Chapter 2

The Serial Concatenation of Rate-1 Codes Through Uniform Interleavers

2.1 Introduction

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

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

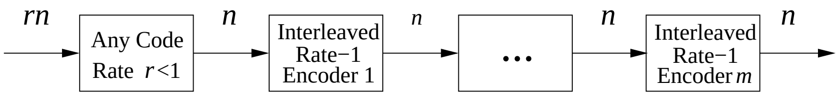


Figure 2.1.1: Our system consists of any rate $r < 1$ code followed by m rate-1 codes.

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

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

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

The outline of the chapter is as follows. In Section 2.2, we review the *weight enumerator* (WE) of linear block codes and the union bound on the probability of error for maximum likelihood decoding. We also review the average weight enumerator for the serial concatenation of two linear block codes through a URI, and relate serial concatenation to matrix multiplication using a normalized form of each code’s *input output weight enumerator* (IOWE). In Section 2.3, we introduce our system, shown in Fig. 2.1.1, compute its average output WE, and compare this WE to that of random codes. In Section 2.4, we consider some properties of rate-1 codes which affect the performance of our system. In Section 2.5, we discuss a probabilistic bound on the minimum distance of any code, taken from an ensemble, in terms of the ensemble averaged WE. Applying this bound to the WE from Section 2.3 gives an expression that is very similar to the Gilbert-Varshamov Bound (GVB) and that is asymptotically equal to the GVB for large block lengths. We also evaluate this bound numerically for various CA^m codes and observe that 3 or 4 “accumulate” codes seem to be sufficient to achieve the bound derived for asymptotically large m. In Section 2.6, we evaluate the performance of those same CA^m codes using bounds on the MLD error probability and simulations for iterative decoding error probability. Finally, in Section 2.7, we share some conclusions and discuss the direction of our future work.

2.2 Weight Enumerators and Serial Concatenation

2.2.1 The Union Bound

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

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

and the MLD probability of bit error is upper bounded by

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

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

2.2.2 Serial Concatenation through a Uniform Interleaver

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

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

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

$$Pr\,(w \to h) \overset{def}{=} \frac{A_{w,h}}{\binom{k}{w}}. \tag{2.2.1}$$

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

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

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

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

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

2.2.3 A Simple Example - The Accumulate Inner Code

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

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

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

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

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

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

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

2.3 Multiple Rate-1 Serial Concatenations

2.3.1 The Input Output Weight Enumerator

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

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

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

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

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

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

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

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

If $\mathbf{P}$ is a finite dimensional stochastic matrix, then we can associate it with a finite-state Markov Chain (MC) with state transition matrix $\mathbf{P}$. In this case, both $\mathbf{P}$ and $\mathbf{Q}$ are finite dimensional stochastic matrices and the association matches states in the MC with input/output weights of the rate-1 UIC. Using some well-known definitions from the theory of MCs, we say that $\boldsymbol{\pi} = [\pi_0, \ldots, \pi_n]$ is a stationary state distribution of the MC with transition probability matrix $\mathbf{P}$ if $\boldsymbol{\pi}\mathbf{P} = \boldsymbol{\pi}$ and $\sum \pi_i = 1$. This allows us to associate a stationary state distribution, $\boldsymbol{\pi}$, of the MC with a stationary weight distribution of the rate-1 UIC. If the average WE, $\overline{A}_h$, of a code ensemble is not changed by encoding every code in the ensemble with the same rate-1 UIC, then $\overline{A}_h$ is a stationary WE of that rate-1 UIC. Using (2.2.2), it is easy to verify that this occurs when

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

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

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

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

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

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

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

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

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

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

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

for $1 \leq h \leq n$.

