Part II

Coding and Decoding

Contents

Chapter 8

Fundamental Limits of Coding and Sequences

Chapters 1-5 studied the construction of modulated signals for transmission, where each symbol generated for transmission was independent of all other symbols. Volume II of this book addresses the situation where transmitted signals are generalized to **sequences** of symbols. Sequences of more than one symbol can be constructed with dependencies between constituent symbols that enlarge the minimum distance. Equivalently, the use of sequences allows higher rates of transmission at the same reliability and with no increase in transmit energy/power. The generation of sequences is usually called **channel coding** or just **coding**, and is studied in this second volume.

Chapter 8 begins in Section 8.1 with a definition of the **sequential encoder**, generalizing the encoder of Chapter 1. The remainder of the chapter then introduces and interprets fundamental performance limits or bounds on digital data transmission with channel coding. In particular, the maximum possible reliable data transmission rate for any particular channel, called the **channel capacity**, is found. The capacity concept was introduced by Shannon in a famous 1948 paper, "The Mathematical Theory of Communication," a paper many credit with introducing the information age. Some of the concepts introduced by Shannon were foreign to intuition, but called for achievable data rates that were well beyond those that could be implemented in practice. Unfortunately, Shannon gave no means by which to achieve the limits. By the late 1990's, communication engineers were able to design systems that essentially achieve the rates promised by Shannon.

Section 8.2 introduces the fundamental information-theory concepts of **entropy** and **mutual information** in deriving heuristically Shannon's capacity. Chapter 8 focuses more on intuition and developing capacity bounds than on the rich and extensive rigorous theory of capacity derivations, to which entire alternative texts have been devoted. Section 8.3 computes capacity for several types of channels and discusses again the AWGN-channel concept of the "gap" from Volume 1. Section 8.4 discusses capacity for a set of parallel channels, a concept used heavily with the gap approximation in Chapter 4, but related to fundamental limits directly. This area of overlap between Volume 1's Chapter 4 and Chapter 8 leads to Section 8.5 on waveform-channel capacity, allowing computation of capacity in bits/second for continuous-time channels, which does not appear in Chapter 4. The appendix for this chapter attempts to provide additional background on the mathematics behind capacity, while providing direction to other work in the area of rigorous development of capacity.

Chapters 9-11 of this Volume II examine specific decoding and coding methods for a single user, essentially illustrating how the bounds established in this Chapter 8 can be very nearly achieved. Chapters 12-15 finish the text with decoding and coding methods for the multi-user channel.

8.1 Encoding into Sequences

The concept of an encoder first appeared in Chapter 1, which emphasized modulators and simple signal sets. An encoder translates the incoming message at time index k, $m_k = 0, ..., M-1$, into the symbol vector $\boldsymbol{x}_k$ that a modulator subsequently converts into a modulated waveform. Most encoders of Chapter 1 had simple "memory-less" data-bits to one- or two-dimensional-symbol translations, e.g., the $x_k = 2m_k - (M-1)$ rule for PAM with $d = 2$. The **sequential encoder** of Figure 8.1 generalizes the encoder so that more formal study of channel coding can ensue. The more general sequential encoder may map message bits into larger dimensionality symbols that can also depend on previous message bits through the state of the encoder. The encoder is designed so that the **sequence** of transmitted symbols has some improved properties with respect to simple independent symbol transmission. The set of sequences that can be output by the encoder for all possible input messages bits is known as a **code** and the individual sequences are known as **codewords**.

8.1.1 The code and sequence

The concept of a code and the constituent sequences or codewords are fundamental to coding:

> **Definition 8.1.1 (Code)** *A* **code** *C is a set of one or more indexed (usually time-indexed is presumed but not required) sequences or* **codewords** *$\boldsymbol{x}_k$ formed by concatenating symbols from an encoder output. Each codeword in the code is uniquely associated with a sequence of encoder-input messages that exist over the same range of indices as the codeword.*

There is thus a one-to-one relationship between the strings of messages on the encoder input and the corresponding codewords on the encoder output in Figure 8.1. The codewords are converted into modulated waveforms by a modulator that is typically not of direct interest in coding theory – rather the designer attempts to design the encoder and consequent code to have some desirable improvement in transmission performance. Codewords are presumed to start at some index 0, and may be finite (and thus called a **block** code) or semi-infinite (and thus called a **tree** or **sliding block** code). Each codeword could be associated with a D transform $\boldsymbol{x}(D) = \sum_k \boldsymbol{x}_k \cdot D^k$ and the input message bit stream could have correspondingly $m(D) = \sum_k m_k \cdot D^k$ where the addition in the transform for the input message is modulo-M, and D is consequently viewed only as a variable indicating a delay of one symbol period. The entire code is then the set of codewords $\{\boldsymbol{x}(D)\}$ that corresponds to all possible input message sequences $\{m(D)\}$.

An example of a simple code would be the majority repetition code for binary transmission that maps a zero bit into the sequence -1 -1 -1 and maps the one bit into the sequence +1 +1 +1 with binary PAM modulation tacitly assumed to translate the 3 successive symbols in a codeword into modulated waveforms as if each successive dimension were a separate symbol for transmission. An ML decoder on the AWGN for this code essentially computes the majority polarity of the received signal. For such a code $\bar{b} = 1/3$ and the minimum distance is $d_{\text{min}} = 2\sqrt{3}$. This is an example of a block code. A simple tree code might instead transmit -1 -1 -1 if the input message bit at time k had changed with respect to time $k-1$ and +1 +1 +1 if there was no change in the input message bit with respect to the last bit. Decoding of such a tree code is addressed in Chapter 9. The tree code has essentially semi-infinite length codewords, while the closely related block code has length-3-symbol codewords. Description of the encoding function can sometimes be written simply and sometimes not so simply, depending on the code. The next subsection formalizes the general concept of a sequential encoder.

8.1.2 The Sequential Encoder

Figure 8.1 illustrates the sequential encoder. The sequential encoder has ν bits that determine its "state," s_k at symbol time k. There are 2^ν states, and the encoding of bits into symbols can vary with the encoder state. For each of the 2^ν possible states, the encoder accepts b bits of input (m_k), corresponding to $M = 2^b$ possible inputs, and outputs a corresponding N-dimensional output vector, $\boldsymbol{x}_k$. This process is repeated once every symbol period, T. The **data rate** of the encoder is

$$R \triangleq \frac{\log_2(M)}{T} = \frac{b}{T} \quad , \tag{8.1}$$

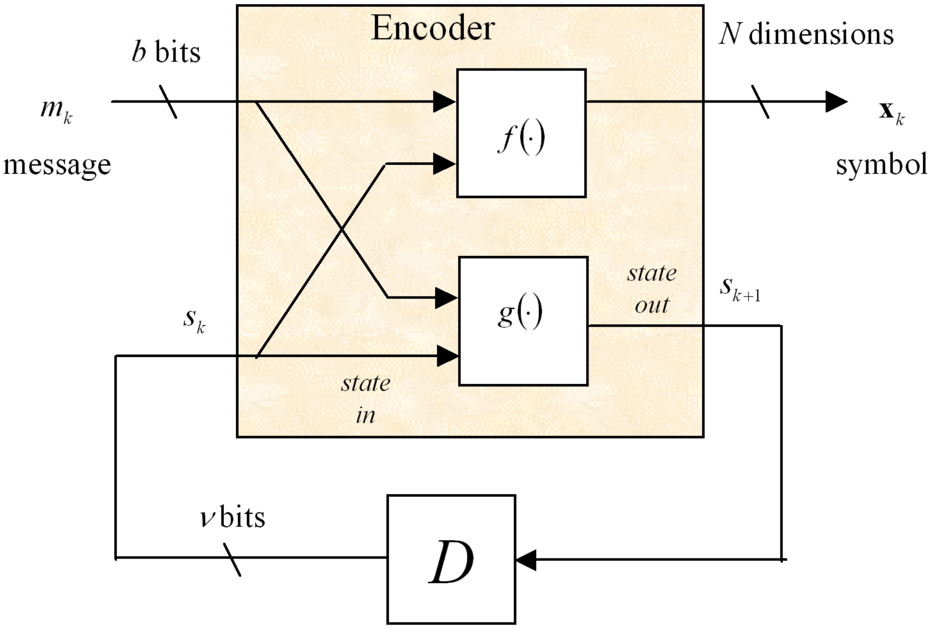

Figure 8.1: The Sequence Encoder.

where T is the symbol period. The number of bits per dimension is

$$\bar{b} \triangleq \frac{b}{N} \quad . \tag{8.2}$$

The output symbol value at symbol time k is $\boldsymbol{x}_k$ and is a function of the input message m_k and the channel state s_k:

$$\boldsymbol{x}_k = f(m_k, s_k) \tag{8.3}$$

and the next state s_{k+1} is also a function of m_k and s_k:

$$s_{k+1} = g(m_k, s_k) \quad . \tag{8.4}$$

The functions f and g are usually considered to be time-invariant, but it is possible that they are time-varying (but this text will not consider time-varying codes).

When there is only one state (ν=0 and $s_k = 0 \; \forall k$), the code is a block code. When there are multiple states ($\nu \geq 1$), the code is a tree code. An encoder description for any given code C may not be unique, although for time-invariant encoders an association of each codeword with each possible input sequence necessarily defines a unique encoder and associated mapping.

EXAMPLE 8.1.1 (QAM) *There is only one state in QAM.* Let $1/T = 2400$ Hz, and $\nu = 0$

- For 4 QAM, $R = 2/T = 2 \cdot 2400 = 4800$bps. $\bar{b}$ is then $2/2 = 1$ bit/dimension.
- For 16 QAM, $R = 4/T = 4 \cdot 2400 = 9600$bps. $\bar{b}$ is then $4/2 = 2$ bits/dimension.
- For 256 QAM, $R = 8/T = 8 \cdot 2400 = 19200$bps. $\bar{b}$ is then $8/2 = 4$ bits/dimension.

4, 16, and 256 QAM are all examples of block codes.

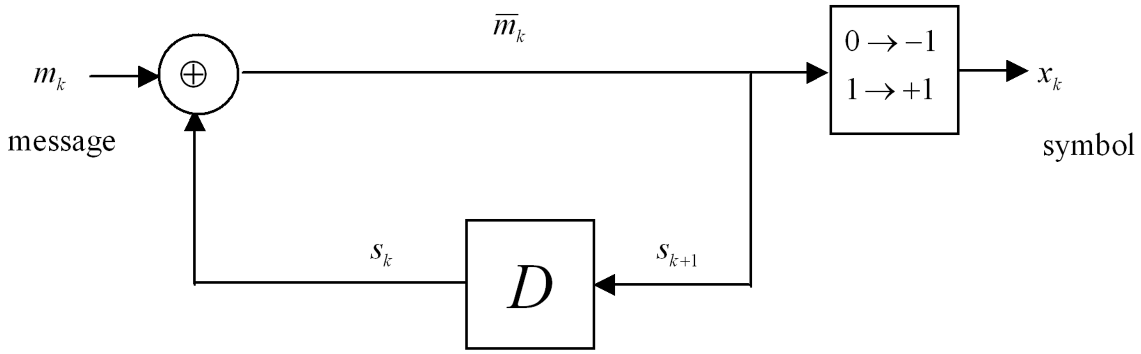

Figure 8.2: A binary PAM differential encoder.

EXAMPLE 8.1.2 (Differential Encoder) The differential encoder of Figure 8.2 has 2 states corresponding to the possible values for the previous single-bit message,

$$\bar{m}_k = m_k \oplus \bar{m}_{k-1} \quad , \tag{8.5}$$

in combination with the encoding rule $0 \rightarrow -1$ and $1 \rightarrow +1$. A differential encoder is an example of a sequential encoder with $\nu = 1$, $N = 1$, $\bar{b} = b = 1$, $m_k = 0, 1$, $s_k = 0, 1$ and $x_k = \pm 1$. This combination is a tree code. More generally, the differential encoder for M-ary PAM replaces the binary adder with a modulo-M adder in Figure 8.2 and the mapping on the right is a conventional $2m - (M-1)$ PAM encoder.

The binary differential encoder maps input bit sequences into output sequences where a change in the sequence at the current symbol time index corresponds to a message of "1" while no change corresponds to a message of "0". The state is simply the last message transmitted $s_k = \bar{m}_{k-1}$. The function f is $x_k = 2 \cdot (m_k \oplus s_k) - 1$. Thus, assuming the D to be initialized to 0, a string of message input bits of 1 1 0 1 0 1 leads to the output sequence or codeword +1 -1 -1 +1 +1 -1. Differential encoding allows a receiver to ignore the sign of the received signal. More generally, the differential encoder encodes the difference modulo-M between successive message inputs to the sequential encoder.

This book imposes the source constraint that there exists some stationary distribution for the N-dimensional output, $\mathrm{p}_{\boldsymbol{x}}(i)$. This is usually possible with systems in practice.

Several definitions are repeated here for convenience: the average energy per dimension is

$$\bar{\mathcal{E}}_{\boldsymbol{x}} = \frac{\mathcal{E}_{\boldsymbol{x}}}{N} \quad , \tag{8.6}$$

(which is not necessarily equal to the energy in any particular dimension). The **energy-per-bit** is

$$\mathcal{E}_b \triangleq \frac{\mathcal{E}_{\boldsymbol{x}}}{b} = \frac{\bar{\mathcal{E}}_{\boldsymbol{x}}}{\bar{b}} \tag{8.7}$$

Therefore,

$$\bar{\mathcal{E}}_{\boldsymbol{x}} = \frac{b}{N}\mathcal{E}_b = \bar{b}\mathcal{E}_b \quad . \tag{8.8}$$

The power is

$$P_x \triangleq \frac{\mathcal{E}_{\boldsymbol{x}}}{T} \quad , \tag{8.9}$$

and thus the energy per bit can be written as

$$\mathcal{E}_b = \frac{P_x}{R} \quad . \tag{8.10}$$

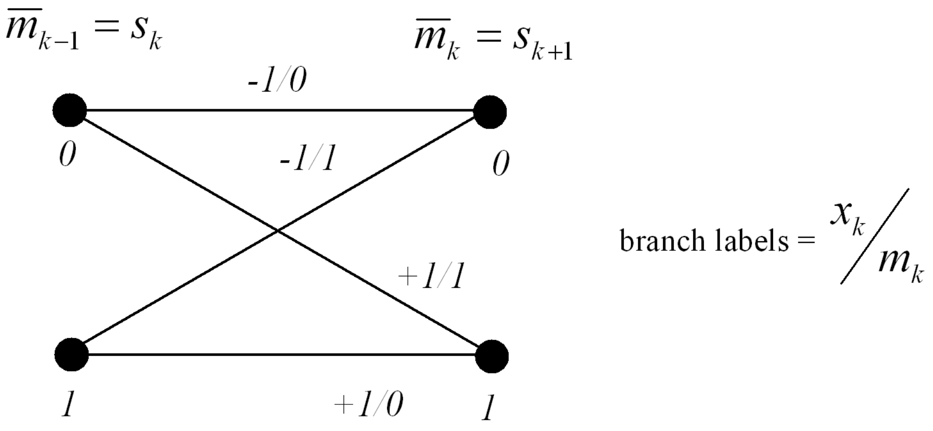
$\overline{m}_{k-1} = s_k$
$\overline{m}_k = s_{k+1}$
-1/0
0
-1/1
0
+1/1
1
+1/0
1
branch labels = x_k / m_k

Figure 8.3: Binary PAM differential encoder trellis.

