. \tag{3.7.3}$$

For a memoryless symmetric channel, with parameter α, we define the DE threshold, α_{DE}, to be the largest α such that $\lim_{l\to\infty} P_l = \Delta_\infty$ (i.e., the fraction of incorrect messages goes to zero). Numerical methods can be used to show that P_l is approaching Δ_∞ as l increases, but actual convergence requires also that Δ_∞ be a stable fixed point of the iteration. This is known as the stability condition, and can be understood by examining the iteration when $P_l = (1-\epsilon)\Delta_\infty + \epsilon Q$ for small ϵ and any Q.

We start by expanding the density update function of the outer code with

$$f\left((1-\epsilon)\Delta_\infty + \epsilon Q\right) = (1-\kappa\epsilon)\Delta_\infty + \kappa\epsilon Q + O(\epsilon^2). \tag{3.7.4}$$

We can compute the coefficient, κ, by analyzing the APP decoder. For any bit in the outer code, consider all of the codewords which have a one in that position. Ignoring the chosen bit, the probability of more than one bit in the remaining bits of the codeword receiving a Q message is $O(\epsilon^2)$. If exactly one other bit in the codeword receives a Q message and the rest receive the Δ_∞ message, then we can compute the output of the APP decoder exactly. For code bits which do not support a weight-2 codeword, this output will always be Δ_∞ because the perfect knowledge of the other bits corrects the error. For code bits which support weight-2 codewords, the output will receive messages from the Q density. Since each weight-2 codeword involving the output bit will contribute one ϵQ, the average output will be $\kappa\epsilon Q$ where κ is the average number of bits involved in weight-2 codewords per input bit. This means that

$$\kappa = \lim_{n\to\infty} \frac{2}{n} A_2^{(o)}(n), \tag{3.7.5}$$

where $A_2^{(o)}$ is the number of weight-2 codewords in the outer code.

Proposition 3.7.1. *Consider a CA code whose outer code has the WE, $A_h^{(o)}(n)$, and let $z(\alpha)$ be the Bhattacharyya parameter of a memoryless symmetric channel with parameter α. The DE threshold is upper bounded by the stability condition, which states that*

$$\alpha_{DE} \leq \sup\left\{\alpha \in \Re^+ | z(\alpha) \leq \frac{1}{2\kappa+1}\right\},$$

where κ is given by (3.7.5).

Proof. We start by expanding (3.7.3) about $P_l = (1-\epsilon)\Delta_\infty + \epsilon Q$ for small ϵ, and this gives

$$P_{l+1} = \left(f\left((1-2\epsilon)\Delta_\infty + 2\epsilon Q + O(\epsilon^2)\right) \oplus \left((1-\epsilon)\Delta_\infty + \epsilon Q\right)\right) \otimes P_0.$$

Using (3.7.4), we can simplify this to

$$P_{l+1} = (1-(2\kappa+1)\epsilon)\,\Delta_\infty + (2\kappa+1)\epsilon Q \otimes P_0 + O(\epsilon^2).$$

If we consider P_{l+n} for large n, we can apply a large deviation principle to the repeated convolution to show that the contribution of Q to P_{l+n} is essentially given by

$$(2\kappa+1)^n z(\alpha)^n \epsilon Q,$$

where $z(\alpha)$ is the Bhattacharyya parameter of the channel [26]. Clearly this will tend to zero if and only if $z(\alpha) < 1/(2\kappa+1)$. □

Example 3.7.2. For parity accumulate codes, the APP decoder for the outer code is given simply by a parity check node. So the decoding graph is equivalent to a particular LDPC code and the stability condition can be derived without considering general outer codes. Assuming a rate $(k-1)/k$ code is used on the AWGN channel, we have

$$e^{-1/(2\sigma^2)} \leq \frac{1}{2k-1}$$

which implies that $E_b/N_0 \geq \frac{k}{k-1}\log(2k-1)$. Using Proposition 3.7.1, we find that the number of weight-2 codewords in the outer code is given by $A_2^{(o)}(n) = (n/k)(k)(k-1)/2$. This makes $\kappa = k-1$ and gives exactly the same condition.

The generalization of (3.7.3) to CA^m codes is straightforward and the details are left to the reader. We do note, however, that CA^m codes are unconditionally stable if $d \geq 3$ or $m \geq 2$. If $d = 2$ and $m = 1$, the stability of iterative decoding depends on the channel parameter and therefore may determine the DE threshold. For example, the true DE threshold of all PA codes is determined by the stability condition. Furthermore, the DE threshold computed via stability condition for PA codes is actually identical to the ML decoding threshold.

For LDPC codes, Richardson and Urbanke also proved a concentration theorem which shows that, for all $\alpha > \alpha_{DE}$, the true probability of bit error probability can be made arbitrarily small by increasing the block length and the number of iterations [26]. We believe this result can be extended to a very general class of sparse graph codes which includes CA^m codes. The DE thresholds of various CA^m codes have been computed and are given in Table 3.1.

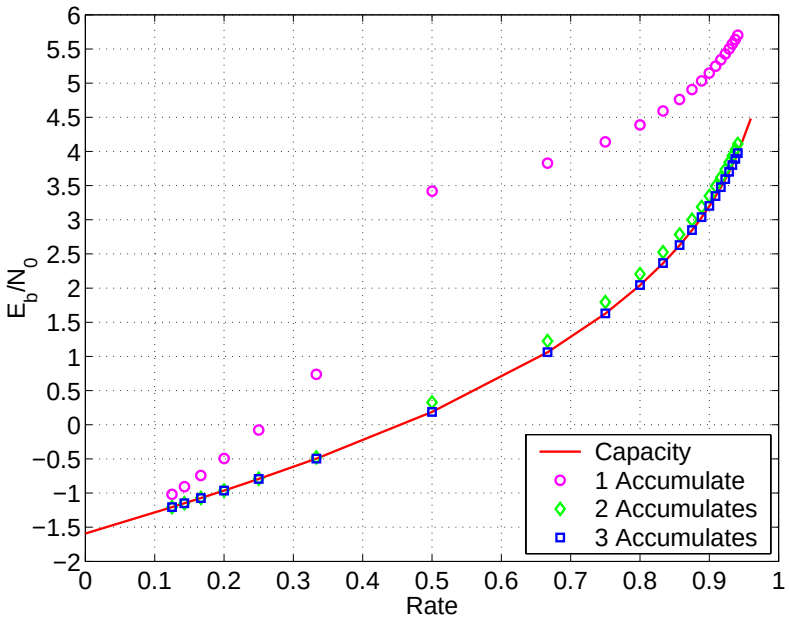


Figure 3.8.1: Typical set decoding E_b/N_0 thresholds for RA^m and PA^m codes in AWGN.

3.8 Numerical Methods for the Spectral Shape

In this section, we outline our numerical method for computing exponentially tight bounds on the spectral shape of CA^m codes. These bounds can be used to compute very good bounds on the noise threshold and minimum distance ratio. These noise thresholds are based on the typical set decoding bounds described in [1] and [16], which can be applied to any binary-input symmetric channel. The minimum distance ratio bounds are based on finding the smallest output weight such that the WE is growing exponentially.

3.8.1 The Quantized Spectral Shape

Our numerical method for computing the spectral shape of CA^m code is based on quantizing the normalized output weight to the grid $0, \Delta, 2\Delta, \ldots, N\Delta$ where $\Delta = 1/N$. Let $\tilde{r}^{(i)}(j\Delta; \mathrm{CA}^m)$ be an estimate of $r^{(i)}(j\Delta; \mathrm{CA}^m)$ based on this quantization. We use the recursive update,

$$\tilde{r}^{(i+1)}(k\Delta; \mathrm{CA}^m) = \max_{0 \le j \le N} \tilde{r}^{(i+1)}(j\Delta; \mathrm{CA}^m) + p(j\Delta, k\Delta),$$

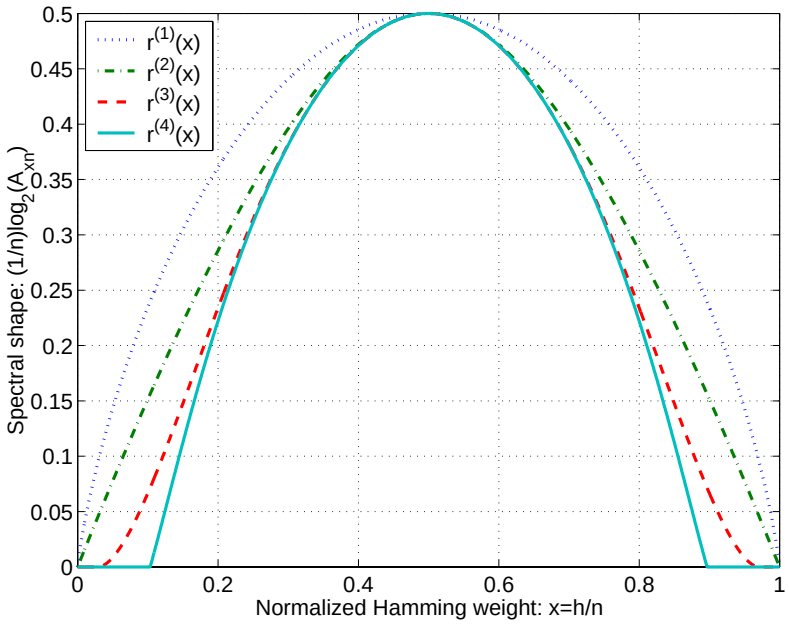


Figure 3.8.2: The spectral shape of a (2,1) single parity code and the associated PA^m codes with $m = 1, 2, 3$.

which is based on (3.6.13) and (3.4.4). The only difficulty lies in estimating $\tilde{r}^{(1)}(j\Delta; \text{TCC})$ from the parametric representation of $r^{(1)}(\delta; \text{TCC})$ given by (3.3.7). We do this by calculating $r(\delta(x); \text{TCC})$ and $\delta(x)$ on an x-grid and then interpolating $r(\delta(x); \text{TCC})$ onto the $0, \Delta, 2\Delta, \ldots, N\Delta$ grid. One problem with this method is that a uniform x-grid may require a very large number of points for reliable estimation of $r^{(1)}(\delta; \text{TCC})$. We have had more success using a non-uniform x-grid, where $x = \sqrt{y}$ and y is uniform on $[0, 1]$.

In general, we have observed that the spectral shape of a CA^m code is continuous and smooth whenever it is positive. Under this assumption, we believe that the error due to quantization, $\left| r^{(i+1)}(j\Delta; \text{CA}^m) - \tilde{r}^{(i+1)}(j\Delta; \text{CA}^m) \right|$, will be $O(1/N)$. The results of this method are shown in Figures 3.8.2 and 3.8.3 for two particular outer codes and $m = 1, 2,$ and 3.

3.8.2 Noise Thresholds

Consider a binary-input symmetric channel with a single parameter, α. The typical pairs decoding threshold, α_T, is given by (3.2.12) of Theorem 3.2.9. It can be computed numerically by finding the α-root of the equation $\max_{0 \le j \le N} \tilde{r}^{(m+1)}(j\Delta; \text{CA}^m) - K(j\Delta, \alpha) = 0$.

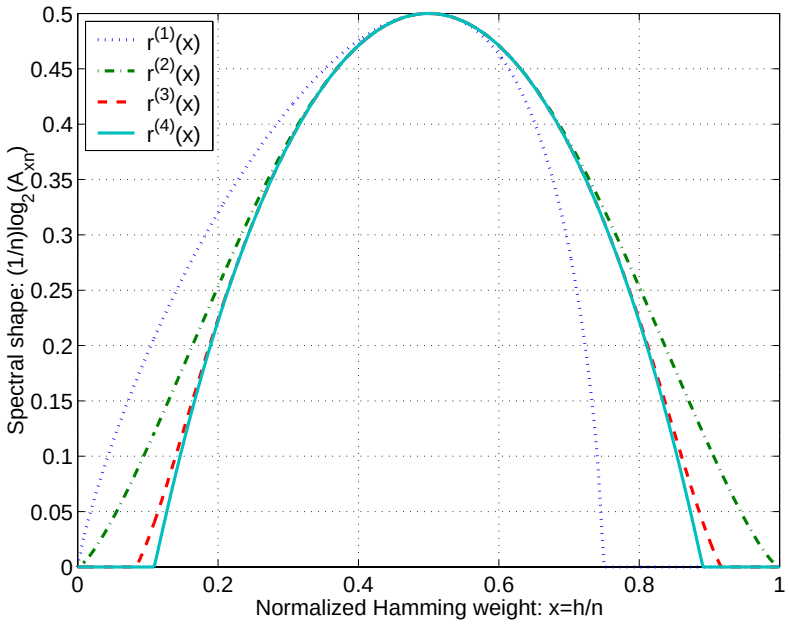


Figure 3.8.3: The spectral shape of a $[1, 1 + D]$ CC and the associated CA^m codes with $m = 1, 2, 3$.

Standard root finding methods such as bisection can be used to solve this problem. Since the most time consuming part of this calculation is computing $K(j\Delta, \alpha)$, one can precompute this quantity on an α-grid of sufficient accuracy, for each j.

We have also found that AWGN thresholds computed using $N = 1000$ typically do not change by more than $0.005\,dB$ for $N > 1000$. Also, thresholds computed using this method for $m = 1$ match other published results in all significant digits [16]. Finally, we note that the thresholds of CA^m with $d = 2$ and $m = 1$ are usually determined by $\lim_{\delta\to 0^+} r^{(2)}(\delta; \text{CA})/\delta$ and will not be correctly estimated using this method. In this case, thresholds can and should be calculated by analytically expanding $r^{(1)}(\delta; \text{TCC})$ about $\delta = 0$ and computing $\lim_{\delta\to 0^+} r^{(2)}(\delta; \text{CA})/\delta$ analytically.

This method was applied to RA^m and PA^m codes on the AWGN channel. The E_b/N_0 thresholds are shown in Figure 3.8.1 and listed in Table 3.1. In the table, γ^* denotes the Shannon limit and γ_m denotes the typical set decoding threshold. The table also lists thresholds for CA^m whose outer codes are the $(8, 4)$ Hamming code and the $[1, 1 + D]$ CC.

3.8.3 Minimum Distance Ratio

In Section 3.6.7, it was shown that the minimum distance of a CA^m code grows linearly with the block length for $m \geq 2$. Let δ_m^* be the smallest $\delta > 0$ such that $r^{(m+1)}(\delta; \text{CA}^m) > 0$. Except for the case of $d = 2$ and $m = 2$, we believe that the growth rate of the minimum distance with block length will be at least δ_m^*. The case of $d = 2$ and $m = 2$ is discussed more thoroughly in Remark 3.6.7. Since we can use our numerical method to estimate δ_m^* with arbitrary accuracy, this provides a useful method for considering the minimum distance ratios of CA^m codes. Furthermore, the minimum distance ratios computed using this method are quite close to the empirical growth rates observed via the exact calculation of the average WE for finite block lengths [23]. The δ_m^* value for $m = 2, 3$ is given in Table 3.1 for each code considered.

3.9 Concluding Remarks

In this chapter, we give a fairly complete analytical picture of the properties and performance of CA^m codes. While the iterative decoding of these codes cannot compete with that of turbo codes or optimized LDPC codes [25], their ability to approach channel capacity under ML decoding is quite astounding. Theoretically, these results offer some insight into the structure of CA^m codes, and a number of new mathematical tools of more general use. From a practical point of view, this work shows that the future of CA^m codes depends on either improving their performance with iterative decoding or, more ambitiously, finding new decoding methods which approach the performance of ML decoding.

C	R	γ^*	γ_1	γ_2	γ_3	δ^*_{GV}	δ^*_2	δ^*_3	α_1	α_2	α_3
RA	1/8	-1.207	-1.102	-1.206	-1.207	.295	.291	.295	0.29	3.86	6.85
RA	1/7	-1.150	-0.905	-1.149	-1.150	.281	.275	.282	0.19	3.52	6.41
RA	1/6	-1.073	-0.742	-1.072	-1.073	.264	.255	.265	0.11	3.13	5.9
RA	1/5	-0.964	-0.494	-0.962	-0.963	.243	.229	.243	0.06	2.69	5.31
RA	1/4	-0.794	-0.078	-0.790	-0.794	.215	.192	.215	0.12	2.20	4.61
RA	1/3	-0.495	0.739	-0.478	-0.495	.174	.133	.174	0.50	1.65	3.76
PA	1/2	0.187	3.419	0.327	0.188	.110	.0287	.104	3.42	1.23	2.72
PA	2/3	1.059	3.828	1.224	1.062	.061	.0101	.054	3.83	1.83	2.86
PA	3/4	1.626	4.141	1.794	1.630	.042	.0052	.035	4.14	2.27	3.12
PA	4/5	2.040	4.388	2.206	2.044	.031	.0032	.031	4.39	2.62	3.36
PA	5/6	2.362	4.590	2.526	2.366	.025	.0021	.019	4.59	2.89	3.57
PA	6/7	2.625	4.760	2.785	2.629	.020	.0015	.016	4.76	3.12	3.75
PA	7/8	2.845	4.906	3.001	2.849	.017	.0011	.012	4.91	3.32	3.90
PA	8/9	3.033	5.034	3.187	3.037	.015	.0009	.011	5.03	3.49	4.04
PA	9/10	3.198	5.148	3.349	3.202	.013	.0007	.009	5.15	3.63	4.16
PA	10/11	3.343	5.249	3.492	3.348	.012	.0006	.008	5.25	3.76	4.27
PA	11/12	3.474	5.341	3.620	3.478	.010	.0005	.007	5.34	3.88	4.37
PA	12/13	3.591	5.425	3.736	3.596	.009	.0004	.006	5.43	3.99	4.46
PA	13/14	3.699	5.502	3.841	3.703	.009	.0004	.006	5.50	4.08	4.55
PA	14/15	3.797	5.572	3.938	3.801	.008	.0003	.005	5.57	4.17	4.63
PA	15/16	3.887	5.638	4.027	3.892	.007	.0003	.005	5.64	4.26	4.70
PA	16/17	3.971	5.700	4.109	3.976	.007	.0002	.004	5.70	4.33	4.77
HA	4/8	0.187	0.690	0.191	0.187	.110	.090	.110	N/A	N/A	N/A
CA	1/2	0.187	0.909	0.199	0.187	.110	.084	.110	N/A	N/A	N/A

Table 3.1: Numerical results for various CA^m codes. (C = outer code, R = code rate, γ^*= Shannon limit, γ_m= typical set decoding threshold with m accumulates, δ^*_{GV}= Gilbert-Varshamov bound, δ^*_m= normalized distance threshold with m accumulates, and α_m= density evolution threshold with m accumulates)

3A Binomial Coefficient Bounds

3A.1 The Product Bound

First, we consider the following well-known upper and lower bounds on the binomial coefficient,

$$\left(\frac{n}{k}\right)^k \leq \binom{n}{k} \leq \left(\frac{ne}{k}\right)^k. \tag{3A.1}$$

Although these bounds are somewhat loose, their simplicity makes them surprisingly useful. The proof of the lower bound is based on the fact that

$$\binom{n}{k} = \frac{n(n-1)\cdots(n-k+1)}{k(k-1)\cdots(1)} = \left(\frac{n}{k}\right)^k \prod_{i=0}^{k-1} \frac{1-i/n}{1-i/k},$$

and that $(1-i/n) \geq (1-i/k)$. The proof of the upper bound is based on the trivial upper bound

$$\binom{n}{k} \leq \frac{n^k}{k!},$$

and a corollary of Stirling's formula that says $\ln k! \geq \int_0^k \ln(x)\,dx = \ln(k^k e^{-k})$.

3A.2 The Entropy Bound

Let the binary entropy function be $H(x) = -x\log_2 x - (1-x)\log_2(1-x)$, then we have

$$\frac{2^{nH(k/n)}}{n+1} \leq \binom{n}{k} \leq 2^{nH(k/n)}, \tag{3A.2}$$

for $0 \leq k \leq n$. A simple information theoretic proof of this can be found in [6, p. 284]. The more detailed analysis of MacWilliams and Sloane can be used to improve these to

$$\frac{1}{\sqrt{8n(k/n)(1-k/n)}} 2^{nH(k/n)} \leq \binom{n}{k} \leq \frac{1}{\sqrt{2\pi n(k/n)(1-k/n)}} 2^{nH(k/n)}. \tag{3A.3}$$

3A.3 Sums of Binomial Coefficients

In this section, we consider bounds on the sum of binomial coefficients,

$$S(n,k) = \sum_{i=0}^{k} \binom{n}{k}. \tag{3A.4}$$

In general, there is no closed form expression for this sum and it arises quite frequently.

The most straightforward bound simply uses a generating function bound (a.k.a. Chernov bound). Starting with the binomial theorem, we have

$$(1+x)^n = \sum_{i=0}^{n} \binom{n}{i} x^i \geq \sum_{i=0}^{k} \binom{n}{i} x^i,$$

for any $0 < x \leq 1$. Lower bounding x^i by x^k and rearranging terms gives

$$S(n,k) \leq (1+x)^n x^{-k}$$

for any $0 < x \leq 1$. Minimizing this bound over x gives the final result of

$$S(n,k) \leq 2^{nH(k/n)}, \tag{3A.5}$$

for $0 \leq k \leq n$. We can simplify (and weaken) the bound further by applying $\log(1-x) \leq -x/\ln 2$ to the entropy function. This results in $H(x) \leq -x\log x - (1-x)(-x/\ln 2)$ and dropping the $-x^2/\ln 2$ term results in the very simple bound

$$S(n,k) \leq \left(\frac{ne}{k}\right)^k. \tag{3A.6}$$

It turns out that even though (3A.5) is only valid for $0 \leq k \leq n$, the weakened version of this bound allows it to hold for $0 \leq k \leq 1.88n$. This can be verified by solving for the largest k such that (3A.6) is greater than or equal to 2^n. Furthermore, it is easy to verify that this upper bound is concave in k because the second derivative is negative for $k > 0$.

Finally, we give the bound,

$$\sum_{i=0}^{k} \binom{n}{i} \leq \frac{(n+1)^k}{k!}, \tag{3A.7}$$

which distinguishes itself from the rest via the $k!$ denominator even though it is numerically very similar to (3A.6). The proof of this bound is via induction, so we define

$$T(n,k) = \frac{(n+1)^k}{k!},$$

and begin by listing the base cases $S(0,0) = T(0,0) = 1$ and $S(n,1) = T(n,1) = n+1$. Next, we prove that $T(n,k) \geq S(n,k)$ assuming that $T(n,k-1) \geq S(n,k-1)$. To do this, we observe that

$$S(n,k) = S(n,k-1) + \binom{n}{k},$$

and

$$T(n,k) = T(n,k-1) + \frac{(n+1)^{k-1}(n-k+1)}{k!}.$$

Since $T(n,k-1) \geq S(n,k-1)$ by assumption and

$$\frac{(n+1)^{k-1}(n-k+1)}{k!} \geq \frac{n(n-1)\cdots(n-k+1)}{k!} = \binom{n}{k},$$

for $0 \leq k \leq n$, it is clear that $T(n,k) \geq S(n,k)$. It turns out that this version of this bound actually holds for $0 \leq k \leq \lfloor 1.72n \rfloor$, since $T(n, \lfloor 1.72n \rfloor) \geq 2^n$. This can be verified by plotting $\log T(n, \lfloor 1.72n \rfloor) - n \log 2$ for $n \geq 1$. Furthermore, this upper bound is concave in k because the second derivative of $\log T(n,k)$ is given by

$$\frac{d^2}{d^2k}\left(k\ln(n+1) - \Gamma(k+1)\right) = -\sum_{i=1}^{\infty} \frac{1}{(k+i)^2},$$

which is negative for $k > 0$.

3B Convolutional Code Bounds

3B.1 Proof of Theorem 3.3.1

Proof of Theorem 3.3.1. Following [18], this proof is based on breaking the output sequence into non-overlapping segments, known as detours, which can be placed in the block independently of each other. A *detour* is defined to be any output sequence generated by a state sequence which starts in the zero state, ends in the zero state, and does not otherwise visit the zero state. Furthermore, all of the weight in an output sequence is contained in the detours. Consider any output sequence consisting of r detours. This output sequence can be uniquely specified by the r detour starting positions and by the r detour output sequences.

So we can count the total number of output sequences by counting the number of ways of choosing the detour starting positions, the detour output sequences, and the number of detours. The number of ways to choose r distinct detour starting positions from n/τ possible starting positions is given by the binomial coefficient $\binom{n/\tau}{r}$. Let $T_h^{(r)}$ be the number of ways to choose r detour output sequences such that the total weight of all detours is h. Since each detour produces an output weight of at least d, the number of detours is at most $\lfloor h/d \rfloor$. Therefore, the

number of output sequences of weight h, $A_h^{(o)}(n)$, is upper bounded by

$$A_h^{(o)}(n) \leq \sum_{r=1}^{\lfloor h/d \rfloor} \binom{n/\tau}{r} T_h^{(r)}. \tag{3B.1}$$

The transfer function, $T(D)$, of a CC is a formal power series which enumerates all detours by weight, and is given by

$$T(D) = \sum_{h=1}^{\infty} T_h D^h,$$

where T_h is the number of distinct detours of weight h. Using basic combinatorics, the formal power series which enumerates distinct r-tuples of detours by total weight is given by

$$[T(D)]^r = \sum_{h=1}^{\infty} T_h^{(r)} D^h,$$

where $T_h^{(r)}$ is the number of ways of independently choosing r detours which have total weight h.

Using these definitions, it is clear that $T(D)$ will be analytic in the neighborhood of $D = 0$ and therefore have a Taylor series which converges for all $D < D_0$, where D_0 is the radius of convergence. Since expansion will also be non-negative and $T_d > 0$, it is also clear that $T(D)$ is monotonic increasing for all $D < D_0$. So we can upper bound $T_h^{(r)}$ using standard asymptotic methods. Starting with

$$[T(D)]^r = \sum_{i=1}^{\infty} T_i^{(r)} D^i \geq T_h^{(r)} D^h,$$

we can rearrange terms to get

$$T_h^{(r)} \leq [T(D)]^r D^{-h}. \tag{3B.2}$$

Let D^* be the unique real positive root of the equation $T(D) = 1$ in the domain $0 < D < D_0$. Since (3B.2) holds for any $0 < D \leq D_0$, we choose $D = D^*$ to get the final bound

$$T_h^{(r)} \leq \left(\frac{1}{D^*}\right)^h. \tag{3B.3}$$

Combining (3B.1) and (3B.3) gives the bound

$$A_h^{(o)}(n) \leq \sum_{t=1}^{\lfloor h/d \rfloor} \binom{n/\tau}{t} \left(\frac{1}{D^*}\right)^h.$$

This bound is generally quite useful in the small output weight regime (e.g., $h \leq dn/(2\tau)$). It does become quite weak for larger output weights, however. We note that the trivial bound, $A_h^{(o)}(n) \leq 2^{nR}$, where R is the rate of the CC, may improve the bound somewhat for large output weights.

The bound on $B_h^{(o)}(n)$ follows from combining our bound on $A_h^{(o)}(n)$ with a bound on input weight, w, for a given output weight, h. Let ρ be the smallest number such that the input weight, w, satisfies $w \leq \rho h$ for all codewords. Since every codeword can be represented by a closed cycle in the state diagram of the encoder, the constant ρ can be computed by finding the maximum value of w/h over all cycles in the state diagram with $w > 0$. If the encoder is non-catastrophic, then $\rho < \infty$ because there will be no cycles with $h = 0$ and $w > 0$. We note that finding ρ is a standard combinatorial optimization problem known as the minimum cycle ratio problem [7]. Starting with (3.2.3), it is easy to verify that

$$B_h^{(o)}(n) = \sum_{w=1}^{k} \frac{w}{k} A_{w,h}^{(o)}(n) \leq \frac{\rho h}{k} A_h^{(o)}(n).$$

Substituting the WE bound for $A_h^{(o)}(n)$ completes the proof. □

3B.2 Proof of Corollary 3.3.2

Proof of Corollary 3.3.2. We start by using (3A.7) to upper bound the binomial sum in (3.3.1). We define the result as

$$f(h,n) = \frac{(n/\tau + 1)^{\lfloor h/d \rfloor}}{\lfloor h/d \rfloor !} g^h,$$

where $g = 1/D^*$. At first, it seems rather straightforward that

$$A_h^{(o)}(n) \leq f(h,n), \tag{3B.4}$$

because we have simply upper bounded the binomial sum. Unfortunately, the binomial sum bound, (3A.7), is designed for cases where the second argument is less than the first. For $f(h,n)$, this corresponds to the condition that $\lfloor h/d \rfloor \leq n/\tau$. If $d \geq \tau$, this means that (3B.4) holds for the entire range, $1 \leq h \leq n$. If $d < \tau$, we can show, with the aid of a few additional assumptions, that (3B.4) also holds for $1 \leq h \leq n$.

We start by noting that (3B.4) actually holds for $1 \leq h \leq h^*$, with $h^* = 1.72dn/\tau$, because (3A.7) holds for $k \leq 1.72n$. Let R be rate of the CC, and recall that we always have

the trivial upper bound $A_h^{(o)}(n) \leq 2^{nR}$. So, if we can show that $f(h,n) \geq 2^{nR}$ for $h^* \leq h \leq n$, then this implies that (3B.4) holds for $1 \leq h \leq n$. Indeed, we show that $f(h,n) \geq 2^{nR}$ for $h^* \leq h \leq n$ by showing that $f(h^*,n) \geq 2^{nR}$ and $f(n,n) \geq 2^{nR}$ and then using the concavity of $f(h,n)$ in h for fixed n.

First, we show that $f(h^*,n) \geq 2^{nR}$ follows from the assumption that $2^{1/\tau} g^{1.72d/\tau} \geq 2^R$. We begin by raising the LHS to the nth power and noting that

$$\frac{(n/\tau+1)^{h^*/d}}{\Gamma(h^*/d+1)} g^{h^*} \geq 2^{n/\tau} g^{1.72dn/\tau}$$

because $T(n,1.72n) \geq 2^n$. Since the LHS is a decreasing function of h^*/d in this range (i.e., $h^*/d \geq n/\tau$), we also have the bound

$$f(h^*,n) = \frac{(n/\tau+1)^{\lfloor h^*/d \rfloor}}{\lfloor h^*/d \rfloor!} \geq \frac{(n/\tau+1)^{h^*/d}}{\Gamma(h^*/d+1)} g^{h^*}.$$

Combining these bounds gives the desired result of $f(h^*,n) \geq 2^{nR}$.

Assuming that $(de/\tau)^{1/d} \left(\sqrt{2\pi n}\right)^{-1/n} g \geq 2^R$, we show now that $f(n,n) \geq 2^{nR}$. We begin by raising the first expression to the nth power and noting that

$$\frac{(n/\tau+1)^{n/d}}{\sqrt{2\pi n}\left(\frac{n}{de}\right)^{n/d}} g^n \geq \frac{\left(\frac{n}{\tau}\right)^{n/d}}{\sqrt{2\pi n}\left(\frac{de}{n}\right)^{-n/d}} g^n = \frac{\left(\frac{de}{\tau}\right)^{n/d}}{\sqrt{2\pi n}} g^n \geq 2^{nR}.$$

Using the fact that $\Gamma(n+1) \leq \sqrt{2\pi n}(n/e)^n$, we substitute terms to get

$$\frac{(n/\tau+1)^{n/d}}{\Gamma(n/d+1)} g^n \geq \frac{(n/\tau+1)^{n/d}}{\left(\frac{n}{de}\right)^{n/d}} g^n.$$

Since the LHS is a decreasing function of n/d in this range (i.e., $n/d \geq n/\tau$), we also have the bound

$$f(n,n) = \frac{(n/\tau+1)^{\lfloor n/d \rfloor}}{\lfloor n/d \rfloor!} g^n \geq \frac{(n/\tau+1)^{n/d}}{\Gamma(n/d+1)} g^n.$$

Combining these bounds gives the desired result of $f(n,n) \geq 2^{nR}$. This completes the proof of the WE bound.

Using the WE bound to upper bound the bit normalized WE, $B_h^{(o)}(n)$, gives

$$B_h^{(o)}(n) \leq \frac{\rho h}{k} \frac{(n/\tau+1)^{\lfloor h/d \rfloor}}{\lfloor h/d \rfloor!} g^h = \frac{\rho}{R\tau} \frac{n+\tau}{n} \frac{h}{\lfloor h/d \rfloor} \frac{(n/\tau+1)^{\lfloor h/d \rfloor - 1}}{(\lfloor h/d \rfloor - 1)!} g^h.$$

For $h \geq d \geq 2$, we use the bound, $h/\lfloor h/d \rfloor \leq 2d$, to obtain (3.3.4). This completes the proof. $\square$

3B.3 Proof of Corollary 3.3.3

Proof of Corollary 3.3.3. We start by using (3A.6) to upper bound the binomial sum in (3.3.1). Let $f(h,n)$ be the resulting bound, which gives

$$f(h,n) = \left(\frac{ne/\tau}{\lfloor h/d \rfloor}\right)^{\lfloor h/d \rfloor} g^h,$$

where $g = 1/D^*$. At first, it seems rather straightforward that

$$A_h^{(o)}(n) \leq f(h,n), \tag{3B.5}$$

because we have simply upper bounded the binomial sum. Unfortunately, the binomial sum bound, (3A.6), is designed for cases where the second argument is less than the first. For $f(h,n)$, this corresponds to the condition that $\lfloor h/d \rfloor \leq n/\tau$. If $d \geq \tau$, this means that (3B.5) holds for the entire range, $1 \leq h \leq n$. If $d < \tau$, we can show, with the aid of a few additional assumptions, that (3B.5) also holds for $1 \leq h \leq n$.

We start by noting that (3B.4) actually holds for $1 \leq h \leq h^*$, with $h^* = 1.88dn/\tau$, because (3A.6) holds for $k \leq 1.88n$. Let R be rate of the CC, and recall that we always have the trivial upper bound $A_h^{(o)}(n) \leq 2^{nR}$. So, if we can show that $f(h,n) \geq 2^{nR}$ for $h^* \leq h \leq n$, then this implies that (3B.4) holds for $1 \leq h \leq n$. Indeed, we show that $f(h,n) \geq 2^{nR}$ for $h^* \leq h \leq n$ by showing that $f(h^*,n) \geq 2^{nR}$ and $f(n,n) \geq 2^{nR}$ and then using the concavity of $f(h,n)$ in h for fixed n.

First, we show that $f(h^*,n) \geq 2^{nR}$ follows from the assumption that $2^{1/\tau} g^{1.88d/\tau} \geq 2^R$. We begin by raising the LHS to the nth power and noting that

$$\left(\frac{ne/\tau}{h^*/d}\right)^{h^*/d} g^{h^*} \geq 2^{n/\tau} g^{1.88dn/\tau}$$

because $(ne/(1.88n))^{1.88n} \geq 2^n$. Since the LHS is a decreasing function of h^*/d in this range (i.e., $h^*/d \geq n/\tau$), we also have the bound

$$f(h^*,n) = \left(\frac{ne/\tau}{\lfloor h^*/d \rfloor}\right)^{\lfloor h^*/d \rfloor} g^{h^*} \geq \left(\frac{ne/\tau}{h^*/d}\right)^{h^*/d} g^{h^*}.$$

Combining these bounds gives the desired result of $f(h^*,n) \geq 2^{nR}$.