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

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

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

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

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

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

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

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

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

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

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

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

2.3.2 A Large Number of Concatenations

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

Remark 2.3.9. An interesting open question is whether the ensemble $\Omega_m(\mathbf{C}^{(o)}, \mathbf{C}^{(i)})$ contains all invertible linear codes, for sufficiently large m. Using the generator matrix definition, (2.3.1), it is possible to give a sufficient condition for this. Let S be the set of all $n \times n$ permutation matrices and define $T = \left\{ \mathbf{\Pi} \mathbf{C}^{(i)} | \mathbf{\Pi} \in S \right\}$. Since $\mathbf{C}^{(i)}$ is invertible by assumption and all permutation matrices are invertible, it is clear that T is a subset of the multiplicative group of $n \times n$ invertible binary matrices denoted $GL_n(\mathbb{F}_2)$. Let $T^m = \{\mathbf{V}_1 \mathbf{V}_2 \ldots \mathbf{V}_m | \mathbf{V}_i \in T\}$ and assume that there exists an m_0 such that $T^{m_0} = GL_n(\mathbb{F}_2)$. In this case, $\Omega_m(\mathbf{C}^{(o)}, \mathbf{C}^{(i)})$ will contain all invertible linear codes for all $m \geq m_0$. Furthermore, the limit, $\lim_{m\to\infty} \Omega_m(\mathbf{C}^{(o)}, \mathbf{C}^{(i)})$, exists and is equal to the ensemble of all invertible linear codes under the uniform distribution. For example, when $\mathbf{C}^{(i)}$ is the "accumulate" code, we have verified that this occurs for $n = 2, 3, 4$ with $m_0 = n + 1$.

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

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

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

Proof. Starting with (2.3.2) gives

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

Applying (2.3.3) gives

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

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

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

For the second statement, we start with

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

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

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

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

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

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

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

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

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

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

for all $m\geq m_0$.

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

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

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

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

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

2.4 Properties of Rate-1 Codes

2.4.1 Conditions for Primitivity

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

Theorem 2.4.1. *Let $\mathbf{h}=h_0,h_1,h_{2,}\ldots$ be the semi-infinite impulse response of a nontrivial, causal, rate-1/1 convolutional code. To avoid degenerate cases, assume that $h_0 = 1$. Define l to be the smallest positive integer such that $h_l=1$. Then, the rate-1 block code formed by truncating this convolutional code, to any length $n\geq l+1$, is primitive.*

Proof. This proof is given in the Appendix. □

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

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

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

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

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

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

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

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

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

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

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

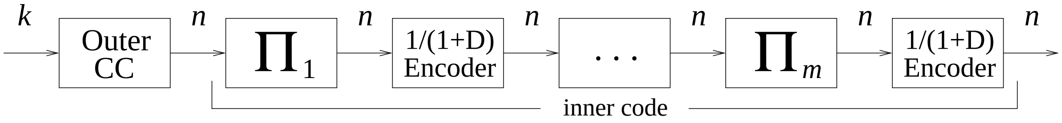


Figure 2.5.1: Encoder for a CA^m Code with the block size indicated at each stage.

