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 $0 \le \delta \le 1$.

We also consider sequences, $\{C_{n_i}\}_{i \ge 0}$, of code ensembles, where each C_{n_i} is an ensemble of (n_i, k_i) codes. We assume that the sequences, $\{n_i\}_{i \ge 0}$ and $\{k_i\}_{i \ge 0}$, 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 $\{n_i\}_{i \ge 0}$.

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 $h \ge 1$ such that $\overline{A}_h(C_{n_i}) > 0$ and let $h_{max}(n_i)$ be the largest $h \le n_i$ 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 $\delta_{min} \le \delta \le \delta_{max}$ 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 $\delta_{min} \le \delta \le \delta_{max}$.

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 $0 \le \delta \le 1$ 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 $\tilde{r}(\delta; C) \le r(\delta; C)$.

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 $g(n) \le cf(n)$ for all $n \ge n_0$. Similarly, the expression $g(n) = \Omega(f(n))$ means that there exist positive constants c and n_0, such that $g(n) \ge cf(n)$ for all $n \ge n_0$. 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 $0 \le \alpha \le \alpha_\bullet$. 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 $0 \le \alpha \le \alpha_\bullet$. 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

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

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

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

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 $e^{-n^{\epsilon}}$.*

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 $e^{-n^{\epsilon}}$. □

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

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

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

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

where

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

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*

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

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

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

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 $\lim_{\delta\to 0^+}\left(r(\delta;C)/\delta\right)$, 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 $\alpha_{TS} = \sup_{\lambda_0 \le \lambda \le 1} \{\alpha | r(\delta;C) < K(\delta,\alpha), \lambda \le \delta \le 1\}$).

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*

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

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 $w \le \rho h$. 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

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

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.

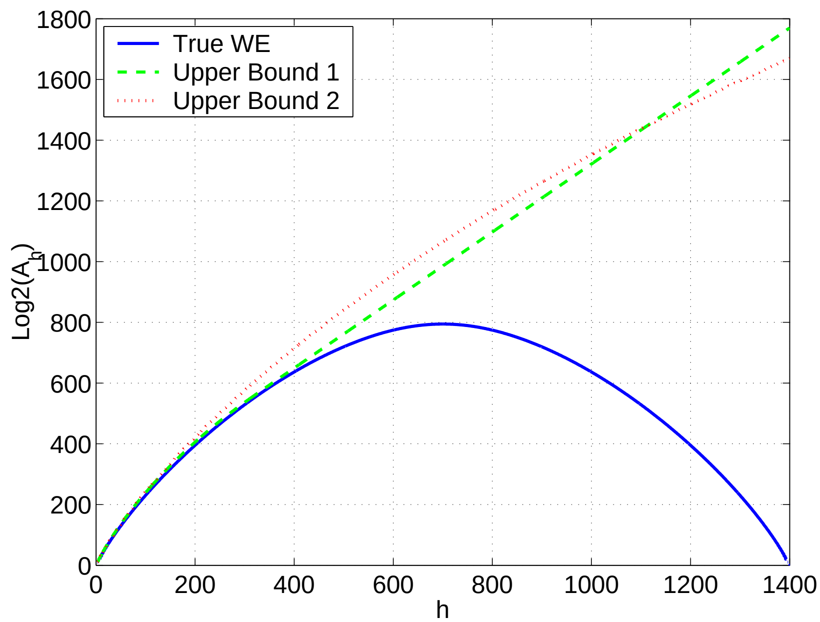


Figure 3.3.1: The true WE and upper bounds for the Hamming (7,3) code.

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.

