3

SOURCE CODING

Communication systems are designed to transmit the information generated by a source to some destination. Information sources may take a variety of different forms. For example, in radio broadcasting, the source is generally an audio source (voice or music). In TV broadcasting, the information source is a video source whose output is a moving image. The outputs of these sources are analog signals and, hence, the sources are called *analog sources*. In contrast, computers and storage devices, such as magnetic or optical disks, produce discrete outputs (usually binary or ASCII characters) and, hence, they are called *discrete sources*.

Whether a source is analog or discrete, a digital communication system is designed to transmit information in digital form. Consequently, the output of the source must be converted to a format that can be transmitted digitally. This conversion of the source output to a digital form is generally performed by the source encoder, whose output may be assumed to be a sequence of binary digits.

In this chapter, we treat source encoding based on mathematical models of information sources and a quantitative measure of the information emitted by a source. We consider the encoding of discrete sources first and then we discuss the encoding of analog sources. We begin by developing mathematical models for information sources.

3-1 MATHEMATICAL MODELS FOR INFORMATION SOURCES

Any information source produces an output that is random, i.e., the source output is characterized in statistical terms. Otherwise, if the source output

were known exactly, there would be no need to transmit it. In this section, we consider both discrete and analog information sources, and we postulate mathematical models for each type of source.

The simplest type of discrete source is one that emits a sequence of letters selected from a finite alphabet. For example, a *binary source* emits a binary sequence of the form 100101110..., where the alphabet consists of the two letters {0, 1}. More generally, a discrete information source with an alphabet of L possible letters, say $\{x_1, x_2, \ldots, x_L\}$, emits a sequence of letters selected from the alphabet.

To construct a mathematical model for a discrete source, we assume that each letter in the alphabet $\{x_1, x_2, \ldots, x_L\}$ has a given probability p_k of occurrence. That is,

$$p_k = P(X = x_k), \quad 1 \leq k \leq L$$

where

$$\sum_{k=1}^{L} p_k = 1$$

We consider two mathematical models of discrete sources. In the first, we assume that the output sequence from the source is statistically independent. That is, the current output letter is statistically independent from all past and future outputs. A source whose output satisfies the condition of statistical independence among output letters in the sequence is said to be *memoryless*. Such a source is called a *discrete memoryless source* (DMS).

If the discrete source output is statistically dependent, as, for example, English text, we may construct a mathematical model based on statistical stationarity. By definition, a discrete source is said to be *stationary* if the joint probabilities of two sequences of length n, say $a_1, a_2, \ldots, a_n$ and $a_{1+m}, a_{2+m}, \ldots, a_{n+m}$ are identical for all $n \geq 1$ and for all shifts m. In other words, the joint probabilities for any arbitrary length sequence of source outputs are invariant under a shift in the time origin.

An *analog* source has an output waveform $x(t)$ that is a sample function of a stochastic process $X(t)$. We assume that $X(t)$ is a stationary stochastic process with autocorrelation function $\phi_{xx}(\tau)$ and power spectral density $\Phi_{xx}(f)$. When $X(t)$ is a bandlimited stochastic process, i.e., $\Phi_{xx}(f) = 0$ for $|f| \geq W$, the sampling theorem may be used to represent $X(t)$ as

$$X(t) = \sum_{n=-\infty}^{\infty} X\left(\frac{n}{2W}\right) \frac{\sin\left[2\pi W\left(t - \frac{n}{2W}\right)\right]}{2\pi W\left(t - \frac{n}{2W}\right)} \tag{3-1-1}$$

where $\{X(n/2W)\}$ denote the samples of the process $X(t)$ taken at the sampling (Nyquist) rate of $f_s = 2W$ samples/s. Thus, by applying the sampling theorem, we may convert the output of an analog source into an equivalent

discrete-time source. Then, the source output is characterized statistically by the joint pdf $p(x_1, x_2, \ldots, x_m)$ for all $m \geq 1$, where $X_n = X(n/2W)$, $1 \leq n \leq m$, are the random variables corresponding to the samples of $X(t)$.

We note that the output samples $\{X(n/2W)\}$ from the stationary sources are generally continuous, and, hence, they cannot be represented in digital form without some loss in precision. For example, we may quantize each sample to a set of discrete values, but the quantization process results in loss of precision, and, consequently, the original signal cannot be reconstructed exactly from the quantized sample values. Later in this chapter, we shall consider the distortion resulting from quantization of the samples from an analog source.

3-2 A LOGARITHMIC MEASURE OF INFORMATION

To develop an appropriate measure of information, let us consider two discrete random variables with possible outcomes x_i, $i = 1, 2, \ldots, n$, and y_i, $i = 1, 2, \ldots, m$, respectively. Suppose we observe some outcome $Y = y_j$ and we wish to determine, quantitatively, the amount of information that the occurrence of the event $Y = y_j$ provides about the event $X = x_i$, $i = 1, 2, \ldots, n$. We observe that when X and Y are statistically independent, the occurrence of $Y = y_j$ provides no information about the occurrence of the event $X = x_i$. On the other hand, when X and Y are fully dependent such that the occurrence of $Y = y_j$ determines the occurrence of $X = x_i$, the information content is simply that provided by the event $X = x_i$. A suitable measure that satisfies these conditions is the logarithm of the ratio of the conditional probability

$$P(X = x_i \mid Y = y_j) \equiv P(x_i \mid y_j)$$

divided by the probability

$$P(X = x_i) \equiv P(x_i)$$

That is, the information content provided by the occurrence of the event $Y = y_j$ about the event $X = x_i$ is defined as

$$I(x_i; y_j) = \log \frac{P(x_i \mid y_j)}{P(x_i)} \tag{3-2-1}$$

$I(x_i; y_j)$ is called the *mutual information* between x_i and y_j.

The units of $I(x_i; y_j)$ are determined by the base of the logarithm, which is usually selected as either 2 or e. When the base of the logarithm is 2, the units of $I(x_i; y_j)$ are bits, and when the base is e, the units of $I(x_i; y_j)$ are called *nats* (natural units). (The standard abbreviation for $\log_e$ is ln.) Since

$$\ln a = \ln 2 \log_2 a = 0.693\,15 \log_2 a$$

the information measured in nats is equal to $\ln 2$ times the information measured in bits.

When the random variables X and Y are statistically independent,

$P(x_i \mid y_j) = P(x_i)$ and, hence, $I(x_i; y_j) = 0$. On the other hand, when the occurrence of the event $Y = y_j$ uniquely determines the occurrence of the event $X = x_i$, the conditional probability in the numerator of (3-2-1) is unity and, hence,

$$I(x_i; y_j) = \log \frac{1}{P(x_i)} = -\log P(x_i) \tag{3-2-2}$$

But (3-2-2) is just the information of the event $X = x_i$. For this reason, it is called the *self-information* of the event $X = x_i$ and it is denoted as

$$I(x_i) = \log \frac{1}{P(x_i)} = -\log P(x_i) \tag{3-2-3}$$

We note that a high-probability event conveys less information than a low-probability event. In fact, if there is only a single event x with probability $P(x) = 1$ then $I(x) = 0$. To demonstrate further that the logarithmic measure of information content is the appropriate one for digital communications, let us consider the following example.

Example 3-2-1

Suppose we have a discrete information source that emits a binary digit, either 0 or 1, with equal probability every τ_s seconds. The information content of each output from source is

$$\begin{aligned} I(x_i) &= -\log_2 P(x_i), \qquad x_i = 0, 1 \\ &= -\log_2 \tfrac{1}{2} = 1 \text{ bit} \end{aligned}$$

Now suppose that successive outputs from the source are statistically independent, i.e., the source is memoryless. Let us consider a block of k binary digits from the source that occurs in a time interval $k\tau_s$. There are $M = 2^k$ possible k-bit blocks, each of which is equally probable with probability $1/M = 2^{-k}$. The self-information of a k-bit block is

$$I(x_i') = -\log_2 2^{-k} = k \text{ bits}$$

emitted in a time interval $k\tau_s$. Thus the logarithmic measure of information content possesses the desired additivity property when a number of source outputs is considered as a block.

Now let us return to the definition of mutual information given in (3-2-1) and multiply the numerator and denominator of the ratio of probabilities by $P(y_j)$. Since

$$\frac{P(x_i \mid y_j)}{P(x_i)} = \frac{P(x_i \mid y_j)P(y_j)}{P(x_i)P(y_j)} = \frac{P(x_i, y_j)}{P(x_i)P(y_j)} = \frac{P(y_j \mid x_i)}{P(y_j)}$$

we conclude that

$$I(x_i; y_j) = I(y_j; x_i) \tag{3-2-4}$$

Therefore the information provided by the occurrence of the event $Y = y_j$ about the event $X = x_i$ is identical to the information provided by the occurrence of the event $X = x_i$ about the event $Y = y_j$.

Example 3-2-2

Suppose that X and Y are binary-valued $\{0, 1\}$ random variables that represent the input and output of a binary-input, binary-output channel. The input symbols are equally likely and the output symbols depend on the input according to the conditional probabilities

$$P(Y = 0 \mid X = 0) = 1 - p_0$$
$$P(Y = 1 \mid X = 0) = p_0$$
$$P(Y = 1 \mid X = 1) = 1 - p_1$$
$$P(Y = 0 \mid X = 1) = p_1$$

Let us determine the mutual information about the occurrence of the events $X = 0$ and $X = 1$, given that $Y = 0$.

From the probabilities given above, we obtain

$$\begin{aligned} P(Y = 0) &= P(Y = 0 \mid X = 0)P(X = 0) + P(Y = 0 \mid X = 1)P(X = 1) \\ &= \tfrac{1}{2}(1 - p_0 + p_1) \\ P(Y = 1) &= P(Y = 1 \mid X = 0)P(X = 0) + P(Y = 1 \mid X = 1)P(X = 1) \\ &= \tfrac{1}{2}(1 - p_1 + p_0) \end{aligned}$$

Then, the mutual information about the occurrence of the event $X = 0$, given that $Y = 0$ is observed, is

$$I(x_1; y_1) = I(0; 0) = \log_2 \frac{P(Y = 0 \mid X = 0)}{P(Y = 0)} = \log_2 \frac{2(1 - p_0)}{1 - p_0 + p_1}$$

Similarly, given that $Y = 0$ is observed, the mutual information about the occurrence of the event $X = 1$ is

$$I(x_2; y_1) = I(1; 0) = \log_2 \frac{2p_1}{1 - p_0 + p_1}$$

Let us consider some special cases: First, if $p_0 = p_1 = 0$, the channel is called *noiseless* and

$$I(0; 0) = \log_2 2 = 1 \text{ bit}$$

Hence, the output specifies the input with certainty. On the other hand, if $p_0 = p_1 = \frac{1}{2}$, the channel is *useless* because

$$I(0; 0) = \log_2 1 = 0$$

However, if $p_0 = p_1 = \frac{1}{4}$, then

$$I(0; 0) = \log_2 \tfrac{3}{2} = 0.587$$

$$I(0; 1) = \log_2 \tfrac{1}{2} = -1 \text{ bit}$$

. In addition to the definition of mutual information and self-information, it is useful to define the *conditional self-information* as

$$I(x_i \mid y_j) = \log \frac{1}{P(x_i \mid y_j)} = -\log P(x_i \mid y_j) \tag{3-2-5}$$

Then, by combining (3-2-1), (3-2-3), and (3-2-5), we obtain the relationship

$$I(x_i; y_j) = I(x_i) - I(x_i \mid y_j) \tag{3-2-6}$$

We interpret $I(x_i \mid y_j)$ as the self-information about the event $X = x_i$ after having observed the event $Y = y_j$. Since both $I(x_i) \geq 0$ and $I(x_i \mid y_j) \geq 0$, it follows that $I(x_i; y_j) < 0$ when $I(x_i \mid y_j) > I(x_i)$, and $I(x_i; y_j) > 0$ when $I(x_i \mid y_j) < I(x_i)$. Hence, the mutual information between a pair of events can be either positive, or negative, or zero.

3-2-1 Average Mutual Information and Entropy

Having defined the mutual information associated with the pair of events (x_i, y_j), which are possible outcomes of the two random variables X and Y, we can obtain the average value of the mutual information by simply weighting $I(x_i; y_j)$ by the probability of occurrence of the joint event and summing over all possible joint events. Thus, we obtain

$$\begin{aligned} I(X; Y) &= \sum_{i=1}^{n} \sum_{j=1}^{m} P(x_i, y_j) I(x_i; y_j) \\ &= \sum_{i=1}^{n} \sum_{j=1}^{m} P(x_i, y_j) \log \frac{P(x_i, y_j)}{P(x_i) P(y_j)} \end{aligned} \tag{3-2-7}$$

as the average mutual information between X and Y. We observe that

$I(X;Y)=0$ when X and Y are statistically independent. An important characteristic of the average mutual information is that $I(X;Y)\geq 0$ (see Problem 3-4).

Similarly, we define the average self-information, denoted by $H(X)$, as

$$H(X)=\sum_{i=1}^{n} P(x_i)I(x_i)$$

$$=-\sum_{i=1}^{n} P(x_i)\log P(x_i) \tag{3-2-8}$$

When X represents the alphabet of possible output letters from a source, $H(X)$ represents the average self-information per source letter, and it is called the *entropy*† of the source. In the special case in which the letters from the source are equally probable, $P(x_i)=1/n$ for all i, and, hence,

$$H(X)=-\sum_{i=1}^{n}\frac{1}{n}\log\frac{1}{n}$$

$$=\log n \tag{3-2-9}$$

In general, $H(X)\leq \log n$ (see Problem 3-5) for any given set of source letter probabilities. In other words, *the entropy of a discrete source is a maximum when the output letters are equally probable.*

Example 3-2-3

Consider a source that emits a sequence of statistically independent letters, where each output letter is either 0 with probability q or 1 with probability $1-q$. The entropy of this source is

$$H(X)\equiv H(q)=-q\log q-(1-q)\log(1-q) \tag{3-2-10}$$

The binary entropy function $H(q)$ is illustrated in Fig. 3-2-1. We observe that the maximum value of the entropy function occurs at $q=\frac{1}{2}$ where $H(\frac{1}{2})=1$.

The average conditional self-information is called the *conditional entropy* and is defined

$$H(X\mid Y)=\sum_{i=1}^{n}\sum_{j=1}^{m} P(x_i,y_j)\log\frac{1}{P(x_i\mid y_j)} \tag{3-2-11}$$

We interpret $H(X\mid Y)$ as the information or uncertainty in X after Y is

†The term *entropy* is taken from statistical mechanics (thermodynamics), where a function similar to (3-2-8) is called (thermodynamic) entropy.

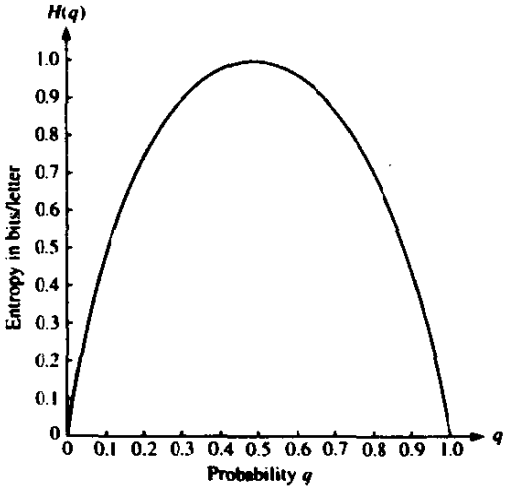


FIGURE 3-2-1 Binary entropy function.

observed. By combining (3-2-7), (3-2-8), and (3-2-11) we obtain the relationship

$$I(X;Y) = H(X) - H(X \mid Y) \tag{3-2-12}$$

Since $I(X;Y) \geq 0$, it follows that $H(X) \geq H(X \mid Y)$, with equality if and only if X and Y are statistically independent. If we interpret $H(X \mid Y)$ as the average amount of (conditional self-information) uncertainty in X after we observe Y, and $H(X)$ as the average amount of uncertainty (self-information) prior to the observation, then $I(X;Y)$ is the average amount of (mutual information) uncertainty provided about the set X by the observation of the set Y. Since $H(X) \geq H(X \mid Y)$, it is clear that conditioning on the observation Y does not increase the entropy.