8.1.3 The Trellis

The "trellis" is a heavily used diagram in coding theory that describes the progression of symbols within a code. It is best illustrated by an example, and we choose the binary differential encoder trellis in Figure 8.3. At each time index k, the possible states of the trellis are indicated in a vertical array of dots, each dot corresponding to one potential value of the state. In the example, there are two states at each time corresponding to the value of $s_k = \bar{m}_{k-1}$. Time-invariant encoder description requires illustration in the trellis of only times k and $k+1$, because the set of states and possible transitions among the states does not change from time to time. A trellis **branch** connects two states and corresponds to a possible input – there are always 2^b branches emanating from any state.

Each branch in the trellis is labeled with the channel symbol and the corresponding input x_k/m_k. Because there are only two states in the example, it is possible to reach any state from any other state, but this is not necessarily true in all trellis diagrams as will be evident in later examples. Any codeword sequence is obtained by following a connected set of branches through the trellis. For example, Figure 8.4 illustrates some possible individual sequences that could occur based on selecting a connected sequence of branches through the trellis. The upper sequence corresponds to a codeword transmitted over 3 successive times k =0, 1, and 2. Nothing is transmitted at or after time 3 in the example, but of course the codeword could be extended to any length by simply following more branches at later times. The upper sequence corresponds to the output $x(D) = -1 + D - D^2$ and the encoder was presumed in state $\bar{m}_{-1} = 0$. The corresponding input bit sequence is $m(D) = D + D^2$. The lower sequence is an alternative of $x(D) = 1 - D - D^2$ and corresponds to the input $m(D) = D$ and a different starting state of $\bar{m}_{-1} = 1$. Usually, the encoder is initialized in one state that is known to both encoder and decoder, but the example here simply illustrates the process of translating the sequence of branches or transitions in the trellis into a codeword. A semi-infinite series of branches that correspond to the system starting in state zero and then always alternating would have transform $x(D) = 1/(1 + D)$.

For the binary differential encoder example, it is possible to see that to determine d_{min}, the designer need only find the two sequences through the trellis that have minimum separation. Those sequences would necessarily be the same for a long period before and after some period of time where the two sequences in the trellis diverged from a common state and then merged later into a common state. For the example, this corresponds to $d^2_{\text{min}} = 4 = (+1 - (-1))^2$. The binary differential encoder provides no increase in d_{min} with respect to uncoded PAM transmission – its purpose is to make the decoder insensitive to a sign ambiguity in transmission because changes decode to 1 and no change decodes to 0, without reference to the polarity of the received signal.

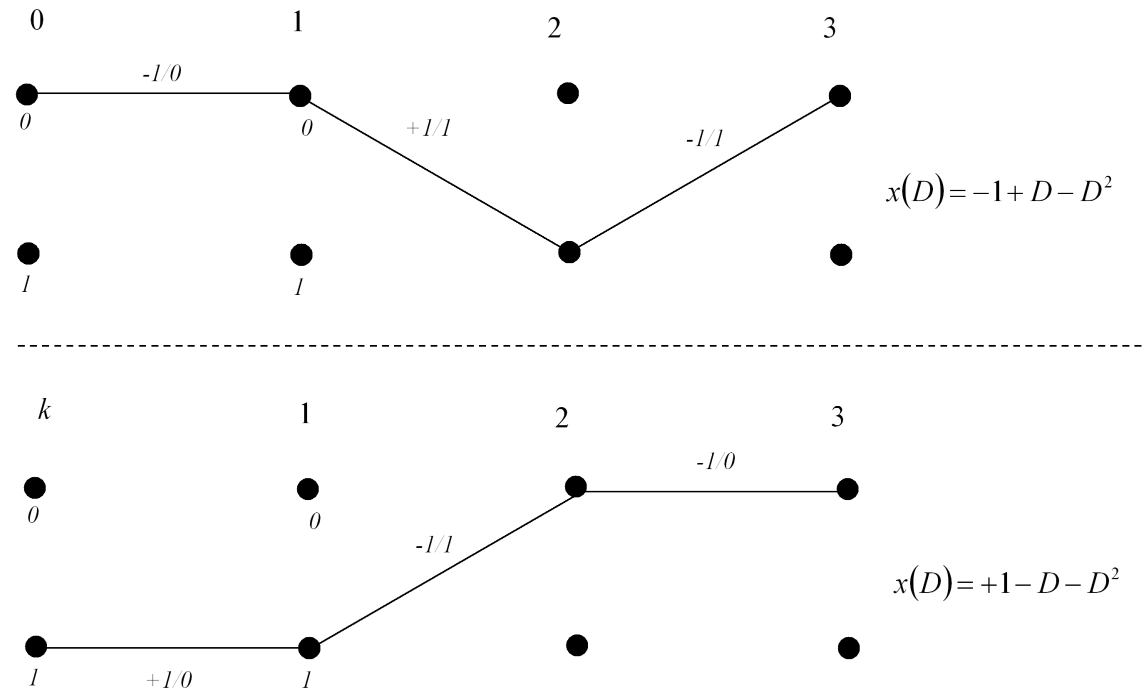

Figure 8.4: Example sequences for the differential encoder.

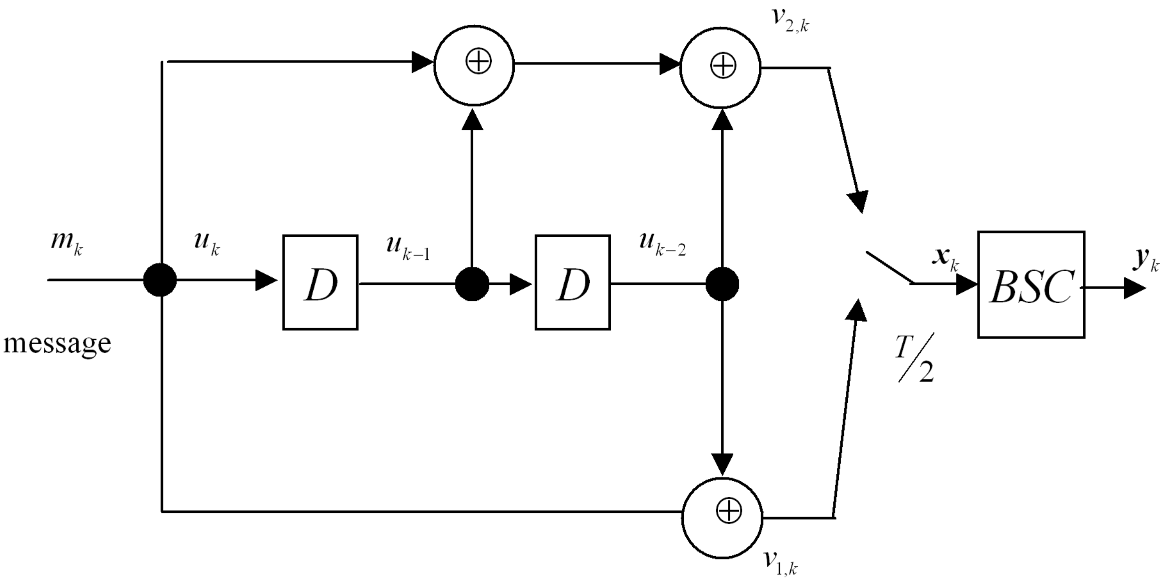

Figure 8.5: An example convolutional code with 4 states and $\bar{b} = .5$.

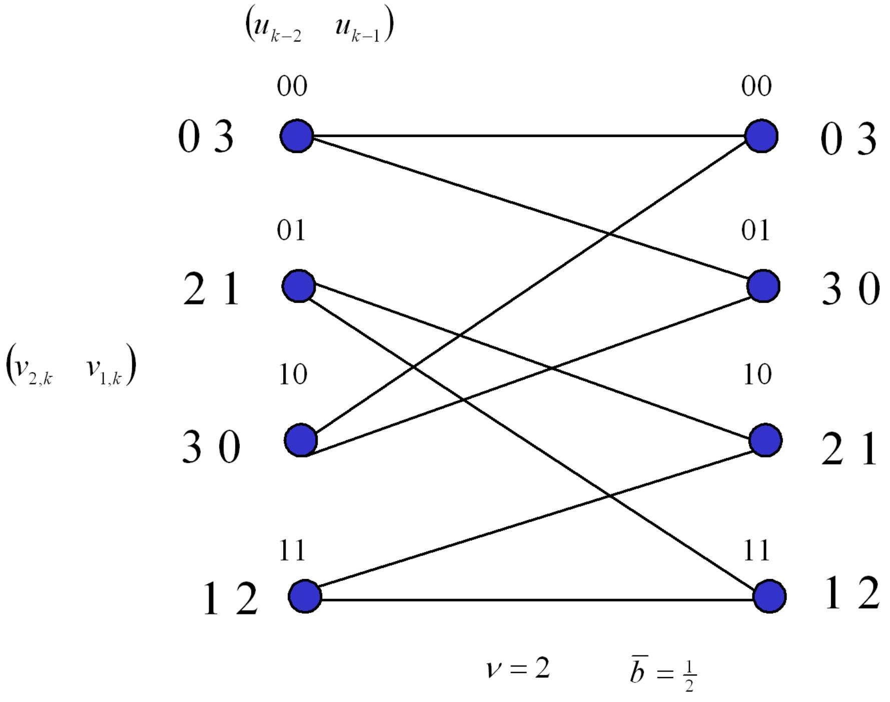

Figure 8.6: Trellis for convolutional code example.

8.1.4 Examples

EXAMPLE 8.1.3 (A simple convolutional code) The convolutional code in Figure 8.5 is based upon a modulo-2 linear combination of current and past input bits. This code's sequential encoder has one input bit and two output bits. The transformation shown is often written with $u(D) = m(D)$ and a "generator" matrix transformation $[v_2(D) \;\; v_1(D)] = u(D) \cdot [1 \oplus D \oplus D^2 \;\; 1 \oplus D^2]$ where the matrix on the right, $G(D) = [1 \oplus D \oplus D^2 \;\; 1 \oplus D^2]$, is often called the "generator matrix." The two output bits are successively transmitted through the channel, which in this example is the binary symmetric channel with parameter p. The parameter p is the probability that isolated and independent bits transmitted through the BSC are in error.

The encoder has 4 states represented by the 4 possible values of the ordered pair (u_{k-2}, u_{k-1}). The number of dimensions on the output symbol is $N = 2$ and thus $\bar{b} = 1/2$ because there is only one input bit per symbol period T.[1]

The trellis diagram for this code is shown in Figure 8.6. The branches are not labeled, and instead the convention is that the upper branch emanating from each state corresponds to an input bit of zero, while the lower branch corresponds to an input bit of one. The outputs transmitted for each state are listed in modulo-4 notation to the left of each starting state, with the leftmost corresponding to the upper branch and the rightmost corresponding to the lower branch. This code can be initialized to start in state 0 at time $k = 0$. Any sequence of input bits corresponds to a connected set of branches through the trellis.

For decoding of sequences on the BSC channel, a little thought reveals that the ML detector simply chooses that sequence of transitions through the trellis that differ least in the trellis-path bits $[v_2(D), v_1(D)]$ from the received two-dimensional output sequence bits $\boldsymbol{y}(D)$. The

[1] Convolutional code theorists often call $r = \bar{b}$ the rate of the code.

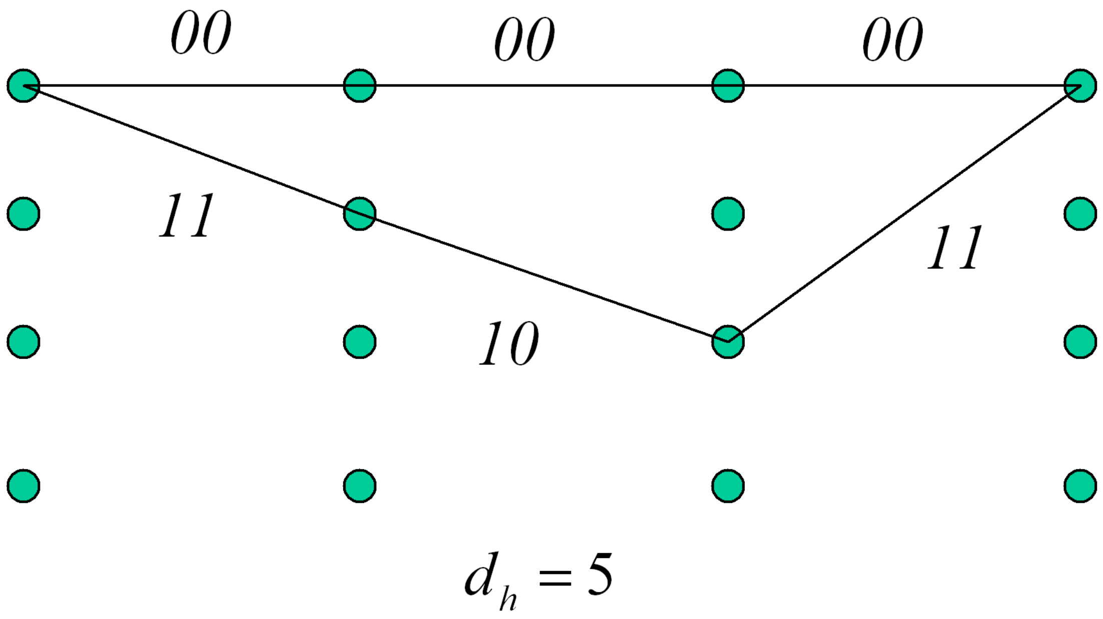

Figure 8.7: Minimum Hamming distance illustration for convolutional code example.

number of bit positions in which two sequences differ is often called the "Hamming distance." The two sequences that differ in the least number of positions (5) through the trellis can be seen in Figure 8.7. That means that at least 3 bit errors must occur in the BSC before those two sequences could be confused. Thus the probability of detecting an erroneous sequence will have a probability on the order of p^3, which for $p < .5$ means the convolutional code has improved the probability of error significantly (albeit at effectively half the bit rate of uncoded transmission).

EXAMPLE 8.1.4 (4-State Trellis Code Example) Trellis codes also use sequential encoding. A trellis for a 4-state encoder is shown in Figure 8.8 along with the 16 QAM signal constellation. This code has $b = 3$ bits per symbol so there are redundant or extra points in the constellation over the minimum needed for transmission. In fact, 16 points is double the 8 points of 8SQ QAM[2] that would be sufficient for "uncoded" QAM transmission. These points are exploited to improve the distance between possible sequences or codewords for this trellis code. The labels Λ_i, $i = 0, 1, 2, 3$ correspond to subsets, each with 4 points, of the 16 QAM constellation. Within each subset of points, the intra-subset minimum distance is large – in fact 3 dB larger than would be the distance for simple 8SQ transmission with $b = 3$. The lines shown in the trellis actually represent 4 parallel transitions between states, with one of the points in the indicated subset for the branch being chosen. Thus, the two lines diverging from each state really correspond to 8 branches, 2 sets of 4 parallel transitions. The particular point in a subset, or equivalently the symbol to be transmitted on each branch, is chosen by 2 input bits, while the 3rd bit specifies which of the two branch/sets to chose. This trellis is very similar to the earlier convolutional code, except that there are now parallel transitions and the interpretation of which point to transmit is different from simple binary PAM modulation of each bit in convolutional encoders.