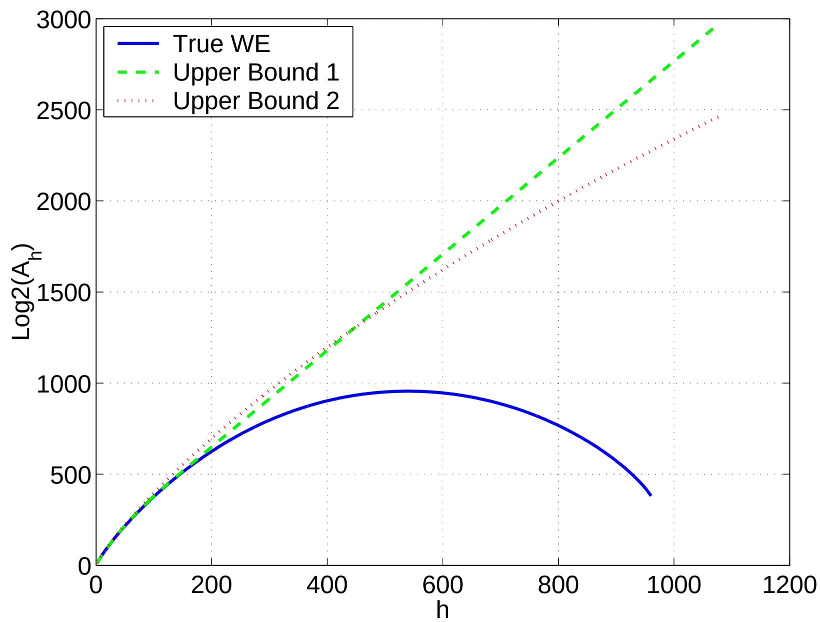


Figure 3.3.2: The true WE and upper bounds for the single parity check (9,8) CC.

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*

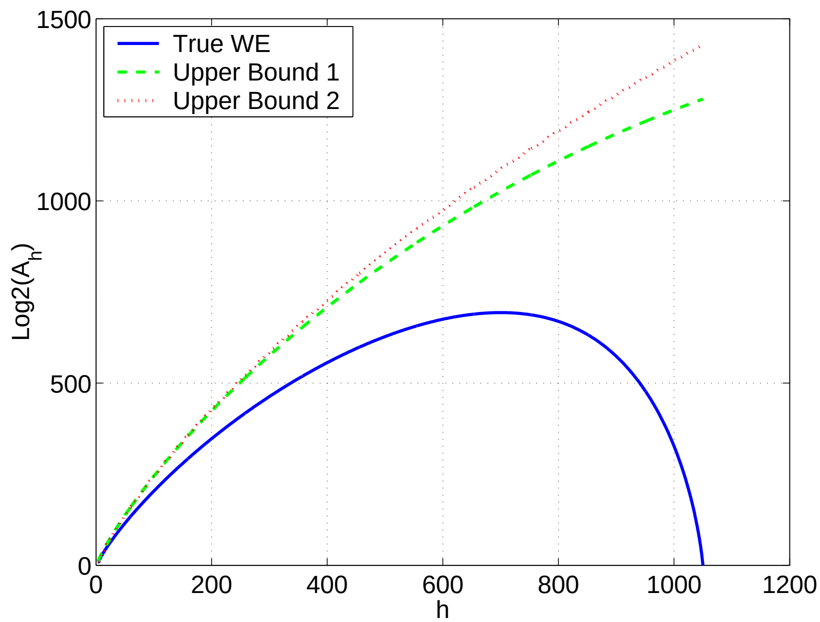


Figure 3.3.3: The true WE and upper bounds for the $G(D) = [1, 1 + D]$ CC.

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

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

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. $\square$

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 $r(\delta;\mathrm{TCC})$ 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 $r(\delta(x);\mathrm{TCC})$.