Example 3-2-4

Let us evaluate the $H(X \mid Y)$ and $I(X;Y)$ for the binary-input, binary-output channel treated previously in Example 3-2-2 for the case where $p_0 = p_1 = p$. Let the probabilities of the input symbols be $P(X=0) = q$ and $P(X=1) = 1-q$. Then the entropy is

$$H(X) \equiv H(q) = -q \log q - (1-q) \log (1-q)$$

where $H(q)$ is the binary entropy function and the conditional entropy $H(X \mid Y)$ is defined by (3-2-11). A plot of $H(X \mid Y)$ as a function of q with

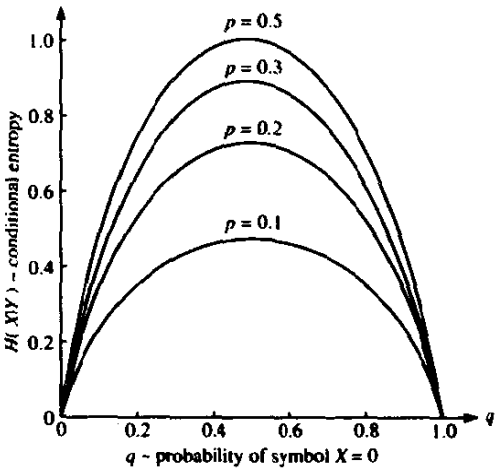


FIGURE 3-2-2 Conditional entropy for binary-input, binary-output symmetric channel.

p as a parameter is shown in Fig. 3-2-2. The average mutual information $I(X; Y)$ is plotted in Fig. 3-2-3.

As in the preceding example, when the conditional entropy $H(X \mid Y)$ is viewed in terms of a channel whose input is X and whose output is Y, $H(X \mid Y)$ is called the *equivocation* and is interpreted as the amount of average uncertainty remaining in X after observation of Y.

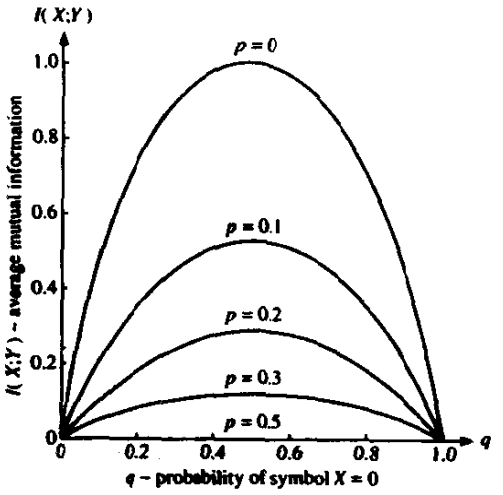


FIGURE 3-2-3 Average mutual information for binary-input, binary-output symmetric channel.

The results given above can be generalized to more than two random variables. In particular, suppose we have a block of k random variables $X_1X_2\cdots X_k$, with joint probability $P(x_1x_2\cdots x_k) \equiv P(X_1 = x_1, X_2 = x_2, \ldots, X_k = x_k)$. Then, the entropy for the block is defined as

$$H(X_1X_2\cdots X_k) = -\sum_{j_1=1}^{n_1}\sum_{j_2=1}^{n_2}\cdots\sum_{j_k=1}^{n_k} P(x_{j_1}x_{j_2}\cdots x_{j_k})\log P(x_{j_1}x_{j_2}\cdots x_{j_k}) \quad (3\text{-}2\text{-}13)$$

Since the joint probability $P(x_1x_2\cdots x_k)$ can be factored as

$$P(x_1x_2\cdots x_k) = P(x_1)P(x_2 \mid x_1)P(x_3 \mid x_1x_2)\cdots P(x_k \mid x_1x_2\cdots x_{k-1}) \quad (3\text{-}2\text{-}14)$$

it follows that

$$\begin{aligned} H(X_1X_2X_3\cdots X_k) &= H(X_1) + H(X_2 \mid X_1) + H(X_3 \mid X_1X_2) \\ &\quad + \ldots + H(X_k \mid X_1\cdots X_{k-1}) \\ &= \sum_{i=1}^{k} H(X_i \mid X_1X_2\cdots X_{i-1}) \end{aligned} \quad (3\text{-}2\text{-}15)$$

By applying the result $H(X) \geq H(X \mid Y)$, where $X = X_m$ and $Y = X_1X_2\cdots X_{m-1}$, in (3-2-15) we obtain

$$H(X_1X_2\cdots X_k) \leq \sum_{m=1}^{k} H(X_m) \quad (3\text{-}2\text{-}16)$$

with equality if and only if the random variables $X_1, X_2, \ldots, X_k$ are statistically independent.

3-2-2 Information Measures for Continuous Random Variables

The definition of mutual information given above for discrete random variables may be extended in a straightforward manner to continuous random variables. In particular, if X and Y are random variables with joint pdf $p(x, y)$ and marginal pdfs $p(x)$ and $p(y)$, the average mutual information between X and Y is defined as

$$I(X; Y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} p(x)p(y \mid x)\log\frac{p(y \mid x)p(x)}{p(x)p(y)}\,dx\,dy \quad (3\text{-}2\text{-}17)$$

Although the definition of the average mutual information carries over to

continuous random variables, the concept of self-information does not. The problem is that a continuous random variable requires an infinite number of binary digits to represent it exactly. Hence, its self-information is infinite and, therefore, its entropy is also infinite. Nevertheless, we shall define a quantity that we call the *differential entropy* of the continuous random variable X as

$$H(X) = -\int_{-\infty}^{\infty} p(x) \log p(x)\, dx \tag{3-2-18}$$

We emphasize that this quantity does *not* have the physical meaning of self-information, although it may appear to be a natural extension of the definition of entropy for a discrete random variable (see Problem 3-6).

By defining the average conditional entropy of X given Y as

$$H(X \mid Y) = -\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} p(x, y) \log p(x \mid y)\, dx\, dy \tag{3-2-19}$$

the average mutual information may be expressed as

$$I(X; Y) = H(X) - H(X \mid Y)$$

or, alternatively, as

$$I(X; Y) = H(Y) - H(Y \mid X)$$

In some cases of practical interest, the random variable X is discrete and Y is continuous. To be specific, suppose that X has possible outcomes x_i, $i = 1, 2, \ldots, n$, and Y is described by its marginal pdf $p(y)$. When X and Y are statistically dependent, we may express $p(y)$ as

$$p(y) = \sum_{i=1}^{n} p(y \mid x_i) P(x_i)$$

The mutual information provided about the event $X = x_i$ by the occurrence of the event $Y = y$ is

$$\begin{aligned} I(x_i; y) &= \log \frac{p(y \mid x_i) P(x_i)}{p(y) P(x_i)} \\ &= \log \frac{p(y \mid x_i)}{p(y)} \end{aligned} \tag{3-2-20}$$

Then, the average mutual information between X and Y is

$$I(X; Y) = \sum_{i=1}^{n} \int_{-\infty}^{\infty} p(y \mid x_i) P(x_i) \log \frac{p(y \mid x_i)}{p(y)}\, dy \tag{3-2-21}$$

Example 3-2-5

Suppose that X is a discrete random variable with two equally probable outcomes $x_1 = A$ and $x_2 = -A$. Let the conditional pdfs $p(y \mid x_i)$, $i = 1, 2$, be gaussian with mean x_i and variance σ^2. That is,

$$p(y \mid A) = \frac{1}{\sqrt{2\pi}\,\sigma} e^{-(y-A)^2/2\sigma^2} \tag{3-2-22}$$

$$p(y \mid -A) = \frac{1}{\sqrt{2\pi}\,\sigma} e^{-(y+A)^2/2\sigma^2} \tag{3-2-22}$$

The average mutual information obtained from (3-2-21) becomes

$$I(X;Y) = \frac{1}{2}\int_{-\infty}^{\infty} \left[p(y \mid A) \log \frac{p(y \mid A)}{p(y)} + p(y \mid -A) \log \frac{p(y \mid -A)}{p(y)} \right] dy \tag{3-2-23}$$

$$p(y) = \tfrac{1}{2}[p(y \mid A) + p(y \mid -A)] \tag{3-2-24}$$

In Chapter 7, it will be shown that the average mutual information $I(X;Y)$ given by (3-2-23) represents the channel capacity of a binary-input additive white gaussian noise channel.

3-3 CODING FOR DISCRETE SOURCES

In Section 3-2 we introduced a measure for the information content associated with a discrete random variable X. When X is the output of a discrete source, the entropy $H(X)$ of the source represents the average amount of information emitted by the source. In this section, we consider the process of encoding the output of a source, i.e., the process of representing the source output by a sequence of binary digits. A measure of the efficiency of a source-encoding method can be obtained by comparing the average number of binary digits per output letter from the source to the entropy $H(X)$.

The encoding of a discrete source having a finite alphabet size may appear, at first glance, to be a relatively simple problem. However, this is true only when the source is memoryless, i.e., when successive symbols from the source are statistically independent and each symbol is encoded separately. The discrete memoryless source (DMS) is by far the simplest model that can be devised for a physical source. Few physical sources, however, closely fit this idealized mathematical model. For example, successive output letters from a machine printing English text are expected to be statistically dependent. On the other hand, if the machine output is a computer program coded in Fortran, the sequence of output letters is expected to exhibit a much smaller dependence. In any case, we shall demonstrate that it is always more efficient to encode blocks of symbol instead of encoding each symbol separately. By making the block size sufficiently large, the average number of binary digits

per output letter from the source can be made arbitrarily close to the entropy of the source.

3-3-1 Coding for Discrete Memoryless Sources

Suppose that a DMS produces an output letter or symbol every τ_s seconds. Each symbol is selected from a finite alphabet of symbols x_i, $i = 1, 2, \ldots, L$, occurring with probabilities $P(x_i)$, $i = 1, 2, \ldots, L$. The entropy of the DMS in bits per source symbol is

$$H(X) = -\sum_{i=1}^{L} P(x_i) \log_2 P(x_i) \leq \log_2 L \tag{3-3-1}$$

where equality holds when the symbols are equally probable. The average number of bits per source symbol is $H(X)$ and the source rate in bits/s is defined as $H(X)/\tau_s$.

Fixed-Length Code Words First we consider a block encoding scheme that assigns a unique set of R binary digits to each symbol. Since there are L possible symbols, the number of binary digits per symbol required for unique encoding when L is a power of 2 is

$$R = \log_2 L \tag{3-3-2}$$

and, when L is not a power of 2, it is

$$R = \lfloor \log_2 L \rfloor + 1 \tag{3-3-3}$$

where $\lfloor x \rfloor$ denotes the largest integer less than x. The code rate R in bits per symbol is now R and, since $H(X) \leq \log_2 L$, it follows that $R \geq H(X)$.

The efficiency of the encoding for the DMS is defined as the ratio $H(X)/R$. We observe that when L is a power of 2 and the source letters are equally probable, $R = H(X)$. Hence, a fixed-length code of R bits per symbol attains 100% efficiency. However, if L is not a power of 2 but the source symbols are still equally probable, R differs from $H(X)$ by at most 1 bit per symbol. When $\log_2 L \gg 1$, the efficiency of this encoding scheme is high. On the other hand, when L is small, the efficiency of the fixed-length code can be increased by encoding a sequence of J symbols at a time. To accomplish the desired encoding, we require L^J unique code words. By using sequences of N binary digits, we can accommodate 2^N possible code words. N must be selected such that

$$N \geq J \log_2 L$$

Hence, the minimum integer value of N required is

$$N = \lfloor J \log_2 L \rfloor + 1 \tag{3-3-4}$$

Now the average number of bits per source symbol is $N/J = R$, and, thus, the

inefficiency has been reduced by approximately a factor of $1/J$ relative to the symbol-by-symbol encoding described above. By making J sufficiently large, the efficiency of the encoding procedure, measured by the ratio $JH(X)/N$, can be made as close to unity as desired.

The encoding methods described above introduce no distortion since the encoding of source symbols or blocks of symbols into code words is unique. This type of encoding is called *noiseless*.

Now, suppose we attempt to reduce the code rate R by relaxing the condition that the encoding process be unique. For example, suppose that only a fraction of the L^J blocks of symbols is encoded uniquely. To be specific, let us select the $2^N - 1$ most probable J-symbol blocks and encode each of them uniquely, while the remaining $L^J - (2^N - 1)$ J-symbol blocks are represented by the single remaining code word. This procedure results in a decoding failure or (distortion) probability of error every time a low probability block is mapped into this single code word. Let P_e denote this probability of error. Based on this block encoding procedure, Shannon (1948a) proved the following source coding theorem.

Source Coding Theorem I

Let X be the ensemble of letters from a DMS with finite entropy $H(X)$. Blocks of J symbols from the source are encoded into code words of length N from a binary alphabet. For any $\varepsilon > 0$, the probability P_e of a block decoding failure can be made arbitrarily small if

$$R \equiv \frac{N}{J} \geq H(X) + \varepsilon \tag{3-3-5}$$

and J is sufficiently large. Conversely, if

$$R \leq H(X) - \varepsilon \tag{3-3-6}$$

then P_e becomes arbitrarily close to 1 as J is made sufficiently large.

From this theorem, we observe that the average number of bits per symbol required to encode the output of a DMS with arbitrarily small probability of decoding failure is lower bounded by the source entropy $H(X)$. On the other hand, if $R < H(X)$, the decoding failure rate approaches 100% as J is arbitrarily increased.

Variable-Length Code Words When the source symbols are not equally probable, a more efficient encoding method is to use variable-length code

TABLE 3-3-1 VARIABLE-LENGTH CODES

Letter	$P(a_k)$	Code I	Code II	Code III
a_1	$\frac{1}{2}$	1	0	0
a_2	$\frac{1}{4}$	00	10	01
a_3	$\frac{1}{8}$	01	110	011
a_4	$\frac{1}{8}$	10	111	111

words. An example of such encoding is the Morse code, which dates back to the nineteenth century. In the Morse code, the letters that occur more frequently are assigned short code words and those that occur infrequently are assigned long code words. Following this general philosophy, we may use the probabilities of occurrence of the different source letters in the selection of the code words. The problem is to devise a method for selecting and assigning the code words to source letters. This type of encoding is called *entropy coding.*

For example, suppose that a DMS with output letters a_1, a_2, a_3, a_4 and corresponding probabilities $P(a_1) = \frac{1}{2}$, $P(a_2) = \frac{1}{4}$, and $P(a_3) = P(a_4) = \frac{1}{8}$ is encoded as shown in Table 3-3-1. Code I is a variable-length code that has a basic flaw. To see the flaw, suppose we are presented with the sequence 001001 .... Clearly, the first symbol corresponding to 00 is a_2. However, the next four bits are ambiguous (not uniquely decodable). They may be decoded either as a_4a_3 or as $a_1a_2a_1$. Perhaps, the ambiguity can be resolved by waiting for additional bits, but such a decoding delay is highly undesirable. We shall only consider codes that are decodable *instantaneously*, that is, without any decoding delay.