The parallel transitions represent possible closest sequences that differ in only one symbol period. The corresponding minimum squared distance for these parallel-transition-closest

[2] 8SQ is another name for 8AMPM.

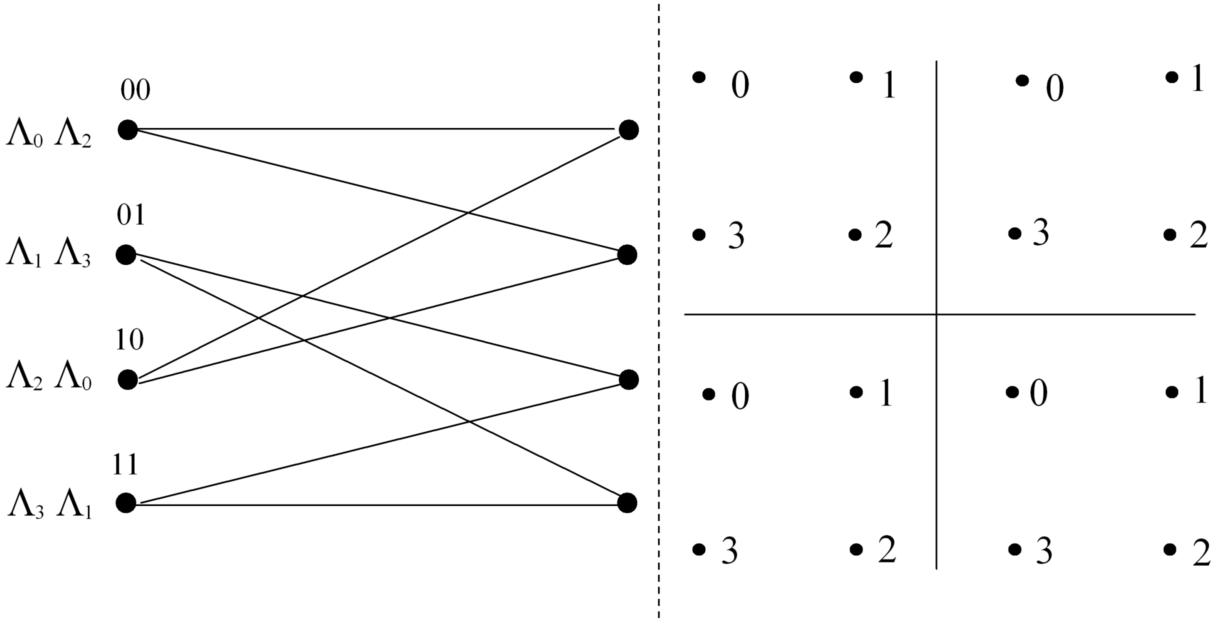

Figure 8.8: 4-State Ungerboeck Trellis Code.

sequences is 2 times greater than uncoded or 8SQ transmission. Another minimum-distance-separation possibility is for two sequences to differ in more than one symbol period. In this latter case, symbol points in different subsets diverging from or merging into any state are either chosen from the even subsets (Λ_0 and Λ_2) or the odd subsets (Λ_1 and Λ_3). Within the odds or the evens, the distance is the same as uncoded 8SQ. Thus diverging at one state and merging at another state forces the squared distance to be $d^2_{8SQ} + d^2_{8SQ} = 2d^2_{8SQ}$, which is also twice the squared distance of uncoded 8SQ transmission with the same transmitted energy.

This code is thus 3 dB better than uncoded 8SQ transmission! It is possible to implement a decoder with finite complexity for this code, as in Chapter 9. The extra 3 dB can be used to improve reliability (reduce probability of error) or increase the data rate. A little reflection suggests that this same code could be used with larger (and smaller) constellations with only the number of parallel transitions changing and no change in the 3 dB basic improvement of the code. While seemingly trivial, this 3 dB improvement escaped the notice of many researchers who were pursuing it after Shannon's work for over 30 years until Gottfried Ungerboeck of IBM found it in 1983. Trellis and the clearly related convolutional codes are studied further in Chapter 10.

EXAMPLE 8.1.5 (Duobinary Partial Response) The partial-response channels of Section 3.7 can be viewed as sequential encoders if the $H(D)$ is viewed as part of the encoder. Figure 8.9 illustrates the trellis for the $H(D) = 1 + D$ channel with binary PAM inputs of ±1 over 3 successive symbol periods beginning with time k. Possible sequences again correspond to connected paths through the trellis. In all such possible duobinary sequences a $+2$ can never follow immediately a -2, but rather at least one 0 must intervene. Thus, not all 3-level sequences are possible, which is not surprising because the input is binary. The ZF-DFE or precoding approach to a partial-response channel does not achieve the maximum performance level because the decoder acts in a symbol-by-symbol sub-optimum manner. Rather, a maximum-likelihood decoder could instead wait and compare all sequences. The closest two sequences are actually $d^2_{\text{min}} = 8$ apart, not $d^2_{\text{min}} = 4$ as with the precoder or DFE, a 3 dB improvement that achieves the matched-filter-bound performance level.

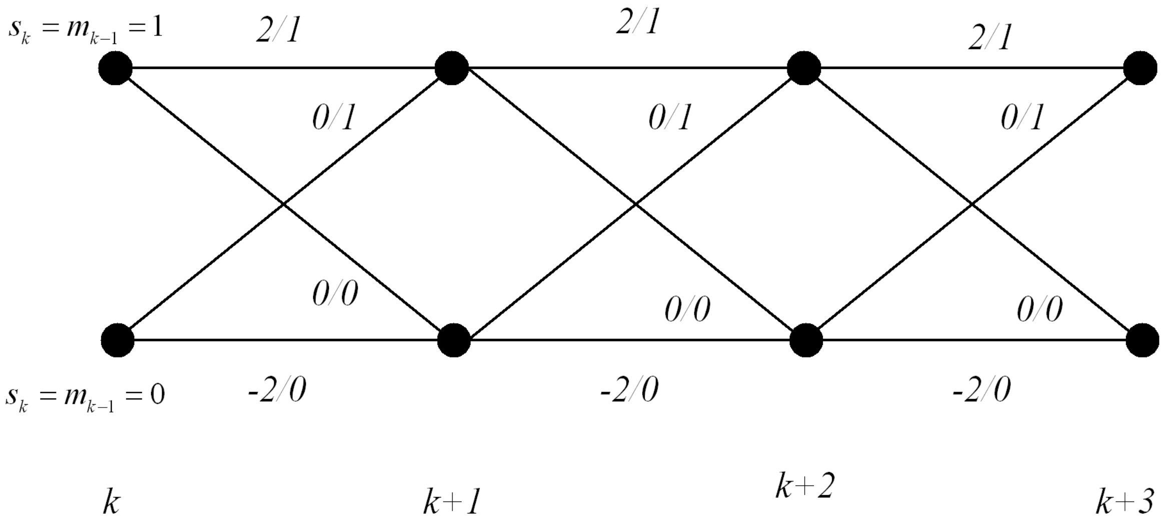

Figure 8.9: Trellis for a duobinary Partial Response Channel

Chapter 9 will introduce the maximum-likelihood sequence detector (MLSD) via the Viterbi Algorithm, which can be implemented with finite complexity on any trellis. For this duobinary partial-response channel, the matched-filter bound performance level can be attained with MLSD, nearly 3 dB better than the best receivers of Chapter 3 for binary transmission.

8.2 Measures of Information and Capacity

This section develops measures of information and bounds on the highest possible data rate that can be reliably transmitted over a channel.

8.2.1 Information Measures

A concept generalizing the number of input bits to an encoder, b, is the **entropy**. We re-emphasize the presumption that a stationary distribution for $\boldsymbol{x}$, $\mathrm{p}_{\boldsymbol{x}}(i)$, exists in N-dimensions. Information measures in this book use a base-2 logarithm and are measured in bits per symbol.[3]

Definition 8.2.1 (Entropy) *The* **entropy** *for an N-dimensional sequential encoder with stationary probability distribution $p_{\boldsymbol{x}}(i)$ $i = 0, ..., M-1$ is:*

$$\begin{aligned} H_{\boldsymbol{x}} &\stackrel{\Delta}{=} -\sum_{i=0}^{M-1} p_{\boldsymbol{x}}(i) \cdot \log_2 \left[p_{\boldsymbol{x}}(i) \right] \quad \textit{bits/symbol} & (8.11) \\ &= E \left\{ \log_2 \left[1/p_{\boldsymbol{x}} \right] \right\} \quad , & (8.12) \end{aligned}$$

The entropy $H_{\boldsymbol{y}}$ of the channel output is also similarly defined for discrete channel-output distributions.

The entropy of a random variable can be interpreted as the "information content" of that random variable, in a sense a measure of its "randomness" or "uncertainty." It can be easily shown that a discrete uniform distribution has the largest entropy, or information – or uncertainty over all discrete distributions. That is, all values are just as likely to occur. A deterministic quantity ($p_{\boldsymbol{x}}(i) = 1$ for one value of i and $p_{\boldsymbol{x}}(j) = 0 \ \forall \ j \neq i$) has no information, nor uncertainty. For instance, a uniform distribution on 4 discrete values has entropy $H_{\boldsymbol{x}} = 2$ bits/symbol. This is the same as b for 4-level PAM or 4 SQ QAM with uniform input distributions. The entropy of a source is the essential bit rate of information coming from that source. If the source distribution is not uniform, the source does not have maximum entropy and more information could have been transmitted (or equivalently transmitted with fewer messages M) with a different representation of the source's message set. Prior to this chapter, most of the message sets considered were uniform in distribution, so that the information carried was essentially b, the base-2 log of the number of messages. In general, $H_{\boldsymbol{x}} \leq \log_2(M)$, where M is the number of values in the discrete distribution. When the input is uniform, the upper bound is attained. Entropy for a discrete distribution is the same even if all the points are scaled in size as long as their probabilities of occurrence remain the same.

In the case of a continuous distribution, the **differential entropy** becomes:

$$H_{\boldsymbol{y}} \stackrel{\Delta}{=} -\int_{-\infty}^{\infty} p_{\boldsymbol{y}}(u) \cdot \log_2 \left[\mathrm{p}_{\boldsymbol{y}}(u) \right] du \quad . \tag{8.13}$$

Theorem 8.2.1 (Maximum Entropy of a Gaussian Distribution) *The distribution with maximum differential entropy for a fixed variance σ_y^2 is Gaussian.*

Proof: Let $g_y(v)$ denote the Gaussian distribution, then

$$\log_2 g_y(v) = -\log_2 \left(\sqrt{2\pi\sigma_y^2} \right) - \left(\frac{v}{\sqrt{2}\sigma_y} \right)^2 \bullet (\ln(2))^{-1} \quad . \tag{8.14}$$

For any other distribution $p_y(v)$ with mean zero and the same variance,

$$-\int_{-\infty}^{\infty} p_y(v) \log_2 \left(g_y(v) \right) dv = \log_2 \left(\sqrt{2\pi\sigma_y^2} \right) + \frac{1}{2\ln(2)} \quad , \tag{8.15}$$

[3] Any other base, p, could be used for the logarithm, and then the measures would be in the $p$its/symbol!

which depends only on σ_y^2. Then, letting the distribution for y be an argument for the entropy,

$$\begin{align}
H_y(g_y) - H_y(p_y) &= \tag{8.16}\\
&= -\int_{-\infty}^{\infty} g_y(v)\log_2(g_y(v))dv + \int_{-\infty}^{\infty} p_y(v)\log_2(p_y(v))dv \tag{8.17}\\
&= -\int_{-\infty}^{\infty} p_y(v)\log_2(g_y(v))dv + \int_{-\infty}^{\infty} p_y(v)\log_2(p_y(v))dv \tag{8.18}\\
&= -\int_{-\infty}^{\infty} p_y(v)\log_2\left(\frac{g_y(v)}{p_y(v)}\right)dv \tag{8.19}\\
&\geq \frac{1}{\ln 2}\int_{-\infty}^{\infty} p_y(v)\left(1-\frac{g_y(v)}{p_y(v)}\right)dv \tag{8.20}\\
&\geq \frac{1}{\ln 2}(1-1) = 0 \quad , \tag{8.21}
\end{align}$$

or[4]

$$H_y(g_y) \geq H_y(p_y) \quad . \tag{8.22}$$

QED.

With simple algebra,

$$H(g_y) = \frac{1}{2}\log_2(2e\pi\sigma_y^2) \quad . \tag{8.23}$$

For baseband complex signals, H_y, in bits per complex (two-dimensional) symbol is often written $H_y = \log_2\left(\pi e\sigma_y^2\right)$ where σ_y^2 becomes the variance of the complex random variable (which is twice the variance of the variance of the real part of the complex variable, when real and imaginary parts have the same variance, as is almost always the case in data transmission). The entropy per real dimension of complex and real Gaussian processes is the same if one recognizes in the formula that the variance of the complex process is double that of the real process.

8.2.2 Conditional Entropy and Mutual Information

Most of the uses of entropy in this text are associated with either the encoder distribution $\mathrm{p}_{\boldsymbol{x}}$, or with the channel output distributions $p_{\boldsymbol{y}}$ or $p_{\boldsymbol{y}/\boldsymbol{x}}$. The normalization of the number of bits per dimension to $\bar{b} = \frac{b}{N}$ tacitly assumes that the successively transmitted dimensions were independent of one another. In the case of independent successive dimensions, $H_{\boldsymbol{x}} = NH_x$. H_x is equal to $\bar{b}$, if the distribution on each dimension is also uniform (as well as independent of the other dimensions). Equivalently in (8.12), $H_{\boldsymbol{x}} = N \cdot H_x$ if each of the dimensions of $\boldsymbol{x}$ is independent.

For instance, 16QAM has entropy $H_{\boldsymbol{x}} = b = 4$ bits/symbol and normalized entropy $\bar{H}_x = \bar{b} = 2$ bits/dimension. However, 32CR has entropy $H_{\boldsymbol{x}} = b = 5$ bits/symbol, but since $p(\pm x = 1) = p(\pm x = 3) = 6/32$ and $p(\pm 5) = 4/32$, the entropy $H_x = 2.56$ bits/dimension $\neq \bar{H}_{\boldsymbol{x}} = \bar{b} = 2.5$ bits/dimension. Note also that the number of one-dimensional distribution values is 6, so $H_x = 2.56 < \log_2(6) = 2.58$.

The differential entropy of a (real) Gaussian random variable with variance $\bar{\sigma}^2$ is

$$H_x = \frac{1}{2}\log_2\left(2\pi e\bar{\sigma}^2\right) \quad \text{bits/symbol.} \tag{8.24}$$

A complex Gaussian variable with variance σ^2 has differential entropy

$$H_{\boldsymbol{x}} = \log_2\left(\pi e\sigma^2\right) \quad \text{bits/symbol.} \tag{8.25}$$

[4] Equation (8.21) uses the bound $\ln(x) \geq x - 1$.

