1
Introduction

Low-Density Parity-Check (LDPC) codes were discovered by Robert G. Gallager in 1963, causing little impact in its contemporary communication systems due to unavailability of data processing technology capable of performing real time decoding. After more than thirty years, the works of MacKay [MKY1997] brought LDPC codes (also called Gallager Codes) to the attention of the scientific community, yielding performance comparable to that of the renowned Turbo Codes [CBG1996].

LPDC codes are described by parity-check matrices that are composed of very few non null elements per row and have very large rank (typically on the order of thousands). The matrices' low density ensures linear time decoding, i.e. the Hamming weight of each row does not grow with the block-length. It has been found that a typical code where each bit influences at least three parity checks has a minimum distance that increases linearly with the block length [RGG1963, pg. 13 -17]. The encoding time, however, tends to grow faster than linearly with the block length. Since generator matrices are obtained by inverting part of the Parity-Check Matrices, they are generally not sparse and cause the encoding times of increasingly large blocks to grow quadratically. Repeat-Accumulate codes are a special case of LDPC codes that allow linear time encoding, they are also called *staircase codes* [MKY2003] for the staircase aspect of their parity-check matrix.

Another improvement on code performance comes from the study of optimal degree distributions. It may seem awkward at first that irregular degree distributions should perform better than regular ones, but it has been proved that [LMS1997] irregular LDPC codes can achieve Channel Capacity, while regular LDPC codes have their decoding threshold theoretically bounded away from the Shannon limit. Irregular Repeat-Accumulate codes were created with the goal of providing linear encoding times and communicating at rates close to the channel capacity.

In this dissertation we apply different methods of graph construction that have been developed for use with LDPC codes to generate practical IRA codes and evaluate their performance through computer simulation on the AWGN channel using BPSK modulation.

Chapter 2 introduces IRA codes, their encoding and decoding algorithms. Chapter 3 explains how a graph may be constructed from a given degree distribution. Chapter 4 presents the results of our simulations and comments on how they may relate to the construction methods and the degree distributions. Finally, Chapter 5 is an overall conclusion on the accomplishments and unanswered questions of this work. Several appendices provide useful information for a better understanding of what is presented in the main chapters.

1.1
Channel Model

In this work we simulate digital transmission over the Additive White Gaussian Noise channel (AWGN). An actual digital communication system maps bits into symbols that modulate continuous waveforms. This continuous waveform system is illustrated in Figure 1.1. Degradation of this analog signal by addition of noise, fading and interference cause detection errors that finally result in bit errors. These disturbing elements constitute a stationnary, memoryless random process. The resulting bit errors can be fixed by forward error correction (FEC) systems that apply channel coding to the message before transmission.

This section describes how the transmitted signals can be modeled by discrete vectors instead of continuous waves in all our simulations.

Figure 1.1: Simplified channel model

We proceed to opening the *Transmission* box in Figure 1.1. The message is assumed to be a memoryless sequence with a uniform *a priori* probability for each bit. This can be accomplished either by source coding or by applying a pseudo random mask to the input stream. The following step applies channel coding to the message $\mathbf{u}_i$ resulting in a valid codeword $\mathbf{c}_i$.

Now we must map the bit stream into an analog waveform that can be transmitted through a physical medium. Nevertheless, our application of LDPC codes is not concerned with this part of the physical layer and we will solely demonstrate the equivalence of the vector channel and the AWGN channel to prove consistency. From the perspective of the soft-decision decoder, the communication system can be modeled as shown in Figure 1.2.

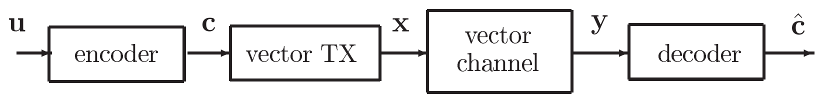

Figure 1.2: Simplified vector channel model

1.2 Vector Channel and Waveform Detection

For each message $\mathbf{u}_i$ there is an associated finite energy analog signal $s_i(t)$. We can relate the message's average bit energy to the transmitted symbol energy if we know the modulation (the set $\{s_i(t)\}_{i=1}^{M}$) and the code rate R.