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

2.5 Bounds on the Minimum Distance

2.5.1 The Minimum Distance Distribution

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

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

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

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

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

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

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

□

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

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

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

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

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

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

where

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

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

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

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

where

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

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

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

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

which can be simplified to

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

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

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

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

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

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

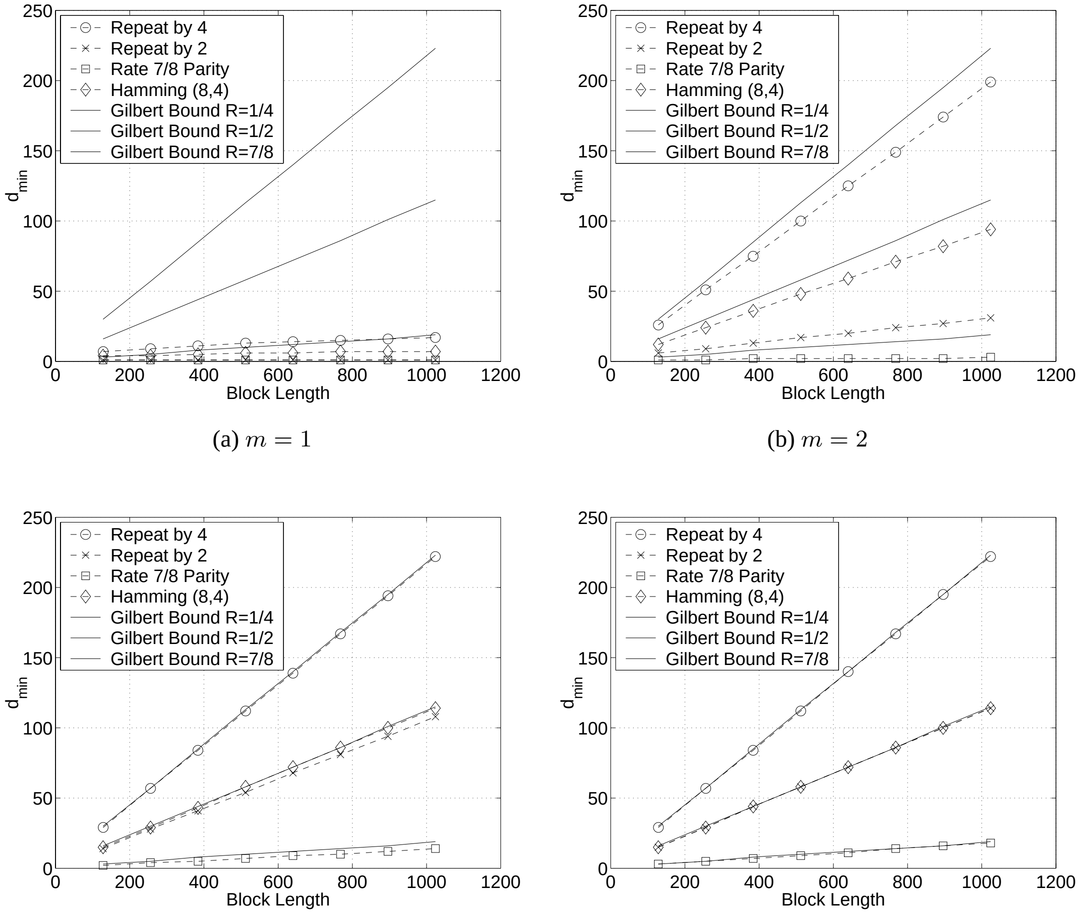
Now, we can define similar normalized distance bounds for the uniform ensemble and for the ensemble, $\Omega_*(\mathbf{C}^{(o)}, \mathbf{C}^{(i)})$. Let $\delta_U(r, \epsilon)$ be the largest δ such that $\lim_{n\to\infty} S(n, \lceil rn \rceil, \delta n)$ $< \epsilon$ and let $\delta_\Omega(r, \epsilon)$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

2.5.2 Convolutional Accumulate-m (CA^m) Codes

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

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

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



(a) $m = 1$

(b) $m = 2$

(c) $m = 3$

(d) $m = 4$

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

fixed rate, a "good" code is defined by a minimum distance which grows linearly with the block length. When examining these results, we will focus on whether or not the minimum distance appears to be growing linearly with block length and on how close the $d_C^*(n_i, 1/2)$ is to the GVB. For ensembles of CA^m codes with $m = 1$, it is known that the minimum distance of almost all of the codes grows like $O(n^{(d^o-2)/d^o})$, where d^o is the free distance of the outer terminated CC [11]. Examining Fig. 2.5.2 for $m = 1$, we see that the minimum distance grows slowly for R4 and H8 (which have $d^o \geq 3$) and not at all for R2 and P9 (which have $d^o = 2$). For $m = 2$, the growth rate of the minimum distance for R4, H8, and R2 appears distinctly linear. It is difficult to determine the growth rate of P9 with $m = 2$ from these results. With $m = 3$, all of the codes appear to have a minimum distance growing linearly with the block length. In fact, the apparent growth rates are very close to $\delta^*_{GVB}(r)$. Finally, with $m = 4$, the bounds on minimum distance and $d^*_{GVB}(n, r)$ are almost indistinguishable. These results are very encouraging and suggest that, over a range of rates, even a few "accumulate" codes are sufficient to approach the behavior of an asymptotically large number.

2.5.3 Expurgated Ensembles

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

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

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

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

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

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

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

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

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

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

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

2.6 Performance

2.6.1 The Error Exponent

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

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

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

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

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

□

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

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

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

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

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

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

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

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

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

This completes the proof. □

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

2.6.2 Maximum Likelihood Decoding Performance

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

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

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

2.6.3 Iterative Decoding Performance

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

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

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

The Interleaver Gain Exponent (IGE) conjecture is based on the observations of Benedetto and Montorsi [2] and is stated rigorously in [4]. It states that the probability of MLD decoding error for turbo-like codes will decay as $O(n^{-\nu})$, where ν depends on the details of the coding system. If the IGE conjecture predicts that the BER (resp. WER) will decay with the block length, then we say that the system has BER (resp. WER) interleaver gain. It is easy to verify that WER interleaver gain implies BER interleaver gain. The IGE and the MLD expurgated bound are quite closely connected. If a system has WER interleaver gain, the probability of picking a code with codewords of fixed weight must decay to zero as the block length increases. Therefore, one would expect the MLD expurgated bound to beat the non-expurgated bound. On the other hand, if a system has only BER interleaver gain, then it is likely that the MLD expurgated bound will equal the non-expurgated bound.