The conditional entropy of one random variable given another is defined according to

$$H_{\boldsymbol{x}/\boldsymbol{y}} \stackrel{\Delta}{=} \sum_{\boldsymbol{v}} \sum_{i=0}^{M-1} \mathrm{p}_{\boldsymbol{x}}(i) \cdot \mathrm{p}_{\boldsymbol{y}/\boldsymbol{x}}(\boldsymbol{v}, i) \cdot \log_2 \frac{1}{\mathrm{p}_{\boldsymbol{x}/\boldsymbol{y}}(\boldsymbol{v}, i)} \tag{8.26}$$

$$H_{\boldsymbol{y}/\boldsymbol{x}} \stackrel{\Delta}{=} \sum_{\boldsymbol{v}} \sum_{i=0}^{M-1} \mathrm{p}_{\boldsymbol{x}}(i) \cdot \mathrm{p}_{\boldsymbol{y}/\boldsymbol{x}}(\boldsymbol{v}, i) \cdot \log_2 \frac{1}{\mathrm{p}_{\boldsymbol{y}/\boldsymbol{x}}(\boldsymbol{v}, i)} \quad , \tag{8.27}$$

with integrals replacing summations when random variables/vectors have continuous distributions. The definition of conditional entropy averages the entropy of the conditional distribution over all possibilities in some given input distribution. Thus, the conditional entropy is a function of both $p_{\boldsymbol{y}/\boldsymbol{x}}$ and $p_{\boldsymbol{x}}$. Conditional entropy measures the residual information or uncertainty in the random variable given the value of another random variable on average. This can never be more than the entropy of the unconditioned random variable, and is only the same when the two random variables are independent. For a communication channel, the conditional entropy, $H_{\boldsymbol{y}/\boldsymbol{x}}$, is basically the uncertainty or information of the "noise." If the conditional distribution is Gaussian, as is often the case in transmission, the conditional entropy of a scalar x becomes

$$H_{x/\boldsymbol{y}} = \begin{cases} \frac{1}{2}\log_2\left(2\pi e \sigma^2_{mmse}\right) & \text{real } x \\ \log_2\left(\pi e \sigma^2_{mmse}\right) & \text{complex } x \end{cases} \tag{8.28}$$

The MMSE in the above equations is that arising from minimum-mean-square-error estimation of x, given $\boldsymbol{y}$. Thus, the conditional entropy then measures the information remaining after the effect of the $\boldsymbol{y}$ has been removed. That is in some sense measuring useless information that a receiver might not be expected to use successfully in estimating $\boldsymbol{x}$.

The information of the "noise," or more generically, the "useless part" of the channel output given a certain input distribution, is not of value to a receiver. Thus, while the entropy of the source is a meaningful measure of the data transmitted, the entropy of the channel output has extra constituents that are caused by the randomness of noise (or other useless effects). Given that it is the output of a channel, $\boldsymbol{y}$, that a receiver observes, only that part of the output that bears the information of the input is of value in recovering the transmitted messages. Thus, the entropy $H_{\boldsymbol{y}} - H_{\boldsymbol{y}/\boldsymbol{x}}$ measures the useful information in the channel output. This information is called the **mutual information**.

> **Definition 8.2.2 (Mutual Information)** *The* **mutual information** *for any N-dimensional signal set with probability distribution $p_{\boldsymbol{x}}(i)$ $i = 0, ..., M-1$, and a corresponding channel description $p_{\boldsymbol{y}/\boldsymbol{x}}(\boldsymbol{v}, i)$, is:*
>
> $$I_{\boldsymbol{y},\boldsymbol{x}} \stackrel{\Delta}{=} H_{\boldsymbol{y}} - H_{\boldsymbol{y}/\boldsymbol{x}} \quad . \tag{8.29}$$

The identity,

$$I_{\boldsymbol{y},\boldsymbol{x}} = H_{\boldsymbol{y}} - H_{\boldsymbol{y}/\boldsymbol{x}} = H_{\boldsymbol{x}} - H_{\boldsymbol{x}/\boldsymbol{y}} \quad , \tag{8.30}$$

easily follows from transposing $\boldsymbol{x}$ and $\boldsymbol{y}$. Using probability distributions directly:

$$I_{\boldsymbol{y},\boldsymbol{x}} \stackrel{\Delta}{=} \sum_{\boldsymbol{v}} \sum_{i=0}^{M-1} p_{\boldsymbol{x}}(i) p_{\boldsymbol{y}/\boldsymbol{x}}(\boldsymbol{v}, i) \log_2 \left[\frac{p_{\boldsymbol{y}/\boldsymbol{x}}(\boldsymbol{v}, i)}{\sum_{m=0}^{M-1} p_{\boldsymbol{y}/\boldsymbol{x}}(\boldsymbol{v}, m) p_{\boldsymbol{x}}(m)} \right] \quad \text{bits/symbol} \tag{8.31}$$

$$= E \log_2 \left[\frac{p_{\boldsymbol{y}/\boldsymbol{x}}}{\mathrm{p}_{\boldsymbol{y}}} \right] \tag{8.32}$$

$$= E \log_2 \left[\frac{p_{\boldsymbol{y},\boldsymbol{x}}}{p_{\boldsymbol{x}} \cdot p_{\boldsymbol{y}}} \right] \tag{8.33}$$

$$= E \log_2 \left[\frac{p_{\boldsymbol{x}/\boldsymbol{y}}}{p_{\boldsymbol{x}}} \right] \quad , \tag{8.34}$$

In the case of a continuous distribution on y and/or x, the summation(s) is (are) replaced by the appropriate integral(s).

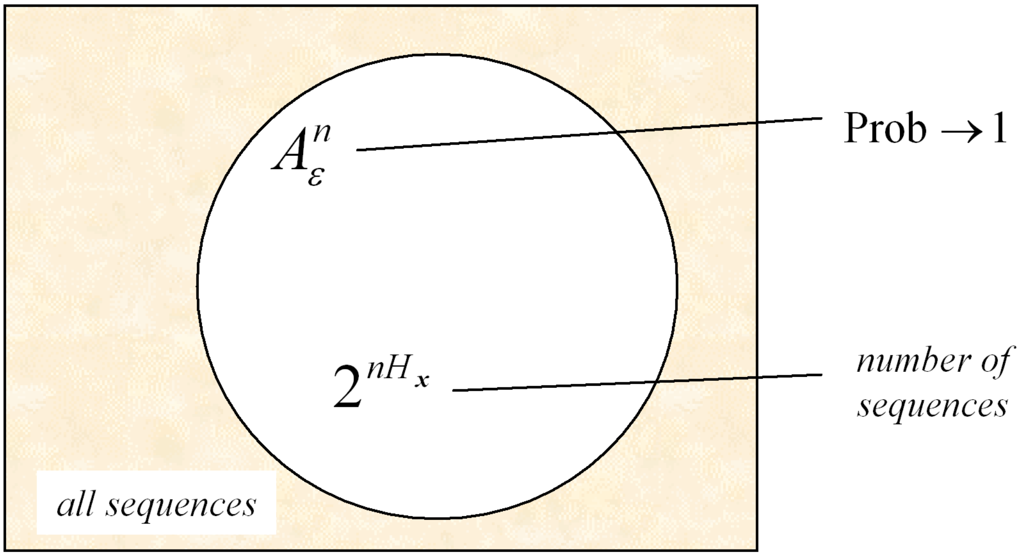

Figure 8.10: Illustration of sequence space and set of equal probability.

8.2.3 Asymptotic Equipartition

The concept of **asymptotic equipartition** addresses the selection of symbols in a sequence of length n from a stationary distribution. As the sequence length grows, there exists a set of "typical" sequences, A_ϵ^n that has probability approaching 1. Each sequence within this set is equally likely and has probability close to $2^{-nH_{\boldsymbol{x}}}$, meaning there are approximately $2^{nH_{\boldsymbol{x}}}$ sequences in this set. Other sequences are essentially not important in that their probability tends to zero.

More formally,

Definition 8.2.3 (typical set) *A* **typical set** *of length-n sequences for a stationary (over time index n) sequential encoder with symbols* $\boldsymbol{x}$ *and entropy* $H_{\boldsymbol{x}}$ *is defined by*

$$A_\epsilon^n \triangleq \left\{ \boldsymbol{x} \mid 2^{-nH_{\boldsymbol{x}}-\epsilon} \leq p(\boldsymbol{x}_0, \boldsymbol{x}_1, ..., \boldsymbol{x}_{n-1}) \leq 2^{-nH_{\boldsymbol{x}}+\epsilon} \right\} \tag{8.35}$$

for any $\epsilon > 0$.

A famous lemma given without proof here is the AEP Lemma:

Lemma 8.2.1 (AEP Lemma) *For a typical set with* $n \to \infty$*, the following are true:*

- $Pr\{A_\epsilon^n\} \to 1$
- *for any sequence* $\boldsymbol{x}(D) \in A_\epsilon^n$*,* $Pr\{\boldsymbol{x}(D)\} \to 2^{-nH_{\boldsymbol{x}}}$

The proof of this lemma follows the argument of this paragraph. Figure 8.10 illustrates the concept with a Venn diagram where the rectangle represents the space of all possible sequences, and the circle within represents a typical set. This typical set tends to dominate the probability as the sequences get longer and longer, and each sequence within this set is equally likely. Trivially if the sequence values are independent and uniform, then each length-n sequence has a probability of $2^{-nH_{\boldsymbol{x}}}$. However, when independent but non uniform, for instance a binary random variable with $p_1 = .9$ and $p_0 = .1$, then $H_x = .469$ bits/dimension, then all sequences have a probability that is the product of n terms of either .1 or .9. Since this decays exponentially to zero, as does also $2^{-nH_{\boldsymbol{x}}} = .722^n$, one can always pick n sufficiently large that any sequence is typical. Sequence probabilities that would not decay exponentially would be difficult to envision. In effect, the stationary presumption of entropy $H_{\boldsymbol{x}}$ for each successive symbol forces the exponential decay and for sufficiently large n all sequences (with increasingly small probabilities) will be close to the almost zero $2^{-nH_{\boldsymbol{x}}}$. Thus, the AEP essentially is a trivial statement

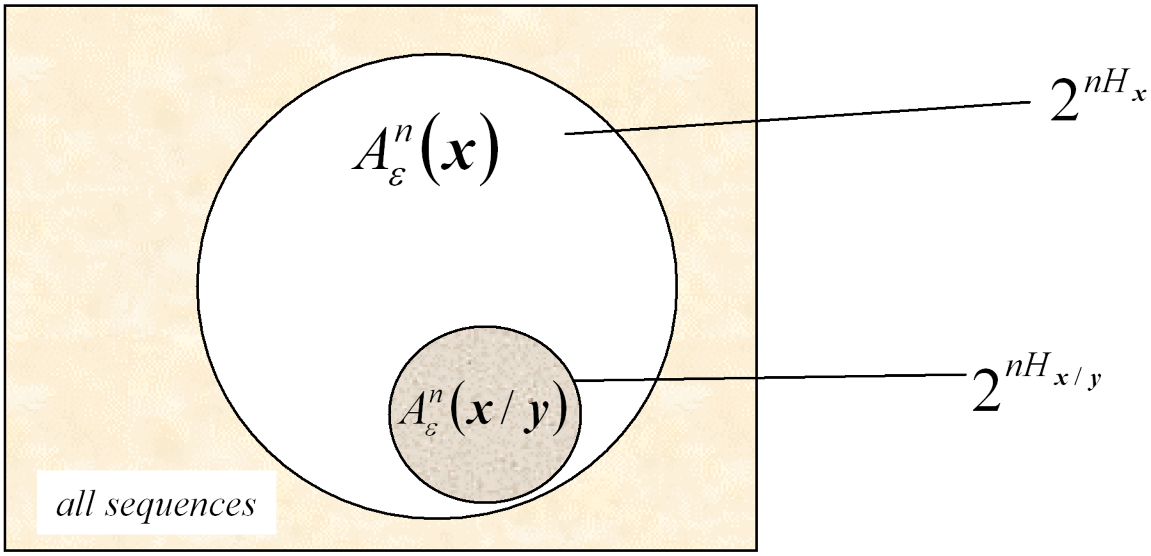

Figure 8.11: Illustration of sets corresponding to $\boldsymbol{x}$ and $\boldsymbol{x}$ conditioned on $\boldsymbol{y}$.

of the stationarity and the exponential decay of any set of encoder symbol values that have probabilities between 0 and 1. If one $\boldsymbol{x}$ value had probability 1 so that $H_{\boldsymbol{x}} = 0$, then its repeated-value sequence would trivially be typical. Sequences having a symbol with zero probability somewhere within would be an example of a non-typical sequence.

Figure 8.11 illustrates that on average the typical set for $\boldsymbol{x}$ given $\boldsymbol{y}$ is smaller (because the entropy is less). Points within the smaller set can be viewed as indistinguishable from one another given $\boldsymbol{y}$ – they are all equally likely to occur given $\boldsymbol{y}$. Thus a MAP receiver would not be able to avoid error well if two possible code sequences were in this smaller set, essentially being forced to resolve a tie by randomly picking any of the multiple points in the set. Good code design then attempts to avoid having more than one codeword in any independent set of size $2^{nH_{\boldsymbol{x}/\boldsymbol{y}}}$. Since there are $2^{nH_{\boldsymbol{x}}}$ sequences as n gets large, the code designer would like to pick code sequences from this larger set so that there is only one in each possible subset of size $2^{nH_{\boldsymbol{x}/\boldsymbol{y}}}$. Thus, if well done, the largest number of distinguishable code sequences is

$$2^{nI(\boldsymbol{x};\boldsymbol{y})} = \frac{2^{nH_{\boldsymbol{x}}}}{2^{nH_{\boldsymbol{x}/\boldsymbol{y}}}} \quad . \tag{8.36}$$

Furthermore, sequences that fall in different sets of size $2^{nH_{\boldsymbol{x}/\boldsymbol{y}}}$ can be distinguished from sequences outside this set with probability tending to one since the AEP Lemma establishes that sequences within the set have probability tending to one. Thus for any given value of $\boldsymbol{y}$, on average, any point outside the set would be eliminated by a decoder from further consideration. This essentially means that a decoder would have a probability of sequence-decision error that tends to zero as long as no two codewords come from the same set. Thus, as $n \to \infty$, $I(\boldsymbol{x};\boldsymbol{y})$ represents the maximum number of bits per symbol that can be reliably transmitted over the communication channel. That is,

$$b \leq I(\boldsymbol{x};\boldsymbol{y}) \quad . \tag{8.37}$$

8.2.4 The Channel Capacity Theorem

The channel capacity is a measure of the maximum data rate that can be transmitted over any given channel reliably. The mutual information essentially measures this data rate (in bits/symbol) for any given input distribution, presuming that an engineer could design a transmitter and receiver that allows the information in the channel output about the channel input to be mapped into the corresponding transmitted messages with sufficiently small probability of error, which is possible given the AEP interpretation. Different input distributions can lead to different mutual information. The best input would then be that which maximizes the mutual information, a concept first introduced by Shannon in his 1948 paper:

Definition 8.2.4 (Channel Capacity) *The channel capacity for a channel described by* $p_{\boldsymbol{y}/\boldsymbol{x}}$ *is defined by*

$$C \triangleq \max_{p_{\boldsymbol{x}}} I(\boldsymbol{y}, \boldsymbol{x}) \quad , \tag{8.38}$$

which is measured in bits per N*-dimensional channel input symbol.*

It is sometimes convenient to normalize C to one dimension, by dividing C by N,

$$\bar{C} \triangleq \frac{C}{N} \quad , \tag{8.39}$$

which is in bits/dimension. The evaluation of the expression in (8.38) can be difficult in some cases, and one may have to resort to numerical techniques to approximate the value for C. Any input constraints are tacitly presumed in the choice of the input vectors $\boldsymbol{x}$ and affect the value computed for capacity. Given the definition of mutual information, Shannon's [1948 BSTJ] famed channel coding theorem is:

Theorem 8.2.2 (The Channel Capacity Theorem) *Given a channel with capacity* C*, then there exists a code with* $\bar{b} < \bar{C}$ *such that* $P_e \leq \delta$ *for any* $\delta > 0$*. Further, if* $\bar{b} > \bar{C}$*, then* $P_e \geq$ *positive constant, which is typically large even for* b *slightly greater than* $\bar{C}$*.*

proof: See the discussion surrounding Equations (8.36) and (8.37).