Definition 1 (Code Rate) *The design rate of a code $R = \frac{k}{n}$ describes the ratio of information — or* message *— bits to the totality of bits contained in each encoded block [TKM2005, Ch. 15].* ◇

Since every non-binary modulation scheme transmits more than one bit per modulation symbol, the first step is to divide the symbol's energy by the number of transmitted bits.

$$\begin{aligned} E_i &= \int_0^T s_i^2(t)dt \\ &= \log_2 M \cdot E_{b'}, \end{aligned} \tag{1-1}$$

where M is the cardinality of the set of symbols used by the modulation scheme constellation, and $E_{b'}$ is the energy per *transmitted* bit. When we evaluate code performance, however, we are often more interested in the average energy per *communicated* bit E_b, which takes the the code into consideration and is calculated by dividing the energy of the n transmitted bits by the k message bits being communicated.

$$E_b \times k = E_{b'} \times n \tag{1-2}$$

$$E_b = \frac{E_{b'}}{R}, \tag{1-3}$$

which gives us

$$E_b = \frac{E_i}{R \log_2 M}. \tag{1-4}$$

The waveform $s_i(t)$ is obtained through the sum of normalized orthogonal signals (see Figure 1.3).

$$s_i(t) = \sum_{j=1}^{N} x_{ij} \cdot \phi_j(t)dt$$

$$\int_{-\infty}^{\infty} \phi_j(t) \cdot \phi_i(t)dt = \begin{cases} 1 & \text{if } j = i \\ 0 & \text{if } j \neq i \end{cases}$$

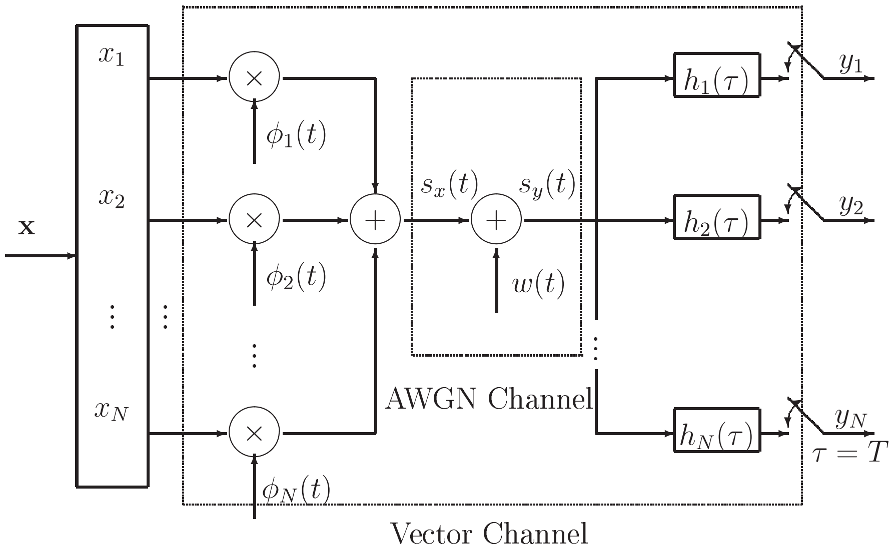

Figure 1.3: Equivalence between the vector Gaussian channel and the continuous AWGN channel

In a QAM scheme we have $N = 2$, the orthonormal functions being segments of sinusoidal waves in phase and quadrature and each element in the transmitted vectors $\mathbf{x}_i$ is assigned to one of $\sqrt[2]{M}$ discrete values. Note that this relation is not valid for 8-PSK, for instance.