Next, we show that $f(n,n) \geq 2^{nR}$ follows from the assumption that $(de/\tau)^{1/d} g \geq 2^R$. We begin by raising the LHS to the nth power and noting that

$$\left(\frac{ne}{\tau}\right)^{n/d} \left(\frac{d}{n}\right)^{n/d} g^n = \left(\frac{de}{\tau}\right)^{n/d} g^n \geq 2^{nR}.$$

Since the LHS is a decreasing function of n/d in this range (i.e., $n/d \geq n/\tau$), we also have the bound

$$f(n,n) = \left(\frac{ne/\tau}{\lfloor n/d \rfloor}\right)^{\lfloor n/d \rfloor} g^n \geq \left(\frac{ne}{\tau}\right)^{n/d} \left(\frac{d}{n}\right)^{n/d} g^n.$$

Combining these bounds gives the desired result of $f(n,n) \geq 2^{nR}$.

Finally, we can simplify the form of $f(h,n)$ by letting $h = i\lfloor h/d \rfloor + r$ and noting that

$$\left(\frac{ne/\tau}{\lfloor h/d \rfloor}\right)^{\lfloor h/d \rfloor} \left(\frac{n}{h}\right)^{-\lfloor h/d \rfloor} \left(\frac{de}{\tau}\right)^{-h/d} = \left(1 + \frac{r}{di}\right)^i \left(\frac{\tau}{de}\right)^{r/d} \leq \left(\frac{\tau}{d}\right)^{(d-1)/d}$$

for $i \geq 1$ (i.e., $h \geq d$). Using this to upper bound $\left(\frac{ne/\tau}{\lfloor h/d \rfloor}\right)^{\lfloor h/d \rfloor}$ gives

$$A_h^{(o)}(n) \leq C \left(\frac{n}{h}\right)^{\lfloor h/d \rfloor} g^h,$$

where $C = \left(\frac{\tau}{d}\right)^{(d-1)/d}$ and $g = \left(\frac{1}{D^*}\right)\left(\frac{de}{\tau}\right)^{1/d}$. This completes the proof of the WE bound.

Using the WE bound to upper bound the bit normalized WE, $B_h^{(o)}(n)$, gives

$$B_h^{(o)}(n) \leq \frac{\rho n}{k} C \left(\frac{n}{h}\right)^{\lfloor h/d \rfloor} g^h = \frac{\rho}{R} C \left(\frac{n}{h}\right)^{\lfloor h/d \rfloor - 1} g^h,$$

and proves (3.3.6). □

3B.4 Proof of Theorem 3.3.6

Proof of Theorem 3.3.6. After treating this problem as a generalization of Gallager's Chernov bounding technique for LDPC codes [12, Eqn. 2.12], a literature search turned up a very mathematical and complete treatment by Miller [21]. We retain our proof of the upper bound since it treats the problem from a coding perspective. For the lower bound and convexity, we refer the reader to [21].

Let $\mathbf{A}(x,p)$ be the state transition matrix for p steps through the trellis be defined by $\mathbf{G}(x)$. It is well-known that trellis sections may be combined by multiplying state transition matrices, and this gives

$$\begin{aligned} \mathbf{A}(x,p) &= [\mathbf{G}(x)]^p \\ &= \sum_{h \geq 0} \mathbf{A}_h(p) x^h, \end{aligned} \tag{3B.6}$$

where each $\mathbf{A}_h(p)$ is an $M \times M$ non-negative matrix. For any $x \geq 0$, we can lower bound (3B.6) by any single term in the sum with $\mathbf{A}_h(p)x^h \leq \sum_{i \geq 0} \mathbf{A}_i(p)x^i$. Solving for $\mathbf{A}_h(p)$ and rearranging terms gives the element-wise matrix inequality

$$\mathbf{A}_h(p) \leq x^{-h} \left[\mathbf{G}(x)\right]^p . \tag{3B.7}$$

One can construct a block code from a CC in a number of ways. Two common methods which preserve the free distance of the code (as the minimum distance of the block code) are trellis termination and trellis tail-biting. We denote the WEs of these two methods by $A_h^{TE}(p)$ and $A_h^{TB}(p)$ respectively, and point out that

$$A_h^{TE} = \left[\mathbf{A}_h(p)\right]_{11} \leq A_h^{TB} = \sum_{i=1}^{M} \left[\mathbf{A}_h(p)\right]_{ii} = Tr\left(\mathbf{A}_h(p)\right). \tag{3B.8}$$

Let $\lambda_i(x)$ be ith eigenvalue of $\mathbf{G}(x)$ in decreasing order by modulus (for $i = 1, \ldots, M$). Using the well-known eigenvalue-sum formula for the trace, we can combine (3B.7) and (3B.8) to get

$$A_h^{TB}(p) \leq Tr\left(x^{-h}\left[\mathbf{G}(x)\right]^p\right) = x^{-h} \sum_{i=1}^{M} \left(\lambda_i(x)\right)^p .$$

Now, we can upper bound the spectral shape with

$$r^{CC}(\delta) \leq \lim_{p \to \infty} \frac{1}{\tau p} \ln A_{\delta n}^{TB}(p).$$

This limit can be evaluated by writing

$$\ln \sum_{i=1}^{M} \left(\lambda_i(x)\right)^p = p \ln \lambda_1(x) + \ln\left(1 + \sum_{i=2}^{M} \left(\frac{\lambda_i(x)}{\lambda_1(x)}\right)^p\right),$$

and noting that the last term is $o(1)$ because $\lambda_1(x) > \lambda_i(x)$ for $i = 2, \ldots, M$. Using that fact results in the upper bound,

$$r^{CC}(\delta) \leq \frac{1}{\tau} \ln \lambda_1(x) - \delta \ln x.$$

This upper bound is valid for any $x > 0$ and can be minimized over x. Setting the derivative with x equal to zero and solving gives

$$\delta(x) = \frac{x\lambda_1'(x)}{\tau \lambda_1(x)},$$

and concludes the proof of the upper bound. □

3C "Accumulate" Code Bounds

3C.1 Lemma 3C.1 and Theorem 3C.2

Lemma 3C.1. *The n term Riemann sum of a function, $f(x)$, on the interval $[a,b]$ is given by*

$$R_n = \frac{b-a}{n}\sum_{i=0}^{n-1} f\left(a + i\frac{b-a}{n}\right). \tag{3C.1}$$

If $f(x)$ is convex and non-decreasing on the interval $[a,b]$, then the sequence $\{R_n\}_{n\geq 1}$ is also non-decreasing. Furthermore, if $f(x)$ is concave and non-increasing on the interval $[a,b]$, then the sequence $\{R_n\}_{n\geq 1}$ is non-increasing

Proof. Using convexity and the fact that $\frac{n-i}{n}\frac{i}{n+1} + \frac{i}{n}\frac{i+1}{n+1} = \frac{i}{n}$, we have

$$f\left(a + i\frac{b-a}{n}\right) \leq \frac{n-i}{n} f\left(a + i\frac{b-a}{n+1}\right) + \frac{i}{n} f\left(a + (i+1)\frac{b-a}{n+1}\right).$$

Now, we can upper bound R_n with a linear combination of $f\left(a + i\frac{b-a}{n+1}\right)$ to get

$$R_n \leq \frac{b-a}{n}\sum_{i=0}^{n-1} \frac{n-i}{n} f\left(a + i\frac{b-a}{n+1}\right) + \frac{i}{n} f\left(a + (i+1)\frac{b-a}{n+1}\right).$$

Rearranging the terms in the sum gives

$$R_n \leq \frac{b-a}{n}\left[\frac{f(a)}{n} + \sum_{i=0}^{n} \frac{n-1}{n} f\left(a + i\frac{b-a}{n+1}\right)\right]. \tag{3C.2}$$

Since $f(x)$ is non-decreasing, we can upper bound $f(a)$ with

$$f(a) \leq \frac{1}{n+1}\sum_{i=0}^{n} f\left(a + i\frac{b-a}{n+1}\right). \tag{3C.3}$$

Substituting the RHS of (3C.3) for $f(a)$ in (3C.2) and rearranging terms gives

$$R_n \leq \frac{b-a}{n+1}\sum_{i=0}^{n} f\left(a + i\frac{b-a}{n+1}\right) = R_{n+1}.$$

This completes the proof for $f(x)$ convex and non-decreasing.

If $f(x)$ is concave and non-increasing on the interval $[a,b]$, then $-f(x)$ is convex and non-decreasing on the same interval. In this case, the original proof can be used to show that $-R_n \leq -R_{n+1}$. Therefore, the sequence is non-increasing. □

Theorem 3C.2. *Let a, b, i, j be integers obeying $0 \leq i \leq a$ and $0 \leq j \leq b$. We have the following inequality,*

$$\frac{\binom{a}{i}\binom{b}{j}}{\binom{a+b}{i+j}} \leq \left(\frac{a}{a+b}\right)^i \left(\frac{b}{a+b}\right)^j \left(\frac{i+j}{i}\right)^i \left(\frac{i+j}{j}\right)^j.$$

In the case of $a = 0$ or $b = 0$, we use the convention that $0^0 = 1$ so that the expression remains well-defined.

Proof. We start by expanding the binomial coefficients in terms of factorials and rearranging terms to get

$$\frac{(a)_i (b)_j}{(a+b)_{i+j}} \frac{(a+b)^{i+j}}{a^i b^j} \leq \frac{(i)_i (j)_j}{(i+j)_{i+j}} \frac{(i+j)^{i+j}}{i^i j^j},$$

where the falling factorial is defined by $(a)_i = a(a-1)\cdots(a-i+1)$. Next, we define the function

$$f_{ij}(a,b) = \frac{(a)_i (b)_j}{(a+b)_{i+j}} \frac{(a+b)^{i+j}}{a^i b^j}$$

for real numbers a, b satisfying $a \geq i$ and $b \geq j$. It is easy to verify that the original inequality is equivalent to the statement $f_{ij}(a,b) \leq f_{ij}(i,j)$. Since $f_{ij}(a,b) = f_{ji}(b,a)$, we assume that $a \geq bi/j$ without loss of generality. We proceed by showing that $f_{ij}(ci, cj)$ is non-increasing for $c \geq 1$ and that $f_{ij}(a,b)$ is non-increasing for $a \geq bi/j$. Since the logarithm preserves order, we will actually consider the logarithm of the function,

$$\log f_{ij}(a,b) = \sum_{x=0}^{i-1} \log\left(\frac{a-x}{a}\right) + \sum_{y=0}^{j-1} \log\left(\frac{b-y}{b}\right) - \sum_{z=0}^{i+j-1} \log\left(\frac{a+b-z}{a+b}\right).$$

First, we show that the derivative of $\log f_{ij}(ci, cj)$ with respect to c is negative for all $c \geq 1$. We start by noting that

$$c\frac{\partial}{\partial c} \log f_{ij}(ci, cj) = \sum_{x=0}^{i-1} \frac{x}{ci - x} + \sum_{y=0}^{j-1} \frac{y}{cj - y} - \sum_{z=0}^{i+j-1} \frac{z}{ci + cj - z}. \tag{3C.4}$$

Now, we note that the first sum can be written as

$$\sum_{x=0}^{i-1} \frac{x}{ci - x} = \sum_{x=0}^{i-1} \frac{x/i}{c - x/i} = iR_i,$$

where R_n is given by (3C.1) with $f(x) = x/(c-x)$, $a = 0$, and $b = 1$. In fact, each sum in (3C.4) can be rewritten in this form to give

$$c\frac{\partial}{\partial c}\log f_{ij}(ci, cj) = iR_i + jR_j - (i+j)R_{i+j},$$

and rearranging terms gives

$$c\frac{\partial}{\partial c}\log f_{ij}(ci, cj) = i(R_i - R_{i+j}) + j(R_j - R_{i+j}).$$

Since $g(x)$ is convex and increasing for $x \in [0, 1)$ and $c \geq 1$, Lemma 3C.1 shows that R_n is non-decreasing. Therefore, the derivative is upper bounded by zero and $\log f_{ij}(ci, cj)$ is non-increasing for all $c \geq 1$.

Next, we show that the derivative of $\log f_{ij}(a, b)$ with respect to a is negative for $a \geq bi/j$. We start by noting that

$$\frac{\partial}{\partial a}\log f_{ij}(a, b) = \sum_{x=0}^{i-1}\frac{x}{a(a-x)} - \sum_{z=0}^{i+j-1}\frac{z}{(a+b)(a+b-z)}.$$

Since $z/(a+b-z)$ is convex and increasing for $z \in [0, a+b)$ and $i \leq i+j$, Lemma 3C.1 shows that

$$\sum_{z=0}^{i+j-1}\frac{z}{(a+b)(a+b-z)} \geq \frac{i+j}{i}\sum_{z=0}^{i-1}\frac{z(i+j)/i}{(a+b)(a+b-z(i+j)/i)} = \sum_{z=0}^{i-1}\frac{z}{c(c-z)},$$

with $c = (a+b)i/(i+j)$. Incorporating this bound gives

$$\frac{\partial}{\partial a}\log f_{ij}(a, b) \leq \sum_{x=0}^{i-1}\frac{x}{a(a-x)} - \sum_{z=0}^{i-1}\frac{z}{c(c-z)}.$$

The RHS of this expression will be non-positive as long as $a \geq c$ (or equivalently $a \geq bi/j$). Therefore, we have shown that $\log f_{ij}(a, b)$ is non-increasing for $a \geq bi/j$.

The conclusion of the theorem follows from the inequality,

$$f_{ij}(a, b) \leq f_{ij}(bi/j, b) \leq f_{ij}(i, j),$$

where the RHS holds because $f_{ij}(ci, cj)$ is non-increasing for $c \geq 1$ and the LHS holds because $f_{ij}(a, b)$ is non-increasing for $a \geq bi/j$. This completes the proof. □

3C.2 Proof of Corollary 3.4.2

Proof of Corollary 3.4.2. This inequality can be verified by hand for the cases of $h \geq w = 0$ and $w \geq h = 0$. For $w \geq 1$ and $h \geq 1$, we start with (3.4.1) and note that

$$P_{w,h}(n) = \frac{\binom{n-h}{\lfloor w/2 \rfloor}\binom{h-1}{\lceil w/2 \rceil - 1}}{\binom{n}{w}} = \frac{\binom{n-h}{\lfloor w/2 \rfloor}\binom{h}{\lceil w/2 \rceil}\frac{\lceil w/2 \rceil}{h}}{\binom{n}{w}}.$$

Applying Theorem 3C.2 to this the RHS gives

$$P_{w,h}(n) \leq \frac{\lceil w/2 \rceil}{h}\left(\frac{n-h}{n}\right)^{\lfloor w/2 \rfloor}\left(\frac{h}{n}\right)^{\lceil w/2 \rceil}\left(\frac{w}{\lfloor w/2 \rfloor}\right)^{\lfloor w/2 \rfloor}\left(\frac{w}{\lceil w/2 \rceil}\right)^{\lceil w/2 \rceil},$$

and the log-sum inequality can be used to show that

$$\left(\frac{w}{\lfloor w/2 \rfloor}\right)^{\lfloor w/2 \rfloor}\left(\frac{w}{\lceil w/2 \rceil}\right)^{\lceil w/2 \rceil} \leq 2^w.$$

Since $h \geq \lceil w/2 \rceil$ whenever $P_{w,h}(n) > 0$, dropping the $\lceil w/2 \rceil / h$ only weakens the bound. This completes the proof. □

3C.3 Proof of Corollary 3.4.4

Proof of Corollary 3.4.4. This inequality can be verified by hand for the cases of $h \geq w = 0$ and $w > h = 0$. For $w \geq 1$ and $h = 1$, the sum has no effect and we must simply verify that

$$P_{w,1}(n) \leq 2^w\left(\frac{1}{n}\right)^{\lceil w/2 \rceil}.$$

This result is easily reproduced by combining (3.4.3) with the fact that $((n-h)/n)^{\lfloor w/2 \rfloor} \leq 1$.

For $w \geq 1$ and $h \geq 2$, we start by writing (3.4.1) as

$$P_{w,h}(n) = \frac{\binom{n-h}{\lfloor w/2 \rfloor}\binom{h-1}{\lceil w/2 \rceil - 1}}{\frac{n}{w}\binom{n-1}{w-1}}$$

because

$$\binom{n}{w} = \binom{n-1}{w-1}\frac{n}{w}.$$

Applying Theorem 3C.2 to this upper bound gives

$$P_{w,h}(n) \leq \frac{w}{n}\left(\frac{n-h}{n-1}\right)^{\lfloor w/2 \rfloor}\left(\frac{h-1}{n-1}\right)^{\lceil w/2 \rceil - 1}\left(\frac{w-1}{\lfloor w/2 \rfloor}\right)^{\lfloor w/2 \rfloor - 1}\left(\frac{w-1}{\lceil w/2 \rceil - 1}\right)^{\lceil w/2 \rceil - 1},$$

and the log-sum inequality can be used to show that

$$\left(\frac{w-1}{\lfloor w/2 \rfloor}\right)^{\lfloor w/2 \rfloor - 1} \left(\frac{w-1}{\lceil w/2 \rceil - 1}\right)^{\lceil w/2 \rceil - 1} \leq 2^{w-1}.$$

Next, we note that

$$\frac{n-h}{n-1} \leq \frac{n-1}{n},$$

for $h \geq 2$. This means that the cumulative IOWTP can be upper bounded by

$$P_{w,\leq h}(n) \leq \sum_{i=1}^{h} \frac{w}{n} 2^{w-1} \left(\frac{h-1}{n}\right)^{\lceil w/2 \rceil - 1},$$

for $w \geq 1$ and $h \geq 2$. Since x^k is strictly increasing with x, the sum can be upper bounded with

$$\sum_{i=1}^{z} (i-1)^k = \sum_{i=1}^{z-1} i^k \leq \int_1^z x^k dx \leq \frac{1}{k+1} z^{k+1}.$$

Finally, we have

$$P_{w,\leq h}(n) \leq \frac{w}{\lceil w/2 \rceil} 2^{w-1} \left(\frac{h}{n}\right)^{\lceil w/2 \rceil},$$

which is easily reduced to (3.4.5) by noting that $w/\lceil w/2 \rceil \leq 2$. □

3C.4 Proof of Corollary 3.4.5

Proof of Corollary 3.4.5. Combining the definition of $P^{(m)}_{h_1,\leq h}(n)$ with the standard formula for serial concatenation through a random interleaver, we get

$$P^{(m)}_{h_1,\leq h}(n) = \sum_{h_2,\ldots,h_{m-1}}^{n} \sum_{h_m=1}^{h} \prod_{i=1}^{m} P_{h_i,h_{i+1}}(n).$$

Using Fact 3.4.1, we can see that all non-zero terms must obey $h_{i+1} \geq \lceil h_i/2 \rceil$ for $i = 1, \ldots, m$. Furthermore, we can upper bound each $P_{h_i,h_{i+1}}(n)$ with $P_{h_i,\leq h_{i+1}}(n)$ and drop the sum over h_m to get

$$P^{(m)}_{h_1,\leq h_m}(n) \leq \sum_{h_2=\lceil h_1/2 \rceil}^{2h_3} \cdots \sum_{h_i=\lceil h_{i-1}/2 \rceil}^{2h_{i+1}} \cdots \sum_{h_{m-1}=\lceil h_{m-2}/2 \rceil}^{2h_m} \prod_{i=1}^{m} \left(\frac{4h_{i+1}}{n}\right)^{\lceil h_i/2 \rceil}.$$

Since all non-zero terms have $h_{i+1} \geq \lceil h_i/2 \rceil$ for $i = 1, \ldots, m$, we have the inductive upper bound $h_i \leq 2^{m+1-i} h_{m+1}$ for non-zero terms. For simplicity, we apply the weaker bound, $h_i \leq 2^{m-1} h_{m+1}$ for $i = 2, \ldots, m$, to get

$$P^{(m)}_{h_1, \leq h_m}(n) \leq \sum_{h_2 = \lceil h_1/2 \rceil}^{2h_3} \cdots \sum_{h_i = \lceil h_{i-1}/2 \rceil}^{2h_{i+1}} \cdots \sum_{h_{m-1} = \lceil h_{m-2}/2 \rceil}^{2h_m} \prod_{i=1}^{m} \left(\frac{2^{m+1} h_{m+1}}{n} \right)^{\lceil h_i/2 \rceil}.$$

Each sum in this expression is essentially a geometric sum which can be upper bounded using

$$\sum_{h_i = \lceil h_{i-1}/2 \rceil}^{2h_{i+1}} \left(\frac{2^{m+1} h_{m+1}}{n} \right)^{\lceil h_i/2 \rceil} \leq 2 \frac{\left(2^{m+1} h_{m+1}/n\right)^{\lceil h_{i-1}/2 \rceil}}{1 - 2^{m+1} h_{m+1}/n},$$

for $h_{m+1} < n/2^{m+1}$. We note that the troublesome $\lceil h_i/2 \rceil$ is handled by repeating each term twice and therefore results in the factor of 2. Applying this bound to the $m-1$ sums results in the expression (3.4.6). □

3D Proof of CA^m Code Bounds

3D.1 WE Bounds for the IGE Conjecture

We use upper and lower bounds to evaluate the limit, $\lim_{n \to \infty} \log_n P_{w,h}(n)$, where $P_{w,h}(n)$ is defined by (3.4.1). Applying (3A.1) to $P_{w,h}(n)$ gives the upper and lower bounds

$$\frac{\left(\frac{(n-h)}{\lfloor w/2 \rfloor}\right)^{\lfloor w/2 \rfloor} \left(\frac{(h-1)}{\lceil w/2 \rceil - 1}\right)^{\lceil w/2 \rceil - 1}}{\left(\frac{ne}{w}\right)^{w}} \leq P_{w,h}(n) \leq \frac{\left(\frac{(n-h)e}{\lfloor w/2 \rfloor}\right)^{\lfloor w/2 \rfloor} \left(\frac{(h-1)e}{\lceil w/2 \rceil - 1}\right)^{\lceil w/2 \rceil - 1}}{\left(\frac{n}{w}\right)^{w}}.$$

Computing the limit of $\log_n$ of these upper and lower bounds is simplified by noticing that all terms not involving n will vanish. Taking only these non-zero terms shows that the two bounds are identical and equal to

$$\lfloor w/2 \rfloor \left(\lim_{n \to \infty} \log_n (n-h) \right) - w = - \lceil w/2 \rceil .$$

Now, consider the limit, $\lim_{n \to \infty} A_h(n)$, where $A_h(n)$ is the WE of a TCC. Using the upper bound, (3.3.1), we can upper bound the limit of $\log_n$ with

$$\lim_{n \to \infty} \log_n A_h(n) \leq \lim_{n \to \infty} \log_n \binom{n/\tau}{\lfloor h/d \rfloor} = \lfloor h/d \rfloor .$$

If we assume that h is an integer multiple of d, then we can also lower bound the number of codewords of weight h in a TCC. We start by assuming that each codeword consists of exactly $\lfloor h/d \rfloor$ minimum distance detours. The number of ways to choose starting positions on these detours is greater than

$$\binom{n/\tau - h}{\lfloor h/d \rfloor}$$

because there are at least $n/\tau - h$ unused trellis steps. This gives a lower bound on the limit of $\log_n$ which is equal to the upper bound.

3D.2 Proof of Lemma 3.6.2

Proof of Lemma 3.6.2. For any integer $h_1 \geq 0$, it is clear that the function $\alpha(h_1, \ldots, h_{m+1}) = \lfloor h_1/d \rfloor - \sum_{i=1}^{m} \lceil h_i/2 \rceil$ is maximized by minimizing $h_2, \ldots, h_m$. Let $\tilde{h}_1, \ldots, \tilde{h}_{m+1}$ be some (but not any) weight path which maximizes the function. Since the maximization is performed over the set of valid weight paths starting at h_1, this means that $\tilde{h}_2, \ldots, \tilde{h}_m$ can be determined by the constraints and that $\tilde{h}_{i+1} = \left\lceil \tilde{h}_i/2 \right\rceil$ for $i = 1, \ldots, m-1$. Using the fact that $\lceil \lceil x/2 \rceil /2 \rceil = \lceil x/4 \rceil$, this can be inductively reduced to $\tilde{h}_i = \left\lceil \tilde{h}_1/2^i \right\rceil$. Therefore, rewriting $\alpha(h_1, \ldots, h_{m+1})$ as a function of h_1 with $h_{i+1} = \lceil h_i/2 \rceil$, for $i = 1, \ldots, m-1$, gives

$$\nu(h_1) = \lfloor h_1/d \rfloor - \sum_{i=1}^{m} \left\lceil h_1/2^i \right\rceil,$$

which is the maximum as a function of h_1.

Now, we consider the maximum of $\nu(h_1)$ for $h_1 \geq 2$. Suppose we start with $h_1 = id$ (i.e., at some integer multiple of d) and consider the sequence $h_1 = id, id+1, \ldots, id+d-1$. Each increase by one cannot increase $\nu(h_1)$ because the positive term is non-increasing while the negative terms are non-decreasing. Now, we can try increasing h_1 by integer multiple of d. In this case, the positive term increases by one while the negative sum contributes a change of $\lceil id/2 \rceil - \lceil (i+1)d/2 \rceil$. For $d \geq 2$ even, it is easy to verify that $\lceil id/2 \rceil - \lceil (i+1)d/2 \rceil = -d/2 \leq -1$. For $d \geq 3$ odd, it is also easy to verify that $\lceil id/2 \rceil - \lceil (i+1)d/2 \rceil \leq -1$. Choosing $i = 1$ as our starting point, this implies that $\nu(h_1) \leq \nu(d)$. This completes the proof that the maximum of $\nu(h_1) = \nu(d)$ for $h_1 \geq 2$. It is also worth noting that $\tilde{h}_{m+1}$ is not constrained by this maximization because it does not appear in $\alpha(h_1, \ldots, h_{m+1})$.

Now, we would like to show, for $d \geq 3$ or $m \geq 2$, that $\nu(4d) \leq \nu(d) - 1$. This will be useful for bounding the number of terms which achieve the maximum exponent of $\nu(d)$. We note that this does not hold for $d = 2$ and $m = 1$, however, because $\nu(h)$ achieves the maximum of zero if h is even.

For $m \geq 2$, we show that $\nu(4d) \leq \nu(d) - 1$ by writing

$$\nu(4d) - \nu(d) = (4 - 1) + \sum_{j=1}^{m} \left\lceil d/2^j \right\rceil - \sum_{i=1}^{m} \left\lceil 4d/2^i \right\rceil .$$

Cancelling the terms where $i = j + 2$ gives

$$\nu(4d) - \nu(d) = 3 - 3d + \sum_{m-1}^{m} \left\lceil d/2^i \right\rceil .$$

For any $m \geq 2$ and $d \geq 2$, it can be verified that $\sum_{m-1}^{m} \left\lceil d/2^i \right\rceil \leq d$, and using this bound gives the final result,

$$\nu(4d) - \nu(d) \leq 3 - 2d \leq -1.$$

For $m = 1$ and $d \geq 3$, we start by writing

$$\nu(4d) - \nu(d) = 4 + \lceil d/2 \rceil - \lceil 4d/2 \rceil .$$

Next, we verify by hand that $\nu(4d) - \nu(d) \leq -1$ for $d = 3$. Applying the bound, $x \leq \lceil x \rceil \leq x + 1$, gives

$$\nu(4d) - \nu(d) \leq 4 - 3d/2,$$

which proves that $\nu(4d) - \nu(d) \leq -1$ for $d \geq 4$. □

3D.3 Proof of Lemma 3.6.3

Proof of Lemma 3.6.3. This proof is based on sequentially choosing the random interleaver and counting the number of ways a minimum weight codeword may be produced during each choice. We start by pointing out that all TCCs have $\Theta(n)$ non-overlapping codewords of minimum weight. For example, if we let μ be the output length of the shortest detour of minimum weight, then there are at least n/μ non-overlapping codewords of minimum weight.

Now, consider all mappings of $d \geq 1$ bits through an "accumulate" code which result in the minimum output weight of $h = \lceil d/2 \rceil$. For d even, these mappings consist of breaking

the d bits into $d/2$ pairs of bits and placing these pairs independently. For d odd, the same basic process is used except that there is a leftover bit. This bit must be placed at the end of the block for the minimum output weight to occur.

Now, consider the sequential process of choosing the random interleaver. We assume that the process is applied to n/μ non-overlapping codewords of weight d. In the ith step, we choose the d bit positions, from the remaining unused positions, where the ith codeword of weight d will be mapped. Consider the event that the placement in ith step supports a minimum weight output given that no previous step has resulted in a minimum weight codeword. We denote this event as E_{i+1} and the overall probability that a minimum weight codeword is produced by these n/μ codewords is

$$P_M(n) = 1 - \prod_{i=0}^{n/\mu-1} \left(1 - Pr(E_i)\right). \tag{3D.1}$$

We can lower bound the probability $Pr(E_i)$ by counting the number of possible way it may occur. After i steps, exactly di bits have been placed and so there are exactly

$$\binom{n-di}{d}$$

ways to place the next w bits. Since a minimum weight output is only generated by breaking the input into pairs, we can lower bound the number of ways this may occur as well. Initially, there are exactly $n-1$ ways to place a pair of bits adjacent to each other. After i steps, there are still at least $n-2di-1$ ways to do this because each bit placed eliminates at most two possible pairs. The number of ways to place the $\lfloor d/2 \rfloor$ pairs can be computed in the same manner as a binomial coefficient, with the exception that each placed pair eliminates at most three of the total possible pairs. There are $\lfloor d/2 \rfloor !$ orders that the pairs may be placed in as well, so the number of ways to place $\lfloor d/2 \rfloor$ adjacent pairs is greater than

$$\frac{\prod_{k=0}^{\lfloor d/2 \rfloor -1}(n-2di-3k-1)}{\lfloor d/2 \rfloor !}.$$

Since the last bit position is special, we only allow the leftover bit to be placed in this position if there is still a chance that a minimum weight codeword may be created. This only reduces the number of ways a minimum distance output may be created and maintains the lower bound. The -1 in the last expression reflects this change and makes it valid for odd w as well,

since there is only one way to place the leftover bit in the last position. This gives the lower bound,

$$Pr(E_i) \geq \frac{\prod_{k=0}^{\lfloor d/2 \rfloor - 1}(n - 2di - 3k - 2)}{\binom{n-di}{d} \ \lfloor d/2 \rfloor !}.$$

Now, we can simplify this expression by weakening the bound to

$$Pr(E_i) \geq \frac{(n - 2di - 3\lfloor d/2 \rfloor + 1)^{\lfloor d/2 \rfloor}}{n^d} \geq \frac{(1/2)^{\lfloor d/2 \rfloor}}{n^{\lceil d/2 \rceil}}, \tag{3D.2}$$

for $i \leq n/4d + 2$. Combining (3D.1) and (3D.2) gives the lower bound

$$P_M(n) \geq 1 - \prod_{i=0}^{\min[n/4d, n/\mu]} \left(1 - \frac{(1/2)^{\lfloor d/2 \rfloor}}{n^{\lceil d/2 \rceil}}\right) = \Omega(n^{1-\lceil d/2 \rceil}).$$

□

Bibliography

[1] S. Aji, H. Jin, A. Khandekar, D. J. C. MacKay, and R. J. McEliece. BSC thresholds for code ensembles based on "typical pairs" decoding. In *Codes, Systems, and Graphical Models*, volume 123 of *the IMA Vol. in Math. and its Appl.*, pages 195–210. Springer, 2001.

[2] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv. Optimal decoding of linear codes for minimizing symbol error rate. *IEEE Trans. Inform. Theory*, 20(2):284–287, March 1974.

[3] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara. Analysis, design, and iterative decoding of double serially concatenated codes with interleavers. *IEEE J. Select. Areas Commun.*, 16(2):231–244, Feb. 1998.

[4] S. Benedetto and G. Montorsi. Unveiling turbo codes: Some results on parallel concatenated coding schemes. *IEEE Trans. Inform. Theory*, 42(2):409–428, March 1996.

[5] G. Cohen, S. Gaubert, and J.-P. Quadrat. Max-plus algebra and system theory: Where we are and where to go now. *Elsevier Annu. Rev. Control*, 23:207–219, 1999.

[6] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley, 1991.

[7] A. Dasdan, S. S. Irani, and R. K. Gupta. Efficient algorithms for optimum cycle mean and optimum cost to time ratio problems. In *Proc. 36th Design Automation Conf.*, pages 37–42, June 1999.

[8] D. Divsalar. A simple tight bound on error probability of block codes with application to turbo codes. *The Telecom. and Mission Oper. Progr. Rep.*, 42(139):1–35, Nov. 1999.

[9] D. Divsalar, S. Dolinar, H. Jin, and R. J. McEliece. AWGN coding theorems from ensemble weight enumerators. In *Proc. IEEE Int. Symp. Information Theory*, page 459, Sorrento, Italy, June 2000.

[10] D. Divsalar, H. Jin, and R. J. McEliece. Coding theorems for "turbo–like" codes. In *Proc. 36th Annual Allerton Conf. on Commun., Control, and Comp.*, pages 201–210, Monticello, IL, USA, Sept. 1998.

[11] D. Divsalar and F. Pollara. Serial and hybrid concatenated codes with applications. In *Proc. Int. Symp. on Turbo Codes & Related Topics*, pages 80–87, Brest, France, Sept. 1997.

[12] R. G. Gallager. *Low-Density Parity-Check Codes*. The M.I.T. Press, Cambridge, MA, USA, 1963.

[13] H. Jin. *Analysis and Design of Turbo-like Codes*. PhD thesis, Caltech, May 2001.

[14] H. Jin and R. J. McEliece. AWGN coding theorems for serial turbo codes. In *Proc. 37th Annual Allerton Conf. on Commun., Control, and Comp.*, pages 893–894, Monticello, IL, USA, Sept. 1999.

[15] H. Jin and R. J. McEliece. RA codes achieve AWGN channel capacity. In *13th International Symposium, AAECC-13*, pages 10–18, Honolulu, HI, USA, Nov. 1999.

[16] H. Jin and R. J. McEliece. Typical pairs decoding on the AWGN channel. In *Int. Symp. Inform. Theory and its Appl.*, volume 1, pages 180–183, Honolulu, HI, USA, Nov. 2000. IEEE.

[17] H. Jin and R. J. McEliece. Coding theorems for turbo code ensembles. *IEEE Trans. Inform. Theory*, 48(6):1451–1461, June 2002.

[18] N. Kahale and R. Urbanke. On the minimum distance of parallel and serially concatenated codes. In *Proc. IEEE Int. Symp. Information Theory*, page 31, Cambridge, MA, USA, Aug. 1998. IEEE.

[19] D. E. Knuth. Big omicron and big omega and big theta. *SIGACT News*, 8(2):18–24, April 1976.

[20] R. J. McEliece, D. J. C. MacKay, and J. Cheng. Turbo decoding as an instance of Pearl's "belief propagation" algorithm. *IEEE J. Select. Areas Commun.*, 16(2):140–152, Feb. 1998.

[21] H. D. Miller. A convexity property in the theory of random variables defined on a finite Markov chain. *Ann. Math. Stats.*, 32:1260–1270, Dec. 1961.

[22] M. Öberg and P. H. Siegel. Performance analysis of turbo-equalized dicode partial-response channel. In *Proc. 36th Annual Allerton Conf. on Commun., Control, and Comp.*, pages 230–239, Monticello, IL, USA, Sept. 1998.