The desired interpretation of this theorem is that reliable transmission can only be achieved when $\bar{b} < \bar{C}$, or equivalently $b < C$.

The capacity for the complex AWGN is probably the best known and most studied. It is determined from

$$I(x; y) = H_x - H_{x/y} = H_x - log_2(\pi e \sigma^2_{mmse}) \tag{8.40}$$

which is maximized when H_x is maximum, which means a Gaussian distribution. Thus,

$$C_{awgn} = log_2(\pi e \mathcal{E}_x) - log_2(\pi e \sigma^2_{mmse}) = \log_2(1 + \text{SNR}_{unbiased}) = \log_2(1 + \text{SNR}) = \max\left(H_y - H_{y/x}\right) \quad . \tag{8.41}$$

Or, for any AWGN (with no ISI), the capacity in bits/dimension is

$$\bar{C} = \frac{1}{2}\log_2(1 + \text{SNR}) \quad . \tag{8.42}$$

8.2.5 Random Coding

A second result that follows the AEP Lemma is that almost any code chosen at random from the best probability distribution will be a capacity achieving code. The random-code construction process assumes that some probability distribution $p_{\boldsymbol{x}}$ is being used (if it is the distribution that achieves capacity, then the data rate is highest; otherwise the highest data rate is the mutual information corresponding to the selected probability distiribution). Each element of a sequence $\boldsymbol{x}(D)$ is selected at random from this distribution to construct the successive values in a sequence. All codewords are so selected at random. If the codeword length n goes to infinity, then the probability tends to one that the code is a good code that achieves zero error probability at the mutual-information data rate that corresponds to the $p_{\boldsymbol{x}}$ used to generate the random code.

The proof of this stunning result again evokes the AEP Lemma: Each selected codeword (sequence) is typically as the block length increases. Furthermore, each of the sets corresponding to the density $p_{\boldsymbol{x}/\boldsymbol{y}}$ has equal size ($2^{nH_{\boldsymbol{x}/\boldsymbol{y}}}$). Thus, any codeword generated at random is thus equally likely to be in any of the $2^{nI(\boldsymbol{x};\boldsymbol{y})}$sets of size $2^{nH_{\boldsymbol{x}/\boldsymbol{y}}}$. Applying the AEP now at a higher level to the random variable associated with "which set is the random codeword/sequence in?" suggests that with probability tending to one, this process of random codeword generated will produce codes or groups of sequences that have one and only codeword in each of the sets.

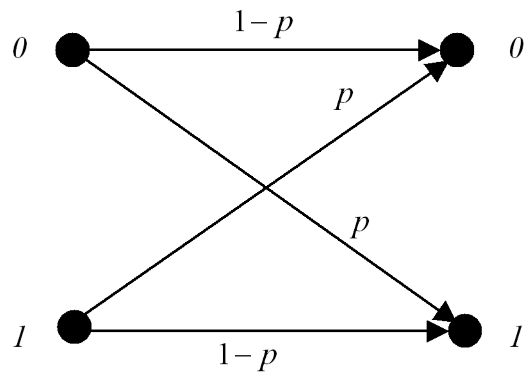

Figure 8.12: The Binary Symmetric Channel, BSC.

8.3 Capacity Calculation for Discrete Channels

This section focuses on a few commonly encountered channel types and the computation of the capacity. Only vector channels are considered, with continuous waveform channels being deferred to the next section.

8.3.1 Discrete Memoryless Channels (DMC)

The discrete memoryless channel was introduced earlier in Chapter 1; both the inputs and the outputs of the DMC are members of discrete finite sets. There are M inputs, $\boldsymbol{x}_0$, ... , $\boldsymbol{x}_{M-1}$ and J outputs $\boldsymbol{y}_0$,..., $\boldsymbol{y}_{J-1}$. The term "memoryless" means that the outputs on the next use of the channel are independent of any previous inputs.

The **binary symmetric channel** (BSC) is probably the most widely cited DMC, and has the probability transition diagram in Figure 8.12. While appearances are similar, this diagram is not a trellis diagram, but rather describes the probability that a certain output is received, given any particular input. Coding theorists interested in error-control codes use this channel to characterize a system such as the equalized systems of Chapter 3, where the engineer focusing on the outer error-control codes would simply view the modulation, equalization, and detection process as an entity from a higher level and just assign (for the BSC)

$$p \triangleq \bar{P}_b = \frac{N_b}{b} \cdot Q(\frac{d_{\min}}{2\sigma}) \quad . \tag{8.43}$$

Such a system is sometimes said to use "hard" decoding, meaning that a decision is made on each symbol before any outer decoding is applied.

The capacity of the BSC can be computed in a straightforward manner by substitution into the mutual information formula:

$$\begin{aligned}
I_{x,y} &= \sum_{m=0}^{1}\sum_{j=0}^{1} p_{\boldsymbol{x}}(m) p_{\boldsymbol{y}/\boldsymbol{x}}(j,m) \log_2\left(\frac{p_{\boldsymbol{y}/\boldsymbol{x}}(j,m)}{p_{\boldsymbol{y}}(j)}\right) && (8.44)\\
&= p_{\boldsymbol{x}}(0)(1-p)\log_2\left(\frac{1-p}{p_{\boldsymbol{x}}(0)(1-p)+p_{\boldsymbol{x}}(1)p}\right) && (8.45)\\
&+ p_{\boldsymbol{x}}(0)(p)\log_2\left(\frac{p}{p_{\boldsymbol{x}}(0)p+p_{\boldsymbol{x}}(1)(1-p)}\right) && (8.46)\\
&+ p_{\boldsymbol{x}}(1)(p)\log_2\left(\frac{p}{p_{\boldsymbol{x}}(0)(1-p)+p_{\boldsymbol{x}}(1)p}\right) && (8.47)\\
&+ p_{\boldsymbol{x}}(1)(1-p)\log_2\left(\frac{1-p}{p_{\boldsymbol{x}}(0)p+p_{\boldsymbol{x}}(1)(1-p)}\right) && (8.48)
\end{aligned}$$

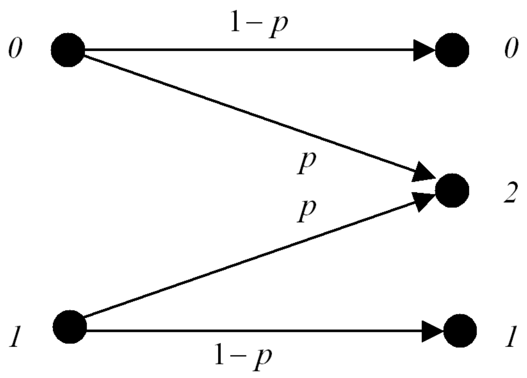

Figure 8.13: The Binary Erasure Channel, BEC.

The input probabilities $p_{\boldsymbol{x}}(0)$ and $p_{\boldsymbol{x}}(1)$ are interchangeable in the above expression. Thus, the maximum must occur when they are equal; $p_{\boldsymbol{x}}(0) = p_{\boldsymbol{x}}(1) = .5$. Then,

$$\begin{aligned} C &= (1-p)\log_2\left[2(1-p)\right] + p\log_2(2p) & (8.49) \\ &= 1 - \mathcal{H}(p) \quad , & (8.50) \end{aligned}$$

where

$$\mathcal{H}(p) \stackrel{\Delta}{=} -p\log_2(p) - (1-p)\log_2(1-p) \quad , \tag{8.51}$$

the entropy of a binary distribution with probabilities p and $1-p$. As $p \to 0$, there are no errors made and $C = \bar{C} \to 1$ bit/symbol (or bit/dimension), otherwise $C \leq 1$ for the BSC.

A second commonly encountered channel is the **binary erasure channel** (BEC), shown in Figure 8.13. The channel is again symmetric in $p_{\boldsymbol{x}}$, so that the maximizing input distribution for the mutual information is $p_{\boldsymbol{x}}(0) = p_{\boldsymbol{x}}(1) = .5$. The capacity is then

$$\begin{aligned} C &= \left[\frac{1}{2}(1-p)\log_2\frac{1-p}{\frac{1}{2}(1-p)}\right]2 + \left[\frac{1}{2}p\log_2\frac{p}{2p\frac{1}{2}}\right]2 & (8.52) \\ &= 1-p & (8.53) \end{aligned}$$

Again, as $p \to 0$, there are no errors made and $C \to 1$ bit/symbol, otherwise $C \leq 1$ for the BEC. When $p \leq 0.5$, then $C_{BEC} \geq C_{BSC}$, which is explored more in Problem 8.11.

More generally, the symmetric DMC may have an $M \times J$ matrix of transition probabilities such that every row is just a permutation of the first row and every column is a permutation of the first column. For instance, the BSC has

$$\begin{bmatrix} 1-p & p \\ p & 1-p \end{bmatrix} \tag{8.54}$$

For the symmetric DMC, the maximizing distribution can be easily shown to be uniform for both C. A special case of interest is the **Universal Discrete Symmetric Channel** has 2^b discrete inputs and the same set of 2^b outputs. The probability of the output being the same as the input is $(1-p_s)$ while the probability of being any of the other possible values is equally likely at $p_s/(2^b-1)$. Because of the symmetry, the maximizing input distribution is again uniform amound the 2^b possible discrete messages. The capacity is

$$C = b - p_s \cdot \log_2\frac{2^b-1}{p_s} + (1-p_s)\cdot\log_2(1-p_s) \leq b \text{ bits.} \tag{8.55}$$

A typical use of this channel is when $b = 8$ or the transmission system is organized to carry an integer number of bytes of information. If the UDSC is constructed from 8 successive uses of the BSC, then

$$p_s = 1 - (1-p)^b \approx b \cdot p = 8p \text{ for small } p \quad . \tag{8.56}$$

Outer codes may be then organized in terms of byte symbols or modulo-256 arithmetic (or more generally modulo-2^b arithmetic).

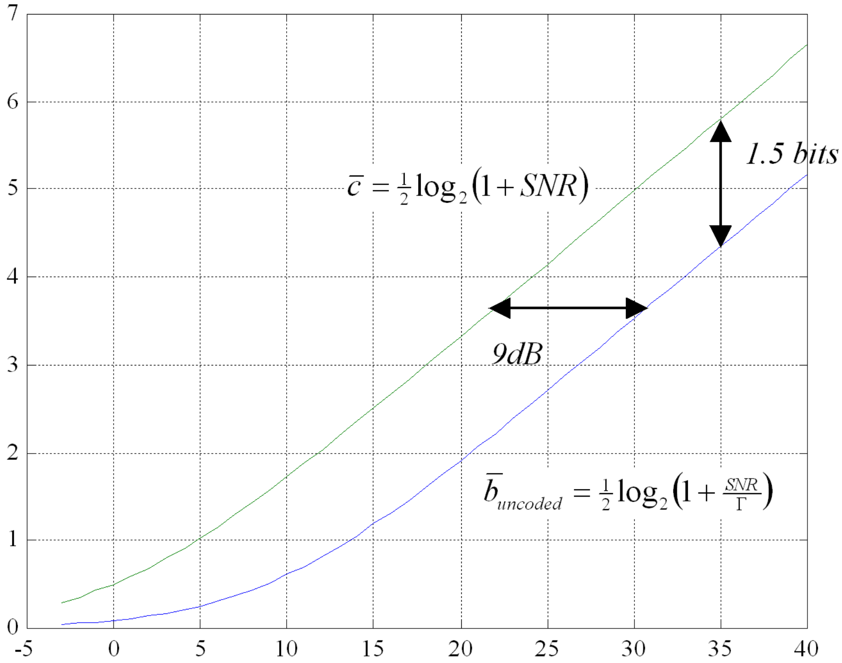

Figure 8.14: Illustration of capacity and gap to uncoded transmission with $\Gamma = 8.8$ dB.

8.3.2 Capacity, Coding, and the Gap

The capacity for the AWGN has been previously computed as $\bar{c} = .5 \cdot log_2(1+\text{SNR})$ and appears in Figure 8.14. Note this semi-log plot shows that as SNR becomes reasonably large (say 20 dB or more), that increase of capacity by 1 bit per dimension requires and additional 6 dB of SNR. Since QAM is the most heavily used transmission format footnoteWithin most coded systems including multicarrier systems like the DMT or GDFE of Chapters 4 and 5, QAM is a consitituent of the codes used so this statement is not an endorsement of wideband QAM use on an ISI channel with notches., a bit per dimension of QAM corresponds to two bits per symbol or an often quoted rule of "3 dB per bit". At low SNR (below 10 dB), this rule no longer applies and for very low SNR (below 0 dB), capacity essentially scales linearly with SNR (instead of logarithmically). This is evident in that

$$\lim_{\text{SNR}\to 0} \bar{c} = \frac{.5}{\ln 2}\text{SNR} \quad . \tag{8.57}$$

since $(log(1+x) \approx x$ for small $x)$.

Recall that for $\bar{b} \geq 1$ in Chapter 1, data rates for PAM and QAM transmission were found experimentally with heuristic theoretical justification to satisfy the formula:

$$\bar{b} = .5\log(1+\frac{\text{SNR}}{\Gamma}) = \bar{c}(\frac{\text{SNR}}{\Gamma}) \quad . \tag{8.58}$$

That is the gap approximation used heavily for uncoded transmission is exactly the capacity formula with the SNR reduced by $\Gamma \geq 1$. For uncoded PAM and QAM transmission and a $\bar{P}_e \approx 10^{-6}$, the gap is a constant 8.8 dB. The gap formula is also plotted in Figure 8.14 where the constant gap is illustrated, and the constant bit/dimension gap of 1.5 bits/dimension is also illustrated.