Code II in Table 3-3-1 is *uniquely decodable* and *instantaneously decodable.* It is convenient to represent the code words in this code graphically as terminal nodes of a tree, as shown in Fig. 3-3-1. We observe that the digit 0 indicates the end of a code word for the first three code words. This characteristic plus the fact that no code word is longer than three binary digits makes this code instantaneously decodable. Note that no code word in this code is a prefix of any other code word. In general, the *prefix condition* requires that for a given code word $\mathbf{C}_k$ of length k having elements $(b_1, b_2, \ldots, b_k)$, there is no other code word of length $l < k$ with elements $(b_1, b_2, \ldots, b_l)$ for $1 \leq l \leq k - 1$. In

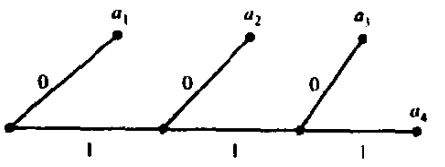


FIGURE 3-3-1 Code tree for code II in Table 3-3-1.

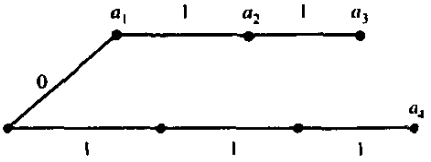


FIGURE 3-3-2 Code tree for code III in Table 3-3-1.

other words, there is no code word of length $l < k$ that is identical to the first l binary digits of another code word of length $k > l$. This property makes the code words instantaneously decodable.

Code III given in Table 3-3-1 has the tree structure shown in Fig. 3-3-2. We note that in this case the code is uniquely decodable but *not* instantaneously decodable. Clearly, this code does *not* satisfy the prefix condition.

Our main objective is to devise a systematic predure for constructing uniquely decodable variable-length codes that are efficient in the sense that the average number of bits per source letter, defined as the quantity

$$\bar{R} = \sum_{k=1}^{L} n_k P(a_k) \tag{3-3-7}$$

is minimized. The conditions for the existence of a code that satisfies the prefix condition are given by the Kraft inequality.

Kraft Inequality A necessary and sufficient condition for the existence of a binary code with code words having lengths $n_1 \leq n_2 \leq \ldots \leq n_L$ that satisfy the prefix condition is

$$\sum_{k=1}^{L} 2^{-n_k} \leq 1 \tag{3-3-8}$$

First, we prove that (3-3-8) is a sufficient condition for the existence of a code that satisfies the prefix condition. To construct such a code, we begin with a full binary tree of order $n = n_L$ that has 2^n terminal nodes and two nodes of order k stemming from each node of order $k-1$, for each k, $1 \leq k \leq n$. Let us select any node of order n_1 as the first code word $\mathbf{C}_1$. This choice eliminates 2^{n-n_1} terminal nodes (or the fraction 2^{-n_1} of the 2^n terminal nodes). From the remaining available nodes of order n_2, we select one node for the second code word $\mathbf{C}_2$. This choice eliminates 2^{n-n_2} terminal nodes (or the fraction 2^{-n_2} of the 2^n terminal nodes). This process continues until the last code word is assigned at terminal node $n = n_L$. Since, at the node of order $j < L$, the fraction of the number of terminal nodes eliminated is

$$\sum_{k=1}^{j} 2^{-n_k} < \sum_{k=1}^{L} 2^{-n_k} \leq 1$$

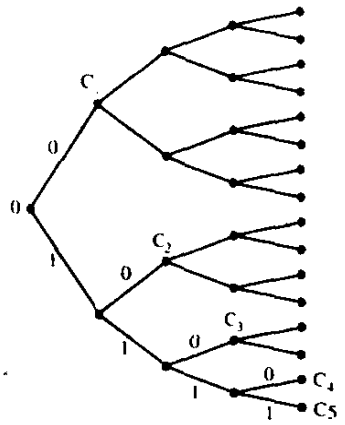


FIGURE 3-3-3 Construction of a binary tree code embedded in a full tree.

there is always a node of order $k > j$ available to be assigned to the next code word. Thus, we have constructed a code tree that is embedded in the full tree of 2^n nodes as illustrated in Fig. 3-3-3, for a tree having 16 terminal nodes and a source output consisting of five letters with $n_1 = 1$, $n_2 = 2$, $n_3 = 3$, and $n_4 = n_5 = 4$.

To prove that (3-3-8) is a necessary condition, we observe that in the code tree of order $n = n_L$, the number of terminal nodes eliminated from the total number of 2^n terminal nodes is

$$\sum_{k=1}^{L} 2^{n-n_k} \leq 2^n$$

Hence,

$$\sum_{k=1}^{L} 2^{-n_k} \leq 1$$

and the proof of (3-3-8) is complete.

The Kraft inequality may be used to prove the following (noiseless) source coding theorem, which applies to codes that satisfy the prefix condition.

Source Coding Theorem II

Let X be the ensemble of letters from a DMS with finite entropy $H(X)$, and output letters x_k, $1 \leq k \leq L$, with corresponding probabilities of occurrence p_k, $1 \leq k \leq L$. It is possible to construct a code that satisfies the prefix condition and has an average length $\bar{R}$ that satisfies the inequalities

$$H(X) \leq \bar{R} < H(X) + 1 \qquad (3\text{-}3\text{-}9)$$

To establish the lower bound in (3-3-9), we note that for code words that have length n_k, $1 \leq k \leq L$, the difference $H(X) - \bar{R}$ may be expressed as

$$H(X) - \bar{R} = \sum_{k=1}^{L} p_k \log_2 \frac{1}{p_k} - \sum_{k=1}^{L} p_k n_k$$

$$= \sum_{k=1}^{L} p_k \log_2 \frac{2^{-n_k}}{p_k} \tag{3-3-10}$$

Use of the inequality $\ln x \leq x - 1$ in (3-3-10) yields

$$H(X) - \bar{R} \leq (\log_2 e) \sum_{k=1}^{L} p_k \left(\frac{2^{-n_k}}{p_k} - 1 \right)$$

$$\leq (\log_2 e) \left(\sum_{k=1}^{L} 2^{-n_k} - 1 \right) \leq 0$$

where the last inequality follows from the Kraft inequality. Equality holds if and only if $p_k = 2^{-n_k}$ for $1 \leq k \leq L$.

The upper bound in (3-3-9) may be established under the constraint that n_k, $1 \leq k \leq L$, are integers, by selecting the $\{n_k\}$ such that $2^{-n_k} \leq p_k < 2^{-n_k+1}$. But if the terms $p_k \geq 2^{-n_k}$ are summed over $1 \leq k \leq L$, we obtain the Kraft inequality, for which we have demonstrated that there exists a code that satisfies the prefix condition. On the other hand, if we take the logarithm of $p_k < 2^{-n_k+1}$, we obtain

$$\log p_k < -n_k + 1$$

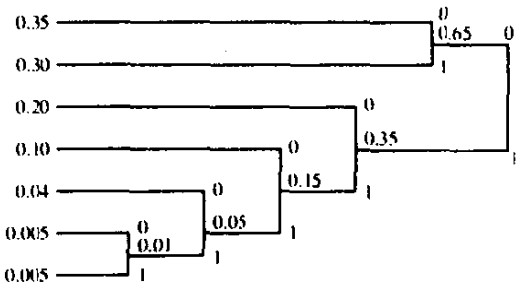
or, equivalently,

$$n_k < 1 - \log p_k \tag{3-3-11}$$

If we multiply both sides of (3-3-11) by p_k and sum over $1 \leq k \leq L$, we obtain the desired upper bound given in (3-3-9). This completes the proof of (3-3-9).

We have now established that variable length codes that satisfy the prefix condition are efficient source codes for any DMS with source symbols that are not equally probable. Let us now describe an algorithm for constructing such codes.

Huffman Coding Algorithm Huffman (1952) devised a variable-length encoding algorithm, based on the source letter probabilities $P(x_i)$, $i = 1, 2, \ldots, L$. This algorithm is optimum in the sense that the average number of binary digits required to represent the source symbols is a minimum, subject to the constraint that the code words satisfy the prefix condition, as defined above, which allows the received sequence to be uniquely and instantaneously decodable. We illustrate this encoding algorithm by means of two examples.



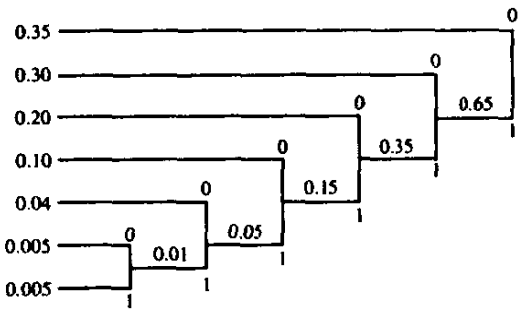
Letter	Probability	Self-information	Code
x_1	0.35	1.5146	00
x_2	0.30	1.7370	01
x_3	0.20	2.3219	10
x_4	0.10	3.3219	110
x_5	0.04	4.6439	1110
x_6	0.005	7.6439	11110
x_7	0.005	7.6439	11111
	$H(X) = 2.11$	$\bar{R} = 2.21$	

FIGURE 3-3-4 An example of variable-length-source encoding for a DMS.

Example 3-3-1

Consider a DMS with seven possible symbols $x_1, x_2, \ldots, x_7$ having the probabilities of occurrence illustrated in Fig. 3-3-4. We have ordered the source symbols in decreasing order of the probabilities, i.e., $P(x_1) > P(x_2) > \ldots > P(x_7)$. We begin the encoding process with the two least probable symbols x_6 and x_7. These two symbols are tied together as shown in Fig. 3-3-4, with the upper branch assigned a 0 and the lower branch assigned a 1. The probabilities of these two branches are added together at the node where the two branches meet to yield the probability 0.01. Now we have the source symbols $x_1, \ldots, x_5$ plus a new symbol, say x_6', obtained by combining x_6 and x_7. The next step is to join the two least probable symbols from the set $x_1, x_2, x_3, x_4, x_5, x_6'$. These are x_5 and x_6', which have a combined probability of 0.05. The branch from x_5 is assigned a 0 and the branch from x_6' is assigned a 1. This procedure continues until we exhaust the set of possible source letters. The result is a code tree with branches that contain the desired code words. The code words are obtained by beginning at the rightmost node in the tree and proceeding to the left. The resulting code words are listed in Fig. 3-3-4. The average number of binary digits per symbol for this code is $\bar{R} = 2.21$ bits/symbol. The entropy of the source is 2.11 bits/symbol.

We make the observation that the code is not necessarily unique. For example, at the next to the last step in the encoding procedure, we have a tie between x_1 and x_3', since these symbols are equally probable. At this point, we chose to pair x_1 with x_2. An alternative is to pair x_2 with x_3'. If we choose this



Letter	Code
x_1	0
x_2	10
x_3	110
x_4	1110
x_5	11110
x_6	111110
x_7	111111

$\bar{R} = 2.21$

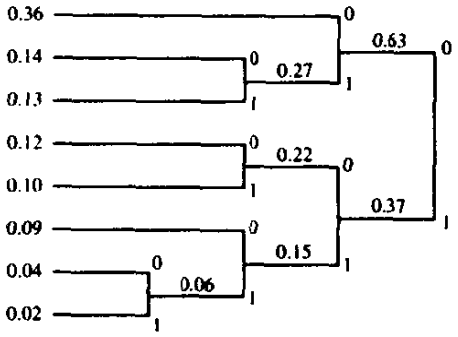
FIGURE 3-3-5 An alternative code for the DMS in Example 3-3-1.

pairing, the resulting code is illustrated in Fig. 3-3-5. The average number of bits per source symbol for this code is also 2.21. Hence, the resulting codes are equally efficient. Secondly, the assignment of a 0 to the upper branch and a 1 to the lower (less probable) branch is arbitrary. We may simply reverse the assignment of a 0 and 1 and still obtain an efficient code satisfying the prefix condition.

Example 3-3-2

As a second example, let us determine the Huffman code for the output of a DMS illustrated in Fig. 3-3-6. The entropy of this source is $H(X) = 2.63$ bits/symbol. The Huffman code as illustrated in Fig. 3-3-6 has an average length of $\bar{R} = 2.70$ bits/symbol. Hence, its efficiency is 0.97.

FIGURE 3-3-6 Huffman code for Example 3-3-2.



Letter	Code
x_1	00
x_2	010
x_3	011
x_4	100
x_5	101
x_6	110
x_7	1110
x_8	1111

$H(X) = 2.63$ $\bar{R} = 2.70$

The variable-length encoding (Huffman) algorithm described in the above examples generates a prefix code having an $\bar{R}$ that satisfies (3-3-9). However, instead of encoding on a symbol-by-symbol basis, a more efficient procedure is to encode blocks of J symbols at a time. In such a case, the bounds in (3-3-9) of source coding theorem II become

$$JH(X) \leq \bar{R}_J < JH(X) + 1, \tag{3-3-12}$$

since the entropy of a J-symbol block from a DMS is $JH(X)$, and $\bar{R}_J$ is the average number of bits per J-symbol blocks. If we divide (3-3-12) by J, we obtain

$$H(X) \leq \frac{\bar{R}_J}{J} < H(X) + \frac{1}{J} \tag{3-3-13}$$

where $\bar{R}_J/J \equiv \bar{R}$ is the average number of bits per source symbol. Hence $\bar{R}$ can be made as close to $H(X)$ as desired by selecting J sufficiently large.

Example 3-3-3

The output of a DMS consists of *letters* x_1, x_2, and x_3 with probabilities 0.45, 0.35, and 0.20, respectively. The entropy of this source is $H(X) = 1.518$ bits/symbol. The Huffman code for this source, given in Table 3-3-2, requires $\bar{R}_1 = 1.55$ bits/symbol and results in an efficiency of 97.9%. If pairs of symbols are encoded by means of the Huffman algorithm, the resulting code is as given in Table 3-3-3. The entropy of the source output for pairs of letters is $2H(X) = 3.036$ bits/symbol pair. On the other hand, the Huffman code requires $\bar{R}_2 = 3.0675$ bits/symbol pair. Thus, the efficiency of the encoding increases to $2H(X)/\bar{R}_2 = 0.990$ or, equivalently, to 99.0%.

In summary, we have demonstrated that efficient encoding for a DMS may be done on a symbol-by-symbol basis using a variable-length code based on

TABLE 3-3-2 HUFFMAN CODE FOR EXAMPLE 3-3-3

Letter	Probability	Self-information	Code
x_1	0.45	1.156	1
x_2	0.35	1.520	00
x_3	0.20	2.330	01

$H(X) = 1.518$ bits/letter
$\bar{R}_1 = 1.55$ bits/letter
Efficiency = 97.9%

TABLE 3-3-3 HUFFMAN CODE FOR ENCODING PAIRS OF LETTERS

Letter pair	Probability	Self-information	Code
x_1x_1	0.2025	2.312	10
x_1x_2	0.1575	2.676	001
x_2x_1	0.1575	2.676	010
x_2x_2	0.1225	3.039	011
x_1x_3	0.09	3.486	111
x_3x_1	0.09	3.486	0000
x_2x_3	0.07	3.850	0001
x_3x_2	0.07	3.850	1100
x_3x_3	0.04	4.660	1101
	$2H(X) = 3.036$ bits/letter pair		
	$\bar{R}_2 = 3.0675$ bits/letter pair		
	$\frac{1}{2}\bar{R}_2 = 1.534$ bits/letter		
	Efficiency = 99.0%		

the Huffman algorithm. Furthermore, the efficiency of the encoding procedure is increased by encoding blocks of J symbols at a time. Thus, the output of a DMS with entropy $H(X)$ may be encoded by a variable-length code with an average number of bits per source letter that approaches $H(X)$ as closely as desired.

3-3-2 Discrete Stationary Sources

In the previous section, we described the efficient encoding of the output of a DMS. In this section, we consider discrete sources for which the sequence of output letters is statistically dependent. We limit our treatment to sources that are statistically stationary.