[23] H. D. Pfister and P. H. Siegel. The serial concatenation of rate-1 codes through uniform random interleavers. In *Proc. 37th Annual Allerton Conf. on Commun., Control, and Comp.*, pages 260–269, Monticello, IL, USA, Sept. 1999.

[24] H. D. Pfister and P. H. Siegel. Coding theorems for generalized repeat accumulate codes. In *Int. Symp. Inform. Theory and its Appl.*, volume 1, pages 21–25, Honolulu, HI, USA, Nov. 2000. IEEE.

[25] T. J. Richardson, M. A. Shokrollahi, and R. L. Urbanke. Design of capacity-approaching irregular low-density parity-check codes. *IEEE Trans. Inform. Theory*, 27(1):619–637, Feb. 2001.

[26] T. J. Richardson and R. L. Urbanke. The capacity of low-density parity check codes under message-passing decoding. *IEEE Trans. Inform. Theory*, 47(2):599–618, Feb. 2001.

[27] I. Sason and S. Shamai (Shitz). Variations on the Gallager bounds, connections and applications. *IEEE Trans. Inform. Theory*, 48(12), Dec. 2002.

[28] R. M. Tanner. A recursive approach to low complexity codes. *IEEE Trans. Inform. Theory*, 27(5):533–547, Sept. 1981.

[29] J. van Mourik, D. Saad, and Y. Kabashima. Magnetization enumerator for LDPC codes - a statistical physics approach. In *Proc. IEEE Int. Symp. Information Theory*, page 256, Lausanne, Switzerland, June 2002.

[30] A. J. Viterbi and J. K. Omura. *Principles of Digital Communication and Coding*. McGraw-Hill, New York, NY, USA, 1979.

[31] A. M. Viterbi and A. J. Viterbi. Improved union bound on linear codes for the input-binary AWGN channel, with applications to turbo codes. In *Proc. IEEE Int. Symp. Information Theory*, volume 1, page 29, Cambridge, MA, USA, Sept. 1998. IEEE.

[32] N. Wiberg. *Codes and Decoding on General Graphs*. PhD thesis, Linköping University, S-581 83 Linköping, Sweden, 1996.

Chapter 4

The Capacity of Finite State Channels

4.1 Introduction

Determining the achievable rates at which information can be reliably transmitted across noisy channels has been one of the central pursuits in information theory since Shannon invented the subject in 1948. In this chapter, we consider these rates for the class of channels known as finite state channels (FSC). A FSC is a discrete-time channel where the distribution of the channel output depends on both the channel input and the underlying channel state. This allows the channel output to depend implicitly on previous inputs and outputs via the channel state.

In practice, there are three types of channel variation which FSCs are typically used to model. A *flat fading* channel is a time-varying channel whose state is independent of the channel inputs. An *intersymbol-interference* (ISI) channel is a time-varying channel whose state is a deterministic function of the previous channel inputs. Channels which exhibit both fading and ISI can also modeled, and their state is a stochastic function of the previous channel inputs.

A number of other authors have dealt with FSCs in the past, and we review some of their important contributions. Since it is easy to construct degenerate FSCs, most of these results are limited to a particular set of well behaved FSCs. A FSC in this particular set is referred to as an indecomposable FSC (IFSC). Blackwell, Breiman, and Thomasian introduced IFSCs in [7] and proved the natural analogue of the channel coding theorem for them. Birch discusses the achievable information rates of IFSCs in [5], and computes bounds for a few simple examples. In [14, p.100], Gallager gives an elegant derivation of the coding theorem and provides a method

to explicitly compute the capacity when the receiver has perfect channel state information. This method cannot be applied, however, when the receiver only has imperfect state estimates computed from the previous channel outputs. Hirt considers linear filter channels with additive white Gaussian noise (AWGN) and equiprobable binary inputs in [16], and develops a Monte Carlo method for estimating achievable rates. In [15], Goldsmith and Varaiya take a different approach and provide an explicit method of estimating the capacity of flat fading IFSCs (i.e., where the state sequence is independent of the transmitted sequence). In this chapter, we provide a simple Monte Carlo method of estimating the achievable information rates of any IFSC and we focus on the problem of estimating the capacity of IFSCs with ISI (i.e., where the state sequence is a deterministic function of the transmitted sequence).

It is worth noting that this method, reported in [24], was discovered independently by Arnold and Loeliger in [1] and by Sharma and Singh[1]. It is quite surprising, in fact, that this method was not proposed earlier. It is simply an efficient application of the famous Shannon-McMillan-Breiman theorem. Nonetheless, [1], [27][1], and [24] represent the first publications where the achievable information rates of a general IFSC are computed to 3 or 4 digits of accuracy. Furthermore, these advances stimulated new interest in the subject which led Kavčić to formulate a very elegant generalization of the Arimoto-Blahut algorithm for finite state channels in [19].

The achievable information rate of an IFSC, for a given input process, is equal to the mutual information rate between the stochastic input process and the stochastic output process. This mutual information rate, $I(\mathcal{X};\mathcal{Y})$, is given by

$$I(\mathcal{X};\mathcal{Y}) = H(\mathcal{X}) + H(\mathcal{Y}) - H(\mathcal{X},\mathcal{Y}), \tag{4.1.1}$$

where $H(\mathcal{X})$, $H(\mathcal{Y})$, and $H(\mathcal{X},\mathcal{Y})$ are the respective entropy rates of the input process, the output process, and the joint input-output process. The symmetric information rate (SIR) of an IFSC is the maximum rate achievable by an input process which chooses each input independently and equiprobably from the source alphabet. The capacity of an IFSC is the largest rate achievable by any input process.

Our simple Monte Carlo method is based on estimating each of the entropy rates in (4.1.1). These entropy rates are estimated by simulating a long realization of the process and

[1] While the Monte Carlo method is introduced correctly in [27], it appears that most of the other results in their paper, based on regenerative theory, are actually incorrect. A correct analytical treatment can be found in Section 4.4.4.

Channel	Transfer Function	Normalized Response
Dicode	$(1-D)$	[1 -1]/$\sqrt{2}$
EPR4	$(1-D)(1+D)^2$	[1 1 -1 -1]/2
E^{2}PR4	$(1-D)(1+D)^3$	[1 2 0 -2 -1]/$\sqrt{10}$

Table 4.1: The transfer function and normalized response of a few partial response targets.

computing its probability using the forward recursion of the well known BCJR algorithm [2]. The fact that this probability can be used to estimate the entropy rate is a consequence of the Shannon-McMillan-Breiman theorem [10, p. 474]. Furthermore, this approach is general enough to allow the mutual information rate to be maximized over Markov input distributions of increasing length, and thus can be used to estimate a sequence of non-decreasing lower bounds on capacity.

This chapter is organized as follows. In Section 4.2, we introduce a few example finite state channels which are discussed throughout the chapter. Mathematical definitions and notation for the chapter are introduced in Section 4.3. In Section 4.4, we address the problem of estimating entropy rates. In particular, this section discusses our simple Monte Carlo method, a general analytical method, and an interesting connection with Lyapunov exponents. Section 4.5 uses the results of the previous section to discuss upper and lower bounds on the capacity of finite state channels. In Section 4.6, we give the numerical results of applying the Monte Carlo method to the example channels. Exact information rates are derived for the dicode erasure channel in Section 4.7. A pseudo-analytical method of estimating information rates based on density evolution, which is quite efficient for two state channels, is also described. Finally, in Section 4.8, we provide some concluding remarks.

4.2 Channel Models

4.2.1 Discrete-Time Linear Filter Channels with AWGN

A very common subset of IFSCs is the set of discrete-time linear filter channels with additive white Gaussian noise (AWGN), which are described by

$$y_k = \sum_{i=0}^{\nu} h_i x_{k-i} + n_k, \tag{4.2.1}$$

where ν is the channel memory, $\{x_k\}$ is the channel input (taken from a discrete alphabet), $\{y_k\}$ is the channel output, and $\{n_k\}$ is i.i.d. zero mean Gaussian noise with variance σ^2. Bounds on the capacity and SIR of this channel have been considered by many authors. In particular, we note the analytical results of Shamai *et al.* in [26] and the original Monte Carlo results of Hirt in [16]. Some examples of these channels are listed in Table 4.1, and were chosen from the class of of binary-input channels which are used to model equalized magnetic recording channels. The state diagram for the noiseless dicode channel (i.e., before the AWGN) is shown in Figure 4.2.1. A formal mathematical definition of these channels is given in Appendix 4A.1.

Computing the achievable information rates of these channels can also be simplified by writing the mutual information rate (4.1.1) as

$$I(\mathcal{X};\mathcal{Y}) = H(\mathcal{Y}) - H(\mathcal{Y}|\mathcal{X}).$$

This is because the second term is simply the entropy of the Gaussian noise sequence, $\{n_k\}$, which can be written in closed form [10, p. 225] as

$$H(\mathcal{Y}|\mathcal{X}) = \frac{1}{2}\log(2\pi e\sigma^2).$$

Therefore, estimating the SIR of these channels reduces to estimating $H(\mathcal{Y})$, and estimating the capacity of this channel reduces to estimating the supremum of $H(\mathcal{Y})$ over all input processes.

4.2.2 The Dicode Erasure Channel

Since it is difficult, if not impossible, to derive a closed form expression for the entropy rate of the dicode channel with AWGN, we also consider the somewhat artificial dicode erasure channel (DEC). This is a simple channel based on the $1 - D$ linear ISI channel whose noiseless state diagram is shown in Figure 4.2.1. The DEC corresponds to taking the output of the dicode

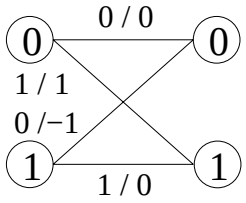


Figure 4.2.1: The state transition diagram of the dicode channel.

channel, $(+1, 0, -1)$, and either erasing it with probability ϵ or transmitting it perfectly with probability $1 - \epsilon$. The state diagram for the noiseless dicode channel is shown in Figure 4.2.1. A formal mathematical definition of the DEC is channel is given in Appendix 4A.2.

The properties of this channel are similar to the dicode channel with AWGN, and again the mutual information rate can be simplified to

$$I(\mathcal{X}; \mathcal{Y}) = H(\mathcal{Y}) - H(\mathcal{Y}|\mathcal{X}).$$

In this case, the second term is simply the entropy of the erasure position sequence which can be written in closed form as $H(\mathcal{Y}|\mathcal{X}) = -\epsilon \log \epsilon - (1 - \epsilon) \log(1 - \epsilon)$. Therefore, the SIR and capacity of this channel can also be determined by considering only $H(\mathcal{Y})$.

4.2.3 The Finite State Z-Channel

The Z-channel is a well-known discrete memoryless channel (DMC) which models a communications system with one "good" symbol and "bad" symbol. The "good" symbol is transmitted perfectly by the channel and the "bad" symbol is either transmitted correctly (with probability $1 - p$) or swapped with the "good" symbol (with probability p). Consider a finite state analogue of this channel in which the "good" and "bad" symbols are not fixed, but depend on the previous input symbol. One trellis section for such a channel, which we call the finite state Z-channel is shown in Fig. 4.2.2. The edges are labeled with the input bit and the output bits, where $B(p)$ stands for the Bernoulli distribution which produces a one with probability p. A formal mathematical definition of this channel is given in Appendix 4A.3.

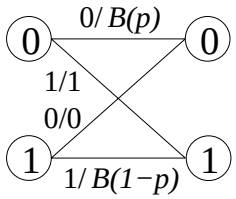


Figure 4.2.2: The state transition diagram of the finite state Z-channel. The symbol $B(p)$ refers to a binary random variable which equals 1 with probability p and 0 with probability $1-p$.

4.3 Definitions

4.3.1 The Indecomposable Finite State Channel

A finite state channel (FSC) is a stochastic mapping from a sequence of inputs, $\{X_t\}_{t\geq 1}$, chosen from the finite input alphabet, $\mathbb{X}$, to a sequence of outputs, $\{Y_t\}_{t\geq 1}$, chosen from the (possibly infinite) output alphabet, $\mathbb{Y}$. Let $\{S_t\}_{t\geq 1}$ be the state sequence of the channel, which takes values in the finite set $\mathcal{S} = \{0, 1, \ldots, N_S - 1\}$. When the output alphabet is countable, the channel statistics are completely defined by the time-invariant conditional probability, $f_{ij}(x,y) \triangleq Pr(Y_t = y, S_{t+1} = j | X_t = x, S_t = i)$. For uncountable $\mathbb{Y}$, we abuse this notation slightly and let, for each j, $f_{ij}(x,y)$ be a continuous density function of the output, y, given starting state i and input x. In this way, we formally define a finite state channel by the triple, $(\mathbb{X}, \mathbb{Y}, \mathbf{F}(\cdot,\cdot))$, where $[\mathbf{F}(x,y)]_{ij} = f_{ij}(x,y)$. Each example channel in Section 4.2 is defined formally using this notation in Appendix 4A.

Since many properties of a FSC can be related to the properties of a finite state Markov chain (FSMC), we start by reviewing some terminology from the theory of FSMCs. A FSMC is *irreducible* if there is a directed path from any state to any other state. If the greatest common divisor of the lengths of all cycles (i.e., paths from a state back to itself) is one, then it is *aperiodic*. A FSMC is ergodic or *primitive* if it is both irreducible and aperiodic. These ideas can also be applied to a non-negative square matrix, $\mathbf{A}$, by associating the matrix with a FSMC which has a path from state i to state j if and only if $[\mathbf{A}]_{ij} > 0$. Using this, we say that a FSC is *indecomposable* if its zero-one connectivity matrix, defined by

$$[\mathbf{F}(*,*)]_{ij} = \begin{cases} 1 & \exists\, x \in \mathbb{X}, y \in \mathbb{Y} \text{ s.t. } f_{ij}(x,y) > 0 \\ 0 & \text{otherwise} \end{cases},$$

is primitive.

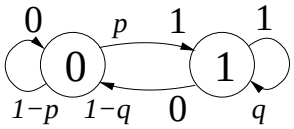


Figure 4.3.1: The state diagram of a two state input process which sends a 1 with probability p from the 0 state and with probability q from the 1 state.

4.3.2 The Markov Input Process

When computing achievable information rates, it is typical to treat the input sequence as a stochastic process as well. Let $\{T_t\}_{t\geq 1}$ be the state sequence of an ergodic FSMC taking values in the finite set $\mathcal{T} = \{0, 1, \ldots, N_T - 1\}$. The statistics of the input process, $\{X_t\}_{t\geq 1}$, are defined by the transition probabilities of the chain, $\theta_{ij} \triangleq Pr(T_{t+1} = j | T_t = i)$, and the edge labels, ϕ_{ij}, with $X_t = \phi_{T_t, T_{t+1}}$. We refer to this type of input process as a Markov input process, and denote it by the pair $(\mathbf{\Theta}, \mathbf{\Phi})$, where $[\mathbf{\Theta}]_{ij} = \theta_{ij}$ and $[\mathbf{\Phi}]_{ij} = \phi_{ij}$.

For example, the state diagram of a general two state Markov input process in shown in Figure 4.3.1. The formal definition of this same process is given by $(\mathbf{\Theta}, \mathbf{\Phi})$ where $\theta_{0,1} = 1 - \theta_{0,0} = p$, $\theta_{1,1} = 1 - \theta_{1,0} = q$, $\phi_{0,0} = \phi_{1,0} = 0$, and $\phi_{0,1} = \phi_{1,1} = 1$.

4.3.3 Combining the Input Process and the Finite State Channel

When the channel inputs are generated by a Markov input process, the channel output, $\{Y_t\}_{t\geq 1}$, can be viewed as coming from stochastic process. In this case, the distribution of Y_t depends only on state transitions in the combined state space of the channel and input. Let $\mathcal{Q} = \{0, 1, \ldots, N_T N_S - 1\}$ and, for any $q \in \mathcal{Q}$, let the $\mathcal{T}$-state of q be $r(q) = \lfloor q/N_S \rfloor$ and the $\mathcal{S}$-state of q by $s(q) = q \mod N_S$. Using this, we can write the state transition probabilities of the combined process as

$$p_{ij} \triangleq Pr(Q_{t+1} = j | Q_t = i) = \theta_{r(i),r(j)} \int_{\mathbb{Y}} f_{s(i),s(j)}(\phi_{r(i),r(j)}, y) dy,$$

where the integral is taken to be a sum if $\mathbb{Y}$ is countable. We also define the conditional observation probability of y, given the transition, to be

$$g_{ij}(y) \triangleq Pr(Y_t = y | Q_{t+1} = j, Q_t = i) = f_{s(i),s(j)}(\phi_{r(i),r(j)}, y).$$

We refer to the stochastic output sequence, $\{Y_t\}_{t\geq 1}$, as a finite state process (FSP) and define it formally in the next section.

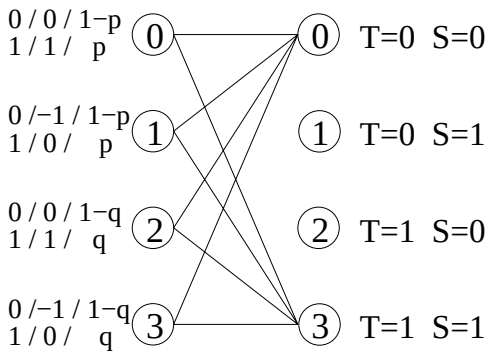


Figure 4.3.2: The combined state diagram for a general two state Markov input process and the dicode channel.

As an example, we show in Figure 4.3.2 the state diagram formed by combining a general two state Markov input process with the dicode channel. The edge labels on the left side of the figure give the input symbol, the output symbol, and the transition probability for each edge. Each state is labeled by its Q-value, and the corresponding S and T values are also shown on the right side of the figure.

Remark 4.3.1. One problem with joining the state spaces of the input and channel processes is that the resulting Markov chain may no longer be primitive. Suppose that the input process and the channel keep the same state variable (e.g., the input process remembers its last output and the channel remembers its last input). The state diagram for the combined process of this type is shown in Figure 4.3.2. The resulting Markov chain is reducible, but it still has a unique ergodic component. Taking only the ergodic component, consisting of states 0 and 3, results in an ergodic finite state process. In other cases, the state diagram for the combined process may actually be disconnected. In general, we will require that the Markov chain associated with the combined process is primitive. Therefore, some care must taken in choosing the input process and/or reducing the combined process. Another example of this problem is given in Appendix 4A.4.

4.3.4 The Finite State Process

Let $\{Q_t\}_{t\geq 1}$ be an ergodic FSMC taking values from the set $\mathcal{Q} = \{0, 1, \ldots, N_Q - 1\}$. The finite state process (FSP), $\{Y_t\}_{t\geq 1}$, is an ergodic stochastic process controlled by $\{Q_t\}_{t\geq 1}$ which takes values from the alphabet $\mathbb{Y}$. The transition probabilities for $\{Q_t\}_{t\geq 1}$ are given

by $Pr(Q_{t+1} = j|Q_t = i) = p_{ij}$, and the dependence of $\{Y_t\}_{t\geq 1}$ on $\{Q_t\}_{t\geq 1}$ is given by $Pr(Y_t = y|Q_{t+1} = j, Q_t = i) = g_{ij}(y)$. While this notation is precise for countable $\mathbb{Y}$, we abuse it slightly for uncountable $\mathbb{Y}$ and let $g_{ij}(y)$ be the continuous density function of the output, y, associated with the state transition from state i to state j. The FSP, $\{Y_t\}_{t\geq 1}$, is defined formally by the triple $(\mathbb{Y}, \mathbf{P}, \mathbf{G}(\cdot))$, where $p_{ij} = [\mathbf{P}]_{ij}$ and $g_{ij}(\cdot) = [\mathbf{G}(\cdot)]_{ij}$.

We note that any FSP can be stationary if the initial state if chosen properly. Let $\boldsymbol{\pi} = [\ \pi_1 \quad \pi_2 \quad \dots \quad \pi_r\ ]$ be the unique stationary distribution of $\{Q_t\}_{t\geq 1}$ which satisfies $\pi_j = \sum_i \pi_i p_{ij}$. If the initial state, Q_1, of the underlying Markov chain is chosen such that $Pr(Q_1 = j) = \pi_j$, then $\{Y_t\}_{t\geq 1}$ is stationary in the sense that $Pr(\mathbf{Y}_1^k = \mathbf{y}_1^k) = Pr(\mathbf{Y}_{t+1}^{t+k} = \mathbf{y}_1^k)$ for all $k \geq 0$ and all $t \geq 1$. This initialization is assumed throughout the discussion of FSPs.

If $\mathbb{Y}$ is a finite set, then an identical process can also be generated as a function of a FSMC. More precisely, this means that there exists a FSMC, $\{X_t\}_{t\geq 1}$, and a mapping ξ, such that $Y_t = \xi(X_t)$. The process $\{Y_t\}_{t\geq 1}$ can also described as the output of a hidden Markov model. We present $\{Y_t\}_{t\geq 1}$ as a FSP because it is the most natural represetation when considering the entropy rate of the process.

4.4 The Entropy Rate of a Finite State Process

Since a number of authors have considered the entropy rate of a FSP in the past, we review some of the key results. Blackwell appears to have been the first to consider the entropy rate of a function of a FSMC. In [6], he gives an explicit formula for the entropy rate, in terms of the solution to an integral equation, and he notes that this result suggests that the entropy rate is "intrinsically a complicated function" of the underlying Markov chain, $\{X_t\}_{t\geq 1}$, and the mapping, ξ. Birch [5] derives a sequence of Markov upper and lower bounds for the entropy rate and shows, under fairly restrictive conditions, that the gap between them converges to zero exponentially fast. The complexity of computing his bounds also grows exponentially, however, making them less useful in practice.

Here, we attack the problem first by introducing an efficient Monte Carlo method of estimating the entropy rate based on the Shannon-McMillan-Breiman theorem. Then, we work towards analytical approaches of computing the entropy rate (via Blackwell's integral equation). We also discuss conditions under which a central limit theorem (CLT) holds for the entropy rate. Under these same conditions, we prove that the gap between sequences of Markov upper

and lower bounds on the entropy rate converges to zero exponentially fast. Finally, we describe a connection between the entropy rate of a FSP and the largest Lyapunov exponent of an associated sequence of random matrices. It is worth noting that the natural logarithm is denoted $\ln$ while the base 2 logarithm is denoted $\log$.

4.4.1 A Simple Monte Carlo Method

Let $\{Y_t\}_{t\geq 1}$ be an ergodic FSP defined by $(\mathbb{Y}, \mathbf{P}, \mathbf{F}(\cdot))$. We start by using the definition of the entropy rate for a stationary process [10, Chap. 4],

$$H(\mathcal{Y}) \triangleq -\lim_{n\to\infty} \frac{1}{n} E\left[\log Pr(\mathbf{Y}_1^n)\right],$$

to define the sample entropy rate as

$$\hat{H}_n(\mathbf{Y}_1^n) = -\frac{1}{n}\log Pr(\mathbf{Y}_1^n). \tag{4.4.1}$$

It is worth noting that $\hat{H}_n(\mathbf{Y}_1^n)$ is a random variable, and the asymptotic convergence of that random variable to the true entropy rate is guaranteed by the Shannon-McMillan-Breiman theorem [10, p. 474]. Mathematically speaking, this theorem states that

$$\lim_{n\to\infty} -\frac{1}{n}\log Pr(\mathbf{Y}_1^n) = H(\mathcal{Y})$$

for almost all realizations of $\mathbf{Y}_1^n$ (i.e., almost surely). While the original proof only holds for finite alphabet processes, it was extended to more general processes by Barron [3].

Efficiently applying the Shannon-McMillan-Breiman theorem to our FSP is equivalent to efficiently computing $\log Pr(\mathbf{Y}_1^n)$ for large n. This quantity has a natural decomposition of the form

$$\log Pr(\mathbf{Y}_1^n) = \sum_{t=1}^{n} \log Pr(Y_t|\mathbf{Y}_1^{t-1}), \tag{4.4.2}$$

and it turns out that the forward recursion of the BCJR algorithm [2] is ideal for computing this quantity. We note that random realizations, $\mathbf{y}_1^n$, of the process, $\mathbf{Y}_1^n$, are generated as a byproduct of any channel simulation. Let us define the forward state probability vector at time t, $\boldsymbol{\alpha}^{(t)}$, in terms of its components,

$$\alpha_i^{(t)} = Pr(Q_t = i|\mathbf{Y}_1^{t-1} = \mathbf{y}_1^{t-1}), \tag{4.4.3}$$

for $i \in \mathcal{Q}$. Using this, the forward recursion of the BCJR algorithm can be written as

$$\alpha_j^{(t+1)} = \frac{1}{A_t} \sum_{i=0}^{N_Q-1} \alpha_i^{(t)} Pr(Y_t = y_t, Q_{t+1} = j | Q_t = i), \tag{4.4.4}$$

where A_t is the standard normalization factor chosen to ensure that $\sum_{j=0}^{N_Q-1} \alpha_j^{(t+1)} = 1$. We note that the probability, $Pr(Y_t = y, Q_{t+1} = j | Q_t = i)$, required by (4.4.4) depends on the FSP and can be written as

$$\begin{aligned} Pr(Y_t = y, Q_{t+1} = j | Q_t = i) &= Pr(Y_t = y | Q_{t+1} = j, Q_t = i) Pr(Q_{t+1} = j | Q_t = i) \\ &= g_{ij}(y) p_{ij}. \end{aligned}$$

Proposition 4.4.1. *The sample entropy rate of a realization, $\mathbf{y}_1^n$, of the FSP, $\{Y_t\}_{t \geq 1}$, is given by*

$$\hat{H}_n(\mathbf{y}_1^n) = -\frac{1}{n} \sum_{t=1}^{n} \log A_t.$$

Proof. From (4.4.4), we see that

$$\begin{aligned} A_t &= \sum_{j=0}^{N_Q-1} \alpha_j^{(t+1)} \\ &= \sum_{j=0}^{N_Q-1} \left(\sum_{i=0}^{N_Q-1} \alpha_i^{(t)} Pr(Y_t = y_t, Q_{t+1} = j | Q_t = i) \right) \\ &= Pr(Y_t = y_t | \mathbf{Y}_1^{t-1} = \mathbf{y}_1^{t-1}), \end{aligned}$$

which means that $\log Pr(\mathbf{Y}_1^n)$ can be computed using (4.4.2). Combining this with (4.4.1) completes the proof. □

Remark 4.4.2. The complexity of this method is linear in the number of states, N_Q, and linear in the length of the realization, n. Furthermore, if a central limit theorem holds for the entropy rate, then the variance of the estimate will decay like $O\left(n^{-1/2}\right)$.

We believe that the rapid mixing of the underlying Markov chain and the form of (4.4.2) leads naturally to a central limit theorem for the sample entropy rate. The following conjecture makes this notion precise. We note that the conclusion of this conjecture is proven, under more restrictive conditions, in Section 4.4.6.

Conjecture 4.4.3. *Let $\{Y_t\}_{t\geq 1}$ be an ergodic FSP which gives rise to the conditional probability sequence, $\{A_t\}_{t\geq 1}$, where $A_t = Pr(Y_t|\mathbf{Y}_1^{t-1})$. If (i) $\lim_{t\to\infty} E\left[(-\log A_t)^{2+\epsilon}\right] < \infty$, then the sample entropy rate obeys a central limit theorem of the form*

$$\sqrt{n}\left[\hat{H}_n(\mathcal{Y}) - H(\mathcal{Y})\right] \xrightarrow{d} N(0,\sigma^2).$$

The variance, σ^2, of the estimate is given by

$$\sigma^2 = R(0) + 2\sum_{\tau=1}^{\infty} R(\tau), \tag{4.4.5}$$

where $R(\tau) = \lim_{t\to\infty} E\left[(\log A_t + H(\mathcal{Y}))\,(\log A_{t-\tau} + H(\mathcal{Y}))\right]$. If we also have that (ii) $\lim_{t\to\infty} E\left[(-\log A_t)^{4+\epsilon}\right] < \infty$, then we can estimate the variance using finite truncations of (4.4.5) with $R(\tau)$ set to the sample autocorrelation,

$$\hat{R}_n(\tau) = \frac{1}{n-\tau}\sum_{t=\tau+1}^{n}\left(\log A_t + \hat{H}_n(\mathcal{Y})\right)\left(\log A_{t-\tau} + \hat{H}_n(\mathcal{Y})\right).$$

Motivation. This conjecture is based on the fact that $\{A_t\}_{t\geq 1}$ is asymptotically stationary and our belief that the autocorrelation, $R(\tau)$, decays exponentially with τ. These conditions are generally sufficient to imply a central limit theorem for sums like (4.4.2). □

4.4.2 The Statistical Moments of Entropy

While the entropy of a random variable is usually defined to be $E\left[-\log Pr(Y)\right]$, one might also consider the random variable $Z = -\log Pr(Y)$. We refer to the kth moment of the random variable, Z, as the kth moment of the entropy. One reason for examining these quantities is that most CLTs require that the increments have finite second moments. Here, we show, under mild conditions, that the kth moment of the entropy is bounded, for all finite k.

Let $p(y)$ be the probability density of any absolutely continuous random variable. Since the function $p(y)$ must integrate to one, we know the tails must decay faster than $1/|y|$. If we assume the slightly stronger condition that $p(|y|) = O(|y|^{-1-\epsilon})$, for some $\epsilon > 0$, then we find that all finite moments of the entropy are bounded. Recall that all finite moments of a random variable are finite if the exponential moments, $E\left[e^{sZ}\right] = E\left[Pr(Y)^{-s}\right]$, are finite for some $s > 0$. We can upper bound this expectation with

$$\begin{aligned} E\left[Pr(Y)^{-s}\right] &= \int_{-\infty}^{\infty} p(y)p(y)^{-s}dy \\ &\leq \int_{-a}^{a} p(y)^{1-s}dy + 2C\int_{a}^{\infty} |y|^{(s-1)(1+\epsilon)}dy, \end{aligned}$$

where a is chosen large enough that $p(y) \leq C|y|^{-1-\epsilon}$ for all $|y| > a$. Using the fact that $p(y) < p(y)^{1-s}$ whenever $p(y) > 1$, it is easy to verify that the first term is less than $2a$. The second term will also be finite as long as $(s-1)(1+\epsilon) < 1$ which is equivalent to $s < \epsilon/(1+\epsilon)$. Since this expectation is finite for $s \in [0, \epsilon/(1+\epsilon))$, all finite moments of Z are bounded.

There are also distributions that are poorly behaved with respect to entropy, however. Consider the probability distribution on the integers given by

$$Pr(Y = n) = \frac{1}{Cn(\log n)^{\rho}},$$

for $n \geq 3$. As long as $\rho > 1$, we can compute a finite

$$C = \sum_{n=3}^{\infty} \frac{1}{n(\log n)^{\rho}}$$

which normalizes this distribution. The kth moment of the entropy for this distribution is given by

$$\sum_{n=3}^{\infty} \frac{1}{Cn(\log n)^{\rho}} \left(-\log\left(Cn(\log n)^{\rho}\right)\right)^{k},$$

which can be lower bounded by

$$\sum_{n=n_0}^{\infty} \frac{2^{-k}}{Cn(\log n)^{\rho-k}}$$

if n_0 is chosen large enough that $\log\left(Cn(\log n)^{\rho}\right) \geq (\log n)/2$. This lower bound will be finite only if $\rho - k > 1$. So for $1 < \rho \leq 2$, the distribution is well-defined but the entropy and all higher moments are infinite. Likewise, the finite variance condition necessary for a CLT requires that $\rho > 3$.

Now, let us focus on the value of the entropy increment, $-\log A_t$, during a transition from state i to state j. In this case, the true distribution of Y_t is given by $g_{ij}(y)$, but the simulation method computes A_t based on the assumed distribution,

$$P_{\boldsymbol{\alpha}}(y) = \sum_{i,j} \alpha_i^{(t)} h_i(y),$$

where $h_i(y) = Pr(Y_t = y | Q_t = i) = \sum_j p_{ij} g_{ij}(y)$. For a particular transition and forward state probabilities, the expectation of $-\log A_t$ can now be written as

$$E\left[-\log A_t | Q_t = i, Q_{t+1} = j, \boldsymbol{\alpha}\right] = E_{g_{ij}(Y)} - \left[\log P_{\boldsymbol{\alpha}}(Y)\right].$$

This general approach can also be used to upper bound the higher moments, $E[-\log A_t]^k$, required by Conjecture 4.4.3. In particular, we consider the bound

$$E_{g_{ij}(Y)}\left[(-\log P_{\boldsymbol{\alpha}}(Y))^k\right] \leq E_{g_{ij}(Y)}\left[\left(-\log\left(\min_i h_i(Y)\right)\right)^k\right],$$

which is based on maximizing the LHS over $\boldsymbol{\alpha}$.

Suppose that the output alphabet is finite (or a bounded continuous set) and there is an $\epsilon > 0$ such that $\min_i \inf_y h_i(y) \geq \epsilon$. In that case, the magnitude of the kth moment can be upper bounded with $E[-\log A_t]^k \leq (-\log \epsilon)^k$. If the output alphabet is countably infinite (or an unbounded continuous set) and $h_i(y) > 0$ for all bounded y, then the magnitude of the kth moment will depend only on the tails of the $h_i(y)$. Let $g(y)$ be the $g_{ij}(y)$ whose tail decays most slowly and $h(y)$ be the $h_i(y)$ whose tail decays most quickly. The magnitude of the kth moment will be finite if

$$E_{g(Y)}\left[(-\log h(Y))^k\right] < \infty.$$

Example 4.4.4. Suppose all of the $g_{ij}(y)$ are Gaussian densities with finite mean and variance, We assume that the particular mean and variance depends on the transition $i \to j$. In this case, the tails of each density decay like $O(e^{-ay^2})$ where a depends on the variance. The magnitude of the kth cross-moment for any two Gaussians is upper bounded by

$$\begin{aligned}\int_{-\infty}^{\infty} C_1 e^{-ay^2}\left(-\log(C_2 e^{-by^2})\right)^k dy &= \int_{-\infty}^{\infty} C_1 e^{-ay^2}\left(-\log C_2 + \frac{by^2}{\ln 2}\right)^k dy \\ &= \sum_{i=0}^{k}(-\log C_2)^i \left(\frac{b}{\ln 2}\right)^{k-i}\int_{-\infty}^{\infty} C_1 e^{-ay^2} y^{2(k-i)} dy.\end{aligned}$$

Since the integral really just computes the $2(k-i)$th moment of a Gaussian, the expression is bounded for all finite k.

4.4.3 A Matrix Perspective

In this section, we introduce a natural connection between the product of random matrices and the entropy rate of a FSP. This connection is interesting in its own right, but will also be very helpful in understanding the results of the next few sections.

Definition 4.4.5. For any $y \in \mathbb{Y}$, the *transition-observation probability matrix*, $\mathbf{M}(y)$, is an $N_Q \times N_Q$ matrix defined by

$$[\mathbf{M}(y)]_{ij} \triangleq Pr(Y_t = y, Q_{t+1} = j | Q_t = i) = p_{ij} f_{ij}(y).$$

These matrices behave similarly to transition probability matrices because their sequential products compute the n-step transition observation probabilities of the form,

$$[\mathbf{M}(y_k)\mathbf{M}(y_{k+1}) \dots \mathbf{M}(y_{k+n})]_{ij} = Pr(\mathbf{Y}_k^{k+n} = \mathbf{y}_k^{k+n}, Q_{k+n+1} = j | Q_k = i).$$

This means that we can write $Pr(\mathbf{Y}_1^n)$ as the matrix product

$$Pr(\mathbf{Y}_1^n) = \boldsymbol{\pi}\mathbf{M}(y_1)\mathbf{M}(y_2) \dots \mathbf{M}(y_n)\mathbf{1}, \tag{4.4.6}$$

where $\boldsymbol{\pi}$ is the row vector associated with the unique stationary distribution of $\{Q_t\}_{t \geq 1}$ and $\mathbf{1}$ is a column vector of all ones.