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

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

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

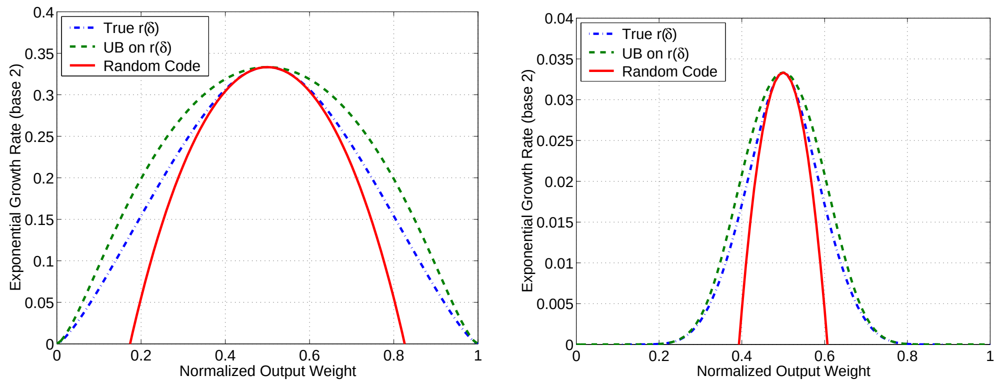
$$\begin{aligned} \overline{A}_{\delta n}^{\text{RA}}(n) &\le \sum_{i=1}^{n/q} \binom{n/q}{i} \left(2^q \delta^{q/2} (1-\delta)^{q/2}\right)^i \\ &= \left(1 + (4\delta(1-\delta))^{q/2}\right)^{n/q} - 1. \end{aligned} \tag{3.5.1}$$

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

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



(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

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

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,

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

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

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

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, $h_1,\dots,h_{m+1}$, 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, $h_1,\dots,h_{m+1}$, is valid if $h_1 \geq d$ and $h_{i+1} \geq \lceil h_i/2 \rceil$ for $i = 1,\dots,m-1$. 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

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

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 $i = 1,\dots m$.

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

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

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

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

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 $h_{i+1} \ge \lceil h_i/2 \rceil$ 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 $h_1 \ge 2$ is equal to

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

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

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 $\beta_M = \max_{h_{m+1}\ge 1} \alpha(h_{m+1}) = \nu(d)$, so the maximum exponent, ν, is given by $\nu = \nu(d)$ or

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

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

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

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 $\lim_{\delta \to 0^+} \left(r^{(m+1)}(\delta; \mathrm{CA}^m)/\delta\right) < \infty$. It is easy to verify that the limit is given by

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

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 $\lim_{\delta \to 0^+} \left(r^{(m+1)}(\delta; \mathrm{CA}^m)/\delta\right) < \infty$, 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 $P_B = \Theta(n^{\nu-1})$. $\square$

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 $\max_{0\leq x\leq 1} [H(x) + C + p(x,y)] = H(y) + C$. Using (3.4.4) to expand the $p(x,y)$ on the LHS of this expression gives

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

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

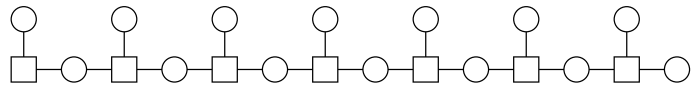


Figure 3.7.1: A GTW graph for the rate-1 “accumulate” code.

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

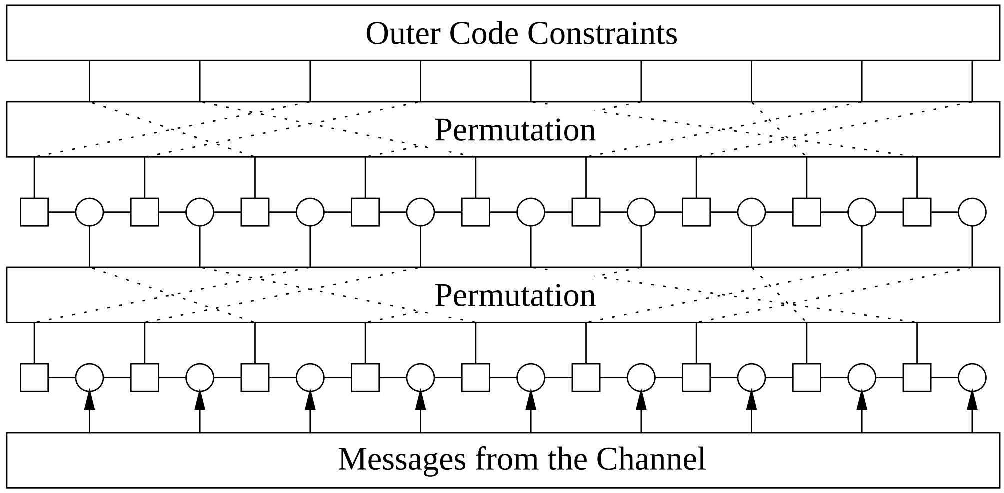


Figure 3.7.2: A Tanner graph for an arbitrary CA^2 code.