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

2.7 Conclusions and Future Work

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

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

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

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

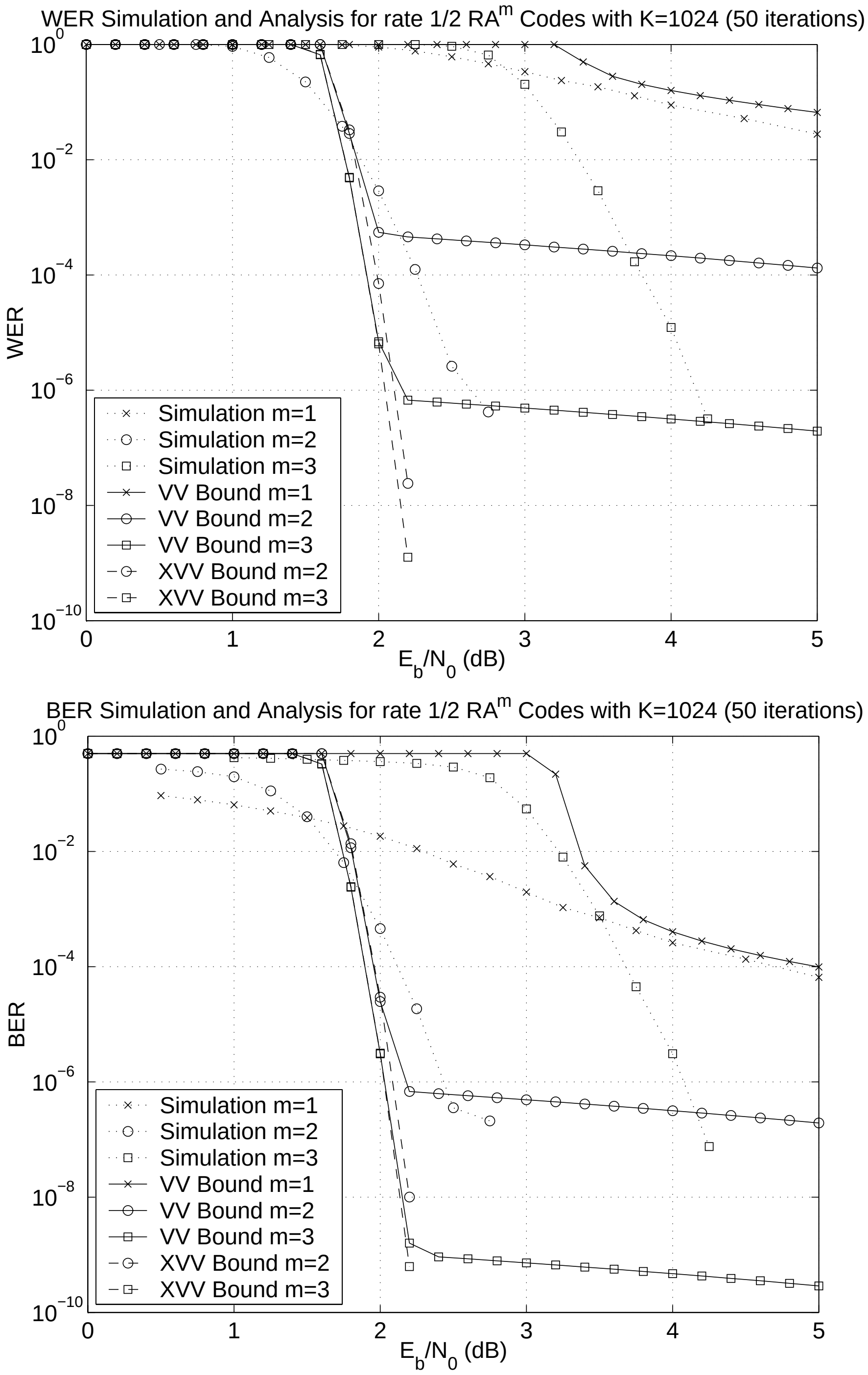
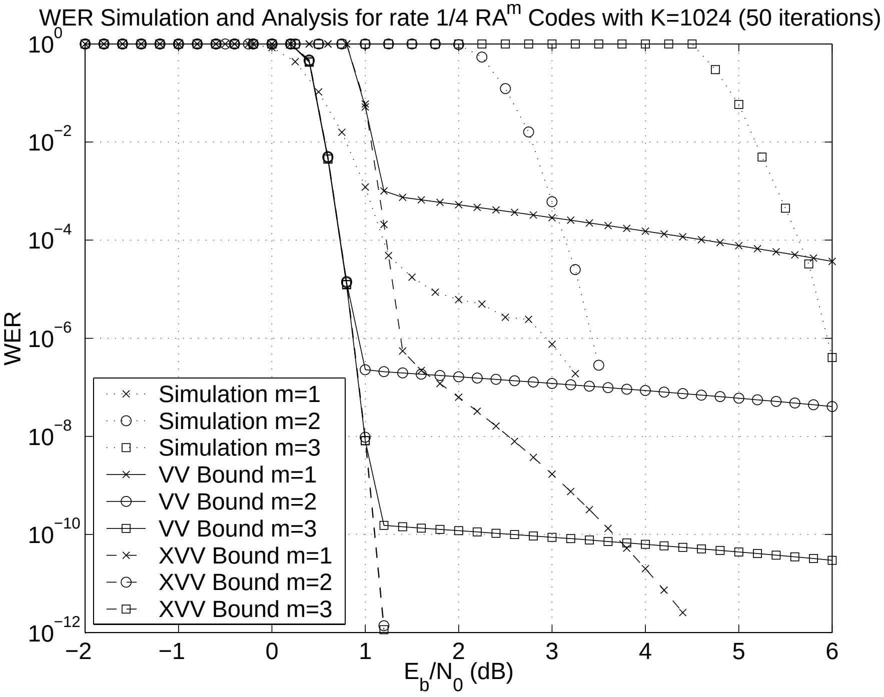