Let us evaluate the entropy of any sequence of letters from a stationary source. From the definition in (3-2-13) and the result given in (3-2-15), the entropy of a block of random variables $X_1X_2 \cdots X_k$ is

$$H(X_1X_2 \cdots X_k) = \sum_{i=1}^{k} H(X_i \mid X_1X_2 \cdots X_{i-1}) \tag{3-3-14}$$

where $H(X_i \mid X_1X_2 \cdots X_{i-1})$ is the conditional entropy of the ith symbol from the source given the previous $i-1$ symbols. The entropy per letter for the k-symbol block is defined as

$$H_k(X) = \frac{1}{k} H(X_1X_2 \cdots X_k) \tag{3-3-15}$$

We define the information content of a stationary source as the entropy per letter in (3-3-15) in the limit as $k \to \infty$. That is,

$$H_\infty(X) = \lim_{k\to\infty} H_k(X) = \lim_{k\to\infty} \frac{1}{k} H(X_1X_2 \cdots X_k) \tag{3-3-16}$$

The existence of this limit is established below.

As an alternative, we may define the entropy per letter from the source in terms of the conditional entropy $H(X_k \mid X_1X_2 \cdots X_{k-1})$ in the limit as k approaches infinity. Fortunately, this limit also exists and is identical to the limit in (3-3-16). That is,

$$H_\infty(X) = \lim_{k \to \infty} H(X_k \mid X_1X_2 \cdots X_{k-1}) \tag{3-3-17}$$

This result is also established below. Our development follows the approach in Gallager (1968).

First, we show that

$$H(X_k \mid X_1X_2 \cdots X_{k-1}) \leq H(X_{k-1} \mid X_1X_2 \cdots X_{k-2}) \tag{3-3-18}$$

for $k \geq 2$. From our previous result that conditioning on a random variable cannot increase entropy, we have

$$H(X_k \mid X_1X_2 \cdots X_{k-1}) \leq H(X_k \mid X_2X_3 \cdots X_{k-1}) \tag{3-3-19}$$

From the stationarity of the source, we have

$$H(X_k \mid X_2X_3 \cdots X_{k-1}) = H(X_{k-1} \mid X_1X_2 \cdots X_{k-2}) \tag{3-3-20}$$

Hence, (3-3-18) follows immediately. This result demonstrates that $H(X_k \mid X_1X_2 \cdots X_{k-1})$ is a nonincreasing sequence in k.

Second, we have the result

$$H_k(X) \geq H(X_k \mid X_1X_2 \cdots X_{k-1}) \tag{3-3-21}$$

which follows immediately from (3-3-14) and (3-3-15) and the fact that the last term in the sum of (3-3-14) is a lower bound on each of the other $k - 1$ terms.

Third, from the definition of $H_k(X)$, we may write

$$\begin{aligned} H_k(X) &= \frac{1}{k}[H(X_1X_2 \cdots X_{k-1}) + H(X_k \mid X_1 \cdots X_{k-1})] \\ &= \frac{1}{k}[(k-1)H_{k-1}(X) + H(X_k \mid X_1 \cdots X_{k-1})] \\ &\leq \frac{k-1}{k} H_{k-1}(X) + \frac{1}{k} H_k(X) \end{aligned}$$

which reduces to

$$H_k(X) \leq H_{k-1}(X) \tag{3-3-22}$$

Hence, $H_k(X)$ is a nonincreasing sequence in k.

Since $H_k(X)$ and the conditional entropy $H(X_k \mid X_1 \cdots X_{k-1})$ are both

nonnegative and nonincreasing with k, both limits must exist. Their limiting forms can be established by using (3-3-14) and (3-3-15) to express $H_{k+j}(X)$ as

$$\begin{aligned} H_{k+j}(X) = \frac{1}{k+j} H(X_1 X_2 \cdots X_{k-1}) \\ + \frac{1}{k+j}[H(X_k \mid X_1 \cdots X_{k-1}) + H(X_{k+1} \mid X_1 \cdots X_k) \\ + \ldots + H(X_{k+j} \mid X_1 \cdots X_{k+j-1})] \end{aligned}$$

Since the conditional entropy is nonincreasing, the first term in the square brackets serves as an upper bound on the other terms. Hence,

$$H_{k+j}(X) \leq \frac{1}{k+j} H(X_1 X_2 \cdots X_{k-1}) + \frac{j+1}{k+j} H(X_k \mid X_1 X_2 \cdots X_{k-1}) \tag{3-3-23}$$

For a fixed k, the limit of (3-3-23) as $j \to \infty$ yields

$$H_\infty(X) \leq H(X_k \mid X_1 X_2 \cdots X_{k-1}) \tag{3-3-24}$$

But (3-3-24) is valid for all k; hence, it is valid for $k \to \infty$. Therefore,

$$H_\infty(X) \leq \lim_{k \to \infty} H(X_k \mid X_1 X_2 \cdots X_{k-1}) \tag{3-3-25}$$

On the other hand, from (3-3-21), we obtain in the limit as $k \to \infty$,

$$H_\infty(X) \geq \lim_{k \to \infty} H(X_k \mid X_1 X_2 \cdots X_{k-1}) \tag{3-3-26}$$

which establishes (3-3-17).

Now suppose we have a discrete stationary source that emits J letters with $H_J(X)$ as the entropy per letter. We can encode the sequence of J letters with a variable-length Huffman code that satisfies the prefix condition by following the procedure described in the previous section. The resulting code has an average number of bits for the J-letter block that satisfies the condition

$$H(X_1 \cdots X_J) \leq \bar{R}_J < H(X_1 \cdots X_J) + 1 \tag{3-3-27}$$

By dividing each term of (3-3-27) by J, we obtain the bounds on the average number $\bar{R} = \bar{R}_J/J$ of bits per source letter as

$$H_J(X) \leq \bar{R} < H_J(X) + \frac{1}{J} \tag{3-3-28}$$

By increasing the block size J, we can approach $H_J(X)$ arbitrarily closely, and in the limit as $J \to \infty$, $\bar{R}$ satisfies

$$H_\infty(X) \leq \bar{R} < H_\infty(X) + \varepsilon \tag{3-3-29}$$

where ε approaches zero as $1/J$. Thus, efficient encoding of stationary sources is accomplished by encoding large blocks of symbols into code words. We should emphasize, however, that the design of the Huffman code requires knowledge of the joint pdf for the J-symbol blocks.

ıe Lempel–Ziv Algorithm

From our preceding discussion, we have observed that the Huffman coding algorithm yields optimal source codes in the sense that the code words satisfy the prefix condition and the average block length is a minimum. To design a Huffman code for a DMS, we need to know the probabilities of occurrence of all the source letters. In the case of a discrete source with memory, we must know the joint probabilities of blocks of length $n \geq 2$. However, in practice, the statistics of a source output are often unknown. In principle, it is possible to estimate the probabilities of the discrete source output by simply observing a long information sequence emitted by the source and obtaining the probabilities empirically. Except for the estimation of the marginal probabilities $\{p_k\}$, corresponding to the frequency of occurrence of the individual source output letters, the computational complexity involved in estimating joint probabilities is extremely high. Consequently, the application of the Huffman coding method to source coding for many real sources with memory is generally impractical.

In contrast to the Huffman coding algorithm, the Lempel–Ziv source coding algorithm is designed to be independent of the source statistics. Hence, the Lempel–Ziv algorithm belongs to the class of *universal source coding algorithms*. It is a variable-to-fixed-length algorithm, where the encoding is performed as described below.

In the Lempel–Ziv algorithm, the sequence at the output of the discrete source is parsed into variable-length blocks, which are called *phrases*. A new phrase is introduced every time a block of letters from the source differs from some previous phrase in the last letter. The phrases are listed in a dictionary, which stores the location of the existing phrases. In encoding a new phrase, we simply specify the location of the existing phrase in the dictionary and append the new letter.

As an example, consider the binary sequence

$$1010110100100111010100001100111010110001101 1$$

Parsing the sequence as described above produces the following phrases:

1, 0, 10, 11, 01, 00, 100, 111, 010, 1000, 011, 001, 110, 101, 10001, 1011

We observe that each phrase in the sequence is a concatenation of a previous phrase with a new output letter from the source. To encode the phrases, we

TABLE 3-3-4 DICTIONARY FOR LEMPEL-ZIV ALGORITHM

	Dictionary location	Dictionary contents	Code word
1	0001	1	00001
2	0010	0	00000
3	0011	10	00010
4	0100	11	00011
5	0101	01	00101
6	0110	00	00100
7	0111	100	00110
8	1000	111	01001
9	1001	010	01010
10	1010	1000	01110
11	1011	011	01011
12	1100	001	01101
13	1101	110	01000
14	1110	101	00111
15	1111	10001	10101
16		1011	11101

construct a dictionary as shown in Table 3-3-4. The dictionary locations are numbered consecutively, beginning with 1 and counting up, in this case to 16, which is the number of phrases in the sequence. The different phrases corresponding to each location are also listed, as shown. The codewords are determined by listing the dictionary location (in binary form) of the previous phrase that matches the new phrase in all but the last location. Then, the new output letter is appended to the dictionary location of the previous phrase. Initially, the location 0000 is used to encode a phrase that has not appeared previously.

The source decoder for the code constructs an identical table at the receiving end of the communication system and decodes the received sequence accordingly.

It should be observed that the table encoded 44 source bits into 16 code words of five bits each, resulting in 80 coded bits. Hence, the algorithm provided no data compression at all. However, the inefficiency is due to the fact that the sequence we have considered is very short. As the sequence is increased in length, the encoding procedure becomes more efficient and results in a compressed sequence at the output of the source.

How do we select the overall length of the table? In general, no matter how large the table is, it will eventually overflow. To solve the overflow problem, the source encoder and source decoder must agree to remove phrases from the respective dictionaries that are not useful and substitute new phrases in their place.

The Lempel–Ziv algorithm is widely used in the compression of computer files. The "compress" and "uncompress" utilities under the UNIX© operating system and numerous algorithms under the MS-DOS operating system are implementations of various versions of this algorithm.

3-4 CODING FOR ANALOG SOURCES—OPTIMUM QUANTIZATION

As indicated in Section 3-1, an analog source emits a message waveform $x(t)$ that is a sample function of a stochastic process $X(t)$. When $X(t)$ is a bandlimited, stationary stochastic process, the sampling theorem allows us to represent $X(t)$ by a sequence of uniform samples taken at the Nyquist rate.

By applying the sampling theorem, the output of an analog source is converted to an equivalent discrete-time sequence of samples. The samples are then quantized in amplitude and encoded. One type of simple encoding is to represent each discrete amplitude level by a sequence of binary digits. Hence, if we have L levels, we need $R = \log_2 L$ bits per sample if L is a power of 2, or $R = \lfloor \log_2 L \rfloor + 1$ if L is not a power of 2. On the other hand, if the levels are not equally probable, and the probabilities of the output levels are known, we may use Huffman coding (also called *entropy coding*) to improve the efficiency of the encoding process.

Quantization of the amplitudes of the sampled signal results in data compression but it also introduces some distortion of the waveform or a loss of signal fidelity. The minimization of this distortion is considered in this section. Many of the results given in this section apply directly to a discrete-time, continuous amplitude, memoryless gaussian source. Such a source serves as a good model for the residual error in a number of source coding methods described in Section 3-5.

3-4-1 Rate-Distortion Function

Let us begin the discussion of signal quantization by considering the distortion introduced when the samples from the information source are quantized to a fixed number of bits. By the term "distortion," we mean some measure of the difference between the actual source samples $\{x_k\}$ and the corresponding quantized values $\tilde{x}_k$, which we denote by $d\{x_k, \tilde{x}_k\}$. For example, a commonly used distortion measure is the *squared-error distortion*, defined as

$$d(x_k, \tilde{x}_k) = (x_k - \tilde{x}_k)^2 \tag{3-4-1}$$

which is used to characterize the quantization error in PCM in Section 3-5-1. Other distortion measures may take the general form

$$d(x_k, \tilde{x}_k) = |x - \tilde{x}_k|^p \tag{3-4-2}$$

where p takes values from the set of positive integers. The case $p = 2$ has the advantage of being mathematically tractable.

If $d(x_k, \tilde{x}_k)$ is the distortion measure per letter, the distortion between a sequence of n samples $\mathbf{X}_n$ and the corresponding n quantized values $\tilde{\mathbf{X}}_n$ is the average over the n source output samples, i.e.,

$$d(\mathbf{X}_n, \tilde{\mathbf{X}}_n) = \frac{1}{n} \sum_{k=1}^{n} d(x_k, \tilde{x}_k) \tag{3-4-3}$$

The source output is a random process, and, hence, the n samples in $\mathbf{X}_n$ are random variables. Therefore, $d(\mathbf{X}_n, \tilde{\mathbf{X}}_n)$ is a random variable. Its expected value is defined as the distortion D, i.e.,

$$D = E[d(\mathbf{X}_n, \tilde{\mathbf{X}}_n)] = \frac{1}{n} \sum_{k=1}^{n} E[d(x_k, \tilde{x}_k)] = E[d(x, \tilde{x})] \tag{3-4-4}$$

where the last step follows from the assumption that the source output process is stationary.

Now suppose we have a memoryless source with a continuous-amplitude output $\mathbf{X}$ that has a pdf $p(x)$, a quantized amplitude output alphabet $\tilde{\mathbf{X}}$, and a per letter distortion measure $d(x, \tilde{x})$, where $x \in \mathbf{X}$ and $\tilde{x} \in \tilde{\mathbf{X}}$. Then, the minimum rate in bits per source output that is required to represent the output $\mathbf{X}$ of the memoryless source with a distortion less than or equal to D is called the *rate-distortion function* $R(D)$ and is defined as

$$R(D) = \min_{p(\tilde{x}|x): E[d(\mathbf{X}, \tilde{\mathbf{X}})] \leq D} I(\mathbf{X}, \tilde{\mathbf{X}}) \tag{3-4-5}$$

where $I(\mathbf{X}; \tilde{\mathbf{X}})$ is the average mutual information between $\mathbf{X}$ and $\tilde{\mathbf{X}}$. In general, the rate $R(D)$ decreases as D increases or, conversely, $R(D)$ increases as D decreases.

One interesting model of a continuous-amplitude, memoryless information source is the gaussian source model. In this case, Shannon proved the following fundamental theorem on the rate-distortion function.

Theorem: Rate-Distortion Function for a Memoryless Gaussian Source (Shannon, 1959a)

The minimum information rate necessary to represent the output of a discrete-time, continuous-amplitude memoryless gaussian source based on a mean-square-error distortion measure per symbol (single letter distortion measure) is

$$R_g(D) = \begin{cases} \frac{1}{2} \log_2 (\sigma_x^2 / D) & (0 \leq D \leq \sigma_x^2) \\ 0 & (D > \sigma_x^2) \end{cases} \tag{3-4-6}$$

where σ_x^2 is the variance of the gaussian source output.

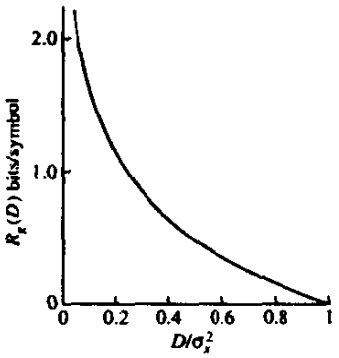


FIGURE 3-4-1 Rate distortion function for a continuous-amplitude memoryless gaussian source.

We should note that (3-4-6) implies that no information need be transmitted when the distortion $D \geq \sigma_x^2$. Specifically, $D = \sigma_x^2$ can be obtained by using zeros in the reconstruction of the signal. For $D > \sigma_x^2$, we can use statistically independent, zero-mean gaussian noise samples with a variance of $D - \sigma_x^2$ for the reconstruction. $R_g(D)$ is plotted in Fig. 3-4-1.