The forward recursion of the BCJR algorithm can also be written in matrix form with

$$\boldsymbol{\alpha}^{(t+1)} = \frac{\boldsymbol{\alpha}^{(t)}\mathbf{M}(y_t)}{\left\|\boldsymbol{\alpha}^{(t)}\mathbf{M}(y_t)\right\|_1}, \tag{4.4.7}$$

where $\boldsymbol{\alpha}^{(t)} = [\ \alpha_1^{(t)} \quad \alpha_2^{(t)} \quad \dots \quad \alpha_{N_Q}^{(t)}\ ]$ and $\|\mathbf{x}\|_1 = \sum_i |x_i|$. This update formula is referred to as the *projective product*, and its properties are discussed at some length in [20]. We note that the order of the matrix-vector product in (4.4.7) is reversed with respect to [20]. The two most important properties of the projective product given by Lemma 2.2 of [20] are: (i) it is Lipschitz continuous if the smallest row sum is strictly greater than zero and (ii) it is a strict contraction if the matrix is positive. We note that these are really the only properties required for a self-contained proof of Theorem 4.4.9 which is stated in the next section.

4.4.4 The Analytical Approach

It appears that the most straightforward analytical approach to the entropy rate problem is the original method proposed by Blackwell [6]. Applying the same approach to this setup gives an integral equation whose solutions are the stationary distributions of the joint Markov chain formed by joining the true state and the forward state probability vector, $\left\{Q_t, \boldsymbol{\alpha}^{(t)}\right\}_{t \geq 1}$. The entropy rate is then computed with

$$\lim_{t \to \infty} E\left[\log Pr(Y_t | \mathbf{Y}_1^{t-1})\right] = \lim_{t \to \infty} E\left[\log \sum_{i=0}^{N_Q - 1} Pr(Y_t | Q_t = i) Pr(Q_t = i | \mathbf{Y}_1^{t-1})\right],$$

where $Pr(Y_t = y|Q_t = i) = \sum_j p_{ij} f_{ij}(y)$ is independent of t and the limit distribution, $\lim_{t\to\infty} Pr(Q_t = i|\mathbf{Y}_1^{t-1})$, depends on the true state, q_t, and is given by a stationary distribution of the joint Markov chain (cf., a solution of Blackwell's integral equation). One problem with this method, besides its general intractability, is the fact that the stationary distribution may not be unique. This is equivalent to saying that the integral equation may not have a unique solution.

Since many of the probability distributions in this section can be rather badly behaved, rigorous treatment requires that we use some measure theory. The following analysis is based on general state space Markov chains as described in [22]. Let $\Omega = \mathcal{Q} \times \mathfrak{D}(\mathcal{Q})$ be the sample space of the joint Markov chain, where $\mathcal{Q} = \{0, 1, \ldots, N_Q - 1\}$ and $\mathfrak{D}(\mathcal{Q})$ is the set of probability distributions (i.e., the set of non-negative vectors of length N_Q which sum to one). Let $\{\mu_t(q, A)\}_{t\geq 1}$ be the probability measure defined by $\mu_t(q, A) = Pr(\boldsymbol{\alpha}^{(t)} \in A, Q_t = q)$ for any $A \in \Sigma$, where Σ is the sigma field of Borel subsets of $\mathfrak{D}(\mathcal{Q})$. The transitions of this Markov chain are described by

$$\mu_{t+1}(j, A) = \sum_{i=0}^{N_Q-1} \int_{\mathfrak{D}(\mathcal{Q})} \mu_t(i, dx) P_{ij}(x, A),$$

where the transition kernel, $P_{ij}(x, A) = Pr(\boldsymbol{\alpha}^{(t+1)} \in A, Q_{t+1} = j|\boldsymbol{\alpha}^{(t)} = x, Q_t = i)$, is a probability measure defined on $A \in \Sigma$. The kernel can be written explicitly as

$$P_{ij}(x, A) = \int_{\{z\in\mathbb{Y}|L(x,y)\in A\}} p_{ij} g_{ij}(dz),$$

where $L(\boldsymbol{\alpha}, y) = \boldsymbol{\alpha}\mathbf{M}(y)/\left\|\boldsymbol{\alpha}\mathbf{M}(y)\right\|_1$ is the forward recursion update.

Before we continue, it is worth discussing some of the standard definitions and notation associated with Markov chains on general state spaces. Our notation, $P_{ij}(x, A)$, for the transition kernel is natural, albeit somewhat non-standard, considering the decomposition of our state space into discrete and continuous components. The n-step transition kernel is denoted $P_{ij}^{(n)}(x, A)$, and the unique stationary distribution is denoted $\pi(i, A)$ if it exists. The transition kernel can also be treated as an operator which maps the set of bounded measurable functions back to itself. The operator notation is given by

$$P^{(n)} r(i, x) = \sum_{j=0}^{N_Q-1} \int_{\mathfrak{D}(\mathcal{Q})} P_{ij}^{(n)}(x, dz) r(j, z),$$

and is useful for discussing the convergence of a Markov chain to a stationary distribution.

A general state space Markov chain is *uniformly ergodic* if it converges in total variation to a unique stationary distribution at a geometric rate which is independent of the starting state [22, p. 382]. This is equivalent to saying that there exists some $\rho < 1$ such that

$$\sup_{i,x} \left| P^{(n)} r(i,x) - \sum_{j=0}^{N_Q - 1} \int_{\mathfrak{D}(\mathcal{Q})} r(j,z)\pi(j,dz) \right| \leq C\rho^n \tag{4.4.8}$$

for all bounded measurable functions, $r(i,A)$, which satisfy $\sup_{i,A} |r(i,A)| \leq 1$. This type of convergence is generally too strong for our problem, however. If (4.4.8) holds only for all bounded continuous functions (in some topology), then the Markov chain converges weakly[2] to a unique stationary distribution. While this behavior is referred to as *geometric ergodicity* in [21], we say instead that the Markov chain is *weakly uniform ergodic* to avoid confusion with the geometric ergodicity defined in [22, p. 354].

Now, we consider the first condition under which the limit distribution, $\pi(s,A) = \lim_{t\to\infty} \mu_t(s,A)$, exists and is unique. This is based on a comment by Blackwell describing when the support of $\pi(s,A)$ is at most countably infinite [7]. Under this condition, Theorem 4.4.7 shows that the joint Markov chain is uniformly ergodic.

Condition 4.4.6. The output alphabet, $\mathbb{Y}$, is countable and there exists a finite output sequence which gives the observer perfect state knowledge (i.e., the joint Markov chain is in true state q with $\alpha_q = 1$). Using the DEC for an example, we see that the output $y_t = 1$ satisfies this condition because it implies with certainty that $s_{t+1} = 1$.

Theorem 4.4.7. *If Condition 4.4.6 holds, then $\pi(s,A)$ exists, is unique, and is supported on a countable set. Furthermore, the joint Markov chain is uniformly ergodic.*

Proof. Let z be state of the joint Markov chain after the output sequence which provides perfect state knowledge. Since this state is reachable from any other state, the *ψ-irreducibility* of this Markov chain is given by Theorem 4.0.1 of [22]. The state, z, also satisfies the conditions of an *atom* as defined in [22, p. 100]. Since any finite output sequence will occur infinitely often with probability 1, the point z is also *Harris recurrent* as defined in [22, p. 200]. Applying Theorem 10.2.2 of [22] shows that $\pi(s,A)$ exists and is unique.

[2] This set of functions provides a metric for the weak convergence of probability measures on a separable metric space [29].

Next, we show that $\pi(s, A)$ is supported on a countable set. Since the return time to state z is finite with probability 1, we assume the joint Markov chain is in state z at time τ and index any state in the support set by its output sequence, $\{y_t\}_{t \geq \tau}$, starting from state z. Therefore, the support set of $\pi(s, A)$ is at most the set of finite strings generated by the alphabet $\mathbb{Y}$, which is countably infinite. In particular, for any $\epsilon > 0$, there is a finite set of strings with total probability greater than $1 - \epsilon$.

Since the underlying Markov chain, $\{Q_t\}_{t \geq 1}$, is primitive, the path to perfect knowledge can start at any time. So, without loss of generality, we assume the output sequence which provides perfect state knowledge starts in any state, takes n steps, ends in state q, and occurs with probability δ. This means that $P_{iq}^{(n)}(x, z) \geq \delta$ for all $i \in \mathcal{Q}$ and all $x \in \mathfrak{D}(\mathcal{Q})$, which is also known as *Doeblin's Condition* [22, p. 391]. Applying Theorem 16.2.3 of [22], we find that the joint Markov chain is uniformly ergodic. □

This leads to the second condition under which the limit distribution, $\lim_{t \to \infty} \mu_t(s, A) = \pi(s, A)$, exists and is unique. This condition is essentially identical to the condition used by Le Gland and Mevel to prove weakly uniform ergodicity in [21].

Condition 4.4.8. Every output has positive probability during every transition. Mathematically, this means that $g_{ij}(y) > 0$ for all $y \in \mathbb{Y}$ and every i, j such that $p_{ij} > 0$. For example, any real output channel with AWGN satisfies this condition.

Since the joint Markov chain implied by Condition 4.4.8 does not, in general, satisfy a minorization condition [22, p. 102], we must turn to methods which exploit the continuity of $P_{ij}(x, A)$. We say that a general state space Markov chain is *(weak) Feller* if its transition kernel maps the set of bounded continuous functions (in some topology) to itself [22, p. 128]. Based on the properties of (4.4.7), one can verify that the joint Markov chain will be weak Feller as long as the minimum row sum of $\mathbf{M}(y)$ is strictly positive for all $y \in \mathbb{Y}$. Unfortunately, the methods of [22] still cannot be used to prove that the joint Markov chain is weakly uniform ergodic because its stationary distribution may not be absolutely continuous. In many cases, it will be singular continuous and concentrated on a set of dimension smaller than that of Ω. For simplicity, we simply adapt the results of [21] to our case. We note, however, that the results of iterated function systems (or iterated random functions) may also be applied to prove this result [12][29].

Theorem 4.4.9 (Le Gland-Mevel). *If Condition 4.4.8, then $\mu_\infty(s, A)$ exists and is unique. Furthermore, the joint Markov chain is weakly uniform ergodic.*

Proof. The analysis in [21] is applied to finite state processes whose output distribution is only a function of the initial state (i.e., $g_{ij}(y) = g_{ik}(y)$ for all j, k). There is a one-to-one correspondence between these two models, however. For example, one can map every transition in our model to a state in their model and represent the same process. Since $g_{ij}(y) > 0$ for all $y \in \mathbb{Y}$ and every i, j, we find that the output distribution of each state in their model will also be positive. Along with the ergodicity of the underlying FSMC, this gives the conditions necessary for Theorem 3.5 of [21]. Therefore, the joint Markov chain is weakly uniform ergodic. □

Now, we address the issue of CLTs for the entropy rate. For uniformly ergodic Markov chains, we use the CLT given by Chen in Theorem II-4.3 of [8]. This CLT is both very general and has the most easily verifiable conditions. For FSPs which satisfy Condition 4.4.8, we use the CLT given by Corollary 4.4.14. One could also prove this directly using the exponential decay of correlation implied by weakly uniform ergodicity, or alternatively, by using the theory of iterated function systems [4]. Unfortunately, all of these methods break down simultaneously if the product $\mathbf{M}(y_t)\mathbf{M}(y_{t+1})\cdots\mathbf{M}(y_{t+n})$ does not become strictly positive for some n.

Theorem 4.4.10 (Chen). *Let $\{X_t\}_{t\geq 1}$ be a uniformly ergodic Markov chain with unique stationary distribution $\pi(x)$. Let $f(x)$ be a measurable function and $S_n = \sum_{t=1}^{n} f(X_t)$. If we assume that (i) $E_\pi[f(X)] = 0$ and (ii) $E_\pi\left[f^2(X)\right] < \infty$, then*

$$S_n/\sqrt{n} \xrightarrow{d} N(0, \sigma^2),$$

where $\sigma^2 = R(0) + 2\sum_{\tau=1}^{\infty} R(\tau) < \infty$ and

$$R(\tau) = \lim_{t\to\infty} E\left[f(X_t)f(X_{t-\tau})\right].$$

Corollary 4.4.11. *Consider the FSP $\{Y_t\}_{t\geq 1}$ and its joint Markov chain $\left\{Q_t, \boldsymbol{\alpha}^{(t)}\right\}_{t\geq 1}$. Suppose (i) the process satisfies the finite variance condition $\lim_{t\to\infty} E\left[\left(\ln Pr(Y_t|\mathbf{Y}_1^{t-1})\right)^2\right]$ and (ii) the joint Markov chain satisfies Condition 4.4.6. In this case, the sample entropy rate, $\hat{H}_n(\mathcal{Y})$, obeys*

$$\sqrt{n}\left[\hat{H}_n(\mathcal{Y}) - H(\mathcal{Y})\right] \xrightarrow{d} N(0, \sigma^2),$$

where σ^2 is finite and given by (4.4.5).

Proof. Using (4.4.4), it is easy to see that $Pr(Y_t|Y_1^{t-1}) = f(Y_t, \boldsymbol{\alpha}^{(t)})$ for some measurable function, f. Now, we introduce the extended Markov chain, $\left\{Q_t, Y_t, \boldsymbol{\alpha}^{(t)}\right\}_{t\geq 1}$, since the function requires the Y_t value. Since the random variable Y_t is conditionally independent of all other quantities given Q_t and Q_{t+1}, it follows that the extended Markov chain inherits the ergodicity properties of the joint Markov chain. Since (i) implies that the joint Markov chain is uniformly ergodic and (ii) implies the finite variance condition of Theorem 4.4.10, we simply apply Theorem 4.4.10 to complete the proof. □

4.4.5 Entropy Rate Bounds

It is well known [10, Chap. 4] that the entropy rate of an ergodic FSP, $\{Y_t\}_{t\geq 1}$, is sandwiched between the Markov upper and lower bounds given by

$$H(Y_k|Y_{k-1}, Y_{k-2}, \ldots, Y_1, Q_1) \leq H(\mathcal{Y}) \leq H(Y_k|Y_{k-1}, Y_{k-2}, \ldots, Y_1). \tag{4.4.9}$$

In fact, Birch proves that the gap between these bounds decays exponentially in k for functions of a FSMC whose transition matrices are strictly positive [5]. The mixing properties of the underlying FSMC make it easy to believe that this gap actually decays exponentially for all FSPs.

Since all three of the quantities in (4.4.9) can be written as integrals over a state distribution of the joint Markov chain, we show that the gap decays to zero exponentially if the joint Markov chain is weakly uniform ergodic. Let $\mu_k(i, A)$ be the state distribution of the joint Markov chain. The entropy of Y_k can be written as a function of $\mu_k(i, A)$ with

$$H(Y_k|\mu_k) = \sum_{i=0}^{N_Q-1} \int_{\mathfrak{D}(\mathcal{Q})} \mu_k(i, dx) V(i, x),$$

where

$$V(i, \boldsymbol{\alpha}) = \int_{\mathbb{Y}} \sum_{j=0}^{N_Q-1} p_{ij} f_{ij}(y) \log \left(\sum_{m=0}^{N_Q-1} \sum_{l=0}^{N_Q-1} \alpha_l p_{lm} f_{lm}(y) \right) dy.$$

The function $V(i, \boldsymbol{\alpha})$ gives the entropy rate of Y conditioned on the true state being i and the state probability vector being $\boldsymbol{\alpha}$. While $V(i, \boldsymbol{\alpha})$ is unbounded as $\alpha_i \to 0$, it is a continuous function of $\boldsymbol{\alpha}$ as long as $\alpha_i > 0$. Fortunately, the probability, $Pr(\alpha_i^{(t)} = 0|Q_t = i)$, must be zero because events with probability zero cannot occur.

Let $\boldsymbol{\pi} = [\ \pi_1 \quad \pi_2 \quad \dots \quad \pi_r\ ]$ be the unique stationary distribution of $\{Q_t\}_{t\geq 1}$ which satisfies $\pi_j = \sum_i \pi_i p_{ij}$. The lower bound, $H(Y_k|Y_{k-1}, Y_{k-2}, \dots, Y_1, Q_1)$, is obtained by starting the chain with the distribution

$$\mu_1(i, \boldsymbol{\alpha}) = \begin{cases} \pi_i & \text{if } \alpha_i = 1 \\ 0 & \text{otherwise} \end{cases} \tag{4.4.10}$$

and taking k steps, because this initial condition corresponds to stationary Q-state probabilities and perfect state knowledge. The upper bound is obtained by starting the chain with the distribution

$$\mu_1(i, \boldsymbol{\alpha}) = \begin{cases} \pi_i & \text{if } \boldsymbol{\alpha} = \boldsymbol{\pi} \\ 0 & \text{otherwise} \end{cases} \tag{4.4.11}$$

and taking k steps because this initial condition corresponds to stationary Q-state probabilities and no state knowledge. The true entropy rate can be computed by using either initialization and letting $k \to \infty$, because all initial conditions eventually converge to unique stationary distribution $\mu_\infty(i, A)$.

If the joint Markov chain is (weakly) uniform ergodic, then the state distribution converges to $\mu_\infty(i, A)$ exponentially fast in k regardless of the initial conditions. Since the upper and lower bounds are only functions of the state distribution, we find that both of these bounds converge to the true entropy rate exponentially fast in k.

4.4.6 Connections with Lyapunov Exponents

Consider any stationary stochastic process, $\{Y_t\}_{t\geq 1}$, equipped with a function, $\mathbf{M}(y)$, that maps each $y \in \mathbb{Y}$ to an $r \times r$ matrix. Let $\mathbf{Z}(\mathbf{Y}_1^n) = \mathbf{M}(Y_1)\mathbf{M}(Y_2)\dots\mathbf{M}(Y_n)$ be the cumulative product of random matrices generated by this process and let $\{\mathbf{y}_1^n\}_{n\geq 1}$ be a sequence of realizations with increasing length. Now, consider the limit

$$\lim_{n\to\infty} \frac{1}{n} \log \|\mathbf{x}\mathbf{Z}(\mathbf{y}_1^n)\|\,,$$

where $\mathbf{x}$ is any non-zero row vector and $\|\cdot\|$ is any vector norm. Oseledec's multiplicative ergodic theorem says that this limit is deterministic for almost all realizations [23]. While the proof takes a very different approach and is quite difficult, one way of thinking about this is that the matrix sequence, $\{\mathbf{Z}(\mathbf{y}_1^n)\}_{n\geq 1}$, can be associated with r eigenvalue sequences which grow (or decay)

exponentially in n. The normalized exponential growth rate of each eigenvalue sequence almost surely has a deterministic limit known as the Lyapunov exponent. The Lyapunov spectrum is the ordered set of Lyapunov exponents, $\gamma_1 > \gamma_2 > \ldots > \gamma_s$, along with their multiplicities, $d_1, d_2, \ldots, d_s$. An earlier ergodic theorem due to Furstenberg and Kesten [13] gives a simple proof for the top Lyapunov exponent, and says that the limit

$$\lim_{n\to\infty} \frac{1}{n} \log \|\mathbf{Z}(\mathbf{Y}_1^n)\| = \gamma_1$$

convergences almost surely, where $\|\cdot\|$ is now taken to the matrix norm induced by the previous vector norm (see [18, p. 303]).

The connection between Lyapunov exponents and the entropy rate of a FSP is given by the following proposition.

Proposition 4.4.12. *The largest Lyapunov exponent, γ_1, of the product of the transition-observation matrices, $\mathbf{M}(y_1)\mathbf{M}(y_2)\ldots\mathbf{M}(y_n)$, is almost surely equal to $-H(\mathcal{Y})$, where $H(\mathcal{Y})$ is the entropy rate of the FSP, $\{Y_t\}_{t\geq 1}$. Mathematically, we have*

$$\lim_{n\to\infty} \frac{1}{n} \log \|\mathbf{M}(Y_1)\mathbf{M}(Y_2)\ldots\mathbf{M}(Y_n)\| = \gamma_1 = -H(\mathcal{Y})$$

for almost all $\mathbf{Y}_1^n$.

Proof. Using (4.4.6), the probability $Pr(\mathbf{Y}_1^n)$ can be written in the form

$$Pr(\mathbf{Y}_1^n) = \sum_{i=1}^{r} \pi_i \sum_{j=1}^{r} [\mathbf{Z}(\mathbf{Y}_1^n)]_{ij} \,. \tag{4.4.12}$$

Applying the matrix norm induced (see [18, p. 303]) by the vector norm, $\|\cdot\|_\infty$, to $\mathbf{Z}(\mathbf{Y}_1^n)$ gives

$$\|\mathbf{Z}(\mathbf{Y}_1^n)\|_\infty = \max_i \sum_{j=1}^{r} [\mathbf{Z}(\mathbf{Y}_1^n)]_{ij} \,,$$

because our matrix is non-negative. Now, we can sandwich $Pr(\mathbf{Y}_1^n)$ with

$$\min_i \pi_i \, \|\mathbf{Z}(\mathbf{Y}_1^n)\|_\infty \leq Pr(\mathbf{Y}_1^n) \leq \|\mathbf{Z}(\mathbf{Y}_1^n)\|_\infty \tag{4.4.13}$$

by replacing the second sum in (4.4.12) by its maximum value to get an upper bound, and then applying the smallest π_i to that upper bound to get a lower bound. The ergodicity of the Markov

chain $\{Q_t\}_{t\geq 1}$ implies that $\min_i \pi_i > 0$, and therefore the inequality (4.4.13) can be rewritten as

$$\frac{1}{n}\log\left\|\mathbf{Z}(\mathbf{Y}_1^n)\right\|_\infty \leq \frac{1}{n}\log Pr(\mathbf{Y}_1^n) \leq \frac{1}{n}\log\left\|\mathbf{Z}(\mathbf{Y}_1^n)\right\|_\infty + \frac{\log\min_i \pi_i}{n}.$$

Finally, this shows that

$$\lim_{n\to\infty}\frac{1}{n}\log\left\|\mathbf{Z}(\mathbf{Y}_1^n)\right\|_\infty = \lim_{n\to\infty}\frac{1}{n}\log Pr(\mathbf{Y}_1^n),$$

which completes the proof. □

The following is a restatement of Theorem 7 from [9] and provides a CLT for the largest Lyapunov exponent.

Theorem 4.4.13 (Cohn-Nermann-Peligrad). *Suppose that $\{\mathbf{M}(Y_t)\}_{t\geq 1}$ is a strictly stationary sequence of $r \times r$ non-negative matrices satisfying: (i) there exists an integer n_0 such that $\mathbf{M}(Y_t)\mathbf{M}(Y_{t+1})\ldots\mathbf{M}(Y_{t+n_0})$ is positive with probability 1, (ii) the process $\{Y_t\}_{t\geq 1}$ is geometrically ergodic, and (iii) $E\left[\min^+\left(\log\left[\mathbf{M}(Y_t)\right]\right)^2\right] < \infty$ and $E\left[\max^+\left(\log\left[\mathbf{M}(Y_t)\right]\right)^2\right] < \infty$ where $\min^+\left(\mathbf{M}(Y_t)\right)$ and $\max^+\left(\mathbf{M}(Y_t)\right)$ are the minimum and maximum over the strictly positive elements of $\mathbf{M}(Y_t)$. Then there exists a $\sigma \geq 0$ such that*

$$n^{-1/2}\left[\log\left|[\mathbf{M}(Y_1)\mathbf{M}(Y_2)\ldots\mathbf{M}(Y_n)]_{ij}\right| - n\gamma_1\right] \xrightarrow{d} N(0,\sigma),$$

converges in distribution and γ_1 is the largest Lyapunov exponent.

The following Corollary provides a CLT for the entropy rate under conditions which are implied by Condition 4.4.8 and a finite variance condition.

Corollary 4.4.14. *Suppose that every observation, $y \in \mathbb{Y}$, is possible during every transition. This implies that $g_{ij}(y) > 0$ for all i, j such that $p_{ij} > 0$. Furthermore, suppose that condition (iii) of Theorem 4.4.13 holds. Then the sample entropy, $\hat{H}_n(\mathcal{Y})$, is asymptotically Gaussian with asymptotic mean $H(\mathcal{Y})$.*

Remark 4.4.15. Using (4.4.7) and the second Lyapunov exponent of the matrix $\mathbf{Z}(\mathbf{Y}_1^n)$, we can also consider the exponential rate at which $\boldsymbol{\alpha}^{(t)}$ forgets its initial condition $\boldsymbol{\alpha}^{(1)}$. Consider the scaled matrix product, $\mathbf{Z}(\mathbf{Y}_1^n)/\left\|\mathbf{Z}(\mathbf{Y}_1^n)\right\|_\infty$, whose maximal row sum will always equal one. The normalized second largest eigenvalue, $|\lambda_2|^{1/n}$, of this scaled matrix product will almost

certainly be equal to $e^{\gamma_2-\gamma_1}$. This is because the normalized eigenvalues of $\mathbf{Z}(\mathbf{Y}_1^n)$ will almost surely be given by the Lyapunov spectrum. Therefore, if the largest Lyapunov exponent is simple (i.e., its multiplicity is one), then $\boldsymbol{\alpha}^{(t)}$ forgets its initial condition almost surely at the positive exponential rate given by $\gamma_2 - \gamma_1$. It is important to note that this is the expected rate at which $\boldsymbol{\alpha}^{(t)}$ forgets its initial condition. This does not necessarily imply that the probability of rare events also decays exponentially.

4.5 Capacity Bounds

The capacity of a FSC is given by

$$C = \lim_{n\to\infty} \frac{1}{n} \max_{Pr(X_1^n)} I(X_1^n; Y_1^n),$$

where the limit always exists and is independent of the initial state [14, Chap. 4]. In terms of mutual information rates, this capacity can also be written as

$$C = \sup_{\mathcal{X}} \left[H(\mathcal{X}) + H(\mathcal{Y}) - H(\mathcal{X},\mathcal{Y}) \right],$$

where the supremum is taken over all stationary ergodic input processes. Unfortunately, the maximization implied by either formula is over an infinite dimensional distribution and impossible to carry out in practice. The capacity can be sandwiched between two computable quantities, however. Using upper and lower bounds on the entropy rates of the FSPs, we illustrate this in Sections 4.5.1, 4.5.2, and 4.5.3.

4.5.1 Lower Bounds

Lower bounds on the capacity are actually quite straightforward to compute because any achievable rate is a lower bound on the capacity. For example, we consider the maximum rate achievable using Markov input distributions with memory η. These distributions have a simple representation because $Pr(X_i|\mathbf{X}_1^{i-1}) = Pr(X_i|\mathbf{X}_{i-\eta}^{i-1})$. Let M_η be the set of all such input distributions Then the sequence $\left\{\underline{C}_\eta\right\}_{\eta\geq 0}$, defined by

$$\underline{C}_\eta = \lim_{n\to\infty} \frac{1}{n} \max_{Pr(\mathbf{X})\in M_\eta} I(\mathbf{X}_1^n; \mathbf{Y}_1^n),$$

is a sequence of lower bounds on the capacity. The sequence of bounds is non-decreasing because any Markov input process in M_η is also in $M_{\eta+1}$. We also note that the information rate, C_η, is referred to as the Markov-η rate of the channel.

Using standard optimization techniques these bounds were computed numerically for linear ISI channels with Gaussian noise in [24] and [1]. The results given in Section 4.6, however, were generated more accurately and efficiently using Kavčić's elegant generalization of the Arimoto-Blahut algorithm [19]. While no rigorous proof exists for the convergence of this algorithm, theoretical and numerical results strongly imply its correctness. In particular, it always returns a valid information rate and, in all cases tested, it gives results numerically equivalent to standard optimization techniques.

4.5.2 The Vontobel-Arnold Upper Bound

Upper bounds on the capacity are somewhat more difficult to compute because the maximization over all input distributions must be treated very carefully. Vontobel and Arnold propose an upper bound on the capacity of finite state channels in [31]. The first step in this upper bound can be seen as a generalization of the standard upper bound [14, Theorem 4.5.1] for DMCs. Let $I_P(X;Y)$ be the mutual information between the inputs and outputs of a DMC for some input distribution $P(x)$. Then, for any fixed channel (i.e., fixed $Pr(Y|X)$), the upper bound states that

$$C = \max_{P_0(x)} I_{P_0}(X;Y) \leq \max_{x} I_{P_1}(X=x;Y), \tag{4.5.1}$$

where

$$I_P(X=x;Y) = E\left[\log \frac{Pr(Y|X)}{\sum_{x'\in\mathbb{X}} Pr(Y|X=x')P(x')}\middle| X=x\right].$$

The natural generalization of this upper bound to channels with memory implies that

$$C = \lim_{n\to\infty} \frac{1}{n} \max_{P_0(\mathbf{x}_1^n)} I_{P_0}(\mathbf{X}_1^n;\mathbf{Y}_1^n) \leq \lim_{n\to\infty} \frac{1}{n} \max_{x_1^n} I_{P_1}(\mathbf{X}_1^n=\mathbf{x}_1^n;\mathbf{Y}_1^n) \tag{4.5.2}$$

for a fixed channel (i.e., fixed $Pr(\mathbf{Y}_1^n|\mathbf{X}_1^n)$) and any $P_1(\mathbf{X}_1^n)$. Vontobel and Arnold start by noting that

$$C \leq \lim_{n\to\infty} \frac{1}{n} \max_{\mathbf{x}_1^n} E\left[\log \frac{Pr(\mathbf{Y}_1^n|\mathbf{X}_1^n)}{R(\mathbf{Y}_1^n)}\middle| \mathbf{X}_1^n=\mathbf{x}_1^n\right] \tag{4.5.3}$$

holds for any distribution $R(\mathbf{Y}_1^n)$. By choosing an $R(\mathbf{Y}_1^n)$ which can be factored according to

$$R(\mathbf{y}_1^n) = R_1^L(\mathbf{y}_1^L) \prod_{i=L+1}^{n} R(y_i|\mathbf{y}_{i-L}^{i-1}),$$

they are able to make this bound computable as well. This distribution, $R(Y_i|\mathbf{Y}_{i-L}^{i-1})$, is generally chosen to be the marginal distribution, $Pr(Y_i|\mathbf{Y}_{i-L}^{i-1})$, because this choice minimizes, for any given L, the quantity

$$E\left[\log \frac{Pr(\mathbf{Y}_1^n|\mathbf{X}_1^n)}{R(\mathbf{Y}_1^n)}\right].$$

Their method of making the bound computable is actually quite clever. It is based upon writing the conditional expectation of (4.5.3) in a form which makes the maximization easy. For FSCs whose state is defined by the previous ν inputs (e.g., any linear ISI channel), we can write

$$E\left[\log \frac{Pr(\mathbf{Y}_1^n|\mathbf{X}_1^n)}{R(\mathbf{Y}_1^n)}\middle|\mathbf{X}_1^n = \mathbf{x}_1^n\right] = K_0(\mathbf{x}_1^{L+\nu}) + E\left[\sum_{i=L+\nu+1}^{n} \log \frac{Pr(Y_i|\mathbf{X}_{i-\nu}^{i})}{R(Y_i|\mathbf{Y}_{i-L}^{i-1})}\middle|\mathbf{X}_1^n = \mathbf{x}_1^n\right],$$

where $K_0(\mathbf{x}_1^{L+\nu})$ absorbs the contribution of the neglected $L+\nu$ initial terms of the sum. The conditional expectation of the ith term in the sum only requires knowledge of $\mathbf{x}_{i-L-\nu}^{i}$. So, using the definition

$$K(\mathbf{x}) = E\left[\log \frac{Pr(Y_i|\mathbf{X}_{i-\nu}^{i})}{R(Y_i|\mathbf{Y}_{i-L}^{i-1})}\middle|\mathbf{X}_{i-L-\nu}^{i} = \mathbf{x}\right], \tag{4.5.4}$$

we have

$$E\left[\log \frac{Pr(\mathbf{Y}_1^n|\mathbf{X}_1^n)}{R(\mathbf{Y}_1^n)}\middle|\mathbf{X}_1^n = \mathbf{x}_1^n\right] = K_0(\mathbf{x}_1^{L+\nu}) + \sum_{i=L+\nu+1}^{n} K(\mathbf{x}_{i-L-\nu}^{i}).$$

Computing the function $F(\mathbf{x}_1^n) = K_0(\mathbf{x}_1^{k_0}) + \sum_{i=k_0+1}^{n} K(\mathbf{x}_{i-i_0}^{i})$ is equivalent to computing the weight of a path (labeled by $\mathbf{x}_1^n$) through an edge-weighted directed graph. Therefore, the Viterbi algorithm can be used to find the maximum of the function over $\mathbf{x}_1^n$. The quantity we actually want, however, is the limiting value

$$\lim_{n\to\infty} \frac{1}{n} \max_{\mathbf{x}_1^n} F(\mathbf{x}_1^n).$$

In the literature, finding this quantity is known as the *minimum mean cycle* problem [11]. The connection is based on fact that the maximum (or minimum) average weight path spends most of

its time walking the same maximum (or minimum) average weight cycle repeatedly. Therefore, the answer is given by the maximum (or minimum) average cycle weight of the graph.

The practical problem of computing the function value, $K(\mathbf{x}_{i-L-\nu}^{i})$, for each binary $(L+\nu)$-tuple can be solved by using a simulation to estimate the expectation in (4.5.4). The complexity of this method linear in the simulation length and exponential in $L+\nu$ because we run one simulation for each $\mathbf{x}$. In some cases, the trade off between complexity and estimation error may also be reduced by using one long simulation and using Bayes' rule to write the conditional expectation as

$$K(\mathbf{x}) = E\left[\frac{Pr(\mathbf{X}_{i-L-\nu}^{i} = \mathbf{x}|\mathbf{Y}_{i-L}^{i-1})}{Pr(\mathbf{X}_{i-L-\nu}^{i} = \mathbf{x})} \log \frac{Pr(Y_i|\mathbf{X}_{i-L-\nu}^{i} = \mathbf{x})}{R(Y_i|\mathbf{Y}_{i-L}^{i-1})}\right].$$

The major drawback of the Vontobel-Arnold Bound is that it can be quite loose for channels whose state sequence is not identifiable from a small number of samples. Consider, for example, a dicode channel with very little noise. The lprobability of observing either a positive or negative transition, after observing L samples near zero, remains large enough to weaken the bound significantly. One might think that the small probability of observing long runs of zeroes at the output would counteract this problem. This is not the case, however, because the maximization over $\mathbf{x}_1^n$ picks the worst-case sequence, regardless of its probability.

4.5.3 A Conjectured Upper Bound

Now, we derive a slightly different expression that we conjecture is also an upper bound on the capacity of IFSCs. This bound also starts with (4.5.2), but uses different simplifications to make the bound computable. We start by considering an IFSC channel, with state sequence $\mathbf{S}_1^n$, driven by a Markov input process with memory $\eta \leq L$. The channel and input state can be combined into a single state variable, and that corresponding state sequence is $\mathbf{Q}_1^n$. The basic idea is that we can upper bound the mutual information by considering a genie-aided decoder which has perfect knowledge of the states Q_{i-L} and Q_{i+L+1} when it is decoding the input X_i. This expression remains a conjectured upper bound because of a subtle gap that remains in our proof.