From Figure 1.3, minding the orthogonality of the base signals, we can define the symbol's energy as

$$E_i = \sum_{j=1}^{N} \int_0^T (x_{ij} \cdot \phi_j(t))^2 dt.$$

and we can define this system as linear time invariant. The time domain expression for the Gaussian channel is

$$s_y(t) = s_x(t) * c(t) + w(t),$$

where we assume the channel doesn't introduce selective fading nor multi-path interference and define $c(t) = A \cdot \delta(t)$, where $\{A \in \Re \mid 0 \leq A \leq 1\}$, as the impulse response of the channel.

Now we must clarify how the effects of noise are represented in the vector channel. The received signal goes through the matched filters. Each element in the vector $\mathbf{y}$ represents the projection of $s_y(t)$ onto one of the base functions, namely,

$$\begin{aligned} y_j &= s_y(t) \, * \, h_j(t)\Big|_{t=T} \\ &= (s_x(t) + w(t)) \, * \, h_j(t)\Big|_{t=T} \\ &= \int_0^T s_x(t) \cdot \phi_j(t)dt + \int_0^T w(t) \cdot \phi_j(t)dt \\ &= x_j + z_j, \end{aligned} \tag{1-5}$$

where $h_j(t) = \phi_j(T - \tau)$ is the matched filter's impulse response. The projection of the Gaussian noise onto each orthogonal base function gives the noise vector $\mathbf{Z}$. Since we know Z_j is the response of a linear time-invariant system to $W(t)$ — a Gaussian stochastic process of zero mean and known variance — it can be proven that $\mathbf{Z}$ is a zero mean Gaussian vector of known variance as well. The variance of the discrete-time noise Z is

$$\begin{aligned} \sigma_Z^2 = E\left[Z_j^2\right] &= E\left[\int\int W(t_1)W(t_2) \cdot h(T - t_1)h(T - t_2)dt_1dt_2\right] \\ &= \int\int R_W(t_1 - t_2) \cdot h(T - t_1)h(T - t_2)dt_1dt_2 \\ &= \frac{\mathcal{N}_0}{2} \int \phi(t_1) \int \delta(t_1 - t_2)\phi(t_2)dt_2dt_1 \\ &= \frac{\mathcal{N}_0}{2} \int \phi(t)^2 dt = \frac{\mathcal{N}_0}{2}, \end{aligned} \tag{1-6}$$

where $R_W(t_1 - t_2) = E[W(t_1)W(t_2)]$ is the time correlation of $W(t)$ in instants t_1 and t_2. Thus we verify that the variance of the noise in the real valued channel output equals the intensity of the power spectral density of the additive white noise $\frac{\mathcal{N}_0}{2}$.

1.3 Soft Decision and Hard Decision

1.3.1 Soft Decision

Proving the equivalence between the noise in the vector channel and the noise in the waveform channel is important because the Gallager decoder is a soft-decision decoder. This means that instead of feeding the decoder with nearest binary values based on the received vector, i.e. hard decision, we feed it with the channel a posteriori probabilities and try to approximate with these values the maximum a posteriori estimate. This means we want to find

$$\hat{\mathbf{x}} = \arg\max_{\mathbf{x}} P\left(\mathbf{x} \mid \mathbf{y}\right).$$

Channel coding introduces a fundamental difference between $P\left(\mathbf{x} \mid \mathbf{y}\right)$ and $\prod_i P\left(x_i \mid y_i\right)$. The first one can only be obtained by investigating the code's properties, while the latter is easily estimated given the received vector, the alphabet of transmitted symbols and the properties of the channel. In a transmission using BPSK modulation through a memoryless AWGN channel with equiprobable input symbols the signal to noise ratio completely characterizes the channel and the a posteriori probabilities can be found using Bayes' rule [TKM2005][Sec. 15.4].

$$\begin{aligned} P\left(x_i \mid y_i\right) &= \frac{P\left(y_i \mid x_i\right) P(x_i)}{P(y_i)} \\ &= \frac{P\left(y_i \mid x_i\right)}{P\left(y_i \mid x_i\right) + P\left(y_i \mid \overline{x_i}\right)}, \end{aligned} \tag{1-7}$$

where $\overline{x_i}$ means the complementary value to the logical variable x_i and $P\left(y_i \mid x_i\right)$ is a short notation for $P\left(Y_i = y \mid X_i = x\right)$.