Figure 2.7.1: Analytical and simulation results for a rate 1/2 RA^m code with $k = 1024$ and $m = 1, 2, 3$. Simulations are completed using 50 decoding iterations and the top plot shows the word error rate (WER) while the bottom plot shows the bit error rate (BER). The label XVV signifies the Viterbi-Viterbi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



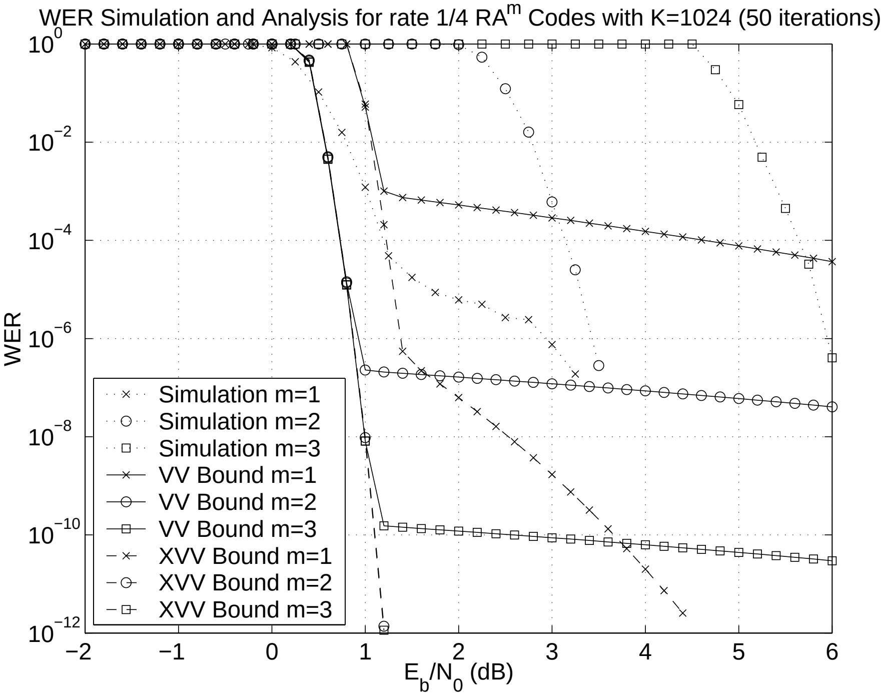


Figure 2.7.2: Analytical and simulation results for a rate 1/4 RA^m code with $k = 1024$ and $m = 1, 2, 3$. Simulations are completed using 50 decoding iterations and the top plot shows the word error rate (WER) while the bottom plot shows the bit error rate (BER). The label XVV signifies the Viterbi-Viterbi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