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*

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

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

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

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 $z(\alpha) < 1/(2\kappa+1)$. $\square$

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.

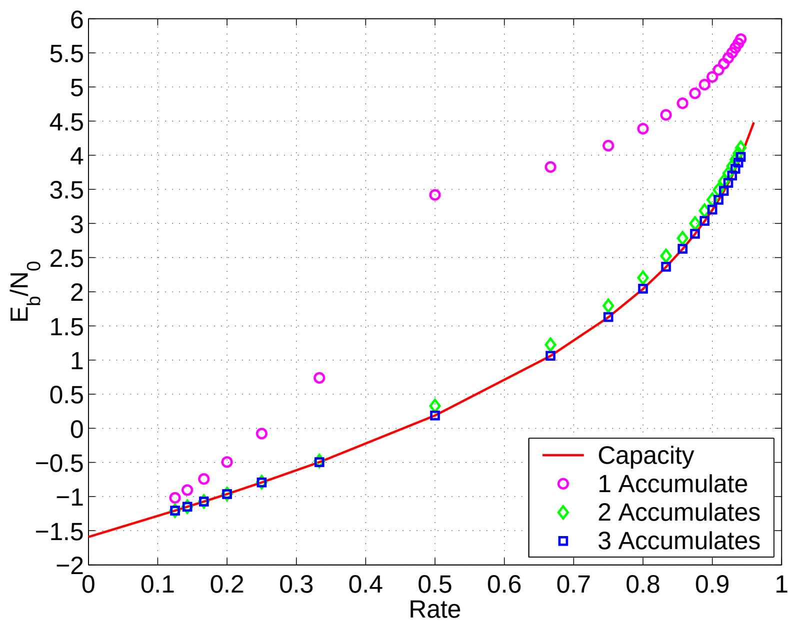


Figure 3.8.1: Typical set decoding E_b/N_0 thresholds for RA^m and PA^m codes in AWGN.

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 $\tilde{r}^{(i)}(j\Delta; \text{CA}^m)$ be an estimate of $r^{(i)}(j\Delta; \text{CA}^m)$ based on this quantization. We use the recursive update,

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

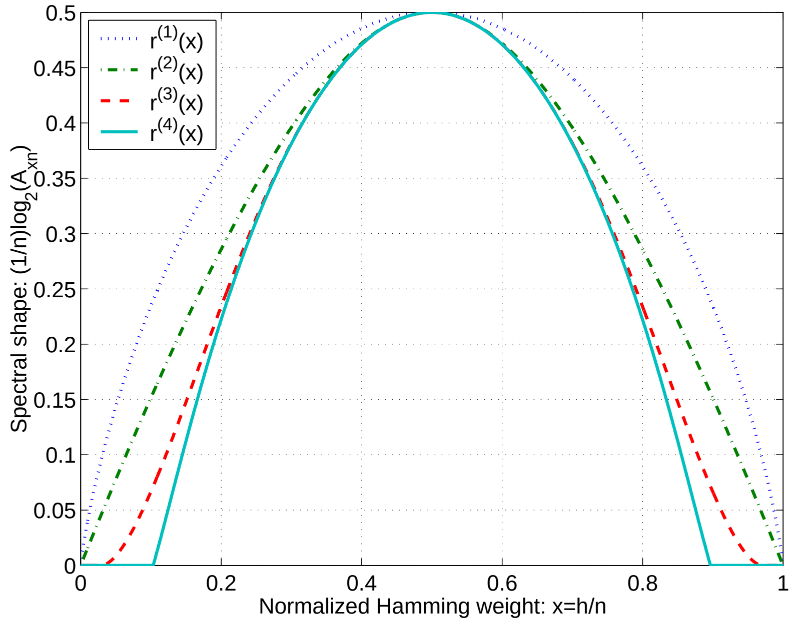


Figure 3.8.2: The spectral shape of a (2,1) single parity code and the associated PA^m codes with $m = 1, 2, 3$.