We begin by upper bounding $I(\mathbf{X}_1^n; \mathbf{Y}_1^n)$ using the chain rule for mutual information

and a genie-aided decoder. The chain rule gives

$$I(\mathbf{X}_1^n;\mathbf{Y}_1^n) = \sum_{i=1}^{n} I(X_i;\mathbf{Y}_1^n|\mathbf{X}_1^{i-1})$$

and, neglecting edge effects, the genie-aided upper bound for each term in the sum is given by

$$I(X_i;\mathbf{Y}_1^n|\mathbf{X}_1^{i-1}) \le I(X_i;Q_{i-L},\mathbf{Y}_1^n,Q_{i+L+1}|\mathbf{X}_1^{i-1}).$$

For any input distribution, $P(\mathbf{x}_1^n)$, we define the genie-aided mutual information to be

$$J_P(\mathbf{X}_1^n;\mathbf{Y}_1^n) = \sum_{\mathbf{x}_1^n} P(\mathbf{x}_1^n) J_P(\mathbf{X}_1^n = \mathbf{x}_1^n;\mathbf{Y}_1^n), \tag{4.5.5}$$

where $J_P(\mathbf{X}_1^n = \mathbf{x}_1^n;\mathbf{Y}_1^n)$ is defined with

$$J_P(\mathbf{X}_1^n = \mathbf{x}_1^n;\mathbf{Y}_1^n) = \sum_{i=1}^{n} E\left[\log \frac{Pr(X_i|\mathbf{X}_1^{i-1},Q_{i-L},\mathbf{Y}_1^n,Q_{i+L+1})}{Pr(X_i|\mathbf{X}_1^{i-1})}\middle|\mathbf{X}_1^n = \mathbf{x}_1^n\right].$$

The Markov nature of the input distribution and the channel allow us to write

$$J_P(\mathbf{X}_1^n = \mathbf{x}_1^n;\mathbf{Y}_1^n) = \sum_{i=1}^{n} E\left[\log \frac{Pr(X_i|\mathbf{X}_{i-L}^{i-1},Q_{i-L},\mathbf{Y}_{i-L}^{i+L},Q_{i+L+1})}{Pr(X_i|\mathbf{X}_{i-\eta}^{i-1})}\middle|\mathbf{X}_{i-L}^{i+L} = \mathbf{x}_{i-L}^{i+L}\right],$$

and this provides an upper bound on $I(\mathbf{X}_1^n;\mathbf{Y}_1^n)$ which is computed as the sum of local functions.

The obvious generalization of the Vontobel-Arnold Bound would imply that $C \le \lim_{n\to\infty} \max_{\mathbf{x}_1^n} J_P(\mathbf{X}_1^n = \mathbf{x}_1^n;\mathbf{Y}_1^n)$. Unfortunately, this does not follow directly from the results that $C \le \lim_{n\to\infty} \max_{\mathbf{x}_1^n} I(\mathbf{X}_1^n = \mathbf{x}_1^n;\mathbf{Y}_1^n)$ and $I_P(\mathbf{X}_1^n;\mathbf{Y}_1^n) \le J_P(\mathbf{X}_1^n;\mathbf{Y}_1^n)$. This is because it is possible that the the genie-aided mutual information could be larger when averaged over all sequences, even though its largest per-sequence value is smaller. Our proof of this bound requires that the chain of inequalities,

$$\max_{P_0(\mathbf{x}_1^n)} I_{P_0}(\mathbf{X}_1^n;\mathbf{Y}_1^n) \le \max_{P_1(\mathbf{x}_1^n)} J_{P_1}(\mathbf{X}_1^n;\mathbf{Y}_1^n) \le \max_{\mathbf{x}_1^n} J_{P_2}(\mathbf{X}_1^n = \mathbf{x}_1^n;\mathbf{Y}_1^n),$$

holds for all $P_2(\mathbf{x}_1^n)$. The LHS inequality indeed holds because $J_P(\mathbf{X}_1^n;\mathbf{Y}_1^n)$ is an upper bound on $I_P(\mathbf{X}_1^n;\mathbf{Y}_1^n)$ for all $P(\mathbf{x}_1^n)$. We conjecture that the RHS inequality holds as well. Numerical results for the dicode channel are encouraging, however, because our lower bounds on the capacity are quite close to the conjectured upper bound but do not surpass it.

Following the Vontobel-Arnold approach, we make the conjectured upper bound computable by writing it as the sum of local functions. For ISI channels whose state sequence is a deterministic function of the input sequence, we can simplify the function $J_P(\mathbf{X}_1^n = \mathbf{x}_1^n; \mathbf{Y}_1^n)$ to

$$F(\mathbf{x}_1^n) = \sum_{i=1}^{n} E\left[\log \frac{Pr(X_i|Q_{i-1}, \mathbf{Y}_i^{i+L}, Q_{i+L+1})}{Pr(X_i|\mathbf{X}_{i-\eta}^{i-1})}\middle| \mathbf{X}_{i-\eta}^{i+L} = \mathbf{x}_{i-\eta}^{i+L}\right].$$

The simplification uses the facts that X_i is conditionally independent of the past given Q_{i-1} and Q_{i-1} is computable from Q_{i-L} and $\mathbf{X}_{i-L}^{i-1}$. Let ν be the memory of the channel, η be the memory of the input, and assume that $\eta \geq \nu$ (without loss of generality). This allows us to write

$$F(\mathbf{x}_1^n) = K_0(\mathbf{x}_1^{L+\eta}) + \sum_{i=\eta+L+1}^{n-L} K(\mathbf{x}_{i-\eta}^{i+L}) + K_1(\mathbf{x}_{n-\eta-L}^{n}),$$

where

$$K(\mathbf{x}) = E\left[\log \frac{Pr(X_i|Q_{i-1}, \mathbf{Y}_i^{i+L}, Q_{i+L+1})}{Pr(X_i|\mathbf{X}_{i-\eta}^{i-1})}\middle| \mathbf{X}_{i-\eta}^{i+L} = \mathbf{x}\right].$$

The terms $K_0(\mathbf{x}_1^{L+\eta})$ and $K_1(\mathbf{x}_{n-\eta-L}^{n})$ are asymptotically irrelevant and will be ignored. Since $F(\mathbf{x}_1^n) = J_P(\mathbf{X}_1^n = \mathbf{x}_1^n; \mathbf{Y}_1^n)$, the conjectured upper bound on capacity is given by

$$C \leq \lim_{n\to\infty} \frac{1}{n} \max_{\mathbf{x}_1^n} F(\mathbf{x}_1^n).$$

Once again, the function $F(\mathbf{x}_1^n)$ can be computed as the weight of a path (labeled by $\mathbf{x}_1^n$) through an edge-weighted directed graph. In this case, the states are labeled by $\mathbf{x}_{i-\eta}^{i+L-1}$ and the edge weights are estimated by stochastic averaging of the expectation in $K(\mathbf{x})$. As with the Vontobel-Arnold bound, the conjectured upper bound is given by the maximum average cycle weight of the graph.

4.6 Monte Carlo Results

4.6.1 Partial Response Channels

We start by giving the results for the power normalized binary-input channels listed in Table 4.1. These channels are known as partial response (PR) channels and are sometimes used to model magnetic recording channels. First, we show the SIR of these channels along with binary-input AWGN capacity in Figure 4.6.1. For each channel, the achievable rate is plotted

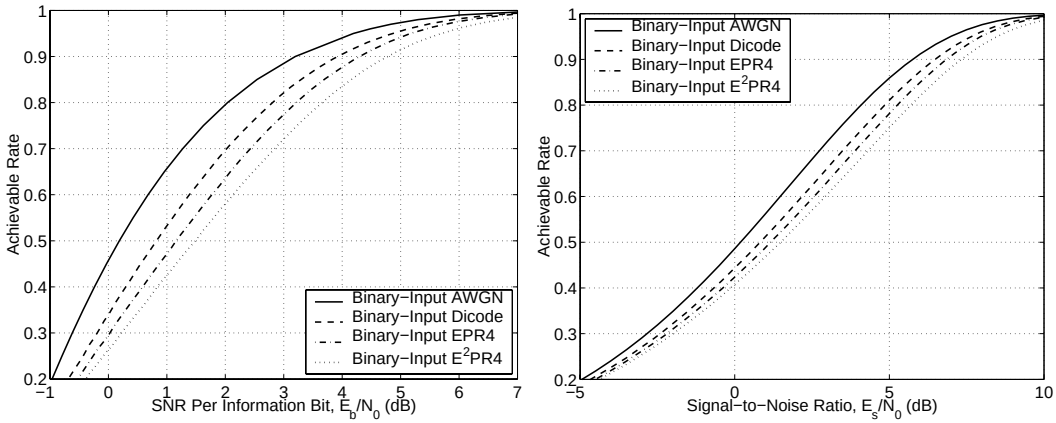


Figure 4.6.1: The SIR for various partial response channels, estimated with $n = 10^7$.

versus both the SNR (E_s/N_0) and the SNR per information bit (E_b/N_0). Notice that increasing the severity of the ISI monotonically decreases the SIR in both cases.

Next, we consider the results attained by optimizing the input distributions for these channels. The elegant generalization of the Arimoto-Blahut algorithm due to Kavčić is used for all of these results [19]. Figure 4.6.2 shows the results for the dicode channel and plots the achievable rate versus E_s/N_0 and E_b/N_0. Figure 4.6.3 shows the results of optimizing input distributions for the EPR4 channel. All of these results show that optimizing the input distribution provides significant gains at low SNR.

Figure 4.6.4 compares the dicode channel lower bound with the Vontobel-Arnold Bound and our own conjectured upper bound. We would like to acknowledge P. Vontobel for providing the data points for the Vontobel-Arnold Bound. Both upper bounds are somewhat loose at low rates, but the conjectured upper bound is actually quite tight at high rates. The conjectured upper bound has an intrinsic advantage at high rates because it is always upper bounded entropy of the input process. We also note that this comparison is not entirely fair because the conjectured upper bound was numerically optimized over the input distribution while the Vontobel-Arnold Bound was not. In fact, without optimization the performance of the conjectured upper bound at low rates does not surpass the Vontobel-Arnold Bound. In the future, we plan to make a fair comparison by optimizing the Vontobel-Arnold Bound as well.

It is worth noting that at low enough SNR, all of the optimized rates actually exceed the capacity of the binary-input AWGN channel. Depending on your perspective, this is either

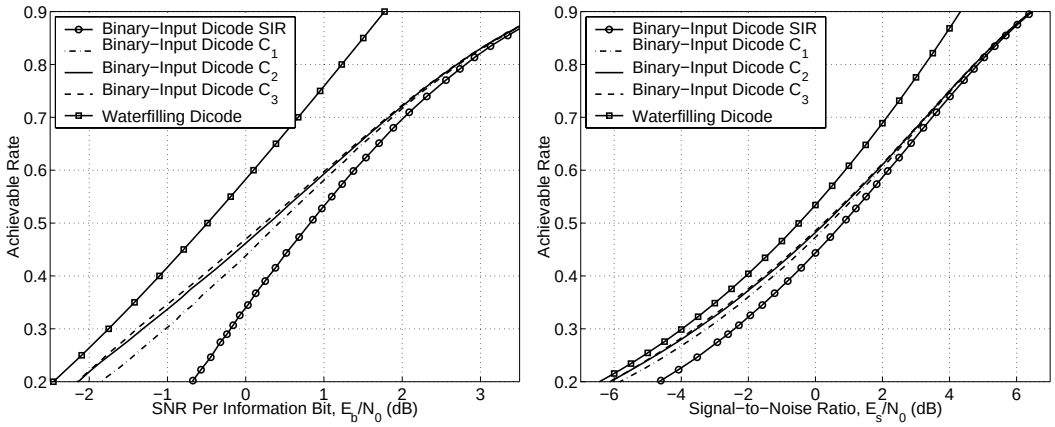


Figure 4.6.2: Monte Carlo lower bounds on the achievable information rate of the dicode channel using optimized Markov input distributions.

an interesting phenomenon or simply a poorly chosen channel normalization. The basic problem is that there is no single normalization which is fair. By convention, we normalize each channel so that the power of a white input signal is unchanged by the channel. This approach seems fair for white input signals such as equiprobable binary inputs. When the channel response is not flat, however, optimizing the input distribution allows the source to concentrate its power around the peaks of the channel response. Now, the signal appears to be receiving a *power gain* from the channel. One solution is to normalize all channels so that the peak of the response is unity. This is merely a different convention, however, and regardless of the chosen normalization, optimizing input distribution will always increase the power output of the channel.

Finally, we remark that at low rates these binary-input ISI channels exhibit a threshold behavior similar to the $-1.59\,dB$ limit of the AWGN channel. Essentially, this means that there is an E_b/N_0 threshold below which reliable communication is impossible. Conversely, it can be shown that, at sufficiently low rates, reliable communication is also possible at any E_b/N_0 larger than this threshold. The threshold is known as the low-rate Shannon limit and is discussed in [28].

4.6.2 The Finite State Z-Channel

The SIR and the Markov-1 rate of the finite state Z-channel are shown in Figure 4.6.5. It is interesting to note that, as with the original Z-channel, one can avoid transmission errors by

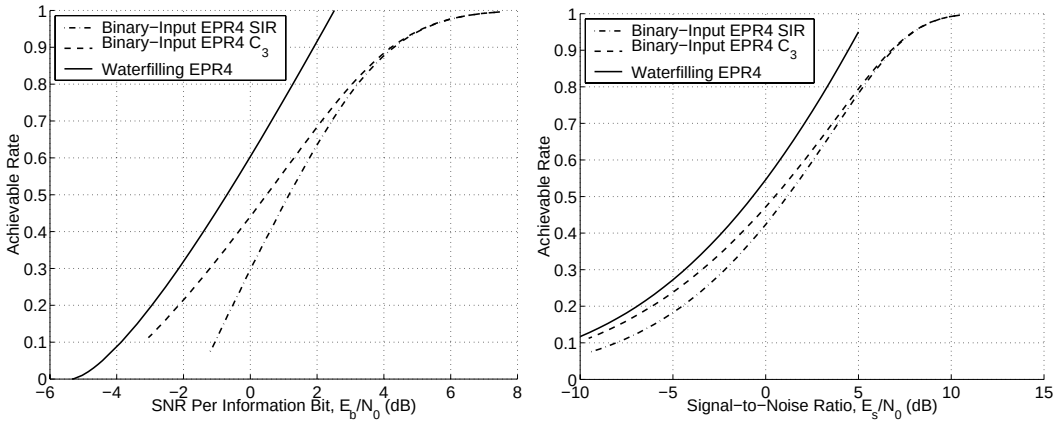


Figure 4.6.3: Monte Carlo lower bounds on the achievable information rate of the EPR4 channel using optimized Markov input distributions.

sending a particular pattern. (e.g., 0101...01). So, while optimizing the input distribution to the dicode channel increases the output power, optimizing the input distribution to the finite state Z-channel decreases the noise. Therefore, the optimized input distribution chooses transitions (i.e., edges from state 0 to state 1 and vice-versa) more frequently than non-transitions (i.e., edges corresponding to self-loops) and thereby incurs fewer channel errors.

4.7 Analytical Results

In this section, we consider analytical methods for computing achievable information rates. We start by using the results of Section 4.4.4 to compute exact information rates for the DEC. This analysis of the DEC is made possible by the fact that the stationary distribution of the joint Markov chain is supported on a countably infinite set. This follows from Theorem 4.4.7 because the reception of a $+$ or $-$ symbol gives the observer perfect state knowledge. Next, we describe a pseudo-analytical method based on density evolution that can be used to estimate rates more efficiently for arbitrary two state channels.

4.7.1 The Symmetric Information Rate of the DEC

Consider a DEC with erasure probability ϵ and equiprobable inputs. Let $\mathbf{X}_1^n$ be the channel input sequence, $\mathbf{S}_1^{n+1}$ be the channel state sequence, and $\mathbf{Y}_1^n$ be the channel output sequence. The input and output alphabets of the DEC are defined so that $X_i \in \{0, 1\}$ and

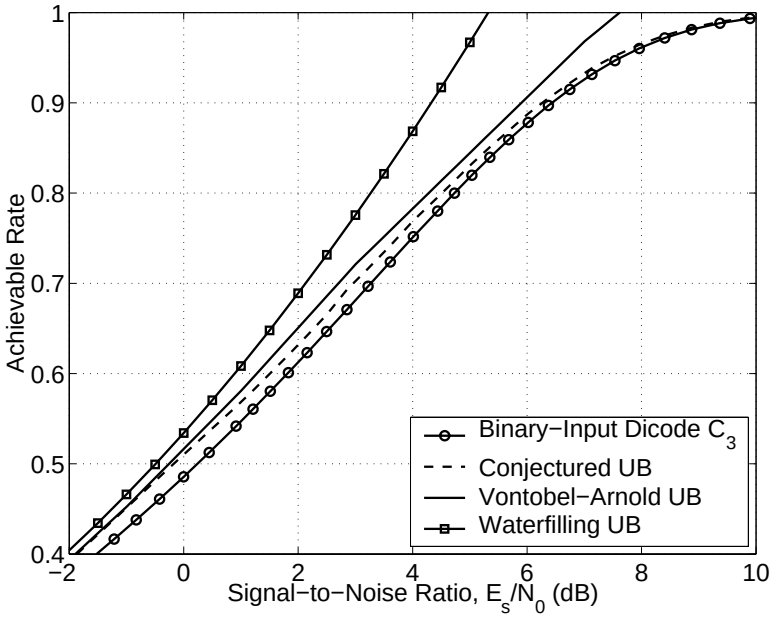


Figure 4.6.4: Monte Carlo upper and lower bounds on the achievable information rate of the dicode channel using optimized Markov input distributions.

$Y_i \in \{-, 0, +, e\}$. In this section, we compute the exact SIR analytically by characterizing the forward recursion of the APP algorithm, which computes $Pr(Y_t|Y_1^{t-1})$, in terms of the random variable $\alpha_i^{(t)} = Pr(S_t = i|Y_1^t)$. Parts of this analysis were motivated by a method used to compute iterative decoding thresholds for turbo codes on the binary erasure channel [30].

Since the channel has only two states, it suffices to consider the quantity $\alpha^{(t)} \triangleq \alpha_0^{(t)} = 1 - \alpha_1^{(t)}$. The real simplification, however, comes from the fact that the distribution of $\alpha^{(t)}$ has finite support when $X \sim B(1/2)$. We can observe this fact by writing the APP recursion as (4.4.7) where $\boldsymbol{\alpha}^{(t)} = \left[\alpha^{(t)} \;\; 1 - \alpha^{(t)}\right]$ and

$$\mathbf{M}(e)=\begin{bmatrix} \frac{\epsilon}{2} & \frac{\epsilon}{2} \\ \frac{\epsilon}{2} & \frac{\epsilon}{2} \end{bmatrix}, \mathbf{M}(0)=\begin{bmatrix} \frac{1-\epsilon}{2} & 0 \\ 0 & \frac{1-\epsilon}{2} \end{bmatrix}, \mathbf{M}(+)=\begin{bmatrix} 0 & \frac{1-\epsilon}{2} \\ 0 & 0 \end{bmatrix}, \mathbf{M}(-)=\begin{bmatrix} 0 & 0 \\ \frac{1-\epsilon}{2} & 0 \end{bmatrix}.$$

Using this, it is easy to verify that we can use instead the simpler recursion,

$$\alpha^{(t+1)} = \begin{cases} 1/2 & \text{if } Y_t = e \\ \alpha^{(t)} & \text{if } Y_t = 0 \\ 0 & \text{if } Y_t = + \\ 1 & \text{if } Y_t = - \end{cases}.$$

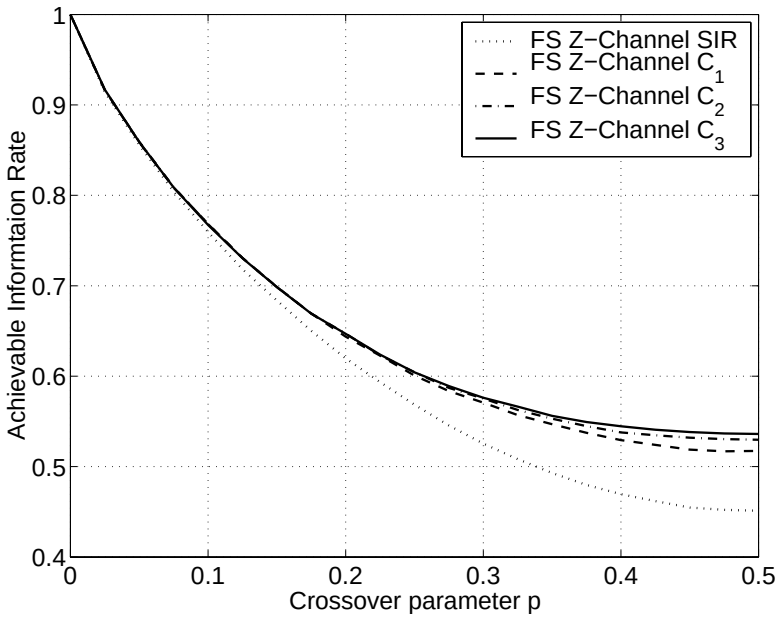


Figure 4.6.5: The SIR and the Markov-1 rate of the finite state Z-channel.

Using this, we see that, for all $t \geq \min\{i \geq 1 | Y_i \neq 0\}$, $\alpha^{(t)}$ will be confined to the finite set $\{0, 1/2, 1\}$.

With the results of Section 4.4.4 in mind, we proceed by finding the stationary distribution of the joint Markov chain, $\left\{Q_t, \boldsymbol{\alpha}^{(t)}\right\}_{t \geq 1}$. Each state of this Markov chain can be indexed by the pair $(q_t, \alpha^{(t)})$, and the stationary distribution is supported on the set $(0,1)$, $(1,0)$, $(0,1/2)$, and $(1,1/2)$. The first two states correspond to the known (K) state condition, while the second two correspond to the unknown (U) state condition. The symmetry of the problem allows us to write $Pr(K)/2 = \pi(0,1) = \pi(1,0)$ and $Pr(U)/2 = \pi(0,1/2) = \pi(1,1/2)$.

Using a two state Markov chain, we can compute the steady state probabilities of the joint Markov chain. First, we note that the joint Markov chain transitions from the known state condition to the unknown state condition only if $Y = e$. Therefore, we have $Pr(K \rightarrow U) = 1 - Pr(K \rightarrow K) = \epsilon$. Secondly, we note that the joint Markov chain transitions from the unknown state condition to the known state condition only if $Y = +$ or $Y = -$. This means that we have $Pr(U \rightarrow K) = 1 - Pr(U \rightarrow U) = (1 - \epsilon)/2$. The steady state probabilities $Pr(K)$

and $Pr(U)$ can be found using the eigenvector equation,

$$\begin{bmatrix} Pr(K) & Pr(U) \end{bmatrix} \begin{bmatrix} 1-\epsilon & \epsilon \\ \frac{1-\epsilon}{2} & \frac{1+\epsilon}{2} \end{bmatrix} = \begin{bmatrix} Pr(K) & Pr(U) \end{bmatrix},$$

whose solution is $Pr(U) = 1 - Pr(K) = 2\epsilon/(1+\epsilon)$.

Now, we can compute the exact entropy rate of Y_1^n using the definition $H(\mathcal{Y}) = \lim_{t\to\infty} H(Y_t|Y_1^{t-1})$. When the joint Markov chain is in a known state, the observer knows that one of only two edges can be traversed in the next step. In this case, the conditional entropy of the next output given the past is denoted $H(Y|K)$ and is given by

$$H(Y|K) = H\left(\left[\epsilon, \frac{1-\epsilon}{2}, \frac{1-\epsilon}{2}\right]\right) = h(\epsilon) + (1-\epsilon),$$

where $H\left([p_1, \ldots, p_k]\right) = -\sum_{i=1}^{k} p_i \log_2 p_i$ and $h(\epsilon) = -\epsilon \log_2 \epsilon - (1-\epsilon)\log_2(1-\epsilon)$. When the joint Markov chain is in an unknown state, the observer must allow the possibility of that any of four edges may actually be traversed in the next step. In this case, the conditional entropy of the next output given the past is denoted $H(Y|U)$ and is given by

$$H(Y|U) = H\left(\left[\epsilon, \frac{1-\epsilon}{4}, \frac{1-\epsilon}{2}, \frac{1-\epsilon}{4}\right]\right) = h(\epsilon) + \frac{3(1-\epsilon)}{2}.$$

Since the stationary distribution of the joint Markov chain determines how often each of these events occurs, we can write

$$H(\mathcal{Y}) = Pr(K)H(Y|K) + Pr(U)H(Y|U).$$

Substituting exact values into this expression and simplifying gives

$$H(\mathcal{Y}) = 1 - \frac{2\epsilon^2}{1+\epsilon} + h(\epsilon).$$

Since the entropy rate of $\mathbf{Y}_1^n$ given $\mathbf{X}_1^n$, $H(\mathcal{Y}|\mathcal{X})$, is simply the entropy rate of the erasure process (i.e., $h(\epsilon)$), the SIR is given by

$$H(\mathcal{Y}) - H(\mathcal{Y}|\mathcal{X}) = H(\mathcal{Y}) - h(\epsilon) = 1 - \frac{2\epsilon^2}{1+\epsilon}.$$

4.7.2 The Markov-1 Rate of the DEC

In this section, we derive the achievable information rate of the DEC using a Markov-1 input distribution. Based on the symmetry of the channel, we use an input distribution which

changes state with probability p and remains in the same state with probability $1-p$. The combined state space of the input process and the channel still only has two states, so again it suffices to consider only $\alpha^{(t)} \triangleq \alpha_0^{(t)} = 1 - \alpha_1^{(t)}$.

In this case, we can write the APP recursion as (4.4.7) where $\boldsymbol{\alpha}^{(t)} = \left[\alpha^{(t)} \;\; 1-\alpha^{(t)}\right]$,

$$\mathbf{M}(e) = \begin{bmatrix} (1-p)\epsilon & p\epsilon \\ p\epsilon & (1-p)\epsilon \end{bmatrix}, \; \mathbf{M}(0) = \begin{bmatrix} (1-p)(1-\epsilon) & 0 \\ 0 & (1-p)(1-\epsilon) \end{bmatrix},$$

$$\mathbf{M}(+) \begin{bmatrix} 0 & p(1-\epsilon) \\ 0 & 0 \end{bmatrix}, \text{ and } \mathbf{M}(-) = \begin{bmatrix} 0 & 0 \\ p(1-\epsilon) & 0 \end{bmatrix}.$$

A simple recursion also exists in this case and is given by

$$\alpha^{(t+1)} = \begin{cases} \alpha^{(t)}(1-p) + (1-\alpha^{(t)})p & \text{if } Y_t = e \\ \alpha^{(t)} & \text{if } Y_t = 0 \\ 0 & \text{if } Y_t = + \\ 1 & \text{if } Y_t = - \end{cases}.$$

The major complication in completing the analysis is the fact that the support set of $\alpha^{(t)}$ is now countably infinite. For example, the $\alpha^{(t)}$ that results from observing a $-$ first and then observing a mixture of k erasures and any number of 0's (but no more $+$'s and $-$'s) is given by $\left(1+(1-2p)^k\right)/2$. Likewise, if the first observation was a $+$, then we would have $\left(1-(1-2p)^k\right)/2$. These two cases, with $k \in \{0, \ldots, \infty\}$, constitute the entire support set of $\alpha^{(t)}$. For simplicity, we refer to these values using the shorthand,

$$\gamma_k^{\pm} = \frac{1 \pm (1-2p)^k}{2}. \tag{4.7.1}$$

Now, we define the countably infinite Markov chain that will be used to help analyze the joint Markov chain. Each state in the new Markov chain is labeled by a letter (A or B) and a non-negative integer. The A_k state corresponds to the event that $(S_t = 0, \alpha^{(t)} = \gamma_k^+)$ or $(S_t = 1, \alpha^{(t)} = \gamma_k^-)$. The symmetry of the system can be used to show these events occur with equal probability. The B_k state corresponds to the event that $(S_t = 0, \alpha^{(t)} = \gamma_k^-)$ or $(S_t = 1, \alpha^{(t)} = \gamma_k^+)$. Again, symmetry forces these events to occur with equal probability. The state probabilities, as a function of time, are defined by

$$\begin{aligned} A_k^{(t)} &= Pr(S_t = 0, \alpha^{(t)} = \gamma_k^+) + Pr(S_t = 1, \alpha^{(t)} = \gamma_k^-) \\ B_k^{(t)} &= Pr(S_t = 0, \alpha^{(t)} = \gamma_k^-) + Pr(S_t = 1, \alpha^{(t)} = \gamma_k^+). \end{aligned}$$

The basic idea of the new Markov chain is to simultaneously track the true state and count the number of erasures since the last instance of perfect knowledge. In doing this, we find that an observed 0 causes the transitions $A_k \to A_k$ and $B_k \to B_k$, an erased 0 causes the transitions $A_k \to A_{k+1}$ and $B_k \to B_{k+1}$, an erased $+$ or $-$ causes the transitions $A_k \to B_{k+1}$ and $B_k \to A_{k+1}$, and an observed $+$ or $-$ causes the transitions $A_k \to A_0$ and $B_k \to A_0$. Using the probabilities of these events gives the recursions

$$\begin{aligned} A_0^{(t+1)} &= A_0^{(t)}(1-\epsilon) + (1-A_0^{(t)})p(1-\epsilon) \\ A_k^{(t+1)} &= A_k^{(t)}(1-p)(1-\epsilon) + A_{k-1}^{(t)}(1-p)\epsilon + B_{k-1}^{(t)}p\epsilon \\ B_k^{(t+1)} &= B_k^{(t)}(1-p)(1-\epsilon) + B_{k-1}^{(t)}(1-p)\epsilon + A_{k-1}^{(t)}p\epsilon. \end{aligned}$$

Solving for the stationary distribution of the new Markov chain gives

$$\begin{aligned} Pr(A_0) &= \frac{p(1-\epsilon)}{p(1-\epsilon)+\epsilon} \\ Pr(A_k) &= Pr(A_0)\omega^k\gamma_k^+ \\ Pr(B_k) &= Pr(A_0)\omega^k\gamma_k^- \end{aligned}$$

where $\omega = \frac{\epsilon}{1-(1-p)(1-\epsilon)}$. Since the forward state probabilities must give a consistent state estimate, the events $(S_t = 0, \alpha^{(t)} = 0)$ and $(S_t = 1, \alpha^{(t)} = 1)$ must have probability zero. This also implies that $B_0^{(t)} = Pr(S_t = 0, \alpha^{(t)} = 0) + Pr(S_t = 1, \alpha^{(t)} = 1) = 0$. Finally, the the new Markov chain is uniformly ergodic by Theorem 4.4.7 and converges to its unique stationary distribution exponentially fast.

Now, we can compute the entropy rate using the limit $H(\mathcal{Y}) = \lim_{t\to\infty} H(Y_t|\mathbf{Y}_1^{t-1})$. When the new Markov chain is in state A_k or B_k, the conditional entropy is denoted by $H(Y|A_k)$ or $H(Y|B_k)$, respectively. These two expressions are given by

$$H(Y|A_k) = -\epsilon\log_2\epsilon - (1-p)(1-\epsilon)\log_2\left((1-p)(1-\epsilon)\right) - p(1-\epsilon)\log_2\left(p(1-\epsilon)\gamma_k^+\right)$$

and

$$H(Y|B_k) = -\epsilon\log_2\epsilon - (1-p)(1-\epsilon)\log_2\left((1-p)(1-\epsilon)\right) - p(1-\epsilon)\log_2\left(p(1-\epsilon)\gamma_k^-\right).$$

The first term of each expression is associated with the observation probability of an erasure, the second term with the observation probability of a 0, and the third term with the observation

probability of either a $+$ or a $-$. Combining these with the stationary distribution of the new Markov chain gives final entropy rate

$$\begin{aligned} H(\mathcal{Y}) &= \sum_{k=0}^{\infty} Pr(A_k)H(Y|A_k) + Pr(B_k)H(Y|B_k) \\ &= D - p(1-\epsilon)\sum_{k=0}^{\infty} Pr(A_0)\omega^k \left(\gamma_k^+ \log_2 \gamma_k^+ + \gamma_k^- \log_2 \gamma_k^-\right), \end{aligned} \tag{4.7.2}$$

where

$$D = -\epsilon \log_2 \epsilon - (1-p)(1-\epsilon)\log_2\left((1-p)(1-\epsilon)\right) - p(1-\epsilon)\log_2\left(p(1-\epsilon)\right).$$

While we could find no closed form solution for this infinite sum, we did find a relatively simple approximation. Using (4.7.1), we can write

$$\log_2 \gamma_k^{\pm} = \log_2\left(1 \pm (1-2p)^k\right) - 1,$$

and then use the two term Taylor expansion, $\log_2(1+x) \approx (x - x^2/2)/\ln 2$, to get

$$\log_2 \gamma_k^{\pm} \approx \pm\frac{(1-2p)^k}{\ln 2} - \frac{(1-2p)^{2k}}{2\ln 2} - 1.$$

This lead us to the approximation

$$\gamma_k^+ \log_2 \gamma_k^+ + \gamma_k^- \log_2 \gamma_k^- \approx \frac{(1-2p)^{2k}}{2\log 2} - 1,$$

which allows the infinite sum (4.7.2) to be approximated in closed form by

$$\sum_{k=1}^{\infty} a_{0,\infty}\omega^k \left(\frac{(1-2p)^{2k}}{2\log 2} - 1\right) = a_{0,\infty}\left(\frac{1}{2\log 2}\frac{\omega(1-2p)^2}{1-\omega(1-2p)^2} - \frac{\omega}{1-\omega}\right).$$

The resulting entropy rate approximation, which we believe is actually an upper bound, is

$$H(\mathcal{Y}) \approx D - \frac{p^2(1-\epsilon)^2}{p(1-\epsilon)+\epsilon}\left(\frac{1}{2\log 2}\frac{\omega(1-2p)^2}{1-\omega(1-2p)^2} - \frac{\omega}{1-\omega}\right).$$

We evaluated the numerical error in this approximation over the rectangle formed by $\epsilon \in [0,1]$ and $p \in [1/2, 2/3]$, and found its maximum value to be roughly 0.0002.

The results of Sections 4.7.1 and 4.7.2 are shown in Figure 4.7.1, along with the capacity of the binary erasure channel (BEC) and the ternary erasure channel (TEC). While one might expect that the SIR and Markov-1 rate should be upper bounded by the capacity of the BEC, we see that this is definitely not the case. This is because the output alphabet of the BEC has only three symbols, while the output alphabet of the DEC has four symbols. Therefore, the rates of the DEC should be upper bounded by the capacity of the TEC. Surprisingly, the achievable rates of the DEC are quite close to the capacity of the TEC when ϵ is close to one.