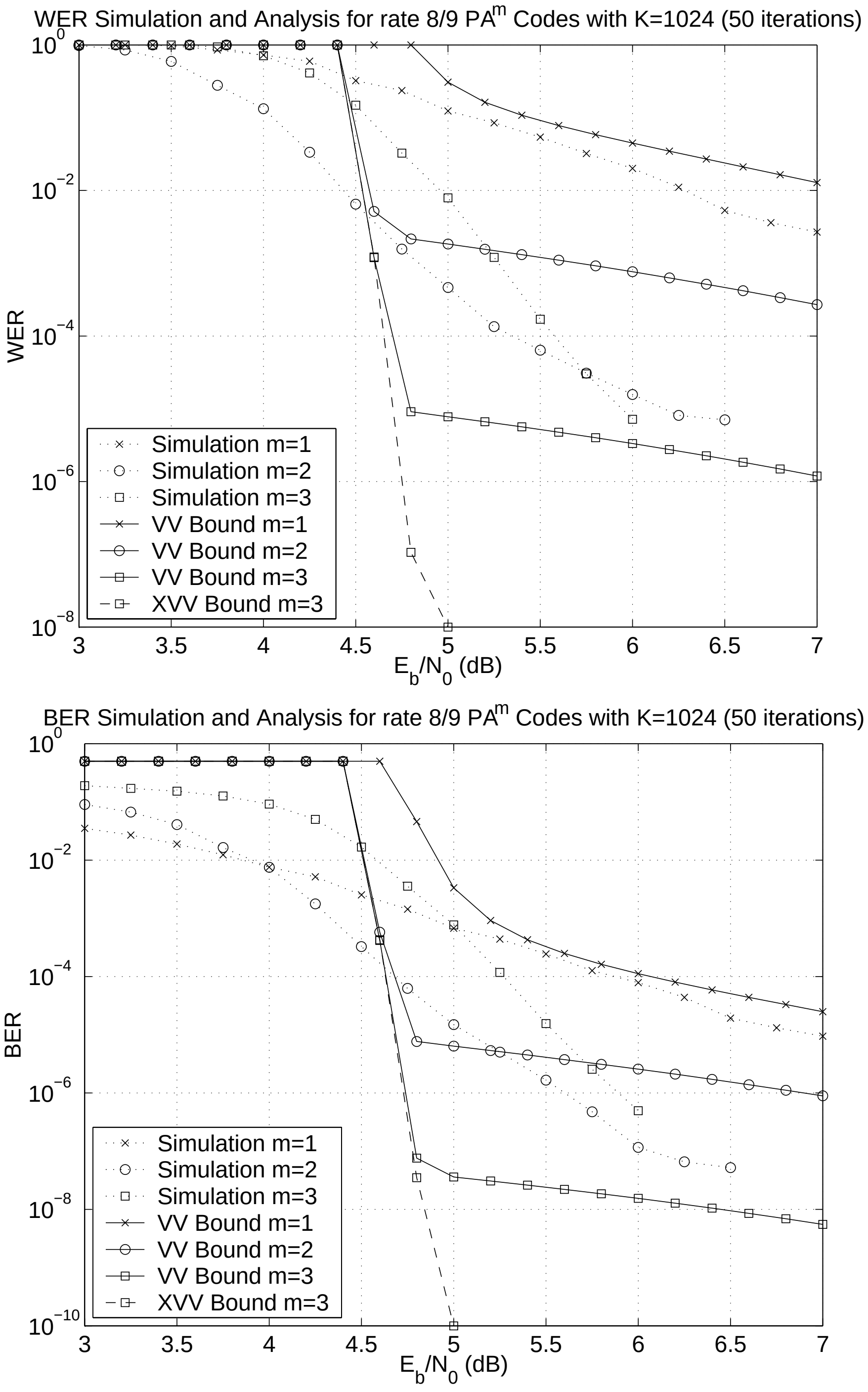


Figure 2.7.3: Analytical and simulation results for a rate 8/9 PA^m with $k = 1024$ and $m = 1, 2, 3$. Simulations are completed using 50 decoding iterations and the top plot shows the word error rate (WER) while the bottom plot shows the bit error rate (BER). The label XVV signifies the Viterbi-Viterbi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