This formula resemblence suggests that at least at a probability of error of 10^{-6}, and reasonably good SNR, the uncoded designs of Volume 1 are about 9 dB of SNR short of capacity. In terms of coded performance, capacity relates that good codes exist over infinite-length sequences (or very long symbol blocks) that can provide up to 9 dB more coding gain at 10^{-6}. Capacity suggests there may be many such codes – Chapters 10 and 11 will find some. Codes that are constructed from PAM and QAM like constellations can recover this lost 9 dB of uncoded transmission. The price will not be a higher error rate (and indeed the probability of error can be driven to zero from the capacity theorem), nor transmit energy increase, but rather a signficantly more complex encoder, and especially, a more complex decoder. The use of 10^{-6} and the corresponding gap of 8.8 dB may seem somewhat arbitrary – one could argue why not 10^{-7} or smaller, where the corresponding larger gaps would suggest yet even higher than 9 dB improvement in SNR is possible. As this text proceeds, it will become increasingly clear that once the probability of error is less than 10^{-6}, an outer concatenated code working on the presumption that the inner AWGN has been well handled and converted to a BSC with probability $p = 10^{-6}$, can easily drive overall probability of bit error close to zero with little data rate loss, so 10^{-6} is an often used design figure for the inner channel and the first decoder.

The concept of coding gain, first addressed in Chapter 1 becomes more important in this second volume, so we repeat concepts of Section 1.6.3 with minor modification here.

Coding Gain

Of fundamental importance to the comparison of two systems that transmit the same number of bits per dimension is the **coding gain**, which specifies the improvement of one constellation over another when used to transmit the same information.

Definition 8.3.1 (Coding Gain) *The* **coding gain** *(or loss), γ, of a particular constellation with data symbols $\{\boldsymbol{x}_i\}_{i=0,\ldots,M-1}$ with respect to another constellation with data symbols $\{\tilde{\boldsymbol{x}}_i\}_{i=0,\ldots,M-1}$ is defined as*

$$\gamma \triangleq \frac{\left(d^2_{min}(\boldsymbol{x})/\bar{\mathcal{E}}_{\boldsymbol{x}}\right)}{\left(d^2_{min}(\tilde{\boldsymbol{x}})/\bar{\mathcal{E}}_{\tilde{\boldsymbol{x}}}\right)} \quad , \tag{8.59}$$

where both constellations are used to transmit $\bar{b}$ bits of information per dimension.

A coding gain of $\gamma = 1$ (0dB) implies that the two systems perform equally. A positive gain (in dB) means that the constellation with data symbols $\boldsymbol{x}$ outperforms the constellation with data symbols $\tilde{\boldsymbol{x}}$.

Signal constellations are often based on N-dimensional structures known as lattices. (A more complete discussion of lattices appears in Chapter 10.) A **lattice** is a set of vectors in N-dimensional space that is closed under vector addition – that is, the sum of any two vectors is another vector in the set. A translation of a lattice produces a **coset** of the lattice. Most good signal constellations are chosen as subsets of cosets of lattices. The **fundamental volume** for a lattice measures the region around a point:

Definition 8.3.2 (Fundamental Volume) *The fundamental volume $\mathcal{V}(\Lambda)$ of a lattice Λ (from which a signal constellation is constructed) is the volume of the decision region for any single point in the lattice. This decision region is also called a* **Voronoi Region of the lattice**. *The Voronoi Region of a lattice, $\mathcal{V}(\Lambda)$, is to be distinguished from the Voronoi Region of the constellation, $\mathcal{V}_{\boldsymbol{x}}$ the latter being the union of M of the former.*

For example, an M-QAM constellation as $M \to \infty$ is a translated subset (coset) of the two-dimensional rectangular lattice Z^2, so M-QAM is a translation of Z^2 as $M \to \infty$. Similarly as $M \to \infty$, the M-PAM constellation becomes a coset of the one dimensional lattice Z. M will be viewed as the number of points in the constellation, which may now with coding exceed 2^b.

The coding gain, γ of one constellation based on $\boldsymbol{x}$ with lattice λ and volume $\mathcal{V}(\Lambda)$ with respect to another constellation with $\tilde{\boldsymbol{x}}$, $\tilde{\Lambda}$, and $\mathcal{V}(\tilde{\Lambda})$ can be rewritten as

$$\begin{aligned} \gamma &= \frac{\left(\frac{d^2_{\min}(\boldsymbol{x})}{\mathcal{V}_{\boldsymbol{x}}{}^{2/N}}\right)}{\left(\frac{d^2_{\min}(\tilde{\boldsymbol{x}})}{\mathcal{V}_{\tilde{\boldsymbol{x}}}{}^{2/N}}\right)} \cdot \frac{\left(\frac{\mathcal{V}_{\boldsymbol{x}}{}^{2/N}}{\bar{\mathcal{E}}_{\boldsymbol{x}}}\right)}{\left(\frac{\mathcal{V}^{2/N}_{\tilde{\boldsymbol{x}}}}{\bar{\mathcal{E}}_{\tilde{\boldsymbol{x}}}}\right)} & (8.60) \\ &= \gamma_f + \gamma_s \quad (dB) & (8.61) \end{aligned}$$

The two quantities on the right in (8.61) are called the **fundamental gain** γ_f and the **shaping gain** γ_s respectively.

Definition 8.3.3 (Fundamental Gain) *The fundamental gain γ_f of a lattice, upon which a signal constellation is based, is*

$$\gamma_f \triangleq \frac{\left(\frac{d^2_{min}(\boldsymbol{x})}{\mathcal{V}^{2/N}_{\boldsymbol{x}}}\right)}{\left(\frac{d^2_{min}(\tilde{\boldsymbol{x}})}{\mathcal{V}^{2/N}_{\tilde{\boldsymbol{x}}}}\right)} \quad . \tag{8.62}$$

The fundamental gain measures the efficiency of the spacing of the points within a particular constellation per unit of fundamental volume surrounding each point.

Definition 8.3.4 (Shaping Gain) *The shaping gain γ_s of a signal constellation is defined as*

$$\gamma_s = \frac{\left(\frac{\mathcal{V}^{2/N}_{\boldsymbol{x}}}{\bar{\mathcal{E}}_{\boldsymbol{x}}}\right)}{\left(\frac{\mathcal{V}^{2/N}_{\tilde{\boldsymbol{x}}}}{\bar{\mathcal{E}}_{\tilde{\boldsymbol{x}}}}\right)} \quad . \tag{8.63}$$

The shaping gain measures the efficiency of the shape of the boundary of a particular constellation in relation to the average energy per dimension required for the constellation.

EXAMPLE 8.3.1 (Ungerboeck 4-state revisited) *Returning to Example 8.1.4, the fundamental and coding gains can both be computed. The coded 16 QAM constellation will be viewd as the lattice of integer ordered pairs, while the uncoded constellation will be that same lattice (rotation by 45 degrees does not change the lattice) The funamental gain is then*

$$\gamma_f = \frac{\left(\frac{d^2_{min}(\boldsymbol{x})}{\mathcal{V}^{2/N}_{\boldsymbol{x}}}\right)}{\left(\frac{d^2_{min}(\tilde{\boldsymbol{x}})}{\mathcal{V}^{2/N}_{\tilde{\boldsymbol{x}}}}\right)} = \gamma_f \triangleq \frac{\left(\frac{4}{(16\cdot 1)^{2/2}}\right)}{\left(\frac{1}{(8\cdot 1)^{2/2}}\right)} = 2 \text{ (}\mathit{3\ dB}\text{)} \quad . \tag{8.64}$$

$$\gamma_s = \frac{\left(\frac{(16\cdot 1)^{2/2}}{\frac{1}{12}(16-1)}\right)}{\left(\frac{(8\cdot 1)^{2/2}}{\frac{1}{12}(16-1)\cdot\frac{1}{2}}\right)} = 1 \text{ (}\mathit{0\ dB}\text{)} \quad . \tag{8.65}$$

A common uncoded reference is often found for listing of fundamental and coding gains, which is a PAM (or SQ QAM) system with $d = 1$ and thus $V(\Lambda) = 1$ and presumed energy $\frac{1}{12}\left(2^{2\bar{b}} - 1\right)$ even for fractional $\bar{b}$ where this energy formula is not correct. Any system could be compared against this uncoded reference. Differences between coded systems could be obtained by subtracting the two fundamental or two shaping (or overall) coding gains with respect to the common uncoded reference. For the 4-state Ungerboeck code, the gains then become with respect to the common reference:

$$\gamma_f = \frac{\left(\frac{d^2_{min}(\boldsymbol{x})}{\mathcal{V}^{2/N}_{\boldsymbol{x}}}\right)}{\left(\frac{d^2_{min}(\tilde{\boldsymbol{x}})}{\mathcal{V}^{2/N}_{\tilde{\boldsymbol{x}}}}\right)} = \gamma_f \triangleq \frac{\left(\frac{4}{(16\cdot 1)^{2/2}}\right)}{\left(\frac{1}{(8\cdot 1)^{2/2}}\right)} = 2 \text{ (}\mathit{3\ dB}\text{)} \quad , \tag{8.66}$$

but

$$\gamma_s = \frac{\left(\frac{(16\cdot 1)^{2/2}}{\frac{1}{12}(16-1)}\right)}{\left(\frac{(8\cdot 1)^{2/2}}{\frac{1}{12}(8-1)}\right)} = 14/15 \text{ (-}\mathit{0.3\ dB}\text{)} \quad . \tag{8.67}$$

Thus the coding gain with respect to the common reference is 3-0.3=2.7 dB. However, 8SQ has $\gamma_f = 0$ dB and $\gamma_s = -0.3$ dB versus the common reference, so that the difference of the two systems, both relative to the common reference becomes $3 - 0.3 - (0 - 0.3)$=3 dB, which is the correct coding gain as before.

Interpretting coding gain and gap

The coding gain then essentially is the amount by which the gap has been reduced at any given fixed probability of error. Once the gain with respect to uncoded transmission for a given coding method is known, the Gap becomes $\Gamma_{code} = 8.8 - 10\log_{10}(\gamma)$.

As an example, consider again the 4-state trellis code of Section 8.1. This code basically gains 3 dB no matter how many points are in the constellation (beyond $\bar{b} \geq 1$). Thus, the gap with use of this code is now 8.8-3 or 5.8 dB. Trellis codes with gaps as low as about 3.3 dB exist but are complex. Section 10.6 and also Chapter 1 discussed shaping gain - the gain in coding that comes exclusively from ensuring that the boundary of points in a large number of dimensions is a hypersphere (and has little to do with sequences but rather just setting the boundaries of the constellation for any given large number of points). Shaping gain contributes up to 1.53 dB of the 8.8 dB in the gap. The remaining 7.3 dB can be attained through sequences and coding. The most powerful turbo and LDPC codes of Chapter 11 can achieve 7 dB of coding gain, or equivalently, their use plus shaping leads to a system that can operate (with large complexity) at data rates very close to capacity.

The gap in Figure 8.14 can also be interpretted vertically as 1.5 bits/dimension between uncoded and coded. For QAM symbols, this is 3 bits/symbol. For a relatively simple code with 3 dB of gain, 1 of these 3 bits is gained with the simple trellis code of Section 8.1. The remaining 2 are increasingly difficult to achieve. Thus, the gap also measures the data rate loss with respect to capacity, as well as an SNR loss with respect to capacity. The bit gain of the 4-state code with respect to uncoded QAM is thus 1 bit/symbol.

8.3.3 Energy per bit and low-rate coding

The concept of the gap is inapplicable below 10 dB. In this range of SNR, typical transmission is at less than $\bar{b} = 1$, and codes such as convolutional codes (rather than trellis codes for instance) are used. There is essentially a varying limit of coding gain at lower SNR with any given type of code.

The capacity formula for the AWGN channel can also be used to derive the minimum energy per bit that is required for reliable data transmission. This is accomplished by writing the capacity result

$$\begin{aligned}
\bar{b} < \bar{C} &= \frac{1}{2}\log_2\left(1 + \frac{\bar{\mathcal{E}}_{\boldsymbol{x}}}{\sigma^2}\right) && (8.68)\\
&= \frac{1}{2}\log_2\left(1 + \frac{\mathcal{E}_{\boldsymbol{x}}}{N\sigma^2}\right) && (8.69)\\
&= \frac{1}{2}\log_2\left(1 + \frac{\bar{b}\mathcal{E}_{\boldsymbol{x}}}{b\sigma^2}\right) && (8.70)\\
&= \frac{1}{2}\log_2\left(1 + \frac{\bar{b}\mathcal{E}_b}{\sigma^2}\right) \quad . && (8.71)
\end{aligned}$$

Solving for $\frac{\mathcal{E}_b}{\sigma^2}$ in (8.71) yields

$$\frac{\mathcal{E}_b}{\sigma^2} = \frac{2^{2\bar{b}} - 1}{\bar{b}} \quad . \qquad (8.72)$$

Equation (8.72) essentially tells us the minimum required $\mathcal{E}_b/\sigma^2$ for any given code rate on the AWGN. Of fundamental interest is the case where $\bar{b} \to 0$ (that is large redundancy or bandwidth in the code). Then (8.72) reduces to

$$\frac{\mathcal{E}_b}{\sigma^2} = 2\ln 2 \ \ (1.4\text{dB}) \quad , \qquad (8.73)$$

meaning that the energy/bit must be above some finite value even if infinite redundancy (or infinite bandwidth) is used if the designer intends to use a code to improve the use of the AWGN. This result is sometimes phrased in terms of the quantity $\mathcal{E}_b/\mathcal{N}_0$= $.5(\mathcal{E}_b/\sigma^2)$, which is equivalent to the statement that the minimum required $\mathcal{E}_b/\mathcal{N}_0$ is -1.6dB, a well-known result.

8.4 Parallel Channels

8.4.1 Parallel WGN Channels and Capacity

A situation that often arises in dealing with channels with linear intersymbol interference is that of parallel independent channels, which is illustrated in Figure 8.15, repeated from Chapter 4. The noise on each channel is independent of the noise on any other channel, and the input energy on the i^{th} channel is $\mathcal{E}_i$. Then,

$$\mathcal{E}_{\boldsymbol{x}} = \sum_{i=1}^{N} \mathcal{E}_i \quad . \tag{8.74}$$

For this situation, the mutual information between the vector of outputs $\boldsymbol{y}$ and the vector of inputs $\boldsymbol{x}$ is the sum of the individual mutual information between inputs and corresponding outputs:

$$I_{\boldsymbol{x},\boldsymbol{y}} = \sum_{i=1}^{N} I_{x_i,y_i} \quad , \tag{8.75}$$

and therefore the sum of the individual capacities is

$$\max_{p_{\boldsymbol{x}}(i)} I = \sum_{i=1}^{N} C_i = \frac{1}{2}\sum_{i=1}^{N} \log_2\left(1+\frac{\mathcal{E}_i}{\sigma_i^2}\right) = \log_2 \prod_{i=1}^{N} \sqrt{\left(1+\frac{\mathcal{E}_i}{\sigma_i^2}\right)} \quad . \tag{8.76}$$

The remaining free variable over which the transmitter can optimize is the energy distribution among the channels, under the constraint that the total energy satisfy (8.74). Thus,

$$C =^{\max}_{\{\mathcal{E}_i\}} \frac{1}{2}\sum_{i=1}^{N} \log_2\left(1+\frac{\mathcal{E}_i}{\sigma_i^2}\right) \tag{8.77}$$

subject to the constraints

$$\sum_{i=1}^{N} \mathcal{E}_i \quad = \quad \mathcal{E}_{\boldsymbol{x}} \tag{8.78}$$

$$\mathcal{E}_i \quad \geq \quad 0 \tag{8.79}$$

Maximization then forms the "Lagrangian"

$$L = \frac{1}{2}\sum_{i=1}^{N} \log_2\left(1+\frac{\mathcal{E}_i}{\sigma_i^2}\right) + \lambda\left(\mathcal{E}_{\boldsymbol{x}} - \sum_{i=1}^{N}\mathcal{E}_i\right) \tag{8.80}$$

and sets the partial derivative with respect to $\mathcal{E}_i$ equal to zero to obtain

$$\frac{1}{2\left(\mathcal{E}_i+\sigma_i^2\right)\ln(2)} = \lambda = \frac{1}{2\lambda'\ln(2)} \tag{8.81}$$

Thus, the solution is the set of energies satisfying

$$\mathcal{E}_i + \sigma_i^2 = \lambda' = \mathit{constant} \tag{8.82}$$

such that $\sum_{i=1}^{N}\mathcal{E}_i = \mathcal{E}_{\boldsymbol{x}}$ and $\mathcal{E}_i \geq 0$. Figure 8.16 illustrates the basic concept. In Figure 8.16, the fixed energy budget is allocated first to those channels with least noise and successively to channels with increasing noise variance, as long as the sum of the noise variance and the channel energy is constant over those channels that are used. Further, the constraint on total symbol energy determines the the constant level λ'. This method is often referred to as **discrete water filling** in analogy with filling a trough with water to a fixed level. The channel with largest noise, σ_4^2, is not used because $\lambda' < \sigma_4^2$.