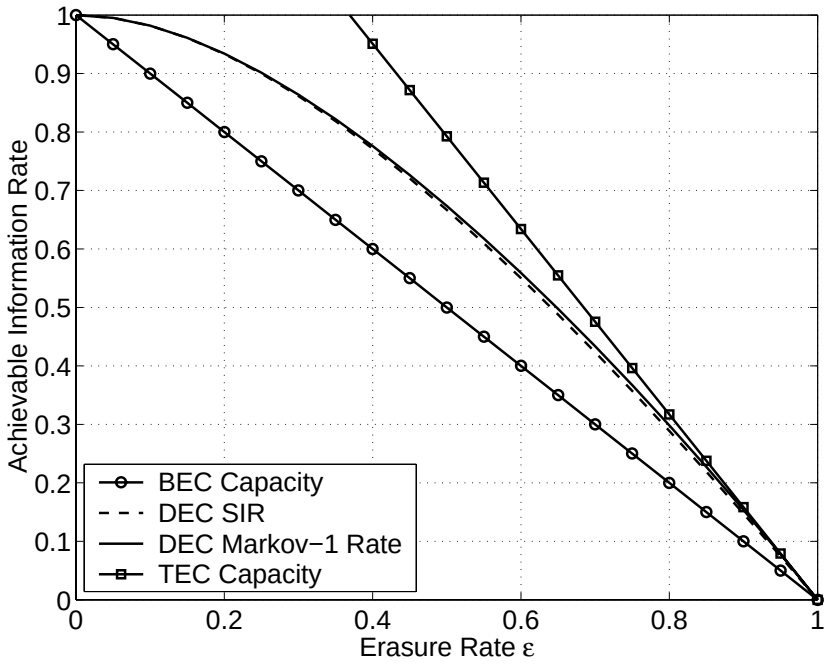


Figure 4.7.1: The SIR and Markov-1 rate of the DEC compared with the capacity of the binary erasure channel (BEC) and the ternary erasure channel (TEC).

4.7.3 Density Evolution for Finite State Channels

Density evolution is a pseudo-analytical method of analyzing LDPC codes that was introduced by Richardson and Urbanke in [25]. It analyzes a decoder by tracking the probabilistic evolution of messages passed around the decoder. In general, it is implemented by quantizing the continuous set of messages to a finite set and then tracking a probability distribution over that set.

Now, we consider a density evolution approach to the forward recursion of the BCJR algorithm. Since the state probability vector acts like a message in the BCJR algorithm, the first step is quantizing these vectors. For a two state channel, the vector is defined by a single parameter, and therefore we can use any scalar quantizer. We note that this idea was applied to two state fading channels by Goldsmith and Varaiya in [15]. For more complicated channels, the natural generalization amounts to using a vector quantizer rather than a scalar quantizer. We note that density evolution on the quantized vectors can be viewed either as an approximate analysis of the true algorithm or an exact analysis of the quantized algorithm.

Let the quantizer, $V(\mathbf{x})$, be a mapping from probability vectors of length N_Q to the index set, $\{1, \ldots, N_V\}$. We abuse notation slightly and define the inverse of this mapping, $V^{-1}(i)$, to be some generalized centroid of the ith quantization cell $\{\mathbf{x} \in \mathfrak{D}(\mathcal{Q}) | V(\mathbf{x}) = i\}$. The forward variable of the quantized algorithm, A_t, is therefore characterized by the update equation

$$A_{t+1} = V\left(\frac{V^{-1}(A_t)\mathbf{M}(Y_t)}{\|V^{-1}(A_t)\mathbf{M}(Y_t)\|_1}\right), \tag{4.7.3}$$

which is simply a quantized version of (4.4.7).

Now, we consider the evolution of $Pr(A_t)$ while the underlying finite state Markov process transitions from state i to state j. In this case, the output, Y_t, is drawn from the distribution $g_{ij}(y)$. This gives rise to the transition matrix, $\mathbf{A}^{(i,j)}$, defined by

$$\left[\mathbf{A}^{(i,j)}\right]_{kl} = Pr(A_{t+1} = l | A_t = k, Q_t = i, Q_{t+1} = j).$$

This matrix can be constructed for channels with a finite output alphabet by evaluating (4.7.3) for all $A_t \in \{1, \ldots, N_V\}$ and $Y_t \in \mathbb{Y}$ and assuming the corresponding probabilities. For channels with continuous output alphabets, one can either integrate over the appropriate regions of $\mathbb{Y}$ or approximate these probabilities by quantizing the output alphabet. The number of non-zero entries in each $\mathbf{A}^{(i,j)}$ matrix is also upper bounded by $N_V |\mathbb{Y}|$, and will therefore be sparse if $N_V \gg |\mathbb{Y}|$.

Next, we analyze the quantized algorithm completely by combining the $\mathbf{A}^{(i,j)}$ matrices with the state transition probabilities, p_{ij}. This allows us to define the $(N_Q N_V) \times (N_Q N_V)$ matrix,

$$\mathbf{A} = \begin{bmatrix} p_{1,1}\mathbf{A}^{(1,1)} & \cdots & p_{1,N_Q}\mathbf{A}^{(1,N_Q)} \\ \vdots & \ddots & \vdots \\ p_{N_Q,1}\mathbf{A}^{(N_Q,1)} & \cdots & p_{N_Q,N_Q}\mathbf{A}^{(N_Q,N_Q)} \end{bmatrix},$$

and point out that

$$[\mathbf{A}]_{(i-1)N_V+k,(j-1)N_V+l} = Pr(A_{t+1} = l, Q_{t+1} = j | A_t = k, Q_t = i).$$

While we expect that the stochastic matrix, $\mathbf{A}$, will generally have a unique stationary distribution, one can also consider the following pair of stationary distributions. Let the lower stationary

distribution, $\underline{\mathbf{v}}$, be defined by $\lim_{n\to\infty} \mathbf{x}\frac{1}{n}\sum_{i=1}^{n} \mathbf{A}^i$ where $\mathbf{x}$ is given by quantizing the distribution (4.4.10). Likewise, let the upper stationary distribution, $\overline{\mathbf{v}}$, be defined by the same limit except that $\mathbf{x}$ is given by quantizing the distribution (4.4.11). These limits are always well defined and can be computed using the eigenvalue decomposition of $\mathbf{A}$. We note that the sparsity of $\mathbf{A}$ may also be exploited to reduce complexity.

Consider the probability vectors, $\mathbf{v}^{(t)}$, defined by $\mathbf{v}^{(t+1)} = \mathbf{v}^{(t)}\mathbf{A}$. Based on the definition of $\mathbf{A}$, these vectors have the implicit definition,

$$\left[\mathbf{v}^{(t)}\right]_{(i-1)N_V+k} = Pr(A_t = k, Q_t = i).$$

Using this, we find that the entropy estimate at time t is given by

$$H(Y_t|\mathbf{Y}_1^{t-1}, W) \approx \sum_{i,j,k} \left[\mathbf{v}^{(t)}\right]_{(i-1)N_V+k} p_{ij} E\left[\log \left\|V^{-1}(k)\mathbf{M}(Y_t)\right\|_1 | Q_t = i, Q_{t+1} = j\right],$$

where W is any random variable which gives rise to the initial distribution, $\mathbf{v}^{(1)}$. This same formula can be used with the stationary distributions $\underline{\mathbf{v}}$ and $\overline{\mathbf{v}}$ to estimate the upper and lower entropy rate bounds.

Since we have a valid probabilistic analysis of the quantized algorithm, we can actually show that any entropy computed in this manner is an upper bound on the same entropy computed via an exact algorithm. For example, suppose we compute the entropy $H(Y_t|\mathbf{Y}_1^{t-1}, W)$ where W is initialized by the vector $\mathbf{v}^{(1)}$. In this case, the entropy computed by the quantized algorithm will always be larger because its state probability estimates are less accurate and therefore increase the entropy.

While the approximation error of this algorithm is quite dependent on the particular quantizer used, we can still make a few general statements. We note that all of these statements are based on the fact that the entropy expression is continuous function on the state probability vector. This means that one would expect the entropy approximation error from using a uniform quantizer to decay like $O\left(N_V^{-1/(N_Q-1)}\right)$. When using an optimized vector quantizer, one would expect the error to decay like $O\left(N_V^{-1/d}\right)$, where d is the (possibly fractal) dimension of the true stationary distribution of the joint Markov chain. This means that this type of analysis may actually be less efficient than Monte Carlo methods when $d > 2$.

This method has been applied successfully to the dicode channel with AWGN. In particular, we used a non-linear scalar quantizer based on the uniform quantization of log-likelihood

ratios. This type of quantization is widely used in the density evolution analysis of LDPC codes [25]. For the dicode channel, the Monte Carlo method can easily achieve tolerances of 10^{-3} while the density evolution approach can achieve tolerances around 10^{-6} with some effort. We note that the density evolution results were in complete agreement with the Monte Carlo results from Section 4.6. The practical value of achieving tolerances less than 10^{-3} is questionable, however.

Remark 4.7.1. While this section discusses only the forward recursion of the BCJR algorithm, the same type of analysis may be applied to the backwards recursion and the output stage. This gives a valid probabilistic analysis of a quantized BCJR algorithm that can be used to approximate the log-likelihood density at the output of the BCJR algorithm. In particular, these densities can be used to optimize LDPC codes and compute information rates for the multilevel coding approach proposed in [24].

4.8 Concluding Remarks

This chapter discusses a number of issues related to entropy rates and capacity for finite state channels. All of the results which are not expressly attributed to other authors were developed independently by us. That said, this field is currently the subject of great interest, and many of the same ideas have recently been developed independently by other authors. For example, the simple Monte Carlo method was published in 2001 by three separate groups [1][24][27]. The formulation of the entropy rate as a Lyapunov exponent was also discovered independently and reported in [17]. Finally, the quantized density evolution approach for information rates is quite natural for two state channels was introduced in [15]. The move to vector quantization is a natural generalization and is also used in a slightly different manner in [32] to help estimate the feedback capacity of finite state channels.

4A Formal Channel Definitions

4A.1 Discrete Input Linear Filter Channels with AWGN

The formal definition, $(\mathbb{X}, \mathbb{Y}, \mathbf{F}(\cdot,\cdot))$, of this finite state channel depends solely on ν, $(h_0, h_1, \ldots, h_\nu)$, σ^2, and $\mathbb{X}$. We start by noting that the number of channel states is given by

$N_S = |\mathbb{X}|^\nu$ and defining the output alphabet, in terms of the input alphabet and channel taps, with

$$\mathbb{Y} = \left\{ y \in \mathbb{R} | y = \sum_{i=0}^{\nu} h_i x_i,\ (x_0, \ldots, x_\nu) \in \mathbb{X}^\nu \right\}.$$

Next, we define $N_X = |\mathbb{X}|$ and let ξ be any one to one mapping from $\mathbb{X}$ to the set $\{0, 1, \ldots, N_X - 1\}$. Although, the channel state is clearly defined by the last $\nu - 1$ inputs, we would also like an integer representation of this quantity. Using a base conversion from $\nu - 1$ digits of $\mathbb{X}$ to the integers, we have the integer state, $S_t = \sum_{i=1}^{\nu} (N_X)^{\nu-i} \xi(X_{t-i})$, and the one step update, $S_{t+1} = \lfloor S_t / N_X \rfloor + \xi(X_t)(N_X)^\nu$. Finally, we define $[\mathbf{F}(x,y)]_{ij} = f_{ij}(x,y)$ with

$$f_{ij}(x,y) = \begin{cases} \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-(y-m_{ij})^2}{2\sigma^2}} & \text{if } j = \lfloor i/N_X \rfloor + \xi(x)(N_X)^\nu \\ 0 & \text{otherwise} \end{cases},$$

where $m_{ij} = h_\nu \xi^{-1}(i \mod N_X) + \sum_{l=0}^{\nu-1} h_l \xi^{-1}\left(\left\lfloor j/(N_X)^{\nu-l-1} \right\rfloor \mod N_X \right)$.

4A.2 Dicode Erasure Channel

The formal definition, $(\mathbb{X}, \mathbb{Y}, \mathbf{F}(\cdot,\cdot))$, of this finite state channel depends only on ϵ. The input and output alphabets are defined by $\mathbb{X} = \{0, 1\}$ and $\mathbb{Y} = \{+, 0, -, e\}$, and $N_S = 2$. The conditional transition-observation probabilities are given by $[\mathbf{F}(x,y)]_{ij} = f_{ij}(x,y)$ where $f_{ij}(x,y) = 0$ unless defined by $f_{00}(0,0) = f_{11}(1,0) = f_{01}(1,+) = f_{10}(0,-) = 1 - \epsilon$ or $f_{00}(0,e) = f_{11}(1,e) = f_{01}(1,e) = f_{10}(0,e) = \epsilon$.

4A.3 Finite State Z-Channel

The formal definition, $(\mathbb{X}, \mathbb{Y}, \mathbf{F}(\cdot,\cdot))$, of this finite state channel has $\mathbb{X} = \mathbb{Y} = \{0, 1\}$ and $N_S = 2$. The conditional transition-observation probabilities are given by $[\mathbf{F}(x,y)]_{ij} = f_{ij}(x,y)$ where $f_{ij}(x,y) = 0$ unless defined by $f_{0,0}(0,1) = f_{1,1}(1,0) = p$, $f_{0,0}(0,0) = f_{1,1}(1,1) = 1 - p$, or $f_{0,1}(1,1) = f_{1,0}(0,0) = 1$.

4A.4 The Finite State Z-Channel with Markov-1 Inputs

Consider a finite state Z-channel with a stochastic input sequence. Using the notation above, we define the input process, $(\mathbf{\Theta}, \mathbf{\Phi})$, with

$$\mathbf{\Theta} = \begin{bmatrix} 1-q & q \\ q & 1-q \end{bmatrix}, \ \mathbf{\Phi} = \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix}.$$

This allows us to define the stochastic output sequence with the triple, $(\mathbb{Y}, \mathbf{P}, \mathbf{G}(\cdot))$, where $\mathbb{Y} = \{0, 1\}$,

$$\mathbf{P} = \begin{bmatrix} 1-q & 0 & 0 & q \\ 1-q & 0 & 0 & q \\ q & 0 & 0 & 1-q \\ q & 0 & 0 & 1-q \end{bmatrix}, \ \mathbf{G}(0) = \begin{bmatrix} 1-p & 0 & 0 & 0 \\ 1 & 0 & 0 & p \\ 1-p & 0 & 0 & 0 \\ 1 & 0 & 0 & p \end{bmatrix}, \ \mathbf{G}(1) = \begin{bmatrix} p & 0 & 0 & 1 \\ 0 & 0 & 0 & 1-p \\ p & 0 & 0 & 1 \\ 0 & 0 & 0 & 1-p \end{bmatrix}.$$

In this case, the transition probability matrix $\mathbf{P}$, and therefore underlying Markov chain, is reducible. Therefore, we simplify our description of the process $\{Y_t\}_{t \geq 1}$by removing any state whose stationary probability is zero. Removing states 1 and 2 results in the simplified description

$$\mathbf{P} = \begin{bmatrix} 1-q & q \\ q & 1-q \end{bmatrix}, \ \mathbf{G}(0) = \begin{bmatrix} 1-p & 0 \\ 1 & p \end{bmatrix}, \ \mathbf{G}(1) = \begin{bmatrix} p & 1 \\ 0 & 1-p \end{bmatrix}.$$

Bibliography

[1] D. Arnold and H. Loeliger. On the information rate of binary-input channels with memory. In *Proc. IEEE Int. Conf. Commun.*, pages 2692–2695, Helsinki, Finland, June 2001.

[2] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv. Optimal decoding of linear codes for minimizing symbol error rate. *IEEE Trans. Inform. Theory*, 20(2):284–287, March 1974.

[3] A. R. Barron. The strong ergodic theorem for densities: Generalized Shannon-McMillan-Breiman theorem. *Ann. Probab.*, 13(4):1292–1303, Nov. 1985.

[4] M. Benda. A central limit theorem for contractive stochastic dynamical systems. *J. Appl. Prob.*, 35:200–205, 1998.

[5] J. J. Birch. On information rates of finite-state channels. *Inform. and Control*, 6:372–380, 1963.

[6] D. Blackwell. Entropy of functions of finite-state Markov chains. *Trans. First Prague Conf. on Inform. Theory, Stat. Dec. Fun., Rand. Processes*, pages 13–20, 1957.

[7] D. Blackwell, L. Breiman, and A. J. Thomasian. Proof of shannon's transmission theorem for finite-state indecomposable channels. *Ann. Math. Stats.*, 29:1209–1220, Dec. 1958.

[8] X. Chen. Limit theorems for functionals of ergodic Markov chains with general state space. *Memoirs of the AMS*, 139(664), May 1999.

[9] H. Cohn, O. Nerman, and M. Peligrad. Weak ergodicity and products of random matrices. *J. Theor. Prob.*, 6:389–405, July 1993.

[10] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley, 1991.

[11] A. Dasdan, S. S. Irani, and R. K. Gupta. Efficient algorithms for optimum cycle mean and optimum cost to time ratio problems. In *Proc. 36th Design Automation Conf.*, pages 37–42, June 1999.

[12] P. Diaconis and D. Freedman. Iterated random functions. *SIAM Review*, 41(1):45–76, Jan. 1999.

[13] H. Furstenberg and H. Kesten. Products of random matrices. *Ann. Math. Stats.*, 31:457–469, June 1960.

[14] R. G. Gallager. *Information Theory and Reliable Communication*. Wiley, New York, NY, USA, 1968.

[15] A. J. Goldsmith and P. P. Varaiya. Capacity, mutual information, and coding for finite-state Markov channels. *IEEE Trans. Inform. Theory*, 42(3):868–886, May 1996.

[16] W. Hirt. *Capacity and Information Rates of Discrete-Time Channels with Memory*. PhD thesis, E.T.H., Zurich, Switzerland, 1988.

[17] T. Holliday, A. Goldsmith, and P. Glynn. Entropy and mutual information for markov channels with general inputs. In *Proc. 40th Annual Allerton Conf. on Commun., Control, and Comp.*, Monticello, IL, USA, Oct. 2002.

[18] R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press, New Jersey, USA, 1985.

[19] A. Kavčić. On the capacity of Markov sources over noisy channels. In *Proc. IEEE Global Telecom. Conf.*, pages 2997–3001, San Antonio, Texas, USA, Nov. 2001.

[20] F. Le Gland and L. Mevel. Basic properties of the projective product with application to products of column-allowable nonnegative matrices. *Math. Control Signals Systems*, 13(1):41–62, July 2000.

[21] F. Le Gland and L. Mevel. Exponential forgetting and geometric ergodicity in hidden Markov models. *Math. Control Signals Systems*, 13(1):63–93, July 2000.

[22] S. P. Meyn and R. L. Tweedie. *Markov Chains and Stochastic Stability*. Springer-Verlag, London, 1993.

[23] V. I. Oseledec. A multiplicative ergodic theorem. Lyapunov characteristic numbers for dynamical systems. *Trans. Moscow Math. Soc.*, pages 197–231, 1968.

[24] H. D. Pfister, J. B. Soriaga, and P. H. Siegel. On the achievable information rates of finite state ISI channels. In *Proc. IEEE Global Telecom. Conf.*, pages 2992–2996, San Antonio, Texas, USA, Nov. 2001.

[25] T. J. Richardson and R. L. Urbanke. The capacity of low-density parity check codes under message-passing decoding. *IEEE Trans. Inform. Theory*, 47(2):599–618, Feb. 2001.

[26] S. Shamai, L. H. Ozarow, and A. D. Wyner. Information rates for a discrete-time Gaussian channel with intersymbol interference and stationary inputs. *IEEE Trans. Inform. Theory*, 37(6):1527–1539, Nov. 1991.

[27] V. Sharma and S. K. Singh. Entropy and channel capacity in the regenerative setup with applications to Markov channels. In *Proc. IEEE Int. Symp. Information Theory*, page 283, Washington, DC, USA, June 2001.

[28] J. B. Soriaga, H. D. Pfister, and P. H. Siegel. On the low rate Shannon limit for binary intersymbol interference channels. submitted to *IEEE Trans. Commun.*, Oct. 2003.

[29] Ö. Stenflo. Ergodic theorems for Markov chains represented by iterated function systems. *Bull. Polish Acad. Sci. Math.*, 49(1):27–43, 2001.

[30] R. Urbanke. Iterative coding systems. http://www.calit2.net/events/2001/courses/ics.pdf, Aug. 2001.

[31] P. O. Vontobel and D. M. Arnold. An upper bound on the capacity of channels with memory and constraint input. In *Proc. IEEE Inform. Theory Workshop*, pages 147–149, Cairns, Australia, Sept. 2001.

[32] S. Yang and A. Kavčić. Markov sources achieve the feedback capacity of finite-state machine channels. In *Proc. IEEE Int. Symp. Information Theory*, page 361, Lausanne, Switzerland, June 2002.

Chapter 5

Joint Iterative Decoding of LDPC Codes and Channels with Memory

5.1 Introduction

Sequences of irregular low-density parity-check (LDPC) codes that achieve the capacity of the binary erasure channel (BEC) under iterative decoding were first constructed by Luby, *et al.* in [11]. This was followed by the work of Chung, *et al., which* provided evidence suggesting that sequences of iteratively decoded LDPC codes can also achieve the channel capacity of the binary-input additive white Gaussian noise (AWGN) channel [3]. Since then, density evolution (DE) [13] has been used to optimize irregular LDPC codes for a variety of memoryless channels (e.g., [6]), and the results suggest, for each channel, that sequences of iteratively decoded LDPC codes can indeed achieve the channel capacity. In fact, the discovery of a channel whose capacity cannot be approached by LDPC codes would be more surprising than a proof that iteratively decoded LDPC codes can achieve the capacity of any binary-input symmetric channel (BISC).

The idea of decoding a code transmitted over a channel with memory via iteration was first introduced by Douillard, *et al.* in the context of turbo codes and is known as *turbo equalization* [4]. This approach can also be generalized to LDPC codes by constructing one large graph which represents the constraints of both the channel and the code. This idea is also referred to as *joint iterative decoding*, and was investigated for partial-response channels by

Figure 5.1.1: Block diagram of the system.

Kurkoski, Siegel, and Wolf in [9].

Until recently, it was difficult to compare the performance of turbo equalization with channel capacity because the binary-input capacity of the channel was unknown. Recently, a new method has gained acceptance for estimating the achievable information rates of finite state channels (FSCs) [1][12][1], and a number of authors have begun designing LDPC based coding schemes which approach the achievable information rates of these channels [8][12][21]. As is the case with DE for general BISCs, the evaluation of code thresholds and the optimization of these thresholds is done numerically. For FSCs, the analysis of this system is quite complex because the BCJR algorithm [2] is used to decode the channel.

Since the capacity of a channel with memory is generally not achievable via equiprobable signaling, one can instead aim for the symmetric information rate (SIR) of the channel. The SIR is defined as the maximum information rate achievable via random coding with equiprobable input symbols. Since linear codes use all inputs equiprobably, the SIR is also the maximum rate directly achievable with linear codes. In this chapter, we introduce a class of channels with memory, which we refer to as generalized erasure channels (GECs). For these channels, we show that DE can be done analytically for the joint iterative decoding of irregular LDPC codes and the channel. This allows us to construct sequences of LDPC degree distributions which appear to achieve the SIR using iterative decoding. As an example, we focus on the dicode erasure channel (DEC), which is simply a binary-input channel with a linear response of $1-D$ and erasure noise.

In Section 5.2, we introduce the basic components of our system. This includes the joint iterative decoder, GECs and the DEC, and irregular LDPC codes. In Section 5.3, we derive a single parameter recursion for the DE of the joint iterative decoder which allows us to give necessary and sufficient conditions for decoder convergence. These conditions are also used to construct code sequences which appear to achieve the SIR. In Section 5.4, we discuss extensions

[1] This method was also introduced by Sharma and Singh in [14]. However, it appears that most of the other results in their paper, based on regenerative theory, are actually incorrect. A correct analytical treatment can be found in Section 4.4.4.

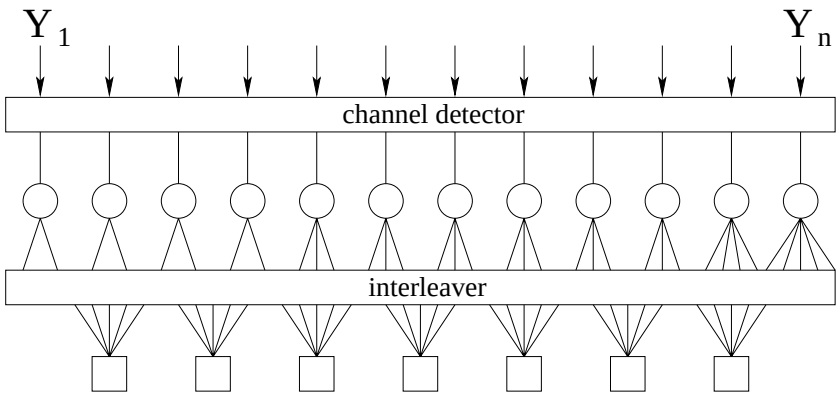


Figure 5.2.1: Gallager-Tanner-Wiberg graph of the joint iterative decoder.

to arbitrary GECs and describe a practical optimization technique based on linear programming. Finally, we offer some concluding remarks in Section 5.5.

5.2 System Model

5.2.1 Description

The system we consider is fairly standard for the joint iterative decoding of an LDPC code and a channel with memory. Equiprobable information bits, $\mathbf{U} = U_1, \ldots, U_k$, are encoded into an LDPC codeword, $\mathbf{X} = X_1, \ldots, X_n$, which is observed through a GEC as the output vector, $\mathbf{Y} = Y_1, \ldots, Y_n$. The decoder consists of an *a posteriori probability* (APP) detector matched to the channel and an LDPC decoder. The first half of decoding iteration i entails running the channel detector on $\mathbf{Y}$ using the *a priori* information from the LDPC code. The second half of decoding iteration i corresponds to executing one LDPC iteration using internal edge messages from the previous iteration and the channel detector output. Figure 5.1.1 shows the block diagram of the system, and Figure 5.2.1 shows the Gallager-Tanner-Wiberg (GTW) graph of the joint iterative decoder.

5.2.2 The Generalized Erasure Channel

Since the messages passed around the GTW graph of the joint decoder are all log-likelihood ratios (LLRs), DE involves tracking the evolution of the distribution of LLR messages

passed around the decoder. Let L be a random variable representing a randomly chosen LLR at the output of the channel decoder. If the distribution of L is supported on the set $\{-\infty, 0, \infty\}$ and $Pr(L = -\infty) = Pr(L = \infty)$, then we refer to it as a *symmetric erasure distribution*. Such distributions are one dimensional, and are completely defined by the erasure probability $Pr(L = 0)$. Our closed form analysis of this system requires that all the densities involved in DE are symmetric erasure distributions.

Definition 5.2.1. A *generalized erasure channel* (GEC) is any channel which satisfies the following condition for i.i.d. equiprobable inputs. The LLR distribution at the output of the channel detector is a symmetric erasure distribution whenever the *a priori* LLR distribution is a symmetric erasure distribution.

This allows DE of the joint iterative decoder to be represented by a single parameter recursion. Let $f(x)$ be a function which maps the erasure probability of the *a priori* LLR distribution, x, to the erasure probability at the output of the detector. The effect of the channel on the DE depends only on $f(x)$, which we refer to as the *erasure transfer function* (ETF) of the GEC. This function is very similar to the mutual information transfer function, $T(I)$, used by the EXIT chart analysis of ten Brink [18]. Since the mutual information of a BEC with erasure probability x is $1 - x$, the mutual information transfer function and $f(x)$ are linked by the identity, $T(I) = 1 - f(1 - I)$.

A remarkable connection between the SIR of a channel, I_s, and its mutual information transfer function was also introduced by ten Brink in [19]. This result requires that $T(I)$ is computed using a symmetric erasure distribution as the *a priori* LLR distribution. A clever application of the chain rule for mutual information shows that

$$\lim_{n\to\infty} \frac{1}{n} I(X_1, \ldots, X_n; Y_1, \ldots, Y_n) = \int_0^1 T(I) dI.$$

Assuming the input process is i.i.d. and equiprobable makes the LHS equal the SIR, and using $T(I) = 1 - f(1 - I)$ allows us to simplify this expression to

$$I_s = \int_0^1 T(I) dI = 1 - \int_0^1 f(x) dx. \tag{5.2.1}$$

Previously, we saw that $f(x)$ completely characterizes the DE properties of a GEC, and now we see that it can also be used to compute the SIR.

5.2.3 The Dicode Erasure Channel

The dicode erasure channel (DEC) is a binary-input channel based on the $1 - D$ linear intersymbol-interference (ISI) dicode channel used in magnetic recording. Essentially, the output of the dicode channel $(+1, 0, -1)$ is erased with probability ϵ and transmitted perfectly with probability $1 - \epsilon$. The precoded DEC is essentially the same, except that the input bits are differentially encoded prior to transmission. This modification simply changes the input labeling of the channel state diagram. The state diagram of the dicode channel is shown with and without precoding in Figure 5.2.2.

The simplicity of the DEC allows the BCJR algorithm for the channel to be analyzed in closed form. The method is similar to the exact analysis of turbo codes on the BEC [20], and the result shows that the DEC is indeed a GEC. Leaving the details to Appendix 5A, we state the ETFs for the DEC with and without precoding. If there is no precoding and the outputs of the DEC are erased with probability ϵ, then the ETF of the channel detector is

$$f(x) = \frac{4\epsilon^2}{(2 - x(1 - \epsilon))^2}. \tag{5.2.2}$$

On the other hand, using a precoder changes this function to

$$f(x) = \frac{4\epsilon^2 x\,(1 - \epsilon(1 - x))}{(1 - \epsilon(1 - 2x))^2}. \tag{5.2.3}$$

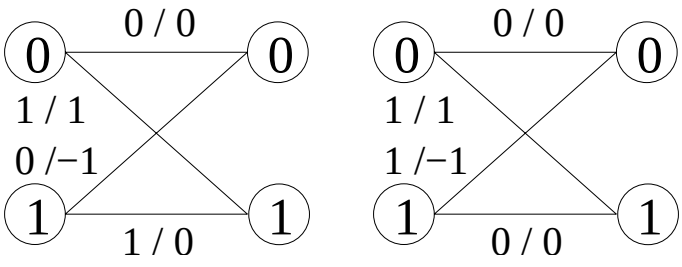
Analyzing only the forward recursion of the BCJR algorithm allows one to compute the SIR of the DEC, and the result, which was computed in Section 4.7.1, is given by

$$I_s(\epsilon) = 1 - \frac{2\epsilon^2}{1 + \epsilon}.$$

It is easy to verify that one can also get this expression for the SIR from either (5.2.2) or (5.2.3) by applying (5.2.1).

5.2.4 Irregular LDPC Codes

Irregular LDPC codes are a generalization of Gallager's LDPC codes [5] that have been shown to perform remarkably well under iterative decoding [13]. They are probably best understood by considering their graphical representation as a bipartite graph, which is shown at bottom of Figure 5.2.1. Iterative decoding is performed by passing messages along the edges of this graph, and the evolution of these messages can be tracked using DE. In general, when we



(a) Without precoding (b) With precoding

Figure 5.2.2: The state diagram of the dicode channel with and without precoding.

speak of an LDPC code we are referring to the ensemble of codes formed by picking a random bipartite graph with the proper degree structure.

For channels with memory, the standard DE assumption of channel symmetry may not hold. Essentially, this means that DE can only be applied to one codeword at a time. In [7], the i.i.d. channel adaptor is introduced as a conceptual device which ensures the symmetry of any channel. If the outer code is a linear code, then this approach is identical to choosing a random coset and treating it as part of the channel. In this work, we employ the i.i.d. channel adaptor approach by assuming that the choice of a random coset vector is embedded in the channel.

The degree distribution of an irregular LDPC code can be viewed either from the edge or node perspective, and the results of this chapter are simplified by using both perspectives. Let $\lambda(x)$ be a polynomial defined by $\lambda(x) = \sum_{\nu \geq 1} \lambda_\nu x^{\nu-1}$, where λ_ν is the fraction of edges attached to a bit node of degree ν. Likewise, let $\rho(x)$ be a polynomial defined by $\rho(x) = \sum_{\nu \geq 1} \rho_\nu x^{\nu-1}$, where ρ_ν is the fraction of edges attached to a check node of degree ν. We refer to $\lambda(x)$ and $\rho(x)$ as the bit and check degree distribution from the edge perspective. Let $L(x)$ be a polynomial defined by $L(x) = \sum_{\nu \geq 1} L_\nu x^\nu$, where L_ν is the fraction of bit nodes with degree ν. Let $R(x)$ be a polynomial defined by $R(x) = \sum_{\nu \geq 1} R_\nu x^\nu$, where R_ν is the fraction of check nodes with degree ν. We refer to $L(x)$ and $R(x)$ as the bit and check degree distributions from the node perspective. The coefficients of all these polynomials represent a fraction of some whole, and that means that $\lambda(1) = \rho(1) = L(1) = R(1) = 1$. Using the definitions of $L(x)$ and $R(x)$, it is also easy to verify that $L(0) = 0$ and $R(0) = 0$. Finally, we note that the possibility of bit and check nodes with degree 1 was intentionally included, so we cannot assume that $\lambda(0) = 0$ or $\rho(0) = 0$.

The average bit degree, a_L, and the average check degree, a_R, are easily computed to be $a_L = \sum_{\nu \geq 1} L_\nu \nu = L'(1)$ and $a_R = \sum_{\nu \geq 1} R_\nu \nu = R'(1)$. One can also switch from the bit to edge perspective by noting that each node of degree ν contributes ν edges to the edge perspective. Counting from the edge perspective and normalizing gives

$$\lambda(x) = \frac{\sum_{\nu \geq 1} L_\nu \nu x^{\nu-1}}{\sum_{\nu \geq 1} L_\nu \nu} = \frac{L'(x)}{a_L} \tag{5.2.4}$$

and

$$\rho(x) = \frac{\sum_{\nu \geq 1} R_\nu \nu x^{\nu-1}}{\sum_{\nu \geq 1} R_\nu \nu} = \frac{R'(x)}{a_R}. \tag{5.2.5}$$

Changing from the edge to bit perspective can be accomplished by integrating both sides of these expressions. This also gives the alternative formulas, $a_L = 1/\int_0^1 \lambda(t)dt$ and $a_R = 1/\int_0^1 \rho(t)dt$. Finally, we note that the rate of an irregular LDPC code is given by $R = 1 - a_L/a_R$.

Iterative decoding of irregular LDPC codes on the BEC, with erasure probability δ, was introduced by Luby *et al.* in [11] and refined in [10]. The recursion for the erasure probability out of the bit nodes is given by

$$x_{i+1} = \delta\lambda\left(1 - \rho(1 - x_i)\right), \tag{5.2.6}$$

while the dual recursion for edges out of the check nodes is given by

$$y_{i+1} = 1 - \rho\left(1 - \delta\lambda(y_i)\right). \tag{5.2.7}$$

Applying linear programming (LP) to these recursions allows one to maximize the code rate over one degree distribution while holding the other one fixed [11]. Although this type of alternating maximization can have convergence problems, it does provide a technique for optimizing degree distribution sequences which works well in practice.

5.3 Analysis of Joint Iterative Decoding

5.3.1 Single Parameter Recursion

Now, we consider a turbo equalization system which performs one channel iteration for each LDPC code iteration. The function, $f(x)$, gives the fraction of erasures produced by the extrinsic output of the channel decoder when the *a priori* erasure rate is x. The update

equations for this system are almost identical to (5.2.6) and (5.2.7). The main difference is that the parameter δ now changes with iteration and is written as δ_i.