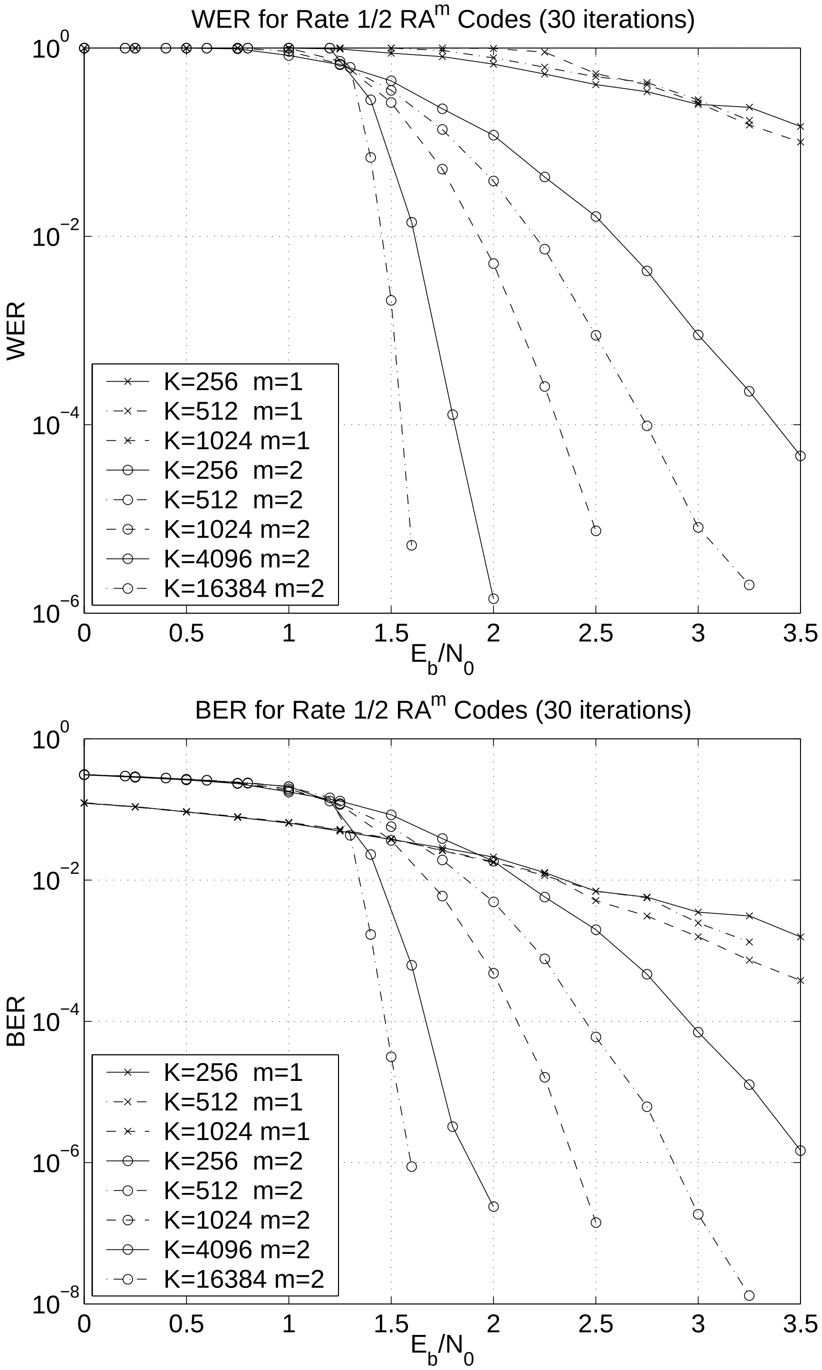


Figure 2.7.4: Simulation results for rate 1/2 RA^m codes for 30 decoding iterations with $m = 1, 2$ and $k = 1024, 2048, 4096, 8192, 16384$. The top plot shows the word error rate (WER) while the bottom plot shows the bit error rate (BER).