The rate distortion function $R(D)$ of a source is associated with the following basic source coding theorem in information theory.

Theorem: Source Coding with a Distortion Measure (Shannon, 1959a)

There exists an encoding scheme that maps the source output into code words such that for any given distortion D, the minimum rate $R(D)$ bits per symbol (sample) is sufficient to reconstruct the source output with an average distortion that is arbitrarily close to D.

It is clear, therefore, that the rate distortion function $R(D)$ for any source represents a lower bound on the source rate that is possible for a given level of distortion.

Let us return to the result in (3-4-6) for the rate distortion function of a memoryless gaussian source. If we reverse the functional dependence between D and R, we may express D in terms of R as

$$D_g(R) = 2^{-2R}\sigma_x^2 \tag{3-4-7}$$

This funcion is called the *distortion-rate function* for the discrete-time, memoryless gaussian source.

When we express the distortion in (3-4-7) in dB, we obtain

$$10 \log_{10} D_g(R) = -6R + 10 \log_{10} \sigma_x^2 \tag{3-4-8}$$

Note that the mean square distortion decreases at a rate of 6 dB/bit.

Explicit results on the rate distortion functions for memoryless non-gaussian sources are not available. However, there are useful upper and lower bounds

on the rate distortion function for any discrete-time, continuous-amplitude, memoryless source. An upper bound is given by the following theorem.

Theorem: Upper Bound on $R(D)$

The rate-distortion function of a memoryless, continuous-amplitude source with zero mean and finite variance σ_x^2 with respect to the mean-square-error distortion measure is upper bounded as

$$R(D) \leq \tfrac{1}{2} \log_2 \frac{\sigma_x^2}{D} \qquad (0 \leq D \leq \sigma_x^2) \tag{3-4-9}$$

A proof of this theorem is given by Berger (1971). It implies that the gaussian source requires the maximum rate among all other sources for a specified level of mean square distortion. Thus, the rate distortion $R(D)$ of any continuous-amplitude, memoryless source with zero mean and finite variance σ_x^2 satisfies the condition $R(D) \leq R_g(D)$. Similarly, the distortion-rate function of the same source satisfies the condition

$$D(R) \leq D_g(R) = 2^{-2R}\sigma_x^2 \tag{3-4-10}$$

A lower bound on the rate-distortion function also exists. This is called the *Shannon lower bound* for a mean-square-error distortion measure, and is given as

$$R^*(D) = H(X) - \tfrac{1}{2} \log_2 2\pi e D \tag{3-4-11}$$

where $H(X)$ is the differential entropy of the continuous-amplitude, memoryless source. The distortion-rate function corresponding to (3-4-11) is

$$D^*(R) = \frac{1}{2\pi e} 2^{-2[R - H(X)]} \tag{3-4-12}$$

Therefore, the rate-distortion function for any continuous-amplitude, memoryless source is bounded from above and below as

$$R^*(D) \leq R(D) \leq R_g(D) \tag{3-4-13}$$

and the corresponding distortion-rate function is bounded as

$$D^*(R) \leq D(R) \leq D_g(R) \tag{3-4-14}$$

The differential entropy of the memoryless gaussian source is

$$H_g(X) = \tfrac{1}{2} \log_2 2\pi e \sigma_x^2 \tag{3-4-15}$$

so that the lower bound $R^*(D)$ in (3-4-11) reduces to $R_g(D)$. Now, if we

express $D^*(R)$ in terms of decibels and normalize it by setting $\sigma_x^2 = 1$ [or dividing $D^*(R)$ by σ_x^2], we obtain from (3-4-12) TABLE 3-4-2

$$10 \log_{10} D^*(R) = -6R - 6[H_g(X) - H(X)] \tag{3-4-16}$$

or, equivalently,

$$10 \log_{10} \frac{D_g(R)}{D^*(R)} = 6[H_g(X) - H(X)] \text{ dB}$$
$$= 6[R_g(D) - R^*(D)] \text{ dB} \tag{3-4-17}$$

The relations in (3-4-16) and (3-4-17) allow us to compare the lower bound in the distortion with the upper bound which is the distortion for the gaussian source. We note that $D^*(R)$ also decreases at -6 dB/bit. We should also mention that the differential entropy $H(X)$ is upper-bounded by $H_g(X)$, as shown by Shannon (1948b).

Table 3-4-1 lists four pdfs that are models commonly used for source signal distributions. The table shows the differential entropies, the differences in rates in bits/sample, and the difference in distortion between the upper and lower bounds. Note that the gamma pdf shows the greatest deviation from the gaussian. The Laplacian pdf is the most similar to the gaussian, and the uniform pdf ranks second of the pdfs shown in the table. These results provide some benchmarks on the difference between the upper and lower bounds on distortion and rate.

Before concluding this section, let us consider a band-limited gaussian source with spectral density

$$\Phi(f) = \begin{cases} \sigma_x^2/2W & (|f| \leq W) \\ 0 & (|f| > W) \end{cases} \tag{3-4-18}$$

When the output of this source is sampled at the Nyquist rate, the samples are uncorrelated and, since the source is gaussian, they are also statistically

TABLE 3-4-1 DIFFERENTIAL ENTROPIES AND RATE DISTORTION COMPARISONS OF FOUR COMMON PDFs FOR SIGNAL MODELS

pdf	$p(x)$	$H(X)$	$R_g(D) - R^*(D)$ (bits/sample)	$D_g(R) - D^*(R)$ (dB)
Gaussian	$\frac{1}{\sqrt{2\pi}\,\sigma_x} e^{-x^2/2\sigma_x^2}$	$\frac{1}{2}\log_2(2\pi e\sigma_x^2)$	0	0
Uniform	$\frac{1}{2\sqrt{3}\,\sigma_x}$, $\lvert x\rvert \leq \sqrt{3}\,\sigma_x$	$\frac{1}{2}\log_2(12\sigma_x^2)$	0.255	1.53
Laplacian	$\frac{1}{\sqrt{2}\,\sigma_x} e^{-\sqrt{2}\lvert x\rvert/\sigma_x}$	$\frac{1}{2}\log_2(2e^2\sigma_x^2)$	0.104	0.62
Gamma	$\frac{\sqrt[4]{3}}{\sqrt{8\pi\sigma_x\lvert x\rvert}} e^{-\sqrt{3}\lvert x\rvert/2\sigma_x}$	$\frac{1}{2}\log_2(4\pi e^{0.423}\sigma_x^2/3)$	0.709	4.25

independent. Hence, the equivalent discrete-time gaussian source is memoryless. The rate-distortion function for each sample is given by (3-4-6). Therefore, the rate-distortion function for the band-limited white gaussian source in bits/s is

$$R_g(D) = W \log_2 \frac{\sigma_x^2}{D} \quad (0 \leq D \leq \sigma_x^2) \tag{3-4-19}$$

The corresponding distortion-rate function is

$$D_g(R) = 2^{-R/W}\sigma_x^2 \tag{3-4-20}$$

which, when expressed in decibels and normalized by σ_x^2, becomes

$$10 \log D_g(R)/\sigma_x^2 = -3R/W \tag{3-4-21}$$

The more general case in which the gaussian process is neither white nor band-limited has been treated by Gallager (1968) and Goblick and Holsinger (1967).

3-4-2 Scalar Quantization

In source encoding, the quantizer can be optimized if we know the probability density function of the signal amplitude at the input to the quantizer. For example, suppose that the sequence $\{x_n\}$ at the input to the quantizer has a pdf $p(x)$ and let $L = 2^R$ be the desired number of levels. We wish to design the optimum scalar quantizer that minimizes some function of the quantization error $q = \tilde{x} - x$, where $\tilde{x}$ is the quantized value of x. To elaborate, suppose that $f(\tilde{x} - x)$ denotes the desired function of the error. Then, the distortion resulting from quantization of the signal amplitude is

$$D = \int_{-\infty}^{\infty} f(\tilde{x} - x)p(x)\,dx \tag{3-4-22}$$

In general, an optimum quantizer is one that minimizes D by optimally selecting the output levels and the corresponding input range of each output level. This optimization problem has been considered by Lloyd (1982) and Max (1960), and the resulting optimum quantizer is usually called the *Lloyd–Max quantizer.*

For a uniform quantizer, the output levels are specified as $\tilde{x}_k = \frac{1}{2}(2k - 1)\Delta$, corresponding to an input signal amplitude in the range $(k - 1)\Delta \leq x < k\Delta$, where Δ is the step size. When the uniform quantizer is symmetric with an even number of levels, the average distortion in (3-4-22) may be expressed as

$$D = 2\sum_{k=1}^{L/2-1} \int_{(k-1)\Delta}^{k\Delta} f(\tfrac{1}{2}(2k - 1)\Delta - x)p(x)\,dx + 2\int_{(L/2-1)\Delta}^{\infty} f(\tfrac{1}{2}(2k - 1)\Delta - x)p(x)\,dx \tag{3-4-23}$$

TABLE 3-4-2 OPTIMUM STEP SIZES FOR UNIFORM QUANTIZATION OF A GAUSSIAN RANDOM VARIABLE

Number of output levels	Optimum step size Δ_{opt}	Minimum MSE D_{min}	$10 \log D_{min}$ (dB)
2	1.596	0.3634	−4.4
4	0.9957	0.1188	−9.25
8	0.5860	0.03744	−14.27
16	0.3352	0.01154	−19.38
32	0.1881	0.00349	−24.57

In this case, the minimization of D is carried out with respect to the step-size parameter Δ. By differentiating D with respect to Δ, we obtain

$$\sum_{k=1}^{L/2-1}(2k-1)\int_{(k-1)\Delta}^{k\Delta} f(\tfrac{1}{2}(2k-1)\Delta - x)p(x)\,dx + (L-1)\int_{-(L/2-1)\Delta}^{\infty} f'(\tfrac{1}{2}(L-1)\Delta - x)p(x)\,dx = 0 \quad (3\text{-}4\text{-}24)$$

where $f'(x)$ denotes the derivative of $f(x)$.

By selecting the error criterion function $f(x)$, the solution of (3-4-24) for the optimum step size can be obtained numerically on a digital computer for any given pdf $p(x)$. For the mean-square-error criterion, for which $f(x) = x^2$, Max (1960) evaluated the optimum step size Δ_{opt} and the minimum mean square error when the pdf $p(x)$ is zero-mean gaussian with unit variance. Some of these results are given in Table 3-4-2. We observe that the minimum mean square distortion D_{min} decreases by a little more than 5 dB for each doubling of the number of levels L. Hence, each additional bit that is employed in a uniform quantizer with optimum step size Δ_{opt} for a gaussian-distributed signal amplitude reduces the distortion by more than 5 dB.

By relaxing the constraint that the quantizer be uniform, the distortion can be reduced further. In this case, we let the output level be $\tilde{x} = \tilde{x}_k$ when the input signal amplitude is in the range $x_{k-1} \leq x < x_k$. For an L-level quantizer, the end points are $x_0 = -\infty$ and $x_L = \infty$. The resulting distortion is

$$D = \sum_{k=1}^{L}\int_{x_{k-1}}^{x_k} f(\tilde{x}_k - x)p(x)\,dx \quad (3\text{-}4\text{-}25)$$

which is now minimized by optimally selecting the $\{\tilde{x}_k\}$ and $\{x_k\}$.

The necessary conditions for a minimum distortion are obtained by differentiating D with respect to the $\{x_k\}$ and $\{\tilde{x}_k\}$. The result of this minimization is the pair of equations

$$f(\tilde{x}_k - x_k) = f(\tilde{x}_{k+1} - x_k), \qquad k = 1, 2, \ldots, L-1 \quad (3\text{-}4\text{-}26)$$

$$\int_{x_{k-1}}^{x_k} f'(\tilde{x}_k - x)p(x)\,dx = 0, \qquad k = 1, 2, \ldots, L \quad (3\text{-}4\text{-}27)$$

TABLE 3-4-3 OPTIMUM FOUR-LEVEL QUANTIZER FOR A GAUSSIAN RANDOM VARIABLE

Level k	x_k	$\tilde{x}_k$
1	−0.9816	−1.510
2	0.0	−0.4528
3	0.9816	0.4528
4	∞	1.510

$D_{\min} = 0.1175$
$10 \log D_{\min} = -9.3$ dB

As a special case, we again consider minimizing the mean square value of the distortion. In this case, $f(x) = x^2$ and, hence, (3-4-26) becomes

$$x_k = \tfrac{1}{2}(\tilde{x}_k + \tilde{x}_{k+1}), \quad k = 1, 2, \ldots, L-1 \tag{3-4-28}$$

which is the midpoint between $\tilde{x}_k$ and $\tilde{x}_{k+1}$. The corresponding equations determining $\{\tilde{x}_k\}$ are

$$\int_{x_{k-1}}^{x_k} (\tilde{x}_k - x)p(x)\,dx = 0, \quad k = 1, 2, \ldots, L \tag{3-4-29}$$

Thus, $\tilde{x}_k$ is the centroid of the area of $p(x)$ between x_{k-1} and x_k. These equations may be solved numerically for any given $p(x)$.

Tables 3-4-3 and 3-4-4 give the results of this optimization obtained by Max

TABLE 3-4-4 OPTIMUM EIGHT-LEVEL QUANTIZER FOR A GAUSSIAN RANDOM VARIABLE (MAX, 1960)

Level k	x_k	$\tilde{x}_k$
1	−1.748	−2.152
2	−1.050	−1.344
3	−0.5006	−0.7560
4	0	−0.2451
5	0.5006	0.2451
6	1.050	0.7560
7	1.748	1344
8	∞	2.152

$D_{\min} = 0.03454$
$10 \log D_{\min} = -14.62$ dB

TABLE 3-4-5 COMPARISON OF OPTIMUM UNIFORM AND NONUNIFORM QUANTIZERS FOR A GAUSSIAN RANDOM VARIABLE (MAX, 1960; PAEZ AND GLISSON, 1972)

R (bits/sample)	$10 \log_{10} D_{min}$	
	Uniform (dB)	Nonuniform (dB)
1	−4.4	−4.4
2	−9.25	−9.30
3	−14.27	−14.62
4	−19.38	−20.22
5	−24.57	−26.02
6	−29.83	−31.89
7	−35.13	−37.81

(1960) for the optimum four-level and eight-level quantizers of a gaussian distributed signal amplitude having zero mean and unit variance. In Table 3-4-5, we compare the minimum mean square distortion of a uniform quantizer to that of a nonuniform quantizer for the gaussian-distributed signal amplitude. From the results of this table, we observe that the difference in the performance of the two types of quantizers is relatively small for small values of R (less than 0.5 dB for $R \leq 3$), but it increases as R increases. For example, at $R = 5$, the nonuniform quantizer is approximately 1.5 dB better than the uniform quantizer.

It is instructive to plot the minimum distortion as a function of the bit rate $R = \log_2 L$ bits per source sample (letter) for both the uniform and nonuniform quantizers. These curves are illustrated in Fig. 3-4-2. The functional dependence of the distortion D on the bit rate R may be expressed as $D(R)$, the distortion-rate function. We observe that the distortion-rate function for the optimum nonuniform quantizer falls below that of the optimum uniform quantizer.