From (1-7) and (1-5), with $x_i = \pm a$, we obtain a numeric expression for the AWGN channel a posteriori probabilities.

$$\begin{aligned} P\left(X_i = a \mid y_i\right) &= \frac{P\left(y_i \mid X_i = a\right)}{P\left(y_i \mid X_i = a\right) + P\left(y_i \mid X_i = -a\right)} \\ &= (1 + e^{\frac{-2ay_i}{\sigma_Z}})^{-1} \end{aligned} \tag{1-8}$$

Keeping real values for soft decision decoding can be specially burdensome in systems that use bit interleaving, where many blocks must be kept before they go through bit de-interleaving and decoding can start. It is also possible though to reduce memory requirements using quantization.

1.3.2 Hard Decision: The BSC and the BEC

Besides the AWGN, we should mention two other important channel models: the Binary Symmetric Channel (BSC) and the Binary Erasure Channel (BEC). The first one arises directly from using hard decision on the AWGN channel, while the latter can be found in communication networks that experience packet loss and is essential to establishing the bounds on the message-passing algorithm [JKM2000].

A hard decision decoder observes the channel output and chooses the symbol that lies closer to the observed value. Therefore, hard decision decoders lead to the BSC, which assumes the channel will cause a transition on the received symbol with a known probability. The transition probability is also independent of the input symbol, therefore symmetric. A transition occurs when the additive noise takes the received value beyond the transition threshold (defined as half of their Euclidean distance) towards the opposite symbol. Using the same example as in (1-8), if the two symbols have a distance of $2a$ the channel transition probability equals $Q(a/\sigma_z)$. Since the Gallager decoder uses the channel a posteriori probabilities (1-7) as an input, we can use the same decoding algorithm for the BSC simply by limiting the channel a posteriori probability values to p and $1-p$.

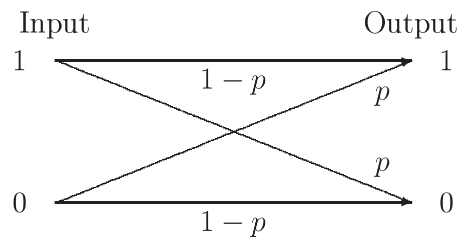

Figure 1.4: Binary Symmetric Channel

The Binary Erasure Channel differs from the BSC by erasing bit values instead of flipping them, as shown in Figure 1.5. This means the decoder will try to find the correct values only for the erased bits while keeping all others intact. The decoding of the erased bits will depend entirely on what is commonly called the extrinsic information, i.e. the information carried by all other nodes in the graph.

Although it may be unlikely that a physical medium will corrupt some bits to the point of making their values completely ambiguous while maintaining all others intact, many of today's communication systems make use of channel coding for the BEC. Communication networks that use reliable links with negligible error rates still have to face the problem of packet loss during routing and switching. Instead of waiting for retransmission of lost packets, this problem can be solved by channel coding for the BEC.

A long stream of data can be grouped in blocks for FEC encoding and then go through a bit interleaver that spreads the contents of each block throughout many packets before transmission. If some of these packets are lost in traffic, the missing bits from the lost packets will appear as spread erasures in the original data blocks after de-interleaving.

Although the Gallager decoder can be used to recover bit erasures, there are codes that have been developed specially for this purpose [LMS1997] [ASH2006], with more appropriate implementations of the message passing algorithm.

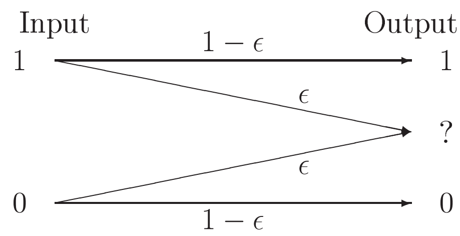

Figure 1.5: BEC