Consider the messages passed from the output of the check nodes to the input of the bit nodes, and let x be the fraction which are erased. Since any non-erasure message passed into a bit node gives perfect knowledge of the bit, the messages at the output of the bit node will only be erased only if all of the messages at the input to the bit node are erased. Therefore, the fraction of erased messages passed back from a bit node of degree ν to the check nodes is given by $\delta_i x^{\nu-1}$. Using this, it is easy to verify that the fraction of erased messages passed back from all the bit nodes to the check nodes is given by $\delta_i \sum_{\nu\geq 1} \lambda_\nu x^{\nu-1} = \delta_i \lambda(x)$.

There is also a fundamental difference between the messages passed from the bit nodes to the check nodes and the messages passed from the bit nodes to the channel detector. This difference is due to the fact that a degree ν bit node sends ν messages to the check nodes and only 1 message to the channel detector. The fraction of erased messages passed from a degree ν bit node to the channel detector is given by x^ν. Combining these two observations, shows that the fraction of erased messages passed from all of the bit nodes to the channel detector is $\sum_{\nu\geq 1} L_\nu x^\nu = L(x)$.

The recursion for the erasure probability out of the bit nodes is now given by

$$x_{i+1} = \delta_i \lambda\left(1-\rho(1-x_i)\right), \tag{5.3.1}$$

where $\delta_i = f\left(L\left(1-\rho(1-x_i)\right)\right)$ and $x_0 = f(1)$. Likewise, the dual recursion for edges out of the check nodes is now given by

$$y_{i+1} = 1-\rho\left(1-\delta_i\lambda(y_i)\right),$$

where $\delta_i = f\left(L(y_i)\right)$ and $y_0 = 1-\rho\left(1-f(1)\right)$.

5.3.2 Conditions for Convergence

Using the recursion (5.3.1), we can derive a necessary and sufficient condition for the erasure probability to converge to zero. This condition is typically written as a basic condition which must hold for $x \in (0,1]$ and an auxiliary stability condition which simplifies the analysis at $x = 0$. The basic condition implies there are no fixed points in the iteration for $x \in (0,1]$ and is given by

$$f\left(L\left(1-\rho(1-x)\right)\right)\lambda\left(1-\rho(1-x)\right) < x. \tag{5.3.2}$$

Verifying this condition numerically for very small x can be difficult, so we require instead that $x = 0$ is a stable fixed point of the recursion. This is equivalent to evaluating the derivative of (5.3.2) at $x = 0$, which gives the stability condition

$$\left(\lambda^2(0)f'(0)a_L + \lambda'(0)f(0)\right)\rho'(1) < 1. \tag{5.3.3}$$

The following facts make it easy to derive (5.3.3) from (5.3.2): $\rho(1) = 1$, $L(0) = 0$, and $L'(x) = a_L\lambda(x)$.

Now, we can use (5.3.2) and (5.3.3) to say something about the code properties required by various channels.

1. If the channel has $f(0) > 0$ and the code has $\lambda(0) > 0$, then $f(0)\lambda(0) > 0$ and (5.3.2) cannot hold near zero. This means that $\lambda(0) = 0$ is required for the satisfaction of (5.3.2), and this implies that the code cannot have degree 1 bit nodes (i.e., bits with very little code protection). In this case, the stability condition simplifies to $\lambda_2 f(0)\rho'(1) \leq 1$.

2. If the channel has $f(0) = 0$, then degree 1 bit nodes do not cause this problem. In this case, the stability condition simplifies to $\lambda_1^2 f'(0)a_L\rho'(1) < 1$.

3. If the channel has $f(1) = 1$ and the code has $\rho(0) = 0$, then iteration cannot proceed beyond the fixed point at $x = 1$. It follows that the code must have $\rho(0) > 0$ to get the iteration started. In this case, the required degree 1 check nodes essentially act very much like pilot bits.

The next step is mapping (5.3.2) into an equivalent condition which is easier to manipulate. Consider the condition

$$f\left(L\left(1-\rho(1-q(x))\right)\right)\lambda\left(1-\rho(1-q(x))\right) < q(x),$$

for any $q(x)$ which is a one-to-one mapping from the interval $(0, 1]$ to the interval $(0, 1]$. This new condition is equivalent to the original condition (5.3.2). Choosing $q(x) = 1 - \rho^{-1}(1 - x)$ collapses the basic condition to

$$f\left(L(x)\right)\lambda(x) < q(x) \tag{5.3.4}$$

because $q^{-1}(x) = 1 - \rho(1 - x)$. Using (5.2.4), we can substitute for $\lambda(x)$ to get

$$f\left(L(x)\right)L'(x) < a_L q(x). \tag{5.3.5}$$

Integrating both sides of this inequality from 0 to x gives

$$F\left(L(x)\right) < a_L Q(x),$$

where $F(x) = \int_0^x f(t)dt$ and $Q(x) = \int_0^x q(t)dt$. We note that the function $F(x)$ is non-decreasing for $x \geq 0$ because the function $f(x)$ is non-negative for $x \geq 0$. This means that $F(x)$ is invertible for $x \geq 0$ and we can solve for $L(x)$ by writing

$$L(x) < F^{-1}\left(a_L Q(x)\right). \tag{5.3.6}$$

It appears that we now have a closed form condition involving only $L(x)$ which can be used to analyze the system. Unfortunately, this condition (5.3.6) does not imply (5.3.2) because the sequence of transformations is not reversible. Integrating both sides of an inequality preserves the inequality, but working backwards requires that we take the derivative of both sides. This does not, in general, preserve the inequality.

One way to work around this problem is to require that each step of the above derivation holds with equality. If we assume that (5.3.6) holds with equality, then we can takes its derivative and solve for $\lambda(x)$ to get

$$\lambda(x) = \frac{q(x)}{f\left(F^{-1}\left(a_L Q(x)\right)\right)}. \tag{5.3.7}$$

It is easy to verify this step using the facts that $\frac{d}{dx}F^{-1}(x) = 1/f\left(F^{-1}(x)\right)$, $Q'(x) = q(x)$, and $L'(x) = a_L\lambda(x)$. Finally, we note that (5.3.7) implies that the basic condition (5.3.2) holds with equality as well.

The following theorem relates these inequalities to the gap between the SIR and the code rate.

Theorem 5.3.1. *Consider any LDPC code ensemble, defined by the degree distributions $\lambda(x)$ and $\rho(x)$, which satisfies (5.3.2) for some GEC with ETF $f(x)$. The gap, Δ, between the rate of the LDPC code and the SIR of the channel, I_s, is given by*

$$\Delta = I_s - R = \int_0^1 g(x)dx,$$

where $g(x) = a_L q(x) - f\left(L(x)\right)L'(x)$ is the non-negative gap between the LHS and the RHS of (5.3.5).

Proof. Evaluating the integral gives

$$\int_0^1 g(x)dx = a_L \left[Q(1) - Q(0)\right] - \left[F\left(L(1)\right) - F\left(L(0)\right)\right],$$

where $Q(0) = 0$, $F(L(0)) = F(0) = 0$, and $F(L(1)) = F(1) = 1 - I_s$. We can compute $Q(1)$ using the geometric fact that

$$\int_0^1 \rho(x)dx + \int_0^1 \rho^{-1}(x)dx = 1,$$

and this gives

$$Q(1) = \int_0^1 \left(1 - \rho^{-1}(1-x)\right) dx = \int_0^1 \left(1 - \rho^{-1}(x)\right) dx = \int_0^1 \rho(x)dx = \frac{1}{a_R}.$$

Putting these together with the fact that $R = 1 - a_L/a_R$ gives

$$\int_0^1 g(x)dx = \frac{a_L}{a_R} - (1 - I_s) = I_s - R.$$

□

5.3.3 Achieving the Symmetric Information Rate

Now, we consider sequences of irregular LDPC code ensembles which can be used to communicate reliably at rates arbitrarily close to the SIR. The code sequence is defined by the sequence of degree distributions $\left\{\rho^{(k)}(x), \lambda^{(k)}(x)\right\}_{k\geq 0}$ and its associated rate sequence $\{R_k\}_{k\geq 0}$ is given by $R_k = 1 - a_L^{(k)}/a_R^{(k)}$ via the results of Section 5.2.4. The main difficulty that we will encounter while using algebraic methods to define code sequences is that the implied degree distributions may not be non-negative and generally have infinite support. We say that a degree distribution is (i) *admissible* if its power series expansion about $x = 0$ has only non-negative coefficients and (ii) *realizable* if it is a polynomial (i.e., finite degree) whose coefficients sum to one. We say that a sequence of degree distributions is *SIR achieving* if, for any $\epsilon > 0$, there exists an k_0 such that, for all $k > k_0$, the kth degree distribution is (i) realizable, (ii) satisfies (5.3.2), and (iii) has rate $R_k > I_s - \epsilon$.

The following corollary of Theorem 5.3.1 provides a necessary and sufficient condition for an SIR achieving sequence of degree distributions.

Corollary 5.3.2. *Consider any sequence of LDPC code ensembles, defined by the sequence* $\left\{\rho^{(k)}(x), \lambda^{(k)}(x)\right\}_{k\geq 0}$ *of realizable degree distributions, which satisfy (5.3.2) for some GEC*

with ETF $f(x)$. This sequence of codes is SIR achieving if and only if the associated sequence of rate gap functions, defined by

$$g^{(k)}(x) = a_L^{(k)} \left[q^{(k)}(x) - f\left(L^{(k)}(x)\right) \lambda^{(k)}(x) \right],$$

converges to zero almost everywhere on $[0,1]$.

Proof. The definition of SIR achieving requires that the associated sequence of rate gaps, defined by $\Delta_k = \int_0^1 g^{(k)}(x)dx$, approaches zero. Since $g^{(k)}(x) > 0$ on $(0,1]$ by assumption, this requires that $\lim_{k\to\infty} g^{(k)}(x) = 0$ almost everywhere on $[0,1]$. □

Remark 5.3.3. For the BEC, Shokrollahi [16] showed that all sequences of capacity achieving codes obey a flatness condition which says that the sequence of non-negative gap functions implied by the basic condition, defined by

$$f\left(L^{(k)}\left(1-\rho^{(k)}(1-x)\right)\right)\lambda^{(k)}\left(1-\rho^{(k)}(1-x)\right) - x,$$

converges (along with all of its derivatives) to zero uniformly on $[0,1]$. We believe this can probably be extended to GECs under the assumption that the power series expansion of $f(x)$ about $x = 0$ converges uniformly on $[0,1]$.

In general, we construct SIR achieving sequences by starting with a sequence of realizable check degree distributions $\left\{\rho^{(k)}(x)\right\}_{k\geq 0}$ and then using a slight variation of (5.3.7) to define a sequence of bit degree distributions $\left\{\widetilde{\lambda}^{(k)}(x)\right\}_{k\geq 0}$. If each bit degree distribution in this sequence is admissible with $\widetilde{\lambda}^{(k)}(1) > 1$, then we can form the sequence of realizable bit degree distributions $\left\{\lambda^{(k)}(x)\right\}_{k\geq 0}$ by truncating the power series of each $\widetilde{\lambda}^{(k)}(x)$ so that $\lambda^{(k)}(1) = 1$. Specifically, we generalize the notation of Section 5.2.4 and let $\lambda_i^{(k)} = \widetilde{\lambda}_i^{(k)}$ for $1 \leq i < N_k$, where N_k is the smallest integer such that $\sum_{i=1}^{N_k} \widetilde{\lambda}_i^{(k)} \geq 1$. The last term $\lambda_{N_k}^{(k)}$ is then chosen so that $\lambda^{(k)}(1) = 1$.

One problem with this method, which does not occur for the BEC [15], is that the truncation may cause the basic condition (5.3.2) to fail. To overcome this problem, we require the codes in sequence to satisfy the slightly stronger condition that

$$(1+\alpha_k) f\left(\widetilde{L}^{(k)}(x)\right)\widetilde{\lambda}^{(k)}(x) = q^{(k)}(x), \tag{5.3.8}$$

where $\widetilde{L}^{(k)}(x) = \int_0^x \widetilde{\lambda}^{(k)}(t)dt / \int_0^1 \widetilde{\lambda}^{(k)}(t)dt$ and $q^{(k)}(x) = 1 - \left(\rho^{(k)}\right)^{-1}(1-x)$. This is essentially the same as designing codes for a sequence of degraded channels given by $f^{(k)}(x) =$

$(1+\alpha_k)f(x)$. Adapting (5.3.6) to our system with equality gives

$$\widetilde{L}^{(k)}(x) = F^{-1}\left(\frac{1}{1+\alpha_k}\widetilde{a}_L^{(k)}Q^{(k)}(x)\right), \tag{5.3.9}$$

where $Q^{(k)}(x) = \int_0^x q^{(k)}(t)dt$. Requiring that $\widetilde{L}^{(k)}(1) = 1$ is the same as choosing $\widetilde{a}_L^{(k)}$ so that, without truncation, the code rate equals the SIR of the degraded channel. This gives

$$\widetilde{a}_L^{(k)} = a_R^{(k)}(1+\alpha_k)F(1), \tag{5.3.10}$$

because from (5.2.1) we have $I_s = 1 - F(1)$. Taking the derivative of (5.3.9) and substituting for $\widetilde{a}_L^{(k)}$ with (5.3.10) gives

$$\widetilde{\lambda}^{(k)}(x) = \frac{q^{(k)}(x)}{(1+\alpha_k)f\left(F^{-1}\left(F(1)a_R^{(k)}Q^{(k)}(x)\right)\right)}. \tag{5.3.11}$$

Notice that the channel only enters this equation via the expression $f\left(F^{-1}\left(F(1)x\right)\right)$ and that varying α_k really only changes the truncation point for $\lambda^{(k)}(x)$. Using the facts that $q^{(k)}(x) = 1$ and $Q^{(k)}(1) = 1/a_R^{(k)}$, we also note that

$$\widetilde{\lambda}^{(k)}(1) = \frac{1}{(1+\alpha_k)f(1)}. \tag{5.3.12}$$

This means that the truncation will work, for small enough α_k, as long as $f(1) < 1$.

Theorem 5.3.4. *Let $\left\{\rho^{(k)}(x)\right\}_{k\geq 0}$ be a sequence of realizable check degree distributions and let $\left\{\widetilde{\lambda}^{(k)}(x)\right\}_{k\geq 0}$ be the sequence of bit degree distributions given by (5.3.11). Suppose that (i) the first derivative of $f(x)$ is bounded on $[0,1]$ and $f(1) < 1$, (ii) each $\widetilde{\lambda}^{(k)}(x)$ given by (5.3.11) with $\alpha_k = 0$ is admissible, and (iii) the average check degree $a_R^{(k)}$ and maximum bit degree N_k satisfy $a_R^{(k)}/N_k \to 0$. Under these conditions, there exists a positive sequence $\{\alpha_k\}_{k\geq 0}$ such that the sequence of degree distributions $\left\{\rho^{(k)}(x), \lambda^{(k)}(x)\right\}_{k\geq 0}$ defined above is SIR achieving.*

Proof. We start by examining the effect of the power series truncation. This gives the the sandwich inequality

$$\widetilde{\lambda}^{(k)}(x) - \left(\widetilde{\lambda}^{(k)}(1) - 1\right)x^{N_k - 1} \leq \lambda^{(k)}(x) < \widetilde{\lambda}^{(k)}(x), \tag{5.3.13}$$

where the LHS holds because $\left(\widetilde{\lambda}^{(k)}(1) - 1\right)x^{N_k-1}$ is an upper bound on the truncated terms and the RHS holds by truncation of positive terms. Now, consider the integral representation,

$L^{(k)}(x) = \int_0^x \lambda^{(k)}(t)dt / \int_0^1 \lambda^{(k)}(t)dt$, implied by integrating (5.2.4). Using this and (5.3.13), we get a sandwich inequality for $L^{(k)}(x)$ given by

$$\frac{\int_0^x \left(\widetilde{\lambda}^{(k)}(t) - \left(\widetilde{\lambda}^{(k)}(1) - 1\right) t^{N_k - 1}\right) dt}{\int_0^1 \widetilde{\lambda}^{(k)}(t)dt} < L^{(k)}(x) < \frac{\int_0^x \widetilde{\lambda}^{(k)}(t)dt}{\int_0^1 \left(\widetilde{\lambda}^{(k)}(t) - \left(\widetilde{\lambda}^{(k)}(1) - 1\right) t^{N_k - 1}\right) dt},$$

where $\lambda^{(k)}(x)$ is upper/lower bounded by the RHS/LHS of (5.3.13) respectively. Evaluating the integrals and rearranging terms reduces this to

$$\widetilde{L}^{(k)}(x) - \beta_k x^{N_k} < L^{(k)}(x) < \frac{1}{1 - \beta_k} \widetilde{L}^{(k)}(x), \tag{5.3.14}$$

where $\beta_k = \left(\widetilde{\lambda}^{(k)}(1) - 1\right) \widetilde{a}_L^{(k)} / N_k$. Using (5.3.10) and (5.3.12), we also see that

$$\beta_k = \left(\frac{1}{(1 + \alpha_k) f(1)} - 1\right) \frac{a_R^{(k)}(1 + \alpha_k)(1 - I_s)}{N_k} \leq \frac{a_R^{(k)}(1 - I_s)}{f(1) N_k} = O\left(\frac{a_R^{(k)}}{N_k}\right). \tag{5.3.15}$$

Now, we use these results to analyze the convergence condition for the true channel. First, we define α_k to take the smallest value such that

$$f\left(L^{(k)}(x)\right) \lambda^{(k)}(x) < q^{(k)}(x) \tag{5.3.16}$$

for all $x \in (0, 1]$, where $L^{(k)}(x)$ and $\lambda^{(k)}(x)$ depend implicitly on α_k through the truncation of (5.3.11). Using this value for α_k means that, by definition, $\left\{\rho^{(k)}(x), \lambda^{(k)}(x)\right\}_{k \geq 0}$ is a sequence of realization degree distributions which satisfies the basic condition. Next, we derive an upper bound on the value of α_k chosen. Using (5.3.13) and (5.3.14), it is easy to verify that the condition,

$$f\left(\frac{1}{1 - \beta_k} \widetilde{L}^{(k)}(x)\right) \widetilde{\lambda}^{(k)}(x) \leq q^{(k)}(x), \tag{5.3.17}$$

is more stringent than (5.3.16) and therefore implies (5.3.16). Using (5.3.8) to substitute for $q^{(k)}(x)$, we can then solve for the smallest α_k that implies (5.3.17). The result is the upper bound,

$$\alpha_k \leq \max_{0 \leq x \leq 1} f\left(\frac{1}{1 - \beta_k} \widetilde{L}^{(k)}(x)\right) / f\left(\widetilde{L}^{(k)}(x)\right) - 1, \tag{5.3.18}$$

because any α_k which satisfies this condition must imply (5.3.17) and (5.3.16).

Finally, we show that the sequence of rate gaps $\Delta_k = I_s - R_k$ converges to zero. Using the LHS of (5.3.13) and the integral form of $a_L^{(k)}$, we can write

$$\frac{1}{a_L^{(k)}} \geq \int_0^1 \left(\widetilde{\lambda}^{(k)}(x) - \left(\widetilde{\lambda}^{(k)}(1) - 1\right) x^{N_k - 1}\right) dx = \frac{1}{\widetilde{a}_L^{(k)}} - \frac{\widetilde{\lambda}^{(k)}(1) - 1}{N_k} = (1 - \beta_k)\frac{1}{\widetilde{a}_L^{(k)}}.$$

Using this and (5.3.10), we can lower bound the code rate with

$$R_k \geq 1 - \frac{\widetilde{a}_L^{(k)}}{a_R^{(k)}(1 - \beta_k)} = 1 - \frac{(1 + \alpha_k)(1 - I_s)}{1 - \beta_k}.$$

This means that the rate gap is upper bounded by

$$\Delta_k \leq \frac{I_s(1 - \beta_k)}{1 - \beta_k} - \frac{(1 - \beta_k) - (1 + \alpha_k)(1 - I_s)}{1 - \beta_k} = \frac{(\alpha_k + \beta_k)(1 - I_s)}{1 - \beta_k}.$$

Combining the assumption that $a_R^{(k)}/N_k \to 0$ with (5.3.15) shows that $\beta_k \to 0$. Since the first derivative of $f(x)$ is bounded, we can also combine $\beta_k \to 0$ with (5.3.18) to show that $\alpha_k \to 0$. In fact, examining (5.3.18) more closely shows that $\alpha_k = O(\beta_k)$ which means the rate gap is also $\Delta_k = O(\beta_k) = O\left(a_R^{(k)}/N_k\right)$. This completes the proof. □

5.3.4 Degree Sequences with Regular Check Distributions

In this section, we limit our scope somewhat by choosing check distributions with a single non-zero coefficient. This type of check distribution is called regular, and is defined by $\rho^{(k)}(x) = x^{k-1}$. This implies that $q^{(k)}(x) = 1 - (1 - x)^{1/(k-1)}$ and $Q^{(k)}(x) = (k - 1)(1 - x)^{k/(k-1)}/k + x$, and we use these to rewrite (5.3.11) as

$$\widetilde{\lambda}^{(k)}(x) = \frac{1 - (1 - x)^{1/(k-1)}}{(1 + \alpha_k) f\left(F^{-1}\left(F(1)\frac{(k-1)(1-x)^{k/(k-1)} + kx}{k}\right)\right)}.$$

As shown in [15], the non-negative power series expansion of $1 - (1 - x)^{1/(k-1)}$ is given by

$$1 - (1 - x)^{1/(k-1)} = \sum_{i=1}^{\infty} \binom{1/(k-1)}{i} (-1)^{i+1} x^i.$$

This means that question of whether or not $\widetilde{\lambda}^{(k)}(x)$ has a non-negative power series expansion is very much linked to the power series expansion of $h(x) \triangleq 1/f\left(F^{-1}(F(1)x)\right)$. While answering this question is difficult, we can still make general comments. For one, this method appears to be doomed if the coefficients of $h(x)$ do not decay to zero. Since the location of the smallest

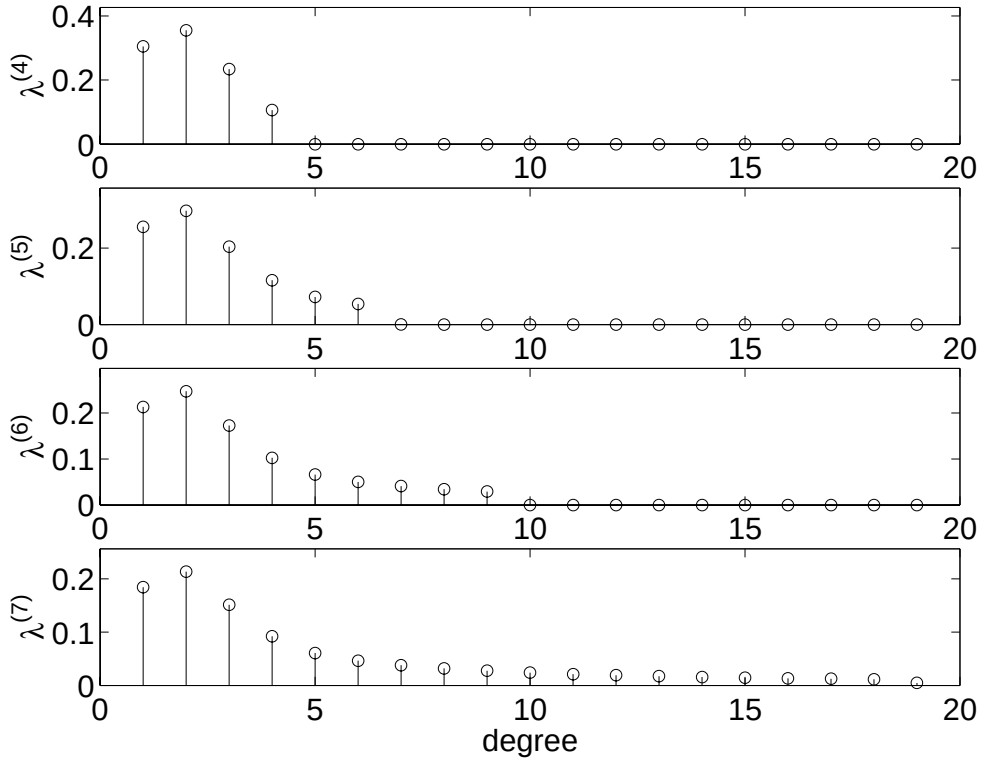


Figure 5.3.1: This shows the bit degree distributions $\lambda^{(k)}(x)$ resulting from constructing check regular codes for the precoded DEC with $\epsilon = 1/2$. The vertical axis of each subplot is scaled differently to highlight their similarity.

zero in the denominator of $h(x)$ determines the exponential growth rate of the coefficients, it makes sense that the modulus of smallest zero in the denominator should be larger than one. This implies the coefficients will decay exponentially.

For the DEC, with and without precoding, we can derive $h(x)$ in closed form. Without precoding, this expression is given by

$$h(x) = \frac{\left(x + \epsilon^2\left(2 - 3x + 2\epsilon^3(1+x)\right)\right)^2}{\epsilon^2(1+\epsilon)^2\left(2\epsilon^2 + x(1-\epsilon)\right)^2}.$$

The addition of a precoder changes this to

$$h(x) = \frac{(1 + 2\epsilon^2\sqrt{x(x+2-2\epsilon)} - \epsilon^2(1-2x))^2}{4\epsilon^3(x + \sqrt{x(x+2-2\epsilon)})(1 - \epsilon^2(1-x) + \epsilon^2\sqrt{x(x+2-2\epsilon)})}.$$

Using the criterion that the smallest zero of the denominator should have modulus greater than one, we have determined that the DEC without precoding requires that $0.5 < \epsilon < 1$ and the DEC

k	$a_R^{(k)}$	$a_L^{(k)}$	R_k	Δ_k	N_k	α_k	β_k	$\overline{\lambda}_1^{(k)}$	$\lambda_1^{(k)}$
4	4	1.595	0.6011	0.0655	4	0.16	0.4844	0.3127	0.3048
5	5	1.903	0.6193	0.0473	7	0.069	0.3937	0.2563	0.2562
6	6	2.102	0.6496	0.0170	9	0.048	0.2671	0.2181	0.2134
7	7	2.411	0.6555	0.0111	19	0.025	0.1681	0.1991	0.1843
8	8	2.718	0.6602	0.0064	33	0.014	0.1063	0.1615	0.1614
9	9	3.030	0.6632	0.0034	56	0.0075	0.0666	0.1436	0.1433
10	10	3.349	0.6651	0.0016	101	0.0042	0.0411	0.1287	0.1286
11	11	3.677	0.6657	0.0009	184	0.0023	0.0249	0.1166	0.1165

Table 5.1: Code construction results for the precoded DEC with $\epsilon = 1/2$.

with precoding requires that $0 < \epsilon \lesssim 0.6309$. While this does not prove that sequences of check regular codes cannot achieve the SIR of arbitrary GECs, it definitely hints at this possibility.

For the precoded DEC with $\epsilon = 0.5$, we constructed the check regular code sequence for $k = 4, \ldots, 11$. While we were unable to prove that the coefficients of each power series expansion are non-negative, we did verify this numerically for the first 200 coefficients. The results of this experiment are shown in Table 5.1 and Figure 5.3.1. Since the choice of α_k in our construction guarantees that each code satisfies the convergence condition for the channel, all of these rates and rate gaps are valid. It is also worth noting that the degree sequences shown in Figure 5.3.1 are all very similar once you account for truncation and scaling. The table also gives the fraction of edges connected to degree 1 nodes, $\lambda_1^{(k)}$, and the maximum value allowed by the stability condition, $\overline{\lambda}_1^{(k)}$. Although, we cannot prove that this sequence of codes satisfies the conditions of Theorem 5.3.4, we can still compare the results to the predictions of the theorem. We find that the constants follow the predictions of Theorem 5.3.4 quite well. For one, N_k appears to be growing exponentially with k and Δ_k seems to be decaying exponentially with k. This type of behavior is well-known for check regular codes on the BEC [16].

For the DEC with $\epsilon = 0.85$ and no precoding, we also constructed the check regular code sequence. In this case, N_k grows so rapidly that we could only construct the codes with $k = 3, 4, 5$. This time, we verified numerically that the first 800 coefficients of each power series are non-negative. The results are shown in Table 5.2, and again the sequences follow the predictions of Theorem 5.3.4 quite well. This table also shows the fraction of edges connected

k	$a_R^{(k)}$	$a_L^{(k)}$	R_k	Δ_k	N_k	α_k	β_k	$\overline{\lambda}_2^{(k)}$	$\lambda_2^{(k)}$
3	3	2.370	0.2101	0.0088	14	0.00053	0.0308	0.6920	0.6916
4	4	3.129	0.2177	0.0011	107	0.00018	0.0054	0.4613	0.4612
5	5	3.906	0.2187	0.0002	757	0.00003	0.0009	0.3460	0.3460

Table 5.2: Code construction results for the DEC with $\epsilon = 0.85$ and no precoding.

to degree 2 bit nodes, $\lambda_2^{(k)}$, and the maximum value allowed by the stability condition, $\overline{\lambda}_2^{(k)}$.

Finally, we should note that values of ϵ and k chosen for these experiments carefully avoided power series expansions with non-negative coefficients. Though not shown here, we have also considered code sequences with arbitrary ETFs such as $f(x) = ax + b$. In all these cases, we have not found any situation where the power series expansion suddenly has a negative term after a long initial sequence of positive terms. In general, the expansions we have found with negative terms expose themselves within the first five terms. Picking larger values of k also seems to help and any expansion usually has negative terms for small enough k.

5.4 Results for Arbitrary GECs

5.4.1 The Existence of Arbitrary GECs

From what we have discussed so far, it is not clear that the set of GECs contains anything more than the DEC with and without precoding. Nothing in our analysis, however, prevents us from considering the much larger family of GECs implied by any non-decreasing $f(x)$ which maps the interval $[0, 1]$ into itself. Moreover, we believe that it is possible to construct, albeit somewhat artificially, a binary-input GEC for any such $f(x)$. This would means that, in some sense, there is a GEC for every well-defined ETF. Similar ideas may also be useful in the context of EXIT charts analysis.

The ETF of a GEC is defined as mapping from the *a priori* erasure probability of the channel decoder to the erasure probability of the extrinsic output. If the *a priori* messages from the code to the channel decoder are divided randomly into two groups of equal size, then erasure probability in the two groups will be the same. Now, suppose that these groups of bits are sent through different GECs. In this case, the extrinsic messages from the first channel will have

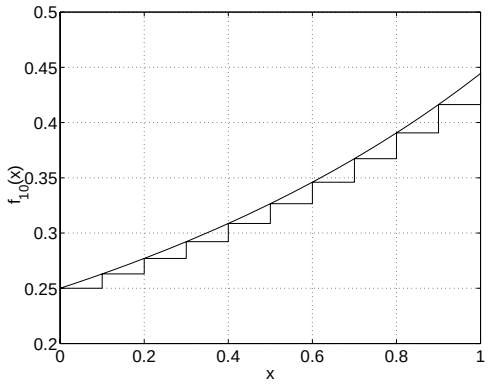


Figure 5.4.1: The results of approximating the non-precoded DEC with $\epsilon = 0.5$ and $n = 10$.

erasure probability $f_1(x)$ and the extrinsic messages from the second channel will have erasure probability $f_2(x)$. Since the two groups were chosen at random, the average erasure probability of all the extrinsic messages passed back to the code will be $\left(f_1(x) + f_2(x)\right)/2$. This idea of linearly combining channels was first introduced in the context of EXIT charts and doping [17]. It also extends naturally to an arbitrary weighted combinations of GECs.

Now, consider the performance of a rate-1/2 systematic linear code when the systematic bits are erased with probability x and the non-systematic bits are erased with probability y. We assume that the code is chosen randomly and that the block length is arbitrarily large. Assuming that the coding theorem holds for this special case, a maximum likelihood decoder should be able to recover all of the systematic bits as long as the average bit erasure rate, $(x+y)/2$, is less than $1/2$. If the average erasure rate is larger than $1/2$, then, with probability 1, there will not be enough information to reconstruct all of the systematic bits. Therefore, we conjecture that the extrinsic erasure rate, at the output of an APP decoder for this code, will be given by

$$u(x,y) = \begin{cases} 0 & 0 \leq x+y < 1 \\ 1 & 1 \leq x+y \leq 2 \end{cases} .$$

This ensemble of codes can be treated as a GEC whose inputs are the systematic bits and whose outputs are the parity bits. In this case, the ETF for the GEC with parameter y and *a priori* erasure rate x is given by $u(x,y)$. It is not too hard to see that we can approximate any non-decreasing $f(x)$, which maps the interval $[0,1]$ into itself, in this fashion. If we define the

sequence of approximations by

$$f_n(x) = \sum_{i=0}^{n} w_n\left(\frac{i}{n}\right) u\left(x, \frac{i}{n}\right)$$

with

$$w_n\left(\frac{i}{n}\right) = f\left(\frac{n-i}{n}\right) - f\left(\frac{n-i-1}{n}\right),$$

then $f_n(x)$ is essentially the nth order lower Riemann approximation of $f(x)$. An example of the approximating function is shown in Figure 5.4.1. If $f(x)$ is differentiable on $[0, 1]$, these approximations will converge, as n goes infinity, to

$$f(x) = \int_0^1 w(y)u(x, y)dy$$

with

$$w(y) = \delta(1-y)f(0) + f'(1-y),$$

where $\delta(y)$ is the Dirac delta function.

5.4.2 Numerical Optimization via Linear Programming

In this section, we attempt to leverage our closed form condition, (5.3.6), into a practical technique for optimizing degree distributions for arbitrary GECs. The result is an algorithm similar to the alternating LP optimization presented in [11]. The optimization algorithm works by choosing a regularly spaced grid of points and using LP to maximize the code rate while satisfying the constraint at each grid point. We use a grid based on the erasure probability passed into a check node, make use of the fact that (5.3.2) need only be satisfied for $x \in (0, f(1)]$. Using this, it is sufficient to use the grid $\mathcal{X} = \{0, s, 2s, \ldots, \lfloor f(1)/s \rfloor s, f(1)\}$, and we note that $s = 0.02$ seems to work well in practice. The arguments passed to the algorithm consist of the set of active bit degrees $\mathcal{L}$, the set of active check degrees $\mathcal{R}$, and the initial check degree distribution $\rho(x)$. The algorithm proceeds by alternately optimizing $\lambda(x)$, for fixed $\rho(x)$, and $\rho(x)$, for fixed $\lambda(x)$.

There are three things which make this algorithm tricky to use in practice, however. The first is that the parameter a_L in (5.3.7) must be guessed correctly to make things work. The second is that the function $q(x)$ generally gets very steep as x approaches 1, so expressions must

be evaluated carefully to avoid numerical problems. The third is that LP is used to maximize the rate under a constraint similar to (5.3.7), but we know this constraint does not imply that (5.3.2) holds as well. The algorithm is stabilized by the fact that the $\rho(x)$ optimization always produces a valid code which satisfies (5.3.2). Furthermore, if the algorithm converges, then (5.3.7) will be satisfied with near equality which implies that (5.3.2) will also be satisfied with near equality. All of these add up to an algorithm that can work quite well, albeit with some tweaking of the initial conditions. Lastly, the convergence is rarely monotonic and the algorithm may initially wander through terrible codes (i.e. negative rate) before finding a very good code.