Since any quantizer reduces a continuous amplitude source into a discrete amplitude source, we may treat the discrete amplitude as letters, say $\tilde{X} = \{\tilde{x}_k, 1 \leq k \leq L\}$, with associated probabilities $\{p_k\}$. If the signal amplitudes are statistically independent, the discrete source is memoryless and, hence, its entropy is

$$H(\tilde{X}) = -\sum_{k=1}^{L} p_k \log_2 p_k \tag{3-4-30}$$

For example, the optimum four-level nonuniform quantizer for the gaussian-distributed signal amplitude results in the probabilities $p_1 = p_4 = 0.1635$ for the two outer levels and $p_2 = p_3 = 0.3365$ for the two inner levels. The entropy for the discrete source is $H(\tilde{X}) = 1.911$ bits/letter. Hence, with

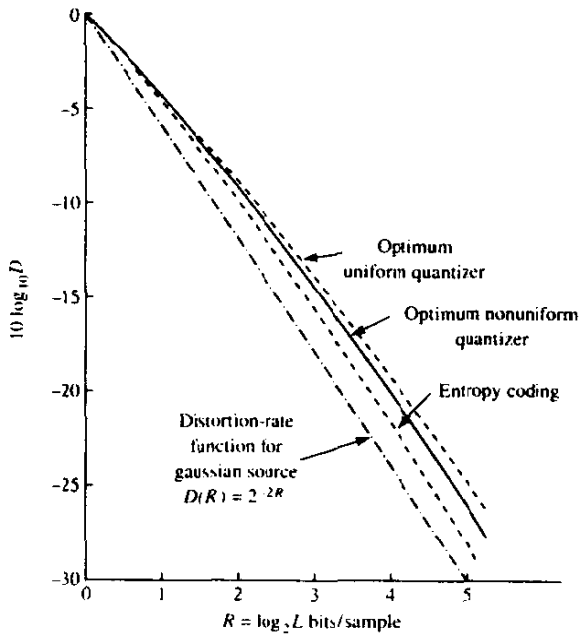


FIGURE 3-4-2 Distortion versus rate curves for discrete-time memoryless gaussian source.

entropy coding (Huffman coding) of blocks of output letters, we can achieve the minimum distortion of −9.30 dB with 1.911 bits/letter instead of 2 bits/letter. Max (1960) has given the entropy for the discrete source letters resulting from quantization. Table 3-4-6 lists the values of the entropy for the nonuniform quantizer. These values are also plotted in Fig. 3-4-2 and labeled *entropy coding.*

From this discussion, we conclude that the quantizer can be optimized when the pdf of the continuous source output is known. The optimum quantizer of $L = 2^R$ levels results in a minimum distortion of $D(R)$, where $R = \log_2 L$

TABLE 3-4-6 ENTROPY OF THE OUTPUT OF AN OPTIMUM NONUNIFORM QUANTIZER FOR A GAUSSIAN RANDOM VARIABLE (MAX, 1960)

$\bar{R}$ (bits/sample)	Entropy (bits/letter)	Distortion $10 \log_{10} D_{min}$
1	1.0	−4.4
2	1.911	−9.30
3	2.825	−14.62
4	3.765	−20.22
5	4.730	−26.02

bits/sample. Thus, this distortion can be achieved by simply representing each quantized sample by R bits. However, more efficient encoding is possible. The discrete source output that results from quantization is characterized by a set of probabilities $\{p_k\}$ that can be used to design efficient variable-length codes for the source output (entropy coding). The efficiency of any encoding method can be compared with the distortion-rate function or, equivalently, the rate-distortion function for the discrete-time, continuous-amplitude source that is characterized by the given pdf.

If we compare the performance of the optimum nonuniform quantizer with the distortion-rate function, we find, for example, that at a distortion of -26 dB, entropy coding is 0.41 bits/sample more than the minimum rate given by (3-4-8), and simple block coding of each letter requires 0.68 bits/sample more than the minimum rate. We also observe that the distortion rate functions for the optimal uniform and nonuniform quantizers for the gaussian source approach the slope of -6 dB/bit asymptotically for large R.

3-4-3 Vector Quantization

In the previous section, we considered the quantization of the output signal from a continuous-amplitude source when the quantization is performed on a sample-by-sample basis, i.e., by scalar quantization. In this section, we consider the joint quantization of a block of signal samples or a block of signal parameters. This type of quantization is called *block* or *vector quantization*. It is widely used in speech coding for digital cellular systems.

A fundamental result of rate-distortion theory is that better performance can be achieved by quantizing vectors instead of scalars, even if the continuous-amplitude source is memoryless. If, in addition, the signal samples or signal parameters are statistically dependent, we can exploit the dependency by jointly quantizing blocks of samples or parameters and, thus, achieve an even greater efficiency (lower bit rate) compared with that which is achieved by scalar quantization.

The vector quantization problem may be formulated as follows. We have an n-dimensional vector $\mathbf{X} = [x_1 \; x_2 \; \cdots \; x_n]$ with real-valued, continuous-amplitude components $\{x_k, 1 \leq k \leq n\}$ that are described by a joint pdf $p(x_1, x_2, \ldots, x_n)$. The vector $\mathbf{X}$ is quantized into another n-dimensional vector $\tilde{\mathbf{X}}$ with components $\{\tilde{x}_k, 1 \leq k \leq n\}$. We express the quantization as $Q(\cdot)$, so that

$$\tilde{\mathbf{X}} = Q(\mathbf{X}) \tag{3-4-31}$$

where $\tilde{\mathbf{X}}$ is the output of the vector quantizer when the input vector is $\mathbf{X}$.

Basically, vector quantization of blocks of data may be viewed as a pattern recognition problem involving the classification of blocks of data into a discrete number of categories or *cells* in a way that optimizes some fidelity criterion, such as mean square distortion. For example, let us consider the quantization

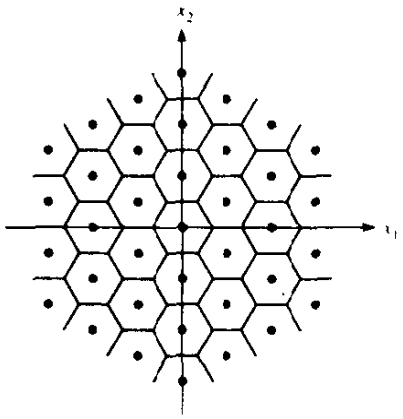


FIGURE 3-4-3 An example of quantization in two-dimensional space.

of two-dimensional vectors $\mathbf{X} = [x_1 \; x_2]$. The two-dimensional space is partitioned into cells as illustrated in Fig. 3-4-3, where we have arbitrarily selected hexagonal-shaped cells $\{C_k\}$. All input vectors that fall in cell C_k are quantized into the vector $\tilde{\mathbf{X}}_k$, which is shown in Fig. 3-4-3 as the center of the hexagon. In this example, there are $L = 37$ vectors, one for each of the 37 cells into which the two-dimensional space has been partitioned. We denote the set of possible output vectors as $\{\tilde{\mathbf{X}}_k, 1 \leq k \leq L\}$.

In general, quantization of the n-dimensional vector $\mathbf{X}$ into an n-dimensional vector $\tilde{\mathbf{X}}$ introduces a quantization error or a distortion $d(\mathbf{X}, \tilde{\mathbf{X}})$. The average distortion over the set of input vectors $\mathbf{X}$ is

$$\begin{aligned} D &= \sum_{k=1}^{L} P(\mathbf{X} \in C_k) E[d(\mathbf{X}, \tilde{\mathbf{X}}_k) \mid \mathbf{X} \in C_k] \\ &= \sum_{k=1}^{L} P(\mathbf{X} \in C_k) \int_{\mathbf{X} \in C_k} d(\mathbf{X}, \tilde{\mathbf{X}}_k) p(\mathbf{X}) \, d\mathbf{X} \end{aligned} \tag{3-4-32}$$

where $P(\mathbf{X} \in C_k)$ is the probability that the vector $\mathbf{X}$ falls in the cell C_k and $p(\mathbf{X})$ is the joint pdf of the n random variables. As in the case of scalar quantization, we can minimize D by selecting the cells $\{C_k, 1 \leq k \leq L\}$ for a given pdf $p(\mathbf{X})$.

A commonly used distortion measure is the mean square error (l_2 norm) defined as

$$d_2(\mathbf{X}, \tilde{\mathbf{X}}) = \frac{1}{n} (\mathbf{X} - \tilde{\mathbf{X}})'(\mathbf{X} - \tilde{\mathbf{X}}) = \frac{1}{n} \sum_{k=1}^{n} (x_k - \tilde{x}_k)^2 \tag{3-4-33}$$

or, more generally, the weighted mean square error

$$d_{2W}(\mathbf{X}, \tilde{\mathbf{X}}) = (\mathbf{X} - \tilde{\mathbf{X}})'\mathbf{W}(\mathbf{X} - \tilde{\mathbf{X}}) \tag{3-4-34}$$

where $\mathbf{W}$ is a positive-definite weighting matrix. Usually, $\mathbf{W}$ is selected to be the inverse of the covariance matrix of the input data vector $\mathbf{X}$.

Other distortion measures that are sometimes used are special cases of the l_p norm defined as

$$d_p(\mathbf{X}, \tilde{\mathbf{X}}) = \frac{1}{n} \sum_{k=1}^{n} |x_k - \tilde{x}_k|^p \tag{3-4-35}$$

The special case $p = 1$ is often used as an alternative to $p = 2$.

Vector quantization is not limited to quantizing a block of signal samples of a source waveform. It can also be applied to quantizing a set of parameters extracted from the data. For example, in linear predictive coding (LPC), described in Section 3-5-3, the parameters extracted from the signal are the prediction coefficients, which are the coefficients in the all-pole filter model for the source that generates the observed data. These parameters can be considered as a block and quantized as a block by application of some appropriate distortion measure. In the case of speech encoding, an appropriate distortion measure, proposed by Itakura and Saito (1968, 1975), is the weighted square error where the weighting matrix $\mathbf{W}$ is selected to be the normalized autocorrelation matrix $\mathbf{\Phi}$ of the observed data.

In speech processing, an alternative set of parameters that may be quantized as a block and transmitted to the receiver is the set of reflection coefficients $\{a_{ii}, 1 \leq i \leq m\}$. Yet another set of parameters that is sometimes used for vector quantization in linear predictive coding of speech comprises the log-area ratios $\{r_k\}$, which are defined in terms of the reflection coefficients as

$$r_k = \log \frac{1 + a_{kk}}{1 - a_{kk}}, \quad 1 \leq k \leq m \tag{3-4-36}$$

Now, let us return to the mathematical formulation of vector quantization and let us consider the partitioning of the n-dimensional space into L cells $\{C_k, 1 \leq k \leq L\}$ so that the average distortion is minimized over all L-level quantizers. There are two conditions for optimality. The first is that the optimal quantizer employs a nearest-neighbor selection rule, which may be expressed mathematically as

$$Q(\mathbf{X}) = \mathbf{X}_k$$

if and only if

$$D(\mathbf{X}, \tilde{\mathbf{X}}_k) \leq D(\mathbf{X}, \tilde{\mathbf{X}}_j), \quad k \neq j, \quad 1 \leq j \leq L \tag{3-4-37}$$

The second condition necessary for optimality is that each output vector $\tilde{\mathbf{X}}_k$ be chosen to minimize the average distortion in cell C_k. In other words, $\tilde{\mathbf{X}}_k$ is the vector in C_k that minimizes

$$D_k = E[d(\mathbf{X}, \tilde{\mathbf{X}}) \mid \mathbf{X} \in C_k] = \int_{\mathbf{X} \in C_k} d(\mathbf{X}, \tilde{\mathbf{X}}) p(\mathbf{X})\, d\mathbf{X} \tag{3-4-38}$$

The vector $\tilde{\mathbf{X}}_k$ that minimizes D_k is called the *centroid* of the cell. Thus, these conditions for optimality can be applied to partition the n-dimensional space

into cells $\{C_k, 1 \leq k \leq L\}$ when the joint pdf $p(\mathbf{X})$ is known. It is clear that these two conditions represent the generalization of the optimum scalar quantization problem to the n-dimensional vector quantization problem. In general, we expect the code vectors to be closer together in regions where the joint pdf is large and farther apart in regions where $p(\mathbf{X})$ is small.

As an upper bound on the distortion of a vector quantizer, we may use the distortion of the optimal scalar quantizer, which can be applied to each component of the vector as described in the previous section. On the other hand, the best performance that can be achieved by optimum vector quantization is given by the rate-distortion function or, equivalently, the distortion-rate function.

The distortion-rate function, which was introduced in the previous section, may be defined in the context of vector quantization as follows. Suppose we form a vector $\mathbf{X}$ of dimension n from n consecutive samples $\{x_m\}$. The vector $\mathbf{X}$ is then quantized to form $\tilde{\mathbf{X}} = Q(\mathbf{X})$, where $\tilde{\mathbf{X}}$ is a vector from the set of $\{\tilde{\mathbf{X}}_k, 1 \leq k \leq L\}$. As described above, the average distortion D resulting from representing $\mathbf{X}$ by $\tilde{\mathbf{X}}$ is $E[d(\mathbf{X}, \tilde{\mathbf{X}})]$, where $d(\mathbf{X}, \tilde{\mathbf{X}})$ is the distortion per dimension, e.g.,

$$d(\mathbf{X}, \tilde{\mathbf{X}}) = \frac{1}{n} \sum_{k=1}^{n} (x_k - \tilde{x}_k)^2$$

The vectors $\{\tilde{\mathbf{X}}_k, 1 \leq k \leq L\}$ can be transmitted at an average bit rate of

$$R = \frac{H(\tilde{\mathbf{X}})}{n} \text{ bits/sample} \tag{3-4-39}$$

where $H(\tilde{\mathbf{X}})$ is the entropy of the quantized source output defined as

$$H(\tilde{\mathbf{X}}) = -\sum_{i=1}^{L} p(\tilde{\mathbf{X}}_i) \log_2 P(\tilde{\mathbf{X}}_i) \tag{3-4-40}$$

For a given average rate R, the minimum achievable distortion $D_n(R)$ is

$$D_n(R) = \min_{Q(\mathbf{X})} E[d(\mathbf{X}, \tilde{\mathbf{X}})] \tag{3-4-41}$$

where $R \geq H(\tilde{\mathbf{X}})/n$ and the minimum in (3-4-41) is taken over all possible mappings $Q(\mathbf{X})$. In the limit as the number of dimensions n is allowed to approach infinity, we obtain

$$D(R) = \lim_{n \to \infty} D_n(R) \tag{3-4-42}$$

where $D(R)$ is the distortion-rate function that was introduced in the previous section. It is apparent from this development that the distortion-rate function can be approached arbitrarily closely by increasing the size n of the vectors.

The development above is predicated on the assumption that the joint pdf $p(\mathbf{X})$ of the data vector is known. However, in practice, the joint pdf $p(\mathbf{X})$ of the data may not be known. In such a case, it is possible to select the

quantized output vectors adaptively from a set of training vectors $\mathbf{X}(m)$. Specifically, suppose that we are given a set of M training vectors where M is much greater than L ($M \gg L$). An iterative clustering algorithm, called the *K means algorithm*, where in our case $K = L$, can be applied to the training vectors. This algorithm iteratively subdivides the M training vectors into L clusters such that the two necessary conditions for optimality are satisfied. The K means algorithm may be described as follows [Makhoul *et al.* (1985)].

K Means Algorithm

Step 1 Initialize by setting the iteration number $i = 0$. Choose a set of output vectors $\tilde{\mathbf{X}}_k(0)$, $1 \leq k \leq L$.

Step 2 Classify the training vectors $\{\mathbf{X}(m),\ 1 \leq m \leq M\}$ into the clusters $\{C_k\}$ by applying the nearest-neighbor rule

$$\mathbf{X} \in C_k(i) \quad \text{iff } D(\mathbf{X}, \tilde{\mathbf{X}}_k(i)) \leq D(\mathbf{X}, \tilde{\mathbf{X}}_j(i)) \quad \text{for all } k \neq j$$

Step 3 Recompute (set i to $i + 1$) the output vectors of every cluster by computing the centroid

$$\tilde{\mathbf{X}}_k(i) = \frac{1}{M_k} \sum_{\mathbf{X} \in C_k} \mathbf{X}(m), \qquad 1 \leq k \leq L$$

of the training vectors that fall in each cluster. Also, compute the resulting distortion $D(i)$ at the ith iteration.