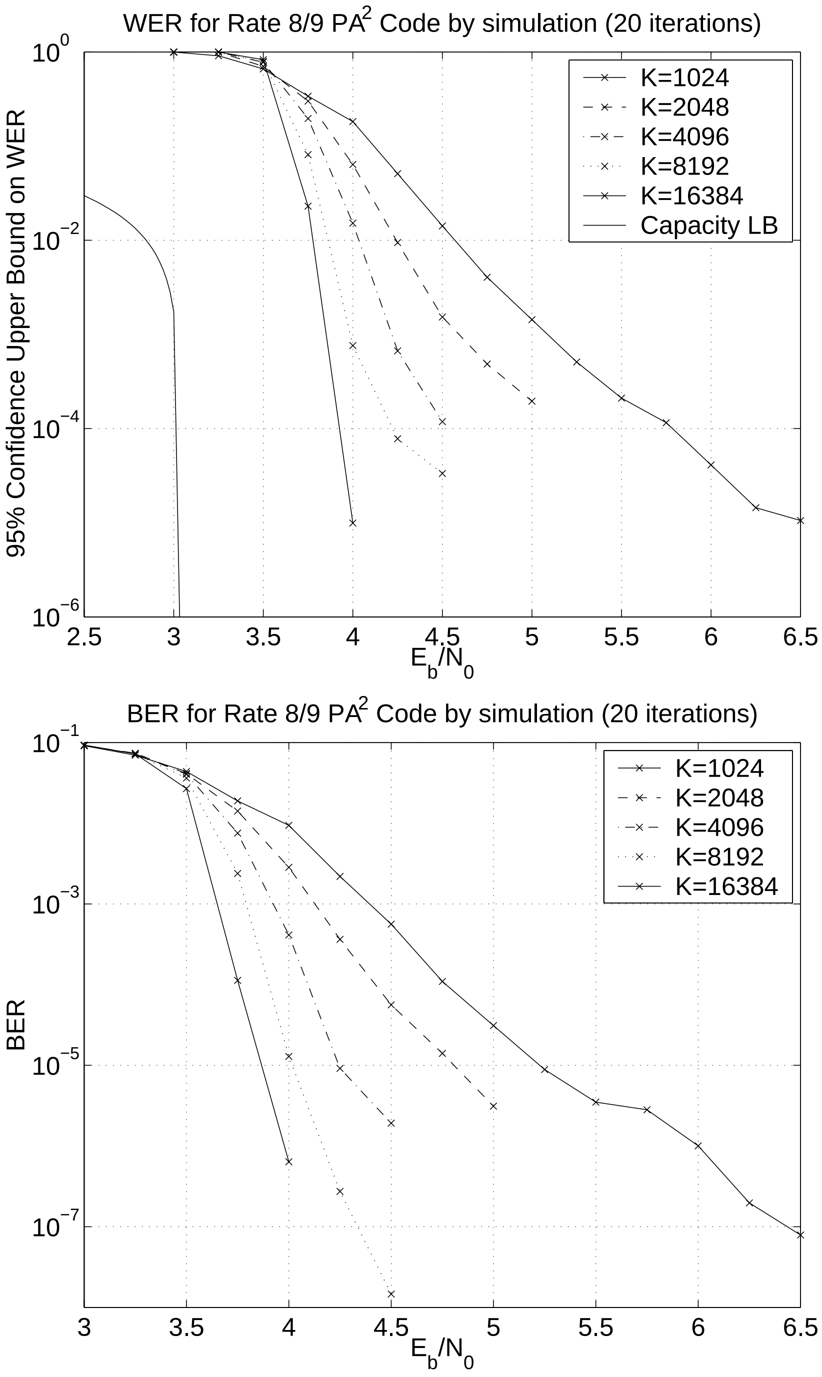


Figure 2.7.5: Simulation results for rate 8/9 PA^2 codes with $k = 1024, 2048, 4096, 8192, 16384$ and 20 decoding iterations. The top plot shows the word error rate (WER) while the bottom plot shows the bit error rate (BER).

2A Proof of Theorem 2.4.1

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

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

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

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

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

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

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

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

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

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

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

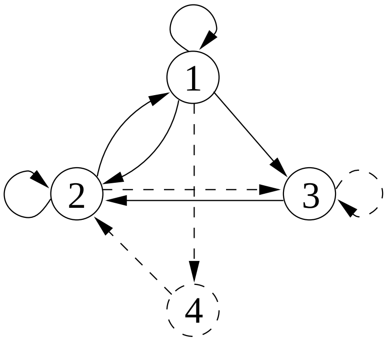


Figure 2A.1: The weight mapping graph, G_4, with the G_3 subgraph drawn in solid lines.

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

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

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

We illustrate the proof technique using the "accumulate" code example from Section

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

Bibliography

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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