Consider optimizing $\rho(x)$ for some fixed $\lambda(x)$. The goal of maximizing the code rate for fixed $\lambda(x)$ is equivalent to minimizing the linear objective function,

$$1/a_R = \sum_{\nu \in \mathcal{R}} \rho_v \frac{1}{\nu}, \tag{5.4.1}$$

because the code rate is $1 - a_L/a_R$. The LP constraints can be derived by starting with (5.3.4) and applying $q^{-1}(x)$ to both sides to get

$$q^{-1}\left(f\left(L(x)\right)\lambda(x)\right) < x.$$

Since $q^{-1}(x) = 1 - \rho(1-x)$, this inequality can be rewritten as

$$\sum_{\nu \in \mathcal{R}} \rho_\nu \left(1 - \left(1 - f\left(L(x)\right)\lambda(x)\right)^{\nu-1}\right) < x,$$

which is linear in the ρ_ν's. In practice, we have found that the numerical robustness is improved by letting $\phi(x)$ equal the $\rho(x)$ from the previous iteration, and requiring that

$$\sum_{\nu \in \mathcal{R}} \rho_\nu \left(1 - \left(1 - f\left(L\left(1-\phi(1-x)\right)\right)\lambda\left(1-\phi(1-x)\right)\right)^{\nu-1}\right) < 1 - \phi(1-x) \tag{5.4.2}$$

be satisfied for all $x \in \mathcal{X}$. The stability condition, (5.3.3), can also be handled via the extra inequality

$$\sum_{\nu \in \mathcal{R}} \rho_\nu(\nu - 1) < \begin{cases} 1/\left(\lambda_1^2 f'(0) a_L\right) & \text{if } f(0) = 0 \\ 1/\left(\lambda_2 f(0)\right) & \text{if } f(0) > 0 \end{cases}.$$

Now, we consider optimizing $\lambda(x)$ for fixed $\rho(x)$. Our goal of maximizing the code rate for fixed $\rho(x)$ is equivalent to maximizing the linear objective function,

$$1/a_L = \sum_{\nu \in \mathcal{L}} \lambda_v \frac{1}{\nu}, \tag{5.4.3}$$

because the code rate is $1 - a_L/a_R$. The LP constraints can be derived by starting with (5.3.7), substituting $1 - \rho(1-x)$ for x, and transforming it into an inequality to get

$$\lambda\left(1-\rho(1-x)\right) < \frac{x}{f\left(F^{-1}\left(a_L Q\left(1-\rho(1-x)\right)\right)\right)}.$$

One subtlety is choosing the a_L so the algorithm performs well. We have found that choosing $a_L = F(1)/a_R$, which implicitly assumes that we will achieve the SIR, does the trick. Adding the relaxation constant, c, and rewriting this gives the linear inequality,

$$\sum_{\nu\in\mathcal{L}} \lambda_\nu \left(1-\rho(1-x)\right)^{\nu-1} < \frac{cx}{f\left(F^{-1}\left(F(1)a_R Q\left(1-\rho(1-x)\right)\right)\right)}, \tag{5.4.4}$$

which must be satisfied for all $x \in \mathcal{X}$. It is worth pointing out the similarity between (5.4.4) and (5.3.11) with c playing the role of α_k. The stability condition, (5.3.3), depends on the channel and is given by

$$\begin{aligned} \lambda_1 &< c\sqrt{\tfrac{1}{f'(0)a_L\rho'(1)}} && \text{if } f(0)=0 \\ \lambda_2 &< \tfrac{c}{f(0)\rho'(1)} && \text{if } f(0)>0 \end{aligned}$$

The relaxation constant is used to improve convergence and we typically use $c = 1-(0.1)^{1+i/4}$ for the ith $\lambda(x)$ optimization.

In our implementation, all of the function evaluations required are done via linear interpolation from sampled function tables. For example, we start by sampling $f(x)$ on a very fine grid. Next, we let $F(x)$ be the integral of linear interpolated $f(x)$, which is given by trapezoidal integration of the $f(x)$ sample table. Inverse functions can also be handled by via linear interpolation. To do this, one simply reverses the role of the sampling grid and the sampled function table. Finally, the function $Q\left(1-\rho(1-x)\right)$ can be computed accurately and efficiently using

$$Q\left(1-\rho(1-x)\right) = \sum_{\nu\in\mathcal{R}} \rho_\nu \left(\frac{1-(1-x)^\nu}{\nu} - x(1-x)^{\nu-1}\right). \tag{5.4.5}$$

Remark 5.4.1. This formula for $Q\left(1-\rho(1-x)\right)$ can be derived by noticing that the rectangle extending from $(0,0)$ to $(x,q(x))$ can be divided into two regions by the curve $q(t)$ with $t \in [0,x]$. Using the fact that the area of these two regions must sum to $xq(x)$, we have

$$\int_0^x q(t)dt + \int_0^{q(x)} q^{-1}(t)dt = xq(x).$$

This allows us to compute the integral of $q(x)$ with

$$\begin{aligned} Q(x) &= \int_0^x q(t)dt \\ &= xq(x) - \int_0^{q(x)} q^{-1}(t)dt \\ &= xq(x) - \int_0^{q(x)} (1-\rho(1-t))\,dt \\ &= xq(x) - \sum_{\nu\geq 1} \rho_\nu \left(q(x) - \frac{1-(1-q(x))^\nu}{\nu} \right). \end{aligned}$$

Finally, we evaluate this expression by substituting $1-\rho(1-y) = q^{-1}(y)$ for x to get

$$Q\left(1-\rho(1-y)\right) = \left(1-\rho(1-y)\right) y - \sum_{\nu\geq 1} \rho_\nu \left(y - \frac{1-(1-y)^\nu}{\nu} \right).$$

Expanding $\rho(1-y)$ and simplifying gives (5.4.5).

5.4.3 A Stability Condition for General Channels

In this section, we discuss one implication that this research has on the joint iterative decoding of LDPC codes and general channels with memory. This implication is that the stability condition for general channels may actually be as simple as the stability condition for memoryless channel. Recall that, as long as $f(0) > 0$, the stability condition for GECs is given by $\lambda_2\rho'(1)f(0) < 1$. This condition is identical to the stability condition for the memoryless erasure channel with erasure probability $f(0)$. Let $F_0(x)$ be the LLR density at the extrinsic output of the channel decoder, for a general channel, when perfect *a priori* information is passed to the decoder. As long as $F_0(x)$ does not have perfect information itself (i.e., it is not equal to a delta function at infinity), then the stability condition is given by applying the memoryless channel condition from [13] to $F_0(x)$. This makes sense because, when the joint decoder is near convergence, the LLRs passed as *a priori* information to the channel decoder are nearly error free. A more rigorous analysis of this phenomenon is given in Appendix 5B.3.

5.5 Concluding Remarks

In this chapter, we consider the joint iterative decoding of irregular LDPC codes and channels with memory. We introduce a new class of erasure channels with memory, known as

generalized erasure channels (GECs). For these channels, we derive a single parameter recursion for density evolution of the joint iterative decoder. This allows us to state necessary and sufficient conditions for decoder convergence and to algebraically construct sequences of LDPC degree distributions which appear to approach the symmetric information rate of the channel. This provides the first possibility of proving that the SIR is actually achievable via iterative decoding. In the future, we hope to prove that the two degree sequences constructed in this chapter actually achieve the SIR. The bigger question is whether or not it is possible to construct degree distribution sequences which achieve the SIR for any GEC.

5A Exact Analysis of the BCJR Decoder for the DEC

In this section, we analyze the behavior of a BCJR decoder for a DEC with erasure rate ϵ. This is achieved by first finding the steady state distributions of the forward and backward recursions, and then computing the extrinsic erasure rate of the decoded input stream. Prior knowledge of the input is taken into account by assuming that it is observed independently through a BEC with erasure rate δ. This approach makes it possible to derive closed form expressions for a system which combines a low density parity check (LDPC) code with a BCJR decoder for this channel. Throughout this section, we assume that the channel inputs are chosen i.i.d. $B(1/2)$.

5A.1 The Dicode Erasure Channel without Precoding

Consider the BCJR algorithm operating on a DEC without precoding. In this section, we compute the extrinsic erasure rate of that decoder as an explicit function of the channel erasure rate, ϵ, and the *a priori* erasure rate, δ. This is done by analyzing the forward recursion, the backward recursion, and the output stage separately.

Expanding our decoder to consider *a priori* information is very similar to expanding the alphabet of our channel. Instead of receiving a single output symbol from the set $\mathbb{Y} = \{-, 0, +, e\}$, we receive a pair of output symbols. One is from the set $\mathbb{Y}$ and the other, which represents the *a priori* symbol, is from the set $\mathbb{W} = \{0, 1, e\}$. Since the channel has only two states, it suffices to consider the quantity $\alpha^{(t)} \triangleq \alpha_0^{(t)} = 1 - \alpha_1^{(t)} = Pr(S_t = 0|\mathbf{W}_1^{t-1}, \mathbf{Y}_1^{t-1})$. The real simplification, however, comes from the fact that the distribution of $\alpha^{(t)}$ has finite

support when $X \sim B(1/2)$. Let W_t and Y_t be the *a priori* symbol and channel output received at time t, respectively. Using this, we can write the forward recursion as

$$\boldsymbol{\alpha}^{(t+1)} = \frac{\boldsymbol{\alpha}^{(t)}\mathbf{M}_\alpha(W_t, Y_t)}{\left\|\boldsymbol{\alpha}^{(t)}\mathbf{M}_\alpha(W_t, Y_t)\right\|_1},$$

where $\boldsymbol{\alpha}^{(t)} = \left[\alpha^{(t)} \;\; 1-\alpha^{(t)}\right]$ and $[\mathbf{M}_\alpha(w,y)]_{ij} = Pr(S_{t+1} = j, W_t = w, Y_t = y | S_t = i)$. It is easy to verify that this recursion is identical to the simpler recursion,

$$\alpha^{(t+1)} = \begin{cases} 1/2 & \text{if } Y_t = e \text{ and } W_t = e \\ \alpha^{(t)} & \text{if } Y_t = 0 \text{ and } W_t = e \\ 0 & \text{if } Y_t = + \text{ or } W_t = 1 \\ 1 & \text{if } Y_t = - \text{ or } W_t = 0 \end{cases}.$$

Using the simple recursion, we see that, for all $t \geq \min\{i \geq 1 | Y_i \neq 0 \text{ or } W_i \neq e\}$, $\alpha^{(t)}$ will be confined to the finite set $\{0, 1/2, 1\}$.

The inherent symmetry of the channel actually allows us to consider even a smaller support set. The real difference between the three α values in the support set is whether the state is known perfectly or not. When $\alpha^{(t)} \in \{0, 1\}$, the state is known with absolute confidence, while $\alpha^{(t)} = 1/2$ corresponds to no prior knowledge.

Using a two state Markov chain, we can compute the steady state probabilities of a new Markov chain which characterizes the forward recursion. To do this, we treat the known state condition (i.e., $\alpha^{(t)} \in \{0, 1\}$) as the K_α state and unknown state condition (i.e., $\alpha^{(t)} = 1/2$) as the U_α state. The new Markov chain transitions from the K_α state to the U_α state only if $W = e$ and $Y = e$. Therefore, we have $Pr(K_\alpha \to U_\alpha) = 1 - Pr(K_\alpha \to K_\alpha) = \epsilon\delta$. The new Markov chain also transitions from the U_α state to the U_α state only if $W = e$ and $Y \in \{e, 0\}$. This means that we have $Pr(U_\alpha \to U_\alpha) = 1 - Pr(U_\alpha \to K_\alpha) = \delta(\epsilon + (1-\epsilon)/2)$. The steady state probabilities $Pr(K_\alpha)$ and $Pr(U_\alpha)$ can be found using the eigenvector equation,

$$\left[\begin{array}{cc} Pr(K_\alpha) & Pr(U_\alpha) \end{array}\right] \left[\begin{array}{cc} 1-\epsilon\delta & \epsilon\delta \\ 1-\frac{\delta(1+\epsilon)}{2} & \frac{\delta(1+\epsilon)}{2} \end{array}\right] = \left[\begin{array}{cc} Pr(K_\alpha) & Pr(U_\alpha) \end{array}\right],$$

whose solution is $Pr(U_\alpha) = 1 - Pr(K_\alpha) = \frac{2\epsilon\delta}{2-\delta(1+\epsilon)+2\epsilon\delta}$.

The backward recursion is analyzed in an almost identical manner. In this case, it suffices to consider the quantity $\beta^{(t)} \triangleq \beta_0^{(t)} = 1 - \beta_1^{(t)} = Pr(S_t = 0 | \mathbf{W}_t^n, \mathbf{Y}_t^n)$. Now, we can

write the backward recursion as

$$\boldsymbol{\beta}^{(t+1)} = \frac{\boldsymbol{\beta}^{(t)}\mathbf{M}_{\beta}(W_t, Y_t)}{\left\|\boldsymbol{\beta}^{(t)}\mathbf{M}_{\beta}(W_t, Y_t)\right\|_1},$$

where $\boldsymbol{\beta}^{(t)} = \left[\beta^{(t)} \;\; 1-\beta^{(t)}\right]$ and $\left[\mathbf{M}_{\beta}(w,y)\right]_{ij} = Pr(S_t = j, W_t = w, Y_t = y | S_{t+1} = i)$. Again, we have a simpler recursion which, in this case, is given by

$$\beta^{(t+1)} = \begin{cases} 1/2 & \text{if } Y_t = e \\ \alpha^{(t)} & \text{if } Y_t = 0 \text{ and } W_t = e \\ 0 & \text{if } Y_t = + \text{ or } (Y_t = 0 \text{ and } W_t = 0) \\ 1 & \text{if } Y_t = - \text{ or } (Y_t = 0 \text{ and } W_t = 1) \end{cases}.$$

Using the simple recursion, we see that, for all $t \geq \min\{i \geq 1 | Y_i \neq 0\}$, $\beta^{(t)}$ will be confined to the finite set $\{0, 1/2, 1\}$.

Now, we can use a two state Markov chain to compute the steady state probabilities of a new Markov chain which characterizes the backward recursion. To do this, we treat the known state condition (i.e., $\beta^{(t)} \in \{0,1\}$) as the K_β state and unknown state condition (i.e., $\beta^{(t)} = 1/2$) as the U_β state. The new Markov chain transitions from the K_β state to the U_β state if $Y = e$. Therefore, we have $Pr(K_\beta \rightarrow U_\beta) = 1 - Pr(K_\beta \rightarrow K_\beta) = \epsilon$. The new Markov chain also transitions from the U_β state to the U_β state if: (i) $Y = e$ or (ii) $W = e$ and $Y = 0$. This means that we have $Pr(U_\beta \rightarrow U_\beta) = 1 - Pr(U_\beta \rightarrow K_\beta) = \epsilon + \delta(1-\epsilon)/2$. The steady state probabilities $Pr(K_\alpha)$ and $Pr(U_\alpha)$ can be found using the eigenvector equation,

$$\left[\begin{array}{cc} Pr(K_\beta) & Pr(U_\beta) \end{array}\right] \left[\begin{array}{cc} 1-\epsilon & \epsilon \\ 1-\epsilon-\frac{\delta(1-\epsilon)}{2} & \epsilon+\frac{\delta(1-\epsilon)}{2} \end{array}\right] = \left[\begin{array}{cc} Pr(K_\beta) & Pr(U_\beta) \end{array}\right],$$

whose solution is $Pr(U_\beta) = 1 - Pr(K_\beta) = \frac{2\epsilon}{(1-\epsilon)(2-\delta)+2\epsilon}$.

Now, we consider the output stage of the BCJR algorithm DEC without precoding. At any point in the trellis, there are now four distinct possibilities for forward/backward state knowledge: $K_\alpha K_\beta$, $K_\alpha U_\beta$, $U_\alpha K_\beta$, and $U_\alpha U_\beta$. At the extrinsic output of the decoder, the respective erasure probability conditioned on each possibility is: 0, ϵ, 0, and $(1+\epsilon)/2$. Therefore,

the erasure probability of the extrinsic output of the BCJR is

$$\begin{aligned} P_e &= Pr(U_\beta)\left(\epsilon Pr(K_\alpha) + \frac{1+\epsilon}{2} Pr(U_\alpha)\right) \\ &= \frac{2\epsilon}{(1-\epsilon)(2-\delta)+2\epsilon}\left(\epsilon\frac{2-\delta(1+\epsilon)}{2-\delta(1+\epsilon)+2\epsilon\delta} + \frac{1+\epsilon}{2}\frac{2\epsilon\delta}{2-\delta(1+\epsilon)+2\epsilon\delta}\right) \\ &= \frac{4\epsilon^2}{(2-\delta(1-\epsilon))^2}. \end{aligned}$$

Decoding without *a priori* information is equivalent to choosing $\delta = 1$, and the corresponding expression simplifies to $4\epsilon^2/(1+\epsilon)^2$.

5A.2 The Dicode Erasure Channel with Precoding

Consider the BCJR algorithm operating on a DEC using a $1/(1+D)$ precoder. In this section, we compute the extrinsic erasure rate of that decoder as an explicit function of the channel erasure rate, ϵ, and the *a priori* erasure rate, δ. This is done by analyzing the forward recursion, the backward recursion, and the output stage separately.

Our approach is basically the same as that of Section 5A.1. Therefore, we start by simplifying things with the reduced forward recursion given by

$$\alpha^{(t+1)} = \begin{cases} 1/2 & \text{if } Y_t = e \text{ and } W_t = e \\ \alpha^{(t)} & \text{if } Y_t = 0 \text{ or } Z_t = 0 \text{ or } Z_t = 1 \\ 0 & \text{if } Y_t = + \\ 1 & \text{if } Y_t = - \end{cases}.$$

Using this, we see that, for all $t \geq \min\{i \geq 1 | Y_i \neq 0\}$, $\alpha^{(t)}$ will be confined to the finite set $\{0, 1/2, 1\}$.

The two state Markov chain that we use to characterize the forward recursion is again based on the known state condition K_α (i.e., $\alpha^{(t)} \in \{0,1\}$) and the unknown state condition U_α (i.e., $\alpha^{(t)} = 1/2$). In this case, the new Markov chain transitions from the K_α state to the U_α state only if $W = e$ and $Y = e$. Therefore, we have $Pr(K_\alpha \to U_\alpha) = 1 - Pr(K_\alpha \to K_\alpha) = \epsilon\delta$. The new Markov chain also transitions from the U_α state to the K_α state only if $Y \in \{+,-\}$. This means that we have $Pr(U_\alpha \to K_\alpha) = 1 - Pr(U_\alpha \to U_\alpha) = (1-\epsilon)/2$. The steady state probabilities $Pr(K_\alpha)$ and $Pr(U_\alpha)$ can be found using the eigenvector equation,

$$\begin{bmatrix} Pr(K_\alpha) & Pr(U_\alpha) \end{bmatrix} \begin{bmatrix} 1-\epsilon\delta & \epsilon\delta \\ \frac{(1-\epsilon)}{2} & \frac{(1+\epsilon)}{2} \end{bmatrix} = \begin{bmatrix} Pr(K_\alpha) & Pr(U_\alpha) \end{bmatrix},$$

whose solution is $Pr(K_\alpha) = 1 - Pr(U_\alpha) = \frac{1-\epsilon}{1-\epsilon+2\epsilon\delta}$.

The precoded case is also simplified by the fact that the state diagram of the precoded channel is such that time reversal is equivalent to negating the sign of the output. Therefore, the statistics of the forward and backward recursions are identical and $Pr(K_\beta) = 1 - Pr(U_\beta) = Pr(K_\alpha)$.

Now, we consider the output stage of the BCJR algorithm for the precoded DEC. At any point in the trellis, there are now four distinct possibilities for forward/backward state knowledge: $K_\alpha K_\beta$, $K_\alpha U_\beta$, $U_\alpha K_\beta$, and $U_\alpha U_\beta$. At the extrinsic output of the decoder, the respective erasure probability conditioned on each possibility is: 0, ϵ, ϵ, and ϵ. Therefore, the erasure probability of the extrinsic output of the BCJR is

$$\begin{aligned} P_e &= \epsilon\left(1 - Pr(K_\alpha)Pr(K_\beta)\right) \\ &= \epsilon\left(1 - \frac{(1-\epsilon)^2}{(1-\epsilon+2\epsilon\delta)^2}\right) \\ &= \frac{4\epsilon^2\delta(1-\epsilon(1-\delta))}{(1-\epsilon(1-2\delta))^2}. \end{aligned}$$

Again, decoding without *a priori* information is equivalent to choosing $\delta = 1$, and the corresponding expression simplifies to $4\epsilon^2/(1+\epsilon)^2$.

5B Joint Iterative Decoding DE for General Channels

While the initial steps of this analysis were focused on GECs, it is entirely possible to write out the DE recursion, such as (5.3.1), for general message passing. Many of these ideas were first introduced by Richardson and Urbanke in [13]. This recursion will track the density function of the messages passed along a particular edge. In general, the messages will be LLRs but most of this analysis does not depend on this. We will, however, make use of notation based on using LLR messages. For example, we use Δ_∞ to denote the message implying perfect knowledge of a "0" bit because, in the LLR domain, this message corresponds to a delta function at infinity. Likewise, the message implying perfect knowledge of a "1" bit is denoted by $\Delta_{-\infty}$. We also use Δ_0 to denote the message implying the complete lack of knowledge because, in the LLR domain, this message corresponds to a delta function at zero.

For channels with memory, the standard DE assumption of channel symmetry may not hold. Essentially, this means that DE can only be applied to one codeword at a time. In [7], the

i.i.d. channel adaptor is introduced as a conceptual device which ensures the symmetry of any channel. If the outer code is a linear code, then this approach is identical to choosing a random coset and treating it as part of the channel. In this section, we use the i.i.d. channel adaptor approach so that DE can be applied to all codewords simultaneously.

Let $\otimes$ be a commutative binary operator on message densities which represents the message combining function for the bit nodes. For example, if the messages consist of LLRs, then this operator will represent convolution because LLRs are added at the bit nodes. In this case, the operator can be defined by noting that $R = P \otimes Q$ implies

$$R(x) = \int_{-\infty}^{\infty} P(y)Q(x-y)dy.$$

Using the notation,

$$P^{\otimes k} = \underbrace{P \otimes \ldots \otimes P}_{k \text{ times}},$$

for exponentials, we can now define $\lambda^{\otimes}(P) = \sum_{\nu \geq 1} \lambda_\nu P^{\otimes \nu - 1}$ and $L^{\otimes}(P) = \sum_{\nu \geq 1} L_\nu P^{\otimes \nu}$. Let $\oplus$ be a commutative binary operator on message densities which represents the message combining function for the check nodes. If min-sum decoding is used in the LLR domain, then this operator can be defined by noting that $R = P \oplus Q$ implies $R(x) = R_+(x)U(x) + R_-(x)U(-x) + R_0(x)\Delta_0(x)$, where

$$\begin{aligned} R_+(x) &= P(x)\int_x^{\infty} Q(y)dy + P(-x)\int_{-\infty}^{-x} Q(-y)dy \\ R_0 &= \int_{0^-}^{0^+} P(y)dy + \int_{0^-}^{0^+} Q(y)dy - \int_{0^-}^{0^+} P(y)Q(y)dy \\ R_-(x) &= P(x)\int_{-\infty}^{x} Q(y)dy + P(-x)\int_x^{\infty} Q(y)dy, \end{aligned}$$

and $U(x)$ is the unit step function. Using similar notation for operator exponentials, we also define $\rho^{\oplus}(P) = \sum_{\nu \geq 1} \rho_\nu P^{\oplus \nu - 1}$. We also require that both of these operators be commutative and associative with respect to scalar multiplication. This means that

$$cP \otimes Q = P \otimes cQ = c(P \otimes Q),$$

and that the respective result holds for $\oplus$.This allows us to apply the natural analogue of the binomial theorem to show that

$$(aP \oplus bQ)^k = \sum_{i=0}^{k} \binom{k}{i} a^i b^{k-i} P^{\oplus i} \oplus P^{\oplus k-i}, \tag{5B.1}$$

and that the respective result holds for $\otimes$. We also note that the zero power gives the identity, so $P^{\otimes 0} = \Delta_0$ and $P^{\oplus 0} = \Delta_\infty$.

Our next assumption, known as *decoder symmetry*, requires that the operators for the bit and check nodes obey a few identities. The bit node operator must satisfy $P \otimes \Delta_0 = P$, $P \otimes \Delta_\infty = \Delta_\infty$, and $P \otimes \Delta_{-\infty} = \Delta_{-\infty}$. This means, in some sense, that Δ_0 acts as an identity and that Δ_∞ (and $\Delta_{-\infty}$) act as zero elements. The check node operator must satisfy $P \otimes \Delta_\infty = P$, $P \otimes \Delta_0 = \Delta_0$, and the fact that $P = Q \otimes \Delta_{-\infty}$ implies $P(x) = Q(-x)$. This means, in some sense, that Δ_∞ acts as the identity, $\Delta_{-\infty}$ acts as a negative identity, and Δ_0 acts as the zero element. We note that most reasonable combining functions for the bit and check nodes satisfy these properties.

Finally, we let $\mathfrak{F}(P)$ represent the mapping from *a priori* messages to extrinsic messages for the channel decoder. Using the same arguments used for (5.3.1), we can now write the DE recursion as

$$P_{i+1} = \mathfrak{F}\left(L^{\otimes}\left(\rho^{\oplus}(P_i)\right)\right) \otimes \lambda^{\otimes}\left(\rho^{\oplus}(P_i)\right). \tag{5B.2}$$

A sufficient condition for convergence can be defined by requiring, for example, that the hard decision error probability is decreased by every iteration. The hard decision error probability for a density is simply a projection of that density onto a scalar, and there are a variety of such projections which can be used to define sufficient conditions for convergence. In the LLR domain, entropy of the bit given the message is another projection that seems to work well.

Using the i.i.d. channel adaptor approach ensures that the expected value of the decoder trajectory does not depend on the codeword transmitted. Since the choice of the random coset is absorbed into the channel, the all zero bit pattern still acts as a codeword and a fixed point of the iteration. Therefore, the message density Δ_∞ is always a fixed point of the iteration and we can discuss its stability. In the same manner as LDPC codes for memoryless channels [13], we can expand this recursion in a series about Δ_∞ by letting $P_i = (1-\epsilon)\Delta_\infty + \epsilon Q$. Our only new assumption is that the channel mapping can be expanded about this density with

$$\mathfrak{F}\left((1-\epsilon)\Delta_\infty + \epsilon Q\right) = (1 - c_F(Q)\epsilon)F_0 + c_F(Q)\epsilon D_F(Q) + O(\epsilon^2),$$

where F_0 is the perfect a priori channel LLR density, $c_F(Q)$ is a scalar valued function of a density, and $D_F(Q)$ is a density valued function of a density. We note that F_0 can be estimated in practice by simulating the channel decoder with perfect *a priori* information. The functions

$c_F(Q)$ and $D_F(Q)$ will usually depend in a complicated way on Q, but we can simplify things by defining $c_F = \sup_Q c_F(Q)$. Using simulations, more accurate results may be possible by estimating $c_F(Q)$ and $D_F(Q)$ along a particular decoding trajectory.

Now, we can use (5B.1) to expand the operators $\rho^{\oplus}$, $\lambda^{\otimes}$, and $L^{\otimes}$ around the density $(1-\epsilon)\Delta_\infty + \epsilon Q$. For $\rho^{\oplus}$, this gives

$$\begin{aligned}\rho^{\oplus}\left((1-\epsilon)\Delta_\infty + \epsilon Q\right) &= \sum_{\nu\geq 1}\rho_\nu \sum_{i=0}^{\nu-1}\binom{\nu-1}{i}(1-\epsilon)^{\nu-1-i}\epsilon^i \Delta_\infty^{\oplus \nu-1-i}\oplus Q^{\otimes i} \\ &= \sum_{\nu\geq 1}\rho_\nu(1-\epsilon)^{\nu-1}\Delta_\infty + \rho'(1)\epsilon Q + O(\epsilon^2),\end{aligned}$$

because $\sum_{\nu\geq 1}\rho_\nu(\nu-1) = \rho'(1)$. For $\lambda^{\otimes}$, this gives

$$\begin{aligned}\lambda^{\otimes}\left((1-\epsilon)\Delta_\infty + \epsilon Q\right) &= \sum_{\nu\geq 1}\lambda_\nu \sum_{i=0}^{\nu-1}\binom{\nu-1}{i}(1-\epsilon)^{\nu-1-i}\epsilon^i \Delta_\infty^{\otimes \nu-1-i}\otimes Q^{\otimes i} \\ &= \lambda_1\Delta_0 + \sum_{\nu\geq 2}\lambda_\nu(1-\epsilon^{\nu-1})\Delta_\infty + \lambda_2\epsilon Q + O(\epsilon^2).\end{aligned}$$

For $L^{\otimes}$, this gives

$$\begin{aligned}L^{\otimes}\left((1-\epsilon)\Delta_\infty + \epsilon Q\right) &= \sum_{\nu\geq 1}L_\nu \sum_{i=0}^{\nu}\binom{\nu}{i}(1-\epsilon)^{\nu-i}\epsilon^i \Delta_\infty^{\otimes \nu-i}\otimes Q^{\otimes i} \\ &= \sum_{\nu\geq 1}L_\nu(1-\epsilon^\nu)\Delta_\infty + a_L\lambda_1\epsilon Q + O(\epsilon^2),\end{aligned}$$

because $L_1 = L'(0) = a_L\lambda_1$.

Now, we can combine these expansions with (5B.2) to estimate P_{i+1} given that $P_i = (1-\epsilon)\Delta_\infty + \epsilon Q$. Working through the details shows that P_{i+1} is given by

$$\left((1-O(\epsilon))F_0 + c_F a_L\lambda_1\rho'(1)\epsilon Q\right)\otimes\left((1-O(\epsilon))\Delta_\infty + \lambda_1\Delta_0 + \lambda_2\rho'(1)\epsilon Q\right) + O(\epsilon^2),$$

which can be simplified to

$$(1-O(\epsilon))\Delta_\infty + (1-O(\epsilon))\lambda_1 F_0 + (1-O(\epsilon))\lambda_2\rho'(1)\epsilon Q\otimes F_0 + c_F a_L\lambda_1^2\rho'(1)\epsilon Q. \quad \text{(5B.3)}$$

Using this, we see that the $f(0) = 0$ condition for the GEC is very similar to the $F_0 = \Delta_\infty$ condition for general channels. In this case, only the last term of (5B.3) matters and an approximate

stability condition is given by $c_F a_L \lambda_1^2 \rho'(1) < 1$. If $F_0 \neq \Delta_\infty$, then we must have $\lambda_1 = 0$ so that the second term of (5B.3) vanishes. When $\lambda_1 = 0$, only the third term of (5B.3) remains and the stability condition is given by

$$\lambda_2 \rho'(1) \int_{-\infty}^{\infty} F_0(x) e^{-x/2} dx < 1.$$

We note that the integral in this equation follows from the stability condition derived for memoryless channels in [13], and the symmetry of $F_0(x)$ implied by the random coset assumption.

Bibliography

[1] D. Arnold and H. Loeliger. On the information rate of binary-input channels with memory. In *Proc. IEEE Int. Conf. Commun.*, pages 2692–2695, Helsinki, Finland, June 2001.

[2] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv. Optimal decoding of linear codes for minimizing symbol error rate. *IEEE Trans. Inform. Theory*, 20(2):284–287, March 1974.

[3] S. Chung, G. D. Forney, Jr., T. J. Richardson, and R. L. Urbanke. On the design of low-density parity-check codes within 0.0045 dB of the Shannon limit. *IEEE Commun. Letters*, 5(2):58–60, Feb. 2001.

[4] C. Douillard, M. Jézéquel, C. Berrou, A. Picart, P. Didier, and A. Glavieux. Iterative correction of intersymbol interference: Turbo equalization. *Eur. Trans. Telecom.*, 6(5):507–511, Sept. – Oct. 1995.

[5] R. G. Gallager. *Low-Density Parity-Check Codes*. The M.I.T. Press, Cambridge, MA, USA, 1963.

[6] J. Hou, P. H. Siegel, and L. B. Milstein. The performance analysis of low density parity-check codes on Rayleigh fading channels. In *Proc. 38th Annual Allerton Conf. on Commun., Control, and Comp.*, volume 1, pages 266–275, Monticello, IL, USA, Oct. 2000.

[7] J. Hou, P. H. Siegel, L. B. Milstein, and H. D. Pfister. Multilevel coding with low-density parity-check component codes. In *Proc. IEEE Global Telecom. Conf.*, pages 1016–1020, San Antonio, Texas, USA, Nov. 2001.

[8] A. Kavčić, X. Ma, and M. Mitzenmacher. Binary intersymbol interference channels: Gallager codes, density evolution and code performance bounds. submitted to *IEEE Trans. Inform. Theory*, 2001.

[9] B. M. Kurkoski, P. H. Siegel, and J. K. Wolf. Joint message-passing decoding of LDPC codes and partial-response channels. *IEEE Trans. Inform. Theory*, 48(6):1410–1422, June 2002.

[10] M. G. Luby, M. Mitzenmacher, and M. A. Shokrollahi. Analysis of random processes via and-or tree evaluation. In *SODA: ACM-SIAM Symposium on Discrete Algorithms*, pages 364–373, Jan. 1998.

[11] M. G. Luby, M. Mitzenmacher, M. A. Shokrollahi, D. A. Spielman, and V. Stemann. Practical loss-resilient codes. In *Proc. of the 29th Annual ACM Symp. on Theory of Comp.*, pages 150–159, 1997.

[12] H. D. Pfister, J. B. Soriaga, and P. H. Siegel. On the achievable information rates of finite state ISI channels. In *Proc. IEEE Global Telecom. Conf.*, pages 2992–2996, San Antonio, Texas, USA, Nov. 2001.

[13] T. J. Richardson, M. A. Shokrollahi, and R. L. Urbanke. Design of capacity-approaching irregular low-density parity-check codes. *IEEE Trans. Inform. Theory*, 27(1):619–637, Feb. 2001.

[14] V. Sharma and S. K. Singh. Entropy and channel capacity in the regenerative setup with applications to Markov channels. In *Proc. IEEE Int. Symp. Information Theory*, page 283, Washington, DC, USA, June 2001.

[15] M. A. Shokrollahi. New sequences of linear time erasure codes approaching the channel capacity. In *Applicable Algebra in Eng., Commun. Comp.*, pages 65–76, 1999.

[16] M. A. Shokrollahi. Capacity-achieving sequences. In *Codes, Systems, and Graphical Models*, volume 123 of *the IMA Vol. in Math. and its Appl.*, pages 153–166. Springer, 2001.

[17] S. ten Brink. Rate one-half code for approaching the Shannon limit by 0.1 dB. *Electronic Letters*, 36(15):1293–1294, July 2000.

[18] S. ten Brink. Convergence behavior of iteratively decoded parallel concatenated codes. *IEEE Trans. Commun.*, 49(10):1727–1737, Oct. 2001.

[19] S. ten Brink. Exploiting the chain rule of mutual information for the design of iterative decoding schemes. In *Proc. 39th Annual Allerton Conf. on Commun., Control, and Comp.*, Oct. 2001.

[20] R. Urbanke. Iterative coding systems. http://www.calit2.net/events/2001/courses/ics.pdf, Aug. 2001.

[21] N. Varnica and A. Kavčić. Optimized LDPC codes for partial response channels. In *Proc. IEEE Int. Symp. Information Theory*, page 197, Lausanne, Switzerland, June 2002. IEEE.