Step 4 Terminate the test if the change $D(i-1) - D(i)$ in the average distortion is relatively small. Otherwise, go to Step 2.

The K means algorithm converges to a local minimum (see Anderberg, 1973; Linde *et al.*, 1980). By beginning the algorithm with different sets of initial output vectors $\{\mathbf{X}_k(0)\}$ and each time performing the optimization described in the K means algorithm, it is possible to find a global optimum. However, the computational burden of this search procedure may limit the search to a few initializations.

Once we have selected the output vectors $\{\tilde{\mathbf{X}}_k,\ 1 \leq k \leq L\}$, each signal vector $\mathbf{X}(m)$ is quantized to the output vector that is nearest to it according to the distortion measure that is adopted. If the computation involves evaluating the distance between $\mathbf{X}(m)$ and each of the L possible output vectors $\{\tilde{\mathbf{X}}_k\}$, the procedure constitutes a *full search*. If we assume that each computation requires n multiplications and additions, the computational requirement for a full search is

$$\mathscr{C} = nL \tag{3-4-43}$$

multiplication and additions per input vector.

If we select L to be a power of 2 then $\log_2 L$ is the number of bits required to represent each vector. Now, if R denotes the bit rate per sample [per component or dimension of $\mathbf{X}(m)$], we have $nR = \log_2 L$, and, hence, the computational cost is

$$\mathscr{C} = n2^{nR} \tag{3-4-44}$$

Note that the number of computations grows exponentially with the dimensionality parameter n and the bit rate R per dimension. Because of this exponential increase of the computational cost, vector quantization has been applied to low-bit-source encoding, such as coding the reflection coefficients or log area ratios in LPC.

The computational cost associated with full search can be reduced by slightly suboptimum algorithms (see Chang *et al.*, 1984; Gersho, 1982).

In order to demonstrate the benefits of vector quantization compared with scalar quantization, we present the following example taken from Makhoul *et al.* (1985).

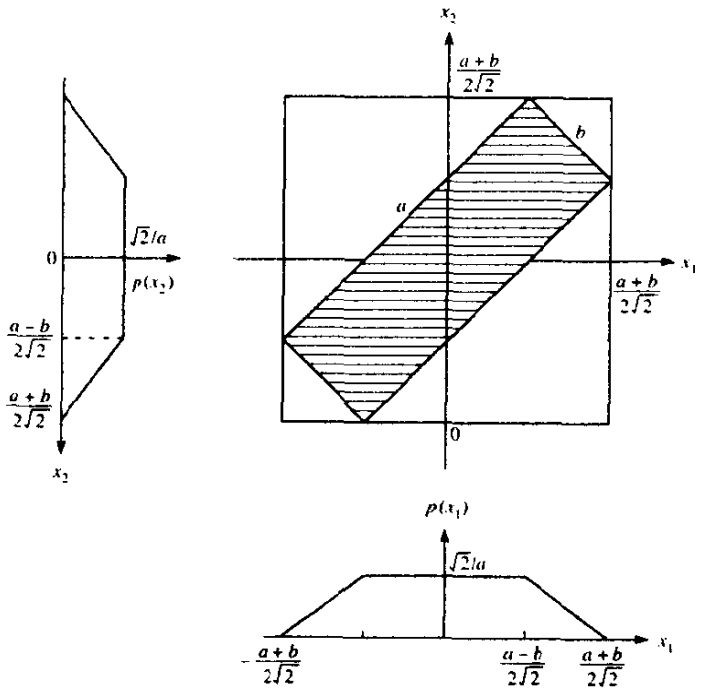
Example 3-4-1

Let x_1 and x_2 be two random variables with a uniform joint pdf

$$p(x_1, x_2) \equiv p(\mathbf{X}) = \begin{cases} \dfrac{1}{ab} & (\mathbf{X} \in \mathbf{C}) \\ 0 & (\text{otherwise}) \end{cases} \tag{3-4-45}$$

where $\mathbf{C}$ is the rectangular region illustrated in Fig. 3-4-4. Note that the rectangle is rotated by 45° relative to the horizontal axis. Also shown in Fig. 3-4-4 are the marginal densities $p(x_1)$ and $p(x_2)$.

FIGURE 3-4-4 A uniform pdf in two dimensions. (*Makhoul* et al., *1985.*)



If we quantize x_1 and x_2 separately by using uniform intervals of length Δ, the number of levels needed is

$$L_1 = L_2 = \frac{a+b}{\sqrt{2}\,\Delta} \tag{3-4-46}$$

Hence, the number of bits needed for coding the vector $\mathbf{X} = [x_1 \;\; x_2]$ is

$$R_x = R_1 + R_2 = \log_2 L_1 + \log_2 L_2$$

$$R_x = \log_2 \frac{(a+b)^2}{2\Delta^2} \tag{3-4-47}$$

Thus, scalar quantization of each component is equivalent to vector quantization with the total number of levels

$$L_x = L_1 L_2 = \frac{(a+b)^2}{2\Delta^2} \tag{3-4-48}$$

We observe that this approach is equivalent to covering the large square that encloses the rectangle by square cells, where each cell represents one of the L_x quantized regions. Since $p(\mathbf{X}) = 0$ except for $\mathbf{X} \in C$, this encoding is wasteful and results in an increase of the bit rate.

If we were to cover only the region for which $p(\mathbf{X}) \neq 0$ with squares having area Δ^2, the total number of levels that will result is the area of the rectangle divided by Δ^2, i.e.,

$$L_x' = \frac{ab}{\Delta^2} \tag{3-4-49}$$

Therefore, the difference in bit rate between the scalar and vector quantization methods is

$$R_x - R_x' = \log_2 \frac{(a+b)^2}{2ab} \tag{3-4-50}$$

For instance, if $a = 4b$, the difference in bit rate is

$$R_x - R_x' = 1.64 \text{ bits/vector}$$

Thus, vector quantization is 0.82 bits/sample better for the same distortion.

It is interesting to note that a linear transformation (rotation by 45°) will decorrelate x_1 and x_2 and render the two random variables statistically independent. Then scalar quantization and vector quantization achieve the same efficiency. Although a linear transformation can decorrelate a vector of random variables, it does not result in statistically independent random variables, in general. Consequently, vector quantization will always equal or exceed the performance of scalar quantization (see Problem 3-40).

Vector quantization has been applied to several types of speech encoding

methods including both waveform and model-based methods which are treated in Section 3-5. In model-based methods such as LPC, vector quantization has made possible the coding of speech at rates below 1000 bits/s (see Buzo *et al.*, 1980; Roucos *et al.*, 1982; Paul 1983). When applied to waveform encoding methods, it is possible to obtain good quality speech at 16 000 bits/s, or, equivalently, at $R = 2$ bits/sample. With additional computational complexity, it may be possible in the future to implement waveform encoders producing good quality speech at a rate of $R = 1$ bit/sample.

3-5 CODING TECHNIQUES FOR ANALOG SOURCES

A number of coding techniques for analog sources have been developed over the past 40 years. Most of these have been applied to the encoding of speech and images. In this section, we briefly describe several of these methods and use speech encoding as an example in assessing their performance.

It is convenient to subdivide analog source encoding methods into three types. One type is called *temporal waveform coding*. In this type of encoding, the source encoder is designed to represent digitally the temporal characteristics of the source waveform. A second type of source encoding is *spectral waveform coding*. The signal waveform is usually subdivided into different frequency bands, and either the time waveform in each band or its spectral characteristics are encoded for transmission. The third type of source encoding is based on a mathematical model of the source and is called *model-based coding*.

3-5-1 Temporal Waveform Coding

There are several analog source coding techniques that are designed to represent the time-domain characteristics of the signal. The most commonly used methods are described in this section.

Pulse Code Modulation† (PCM) Let $x(t)$ denote a sample function emitted by a source and let x_n denote the samples taken at a sampling rate $f_s \geq 2W$, where W is the highest frequency in the spectrum of $x(t)$. In PCM, each sample of the signal is quantized to one of 2^R amplitude levels, where R is the number of binary digits used to represent each sample. Thus the rate from the source is Rf_s bits/s.

The quantization process may be modeled mathematically as

$$\tilde{x}_n = x_n + q_n \tag{3-5-1}$$

where $\tilde{x}_n$ represents the quantized value of x_n and q_n represents the quantization error, which we treat as an additive noise. Assuming that a

† PCM, DPCM, and ADPCM are source coding techniques. They are not digital modulation methods.

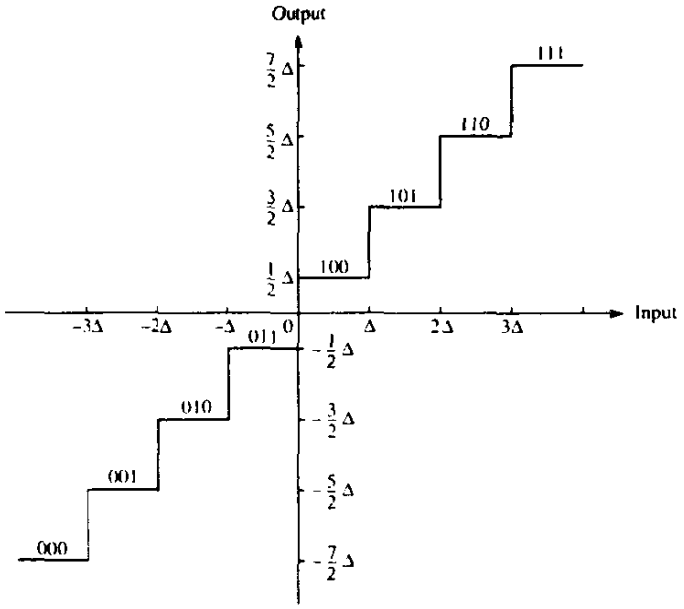


FIGURE 3-5-1 Input–output characteristic for a uniform quantizer.

uniform quantizer is used, having the input–output characteristic illustrated in Fig. 3-5-1, the quantization noise is well characterized statistically by the uniform pdf

$$p(q) = \frac{1}{\Delta}, \qquad -\tfrac{1}{2}\Delta \leq q \leq \tfrac{1}{2}\Delta \tag{3-5-2}$$

where the step size of the quantizer is $\Delta = 2^{-R}$. The mean square value of the quantization error is

$$E(q^2) = \tfrac{1}{12}\Delta^2 = \tfrac{1}{12} \times 2^{-2R} \tag{3-5-3}$$

Measured in decibels, the mean square value of the noise is

$$10 \log \tfrac{1}{12}\Delta^2 = 10 \log (\tfrac{1}{12} \times 2^{-2R}) = -6R - 10.8 \text{ dB} \tag{3-5-4}$$

We observe that the quantization noise decreases by 6 dB/bit used in the quantizer. For example, a 7 bit quantizer results in a quantization noise power of −52.8 dB.

Many source signals such as speech waveforms have the characteristic that small signal amplitudes occur more frequently than large ones. However, a uniform quantizer provides the same spacing between successive levels throughout the entire dynamic range of the signal. A better approach is to employ a nonuniform quantizer. A nonuniform quantizer characteristic is usually obtained by passing the signal through a nonlinear device that compresses the signal amplitude, followed by a uniform quantizer. For

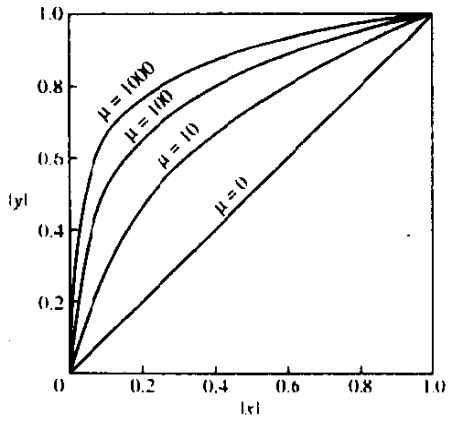


FIGURE 3-5-2 Input–output magnitude characteristic for a logarithmic compressor.

example, a logarithmic compressor has an input–output magnitude characteristics of the form

$$|y| = \frac{\log(1 + \mu|x|)}{\log(1 + \mu)} \tag{3-5-5}$$

where $|x| \leq 1$ is the magnitude of the input, $|y|$ is the magnitude of the output, and μ is a parameter that is selected to give the desired compression characteristic. Figure 3-5-2 illustrates this compression relationship for several values of μ. The value $\mu = 0$ corresponds to no compression.

In the encoding of speech waveforms, for example, the value of $\mu = 255$ has been adopted as a standard in the USA and Canada. This value results in about a 24 dB reduction in the quantization noise power relative to uniform quantization, as shown by Jayant (1974). Consequently, a 7 bit quantizer used in conjunction with a $\mu = 255$ logarithmic compressor produces a quantization noise power of approximately -77 dB compared with the -53 dB for uniform quantization.

In the reconstruction of the signal from the quantized values, the inverse logarithmic relation is used to expand the signal amplitude. The combined compressor–expandor pair is termed a *compandor*.

Differential Pulse Code Modulation (DPCM) In PCM, each sample of the waveform is encoded independently of all the others. However, most source signals sampled at the Nyquist rate or faster exhibit significant correlation between successive samples. In other words, the average change in amplitude between successive samples is relatively small. Consequently, an encoding scheme that exploits the redundancy in the samples will result in a lower bit rate for the source output.

A relatively simple solution is to encode the differences between successive samples rather than the samples themselves. Since differences between samples are expected to be smaller than the actual sampled amplitudes, fewer bits are required to represent the differences. A refinement of this general approach is

to predict the current sample based on the previous p samples. To be specific, let x_n denote the current sample from the source and let $\hat{x}_n$ denote the predicted value of x_n, defined as

$$\hat{x}_n = \sum_{i=1}^{p} a_i x_{n-i} \tag{3-5-6}$$

Thus $\hat{x}_n$ is a weighted linear combination of the past p samples and the $\{a_i\}$ are the predictor coefficients. The $\{a_i\}$ are selected to minimize some function of the error between x_n and $\hat{x}_n$.

A mathematically and practically convenient error function is the mean square error (MSE). With the MSE as the performance index for the predictor, we select the $\{a_i\}$ to minimize

$$\begin{aligned}\mathscr{E}_p = E(e_n^2) &= E\left[\left(x_n - \sum_{i=1}^{p} a_i x_{n-i}\right)^2\right] \\ &= E(x_n^2) - 2\sum_{i=1}^{p} a_i E(x_n x_{n-i}) + \sum_{i=1}^{p}\sum_{j=1}^{p} a_i a_j E(x_{n-i}x_{n-j})\end{aligned} \tag{3-5-7}$$

Assuming that the source output is (wide-sense) stationary, we may express (3-5-7) as

$$\mathscr{E}_p = \phi(0) - 2\sum_{i=1}^{p} a_i \phi(i) + \sum_{i=1}^{p}\sum_{j=1}^{p} a_i a_j \phi(i-j) \tag{3-5-8}$$

where $\phi(m)$ is the autocorrelation function of the sampled signal sequence x_n. Minimization of $\mathscr{E}_p$ with respect to the predictor coefficients $\{a_i\}$ results in the set of linear equations

$$\sum_{i=1}^{p} a_i \phi(i-j) = \phi(j), \qquad j = 1, 2, \ldots, p \tag{3-5-9}$$

Thus, the values of the predictor coefficients are established. When the autocorrelation function $\phi(n)$ is not known *a priori*, it may be estimated from the samples $\{x_n\}$ using the relation†

$$\hat{\phi}(n) = \frac{1}{N}\sum_{i=1}^{N-n} x_i x_{i+n}, \qquad n = 0, 1, 2, \ldots, p \tag{3-5-10}$$

and the estimate $\hat{\phi}(n)$ is used in (3-5-9) to solve for the coefficients $\{a_i\}$. Note that the normalization factor of $1/N$ in (3-5-10) drops out when $\hat{\phi}(n)$ is substituted in (3-5-9).