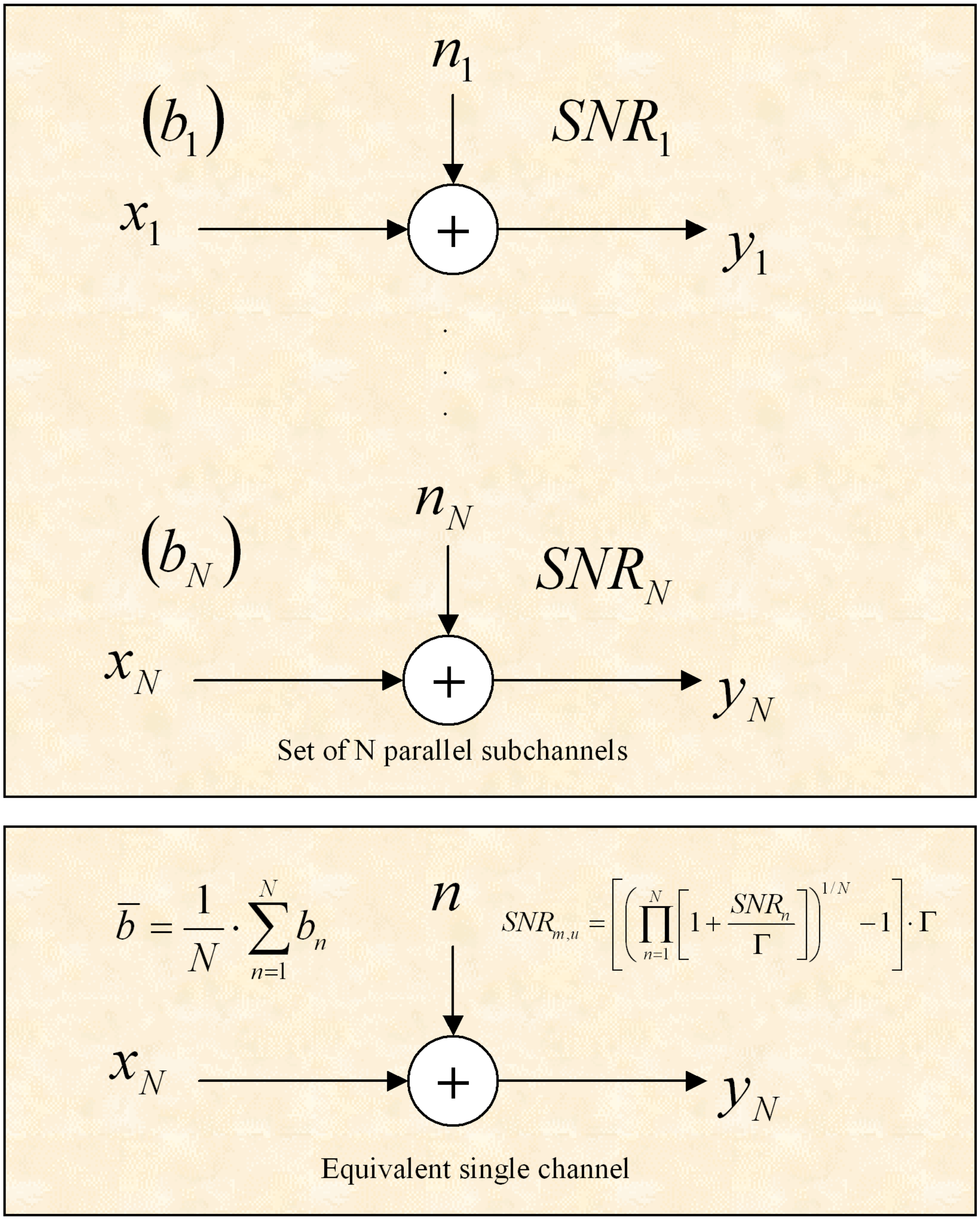
n_1
(b_1)
SNR_1
x_1
+
y_1
n_N
(b_N)
SNR_N
x_N
+
y_N
Set of N parallel subchannels
$\bar{b} = \frac{1}{N} \cdot \sum_{n=1}^{N} b_n$
n
$SNR_{m,u} = \left[\left(\prod_{n=1}^{N} \left[1 + \frac{SNR_n}{\Gamma} \right] \right)^{1/N} - 1 \right] \cdot \Gamma$
x_N
+
y_N
Equivalent single channel

Figure 8.15: Parallel Independent Channels.

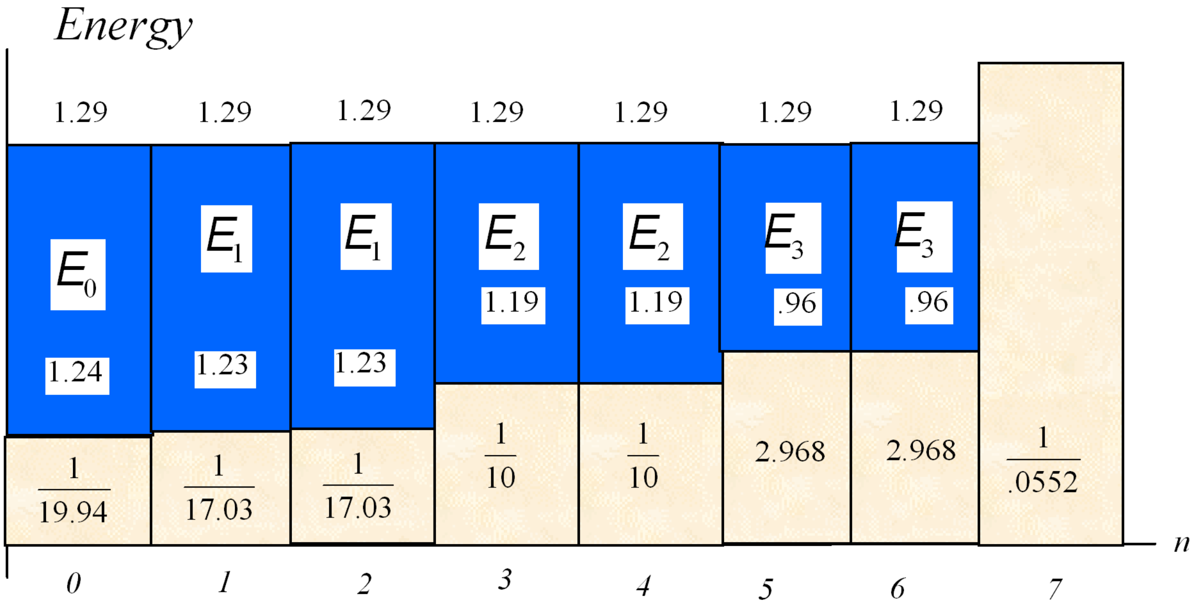

Figure 8.16: Discrete Water filling Illustration.

8.5 Capacity of Waveform Channels

Most of the essential concepts of capacity are contained within the formulae of the last section for $\bar{C}$. In practice, engineers are usually interested in data rates in units of bits/second. This section investigates the conversion of the previous results to units of bits/second.

8.5.1 Capacity Conversion for Memoryless Channels

Both the DMC's discussed previously and the AWGN are instances of memoryless channels in that the current output only depends on the current channel input. In this case, modulation methods such as PAM, QAM, PSK, and others use $2W$ dimensions/second, where W denotes the (positive-frequency-only) "bandwidth" of the system. For instance, with a memoryless channel, the basis function for PAM, $\varphi(t) = \frac{1}{\sqrt{T}}\text{sinc}(t/T)$, can be used without intersymbol interference effects. This basis function requires a bandwidth of $\frac{1}{T} = 2(\frac{1}{2T}) = 2W$ dimensions per second. A similar result follows for QAM and PSK. These waveforms are presumed strictly bandlimited.

The term "bandwidth" is used loosely here, as no system is strictly bandlimitted in practice, and some definition of bandwidth is then required. This text presumes for coding purposes that the system engineer has designed the modulation system for a memoryless (or close to memoryless) channel such that the number of dimensions per second transmitted over the channel is $1/\bar{T}$, where $\bar{T} \stackrel{\Delta}{=} T/N$.

Thus, the memoryless channel with transmit bandwidth W, and $2W = 1/\bar{T}$ output dimensions per second,

$$\mathcal{C} = 2W\bar{C} \tag{8.83}$$

where $\mathcal{C}$ is the capacity in bits/unit time. The AWGN channel then has capacity

$$\mathcal{C}_{awgn} = W \cdot \log_2\left(1 + \frac{\bar{\mathcal{E}}_{\boldsymbol{x}}}{\sigma^2}\right) \text{ bits/unit time } , \tag{8.84}$$

a well-known result.

8.5.2 Waveform Channels with Memory

Most practical channels are not memoryless, so rather than approximate the channel by a memoryless channel using equalization or some other such means, it is of a great deal of interest to evaluate the capacity of the channel with impulse response $h(t)$ and additive Gaussian noise with power spectral density $\mathcal{S}_n(f)$. In this case, the symbol interval T becomes arbitrarily large and analysis considers the channel as a "one-shot" channel with infinite complexity and infinite decoding delay. This generalization of capacity explicitly includes the symbol energy $\mathcal{E}_{\boldsymbol{x}}$ and the symbol period T as arguments, $C \rightarrow C_T(\mathcal{E}_{\boldsymbol{x}})$. Then, the capacity, in bits/second, for a channel with intersymbol interference is

$$\mathcal{C} = \lim_{T\rightarrow\infty} \frac{1}{T} C_T(P_x T) \quad . \tag{8.85}$$

$C_T(P_x T)$ is still the quantity defined in (8.84), but the notation now emphasizes the dependence upon the symbol period T.

The equivalent channel model also shown in Figure 8.15, models an equivalent noise, $n'(t)$, at the impulse response input. As $H(f)$ is one-to-one[5], the ISI-free channel at the filter input can be used for analysis. The power spectral density of the additive Gaussian noise at this point is $\frac{\frac{\mathcal{N}_0}{2}\bar{\mathcal{S}}_n(f)}{|H(f)|^2}$.

The approach to capacity involves decomposing the equivalent channel in Figure 8.15 into an infinite set of parallel independent channels. If the noise $n'(t)$ were white, then this could be accomplished by just transmitting a set of orthogonal signals, as was discussed in Chapter 1. As the noise is not generally white, the decomposition requires what is known as a Karhuenen-Loève decomposition of the noise signal $n'(t)$, which is

$$n'(t) = \sum_{i=1}^{\infty} n'_i \cdot \psi_i(t) \quad , \tag{8.86}$$

[5] Unless $H(f) = 0$ at some frequencies, in which case no energy is transmitted at those frequencies.

where $\{\psi_i(t)\}_{i=1}^{\infty}$ is a set of orthonormal basis functions that satisfy

$$\int_{-\infty}^{\infty} \psi_i(t) \cdot r_{n'}(s-t)dt = \sigma_i^2 \cdot \psi_i(s) \quad , \tag{8.87}$$

and

$$E\left[n_i n_j^*\right] = \left\{ \begin{array}{ll} \sigma_i^2 & i=j \\ 0 & i \neq j \end{array} \right. \quad . \tag{8.88}$$

$\psi_i(t)$ are often also called "eigenfunctions" of the noise autocorrelation function and σ_i^2 are the "eigenvalues," and are the noise variances of the parallel channels. The eigenfunctions constitute a complete set. Thus any waveform satisfying the Paley-Wiener Criterion of Chapter 1 can be represented, so the channel-input waveform $x(t)$ is

$$x(t) = \sum_{i=1}^{\infty} x_i \cdot \psi_i(t) \quad , \tag{8.89}$$

where

$$x_i = \int_{-\infty}^{\infty} x(t) \cdot \psi_i(t) dt \quad . \tag{8.90}$$

The samples at the output of the infinite set of matched filters, $\{\psi_i(-t)\}_{i=0,...,\infty}$, constitute a set of parallel independent channels with noise variances σ_i^2.