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 $\max_{0 \le j \le N} \tilde{r}^{(m+1)}(j\Delta; \text{CA}^m) - K(j\Delta, \alpha) = 0$.

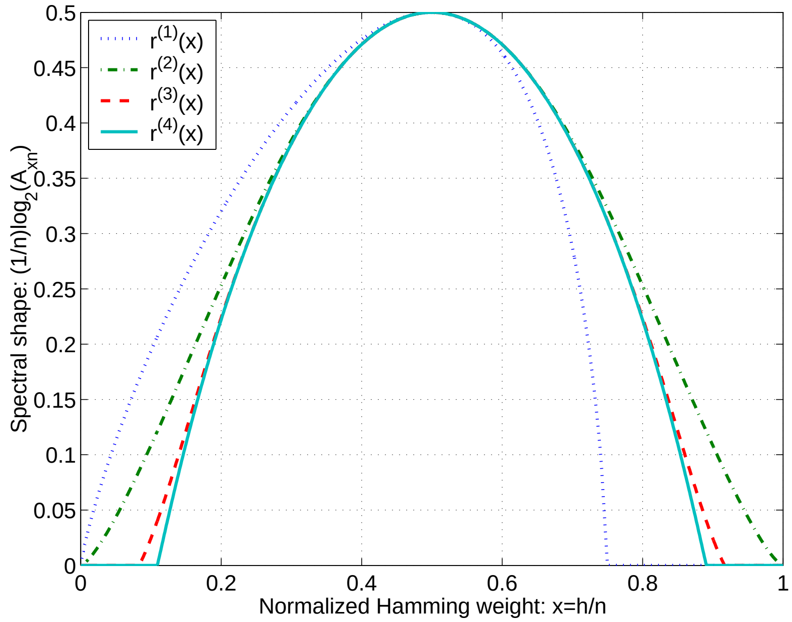


Figure 3.8.3: The spectral shape of a $[1, 1 + D]$ CC and the associated CA^m codes with $m = 1, 2, 3$.

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

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

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. $\square$

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

For $h \ge d \ge 2$, we use the bound, $h/\lfloor h/d \rfloor \le 2d$, 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

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

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 $\lfloor h/d \rfloor \le n/\tau$. If $d \ge \tau$, this means that (3B.5) holds for the entire range, $1 \le h \le n$. If $d < \tau$, we can show, with the aid of a few additional assumptions, that (3B.5) also holds for $1 \le h \le n$.

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

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

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

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

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

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

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

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

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. $\square$

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. $\square$

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 $P_{h_1,\leq h}^{(m)}(n)$ with the standard formula for serial concatenation through a random interleaver, we get

$$P_{h_1,\leq h}^{(m)}(n)=\sum_{h_2,\ldots,h_{m-1}}^{n}\sum_{h_m=1}^{h}\prod_{i=1}^{m}P_{h_i,h_{i+1}}(n).$$

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

$$P_{h_1,\leq h_m}^{(m)}(n)\leq\sum_{h_2=\lceil h_1/2\rceil}^{2h_3}\cdots\sum_{h_i=\lceil h_{i-1}/2\rceil}^{2h_{i+1}}\cdots\sum_{h_{m-1}=\lceil h_{m-2}/2\rceil}^{2h_m}\prod_{i=1}^{m}\left(\frac{4h_{i+1}}{n}\right)^{\lceil h_i/2\rceil}.$$

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). $\square$

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

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

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 $d \geq 4$. $\square$

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.