The application of the parallel channels concepts from Section 8.4 computes a capacity (in bits/dimension)

$$C_T(P_x T) = \sum_{i=1}^{\infty} \max\left[0, \frac{1}{2}\log_2\left(1 + \frac{\mathcal{E}_i}{\sigma_i^2}\right)\right] \tag{8.91}$$

with transmit energy

$$P_x T = \sum_{i=1}^{\infty} \max\left[0, \lambda' - \sigma_i^2\right] \quad . \tag{8.92}$$

Dividing both sides by T and taking limits as $T \rightarrow \infty$ produces

$$\mathcal{C} =_{T\rightarrow\infty}^{\lim} \frac{1}{T} \sum_{i=1}^{\infty} \max\left[0, \frac{1}{2}\log_2\left(\frac{\lambda'}{\sigma_i^2}\right)\right] \quad , \tag{8.93}$$

and

$$P_x =_{T\rightarrow\infty}^{\lim} \frac{1}{T} \sum_{i=1}^{\infty} \max\left[0, \lambda' - \sigma_i^2\right] \quad . \tag{8.94}$$

Both sums above are nonzero over the same range for i, which this text calls $\Omega_{\mathcal{E}}$. In the limit,

$$\sigma_i^2 \rightarrow \frac{\mathcal{S}_n(f)}{|H(f)|^2} \tag{8.95}$$

and

$$\frac{1}{T} \rightarrow df \quad , \tag{8.96}$$

leaving the famous "water-filling" scheme of Gallager/Shannon for capacity calculation of the waveform channel:

$$\mathcal{C} = \frac{1}{2}\int_{\Omega_{\mathcal{E}}} \log_2 \frac{\lambda' |H(f)|^2}{\mathcal{S}_n(f)} df \tag{8.97}$$

and

$$P_x = \int_{\Omega_{\mathcal{E}}} \left(\lambda' - \frac{\mathcal{S}_n(f)}{|H(f)|^2}\right) df \quad , \tag{8.98}$$

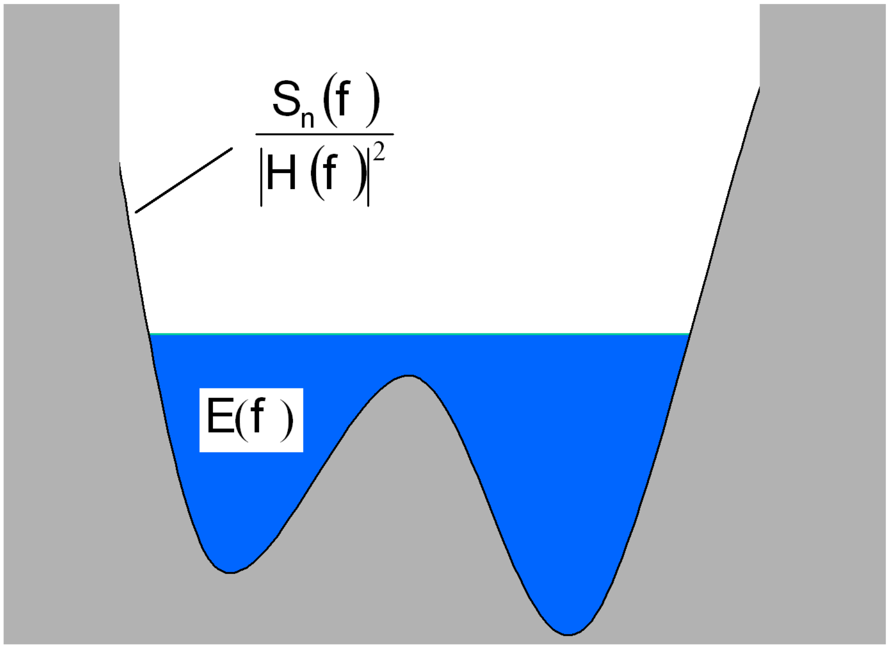

Figure 8.17: Continuous Water Filling.

where the transmit spectrum is chosen to satisfy

$$\lambda' = \frac{S_n(f)}{|H(f)|^2} + \mathcal{E}(f) \quad , \tag{8.99}$$

which results in the equivalent capacity expression

$$C = \frac{1}{2}\int_{\Omega_{\mathcal{E}}} \log_2\left(1 + \frac{\mathcal{E}(f)|H(f)|^2}{S_n(f)}\right) df \quad . \tag{8.100}$$

The continuous water filling concept is illustrated in Figure 8.17. The designer "pours" energy into the inverted channel (multiplied by any noise power spectral density) until no energy remains, which determines both λ' and $\Omega_{\mathcal{E}}$. Then $\mathcal{C}$ is computed through (8.97).

8.5.3 Capacity of the infinite bandwidth channel

An interesting interpretation of the AWGN capacity result presumes infinite bandwidth on the part of the transmitter and a channel that ideally passes all frequencies with equal gain and no phase distortion. In this case, $W \to \infty$ in $\mathcal{C} = W \log_2(1 + \text{SNR})$, or

$$\mathcal{C}_\infty =^{\lim}_{W\to\infty} W \frac{1}{\ln 2} \ln\left(1 + \frac{P_x}{2W\sigma^2}\right) = \frac{1}{\ln 2}\frac{P_x}{2\sigma^2} \quad . \tag{8.101}$$

This result shows that even with infinite bandwidth, if we have a finite-power constraint imposes a finite data rate.

8.5.4 Example of Water-Pouring Capacity Calculation

An example of the continuous water filling is the flat AWGN channel with $H(f) = 1$ and $\mathcal{S}_n(f) = \frac{\mathcal{N}_0}{2}$. Then, one orthonormal set of basis functions is $\frac{1}{\sqrt{T}}\text{sinc}\left(\frac{t-iT}{T}\right)$ $\forall$ i functions and the eigenvalues are the constant $\sigma_i^2 = \frac{\mathcal{N}_0}{2}$. Thus,

$$P_x = (\lambda' - \frac{\mathcal{N}_0}{2})2W \tag{8.102}$$

where $W = 1/2T$. Then,

$$\lambda' = \frac{P_x}{2W} + \frac{\mathcal{N}_0}{2} \tag{8.103}$$

and

$$\mathcal{C} = \left(\frac{1}{2}\log_2\left[\frac{\frac{P_x}{2W}+\frac{\mathcal{N}_0}{2}}{\frac{\mathcal{N}_0}{2}}\right]\right)2W = W\log_2\left(1+\text{SNR}\right) \quad , \tag{8.104}$$

a result obtained earlier.

EXAMPLE 8.5.1 ($1 + .9D^{-1}$ Channel Capacity) A second example is the channel with impulse response $h(t) = \text{sinc}(t) + .9\text{sinc}(t-1)$ from Chapters 3 and 4. Then

$$P_x = \int_{-W}^{W}\left(\lambda' - \frac{.181}{1.81+1.8\cos(\omega)}\right)\frac{d\omega}{2\pi} \tag{8.105}$$

where W is implicitly in radians/second for this example. If $P_x = 1$ with an SNR of 10dB, the integral in (8.105) simplifies to

$$\begin{aligned} \pi &= \int_0^W \left(\lambda' - \frac{.181}{1.81+1.8\cos(\omega)}\right)d\omega & (8.106)\\ &= \lambda' W - .181\left\{\frac{2}{\sqrt{1.81^2-1.8^2}}\arctan\left[\frac{\sqrt{1.81^2-1.8^2}}{1.81+1.8}\tan\left(\frac{W}{2}\right)\right]\right\} & (8.107)\end{aligned}$$

At the bandedge W,

$$\lambda' = \frac{.181}{1.81+1.8\cos(W)} \quad . \tag{8.108}$$

leaving the following transcendental equation to solve by trial and error:

$$\pi = \frac{.181W}{1.81+1.8\cos(W)} - 1.9053\arctan\left(.0526\tan(W/2)\right) \tag{8.109}$$

$W = .88\pi$ approximately solves (8.109).

The capacity is then

$$\begin{aligned} \mathcal{C} &= \frac{2}{2\pi}\int_0^{.88\pi}\frac{1}{2}\log_2\left(\frac{1.33}{.181}(1.81+1.8\cos\omega)\right)d\omega & (8.110)\\ &= \frac{1}{2\pi}\int_0^{.88\pi}\log_2 7.35 d\omega + \frac{1}{2\pi}\int_0^{.88\pi}\log_2\left(1.81+1.8\cos\omega\right)d\omega & (8.111)\\ &= 1.266 + .284 & (8.112)\\ &\approx 1.5\text{bits/second} \quad . & (8.113)\end{aligned}$$

See Chapters 4 and 5 for a more complete development of this example and capacity.

Exercises - Chapter 3

8.1 *Sequence Generation*

Find the bipolar binary outputs (±1 sequences) of the binary differential encoder for the following input sequences: (a input state corresponding to a previous bit of 0 can be assumed and $m(D)$ is a binary sequence of 0's and 1's)

a. $m(D) = 1$

b. $m(D) = D$

c. $m(D) = 1/(1 \oplus D)$ (See Section 3.8.4).

d. the input periodically cycles through the sequence 1 0 1, starting at time 0.

8.2 *Sequence Generation for a trellis code*

Find the sequence of 16 QAM outputs for the trellis code example of Section 8.1 when the input message sequence is as given below. You may assume the encoder starts in state zero and that the 2nd and 3rd input bit in each group of input 3 bits specifies the subset points in clockwise fashion starting with the upper left-hand corner point being 00. Assume the 16QAM constellation has $d = 2$. All addition is mod 8 in this problem.

a. $m(D) = [1\ 0\ 0] + [0\ 1\ 1]D + [0\ 1\ 0]D^2 = 4 + 3D + 2D^2$.

b. $m(D) = 3 + 7D + 5D^2 + D^3$

c. $m(D) = \frac{4+3D+D^2}{1+D^3}$.

d. $m(D) = \frac{4+3D+D^2}{1-D^3}$.

8.3 *Trellis of Extended Partial Response*

a. For binary inputs, draw the trellis for EPR4 (See Section 3.8.3) following the convention that the top branch emanating from each state corresponds to a zero input bit and the other branch is the 1 input. Use the state $(m_{k-3}, m_{k-2}, m_{k-1})$.

b. Determine the minimum distance between any closest two sequences.

c. How many states would the trellis have if the input had 4 levels? Does the minimum distance change?

8.4 *Convolutional Code* Consider only the binary output bits from the convolutional code in the example in Section 8.1, $v_{2,k}$ and $v_{1,k}$ in this problem (that is do not translate them to bipolar signals) and transmission on the BSC channel.

a. For the convolutional code example in Section 8.1, show that the modulo two sum of any two codewords is another codeword.

b. Show that the all zeros sequence is a codeword.

c. Defining the Hamming weight of any codeword as the number of 1's in the sequence, find the smallest Hamming weight for any nonzero codeword.

d. Show that any convolutional code that can be described by the modulo-2 vector multiplication $\boldsymbol{v}(D) = u(D) \cdot G(D)$ satisfies the same 3 properties in parts a,b, and c.

8.5 *Capacity* An AWGN has an input with two-sided power spectral density -40 dbm/Hz and noise power spectral density -60 dBm/Hz.

a. What is the SNR for PAM transmission on this channel? For QAM transmission?

b. What is the capacity in bits/dimension?

c. What is the minimum amount of transmit power necessary to send 1 Mbps over this channel if the QAM symbol rate is 1 MHz? if 2 MHz? (assume sequences are used).

d. If the capacity is known to be 10 Mbps, what is the symbol rate for PAM?

8.6 *Channel Capacity Calculations*

For this problem, assume N = number of dimensions =1.

a. Plot the graphs of the channel capacity of the binary symmetric channel and of the binary erasure channel as a function of the bit error probability p over the range for p of 0 to 1. (2 pts)

b. Explain intuitively why the capacity of the binary symmetric channel decreases monotonically from 1 to (for $p = 0$) to 0 (for $p = .5$) and then increases back to 1 for $p = 1$. (2 pts)

c. In contrast to the BSC, the capacity of the BEC decreases monotonically from 1 (for $p = 0$) to 0 (for $p = 1$). Explain why this is the case. (1 pts)

d. Find the capacity of an AWGN with SNR of 10 dB. (1pt)

e. Find P_b for binary PAM transmission on the channel of part (d). (2pts)

f. Letting p for a BSC be equal to the P_b you found in part (e), find the capacity of this BSC. (2 pts)

g. Compare the capcity in part (f) with the caqpacity in part (d). Why are they different?

8.7 *Coding Gain*

This problem refers to the 4-state Trellis Code example in Section 8.1.

a. Find a way to use the same trellis code if $b = 5$ and the constellation used is 64 QAM. Label your constellation points with subset indices. (2 pts)

b. What is the coding gain of your code in part (a) if the uncoded reference is 32 SQ QAM? (1 pt)

c. What is the coding gain of your code in part (a) if the uncoded reference is 32 CR QAM? (2 pts)

d. What is the coding gain if $b = 1$ and 4 QAM is the constellation used? (you may use BPSK as the reference system). (3pts)

e. Find the fundamental coding gain for this code. (1 pt)

f. Find the shaping gain of 32CR constellations with respect to 32 SQ constellations when $b = 5$ and no trellis code is used. (2 pts)

8.8 *Universal DMC*

Inputs to and outputs from a DMC is presumed to be any one of 256 possible messages and the channel probabilities are given by

$$p(i/j) = \begin{cases} \frac{p_s}{255} & \forall i \neq j \\ 1 - p_s & i = j \end{cases} \tag{8.114}$$

a. Find the input distribution that achieves capacity for this channel. (1 pt)

b. Find the capacity. (2 pts)

c. What is the capacity as $p_s \rightarrow 0$? (1 pt)

d. Why might this channel be of interest? (1 pt)

8.9 *Partial Response and Capacity*

A precoded partial-response system uses $M = 4$ PAM on the $1 - D$ channel with AWGN and $\text{SNR} = \bar{\mathcal{E}}_{\boldsymbol{x}}/\sigma^2$=10 dB.

a. Draw the precoder. (2 pts)

b. Compute P_e and $\bar{P}_b$. Assume that each hard decision is independent of the others as a worst-case. (2 pts)

c. Model this system as a BSC and compute capacity. (2 pts)

d. Model this system as an AWGN and compute capacity, comparing to that in part (c). (You may assume constant transmit power and $T = 1$.) (2 pts)

e. Do you think the answer in part (d) is the highest capacity for this channel? Why or Why not? (variable T may be required.) (2 pts)

8.10 *Gap for Trellis*

A transmission system for the AWGN uses QAM and the 4-state trellis code of Section 8.1 with symbol rate of 1 MHz and operates at $P_e = 10^{-6}$.

a. Estimate the gap in SNR using the fundamental coding gain. (1 pt)

b. Estimate the gap to capacity in bits/symbol at high SNR. (2 pts)

c. Use your SNR gap in part (a) to estimate the data rate this code can achieve for SNRs of 17, 25, and 35 dB. You need not worry about fractional bits per symbol here in your answers. (2 pts)

d. At what new symbol rate would you obtain the largest data rate if the SNR is 25 dB at the symbol rate of 1 MHz and what is the corresponding data rate (assume constant power)? (2 pts)

8.11 *Channel Capacity Calculations.*

For this problem, assume N = number of dimensions = 1.

a. (2 pts) Plot the graphs of the channel capacity of the binary symmetric channel and of the binary erasure channel as a function of the bit error probability p. Note that p varies from 0 to 1!

b. (2 pts) Can you explain intuitively as to why the capacity of the binary symmetric channel at first decreases from 1 (for $p = 0$) to 0 (for $p = 0.5$) and then increases back to 1 (for $p = 1$)?

c. (1 pt) In contrast to the binary symmetric channel, the capacity of the binary erasure channel decreases monotonically from 1 (for $p = 0$) to 0 (for $p = 1$). Explain as to why this seems reasonable.

d. (1 pt) Find the capacity of a flat AWGN channel for an SNR of 10 dB.

e. (2 pts) Notice that the flat AWGN channel, for which you calculated the capacity in part (d), is in fact our channel of chapter 1 (the simplest possible channel!). Suppose that on this channel, we transmit uncoded 2-PAM signals (ie. the usual 2-PAM signals) with an SNR of 10 dB. Find P_b. *Hint*: $P_b = P_e$ for uncoded 2-PAM.

f. (2 pts) Each transmission (and detection) of an uncoded 2-PAM signal is equivalent to a 1 bit transmission. Therefore, we can look upon the 2-PAM based transmission scheme of part (e) as defining a binary symmetric channel with a bit error probability $p = P_b$. For the P_b obtained in part (e), find the capacity of this BSC channel.

g. (3 pts) How does the capacity calculated in part (f) compare with the capacity of the actual AWGN channel, which was calculated in part (d)? Can you think of some reason(s) for the difference?

8.12 *Capacity in bits/sec - Midterm 2001 - 7 pts*

An AWGN channel has SNR=20 dB when the symbol rate is 1 MHz for PAM transmission.

a. What is the capacity in bits/dimension? (1 pt)

b. What is the capacity at any symbol rate a/T where $a > 0$? (2 pts)

c. Find the max bit rate in Mbps that can be transmitted on this channel (you may vary T)? (2 pts)

d. What is the SNR at the symbol rate used in part c? (1 pt) Would you expect this operating condition to be useful in practice? (1 pt)