The linear equations in (3-5-9) for the predictor coefficients are called the *normal equations* or the *Yule–Walker equations*. There is an algorithm developed by Levinson (1947) and Durbin (1959) for solving these equations efficiently. It is described in Appendix A. We shall deal with the solution in greater detail in the subsequent discussion on linear predictive coding.

† The estimation of the autocorrelation function from a finite number of observations $\{x_i\}$ is a separate issue, which is beyond the scope of this discussion. The estimate in (3-5-10) is one that is frequently used in practice.

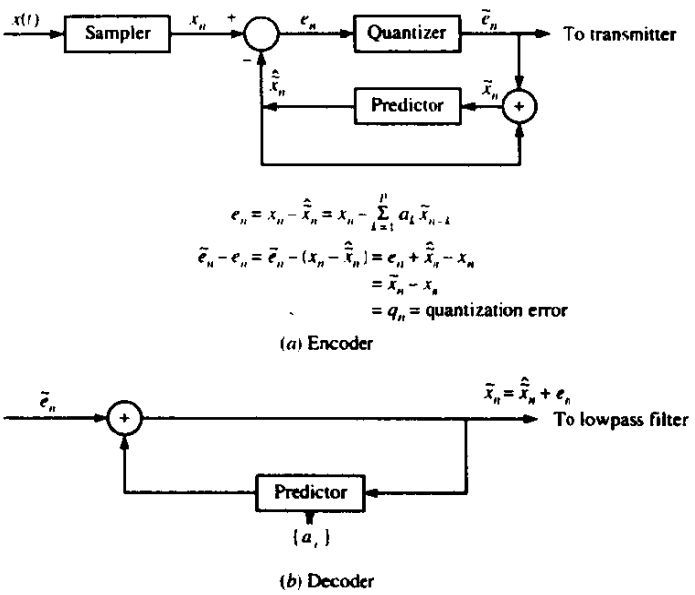


FIGURE 3-5-3 (*a*) Block diagram of a DPCM encoder. (*b*) DPCM decoder at the receiver.

Having described the method for determining the predictor coefficients, let us now consider the block diagram of a practical DPCM system, shown in Fig. 3-5-3(*a*). In this configuration, the predictor is implemented with the feedback loop around the quantizer. The input to the predictor is denoted by $\tilde{x}_n$, which represents the signal sample x_n modified by the quantization process, and the output of the predictor is

$$\hat{x}_n = \sum_{i=1}^{p} a_i \tilde{x}_{n-i} \tag{3-5-11}$$

The difference

$$e_n = x_n - \hat{x}_n \tag{3-5-12}$$

is the input to the quantizer and $\tilde{e}_n$ denotes the output. Each value of the quantized prediction error $\tilde{e}_n$ is encoded into a sequence of binary digits and transmitted over the channel to the destination. The quantized error $\tilde{e}_n$ is also added to the predicted value $\hat{x}_n$ to yield $\tilde{x}_n$.

At the destination, the same predictor that was used at the transmitting end is synthesized and its output $\hat{x}_n$ is added to $\tilde{e}_n$ to yield $\tilde{x}_n$. The signal $\tilde{x}_n$ is the desired excitation for the predictor and also the desired output sequence from which the reconstructed signal $\tilde{x}(t)$ is obtained by filtering, as shown in Fig. 3-5-3(*b*).

The use of feedback around the quantizer, as described above, ensures that the error in $\tilde{x}_n$ is simply the quantization error $q_n = \tilde{e}_n - e_n$ and that there is no

accumulation of previous quantization errors in the implementation of the decoder. That is,

$$\begin{aligned} q_n &= \tilde{e}_n - e_n \\ &= \tilde{e}_n - (x_n - \hat{x}_n) \\ &= \tilde{x}_n - x_n \end{aligned} \tag{3-5-13}$$

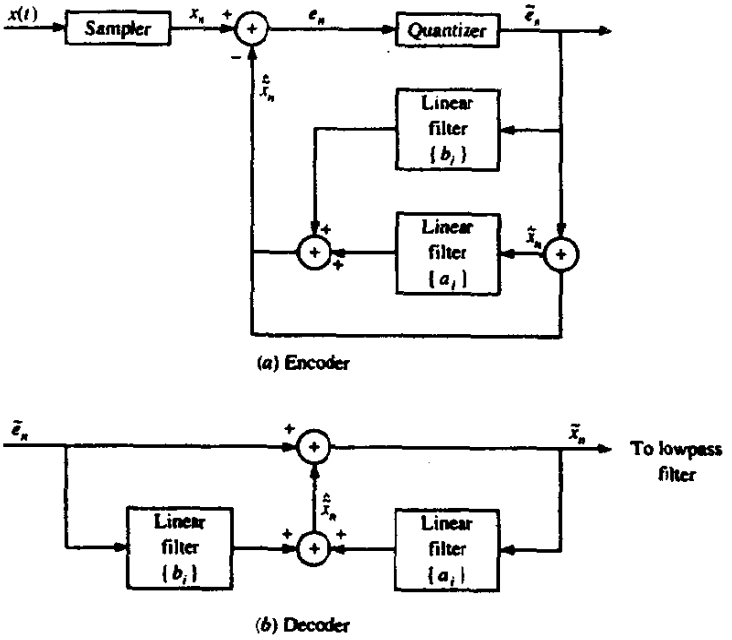
Hence $\tilde{x}_n = x_n + q_n$. This means that the quantized sample $\tilde{x}_n$ differs from the input x_n by the quantization error q_n independent of the predictor used. Therefore, the quantization errors do not accumulate.

In the DPCM system illustrated in Fig. 3-5-3, the estimate or predicted value $\hat{x}_n$ of the signal sample x_n is obtained by taking a linar combination of past values $\tilde{x}_{n-k}$, $k = 1, 2, \ldots, p$, as indicated by (3-5-11). An improvement in the quality of the estimate is obtained by including linearly filtered past values of the quantized error. Specifically, the $\hat{\tilde{x}}_n$ estimate may be expressed as

$$\hat{\tilde{x}}_n = \sum_{i=1}^{p} a_i \tilde{x}_{n-i} + \sum_{i=1}^{m} b_i \tilde{e}_{n-i} \tag{3-5-14}$$

where $\{b_i\}$ are the coefficients of the filter for the quantized error sequence $\tilde{e}_n$. The block diagrams of the encoder at the transmitter and the decoder at the receiver are shown in Fig. 3-5-4. The two sets of coefficients $\{a_i\}$ and $\{b_i\}$ are selected to minimize some function of the error $e_n = x_n - \hat{x}_n$, such as the mean square error.

FIGURE 3-5-4 DPCM modified by the addition of linearly filtered error sequence.



Adaptive PCM and DPCM Many real sources are quasistationary in nature. One aspect of the quasistationary characteristic is that the variance and the autocorrelation function of the source output vary slowly with time. PCM and DPCM encoders, however, are designed on the basis that the source output is stationary. The efficiency and performance of these encoders can be improved by having them adapt to the slowly time-variant statistics of the source.

In both PCM and DPCM, the quantization error q_n resulting from a uniform quantizer operating on a quasistationary input signal will have a time-variant variance (quantization noise power). One improvement that reduces the dynamic range of the quantization noise is the use of an adaptive quantizer. Although the quantizer can be made adaptive in different ways, a relatively simple method is to use a uniform quantizer that varies its step size in accordance with the variance of the past signal samples. For example, a short-term running estimate of the variance of x_n can be computed from the input sequence $\{x_n\}$ and the step size can be adjusted on the basis of such an estimate. In its simplest form, the algorithm for the step-size adjustment employs only the previous signal sample. Such an algorithm has been successfully used by Jayant (1974) in the encoding of speech signals. Figure 3-5-5 illustrates such a (3 bit) quantizer in which the step size is adjusted recursively according to the relation

$$\Delta_{n+1} = \Delta_n M(n) \tag{3-5-15}$$

FIGURE 3-5-5 Example of a quantizer with an adaptive step size. (*Jayant, 1974.*)

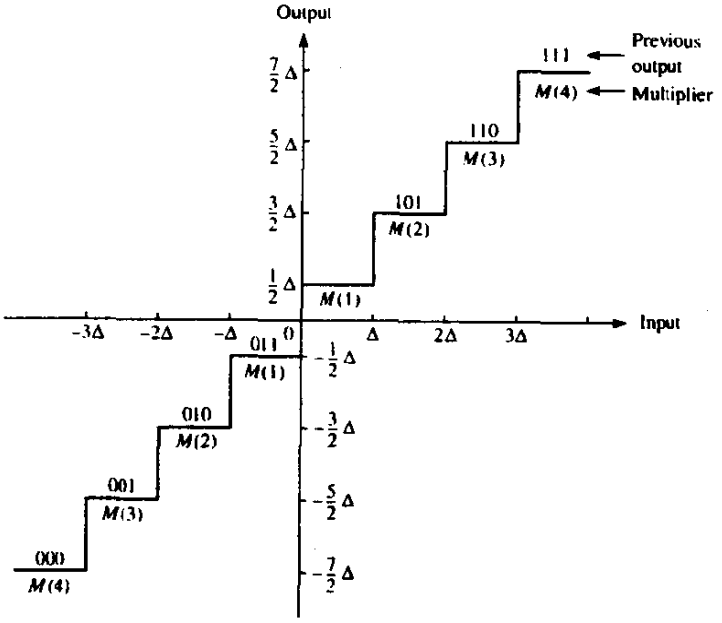


TABLE 3-5-1 MULTIPLICATION FACTORS FOR ADAPTIVE STEP SIZE ADJUSTMENT (JAYANT, 1974)

	PCM			DPCM		
	2	3	4	2	3	4
$M(1)$	0.60	0.85	0.80	0.80	0.90	0.90
$M(2)$	2.20	1.00	0.80	1.60	0.90	0.90
$M(3)$		1.00	0.80		1.25	0.90
$M(4)$		1.50	0.80		1.70	0.90
$M(5)$			1.20			1.20
$M(6)$			1.60			1.60
$M(7)$			2.00			2.00
$M(8)$			2.40			2.40

where $M(n)$ is a factor, whose value depends on the quantizer level for the sample x_n, and Δ_n is the step size of the quantizer for processing x_n. Values of the multiplication factors optimized for speech encoding have been given by Jayant (1974). These values are displayed in Table 3-5-1 for 2, 3, and 4 bit adaptive quantization.

In DPCM, the predictor can also be made adaptive when the source output in quasistationary. The coefficients of the predictor can be changed periodically to reflect the changing signal statistics of the source. The linear equations given by (3-5-9) still apply, with the short-term estimate of the autocorrelation function of x_n substituted in place of the ensemble correlation function. The predictor coefficients thus determined may be transmitted along with the quantized error $\tilde{e}(n)$ to the receiver, which implements the same predictor. Unfortunately, the transmission of the predictor coefficients results in a higher bit rate over the channel, offsetting, in part, the lower data rate achieved by having a quantizer with fewer bits (fewer levels) to handle the reduced dynamic range in the error e_n resulting from adaptive prediction.

As an alternative, the predictor at the receiver may compute its own prediction coefficients from $\tilde{e}_n$ and $\tilde{x}_n$, where

$$\tilde{x}_n = \tilde{e}_n + \sum_{i=1}^{p} a_i \tilde{x}_{n-i} \tag{3-5-16}$$

If we neglect the quantization noise, $\tilde{x}_n$ is equivalent to x_n. Hence, $\tilde{x}_n$ may be used to estimate the autocorrelation function $\phi(n)$ at the receiver, and the resulting estimates can be used in (3-5-9) in place of $\phi(n)$ to solve for the predictor coefficients. For sufficiently fine quantization, the difference between x_n and $\tilde{x}_n$ is very small. Hence, the estimate of $\phi(n)$ obtained from $\tilde{x}_n$ is usually adequate for determining the predictor coefficients. Implemented in this manner, the adaptive predictor results in a lower source data rate.

Instead of using the block processing approach for determining the

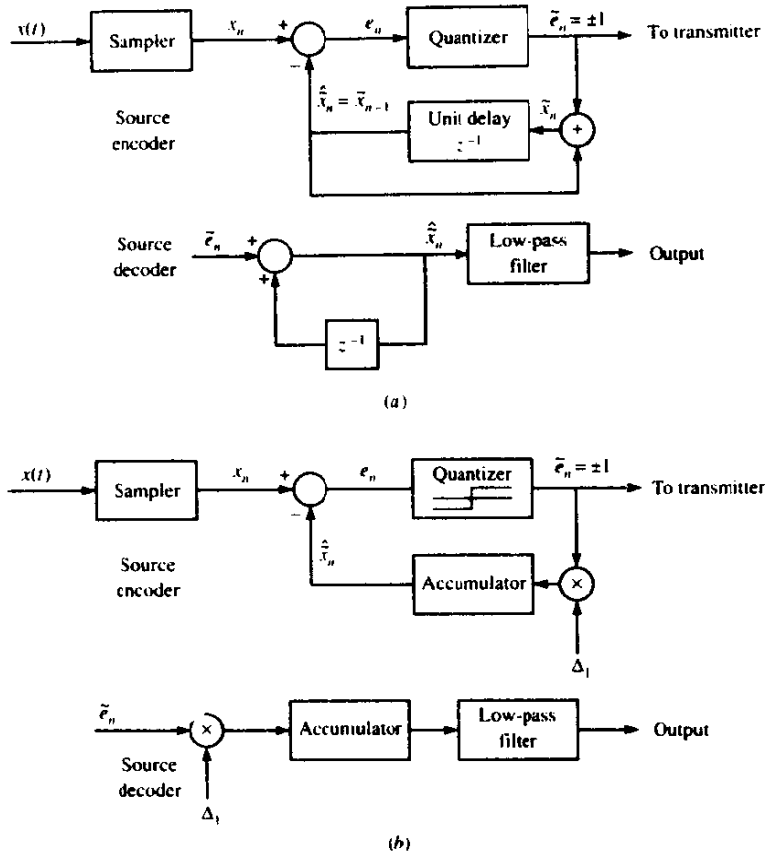


FIGURE 3-5-6 (*a*) Block diagram of a delta modulation system. (*b*) An equivalent realization of a delta modulation system.

predictor coefficients $\{a_i\}$ as described above, we may adapt the predictor coefficients on a sample-by-sample basis by using a gradient-type algorithm, similar in form to the adaptive gradient equalization algorithm that is described in Chapter 11. Similar gradient-type algorithms have also been devised for adapting the filter coefficients $\{a_i\}$ and $\{b_i\}$ of the DPCM system shown in Fig. 3-5-4. For details on such algorithms, the reader may refer to the book by Jayant and Noll (1984).

Delta Modulation (DM) Delta modulation may be viewed as a simplified form of DPCM in which a two-level (1 bit) quantizer is used in conjunction with a fixed first-order predictor. The block diagram of a DM encoder–decoder is shown in Fig. 3-5-6(*a*). We note that

$$\hat{x}_n = \tilde{x}_{n-1} = \hat{x}_{n-1} + \tilde{e}_{n-1} \tag{3-5-17}$$

Since

$$\begin{aligned} q_n &= \tilde{e}_n - e_n \\ &= \tilde{e}_n - (x_n - \hat{\tilde{x}}_n) \end{aligned}$$

It follows that

$$\hat{\tilde{x}}_n = x_{n-1} + q_{n-1}$$

Thus the estimated (predicted) value of x_n is really the previous sample x_{n-1} modified by the quantization noise q_{n-1}. We also note that the difference equation (3-5-17) represents an integrator with an input $\tilde{e}_n$. Hence, an equivalent realization of the one-step predictor is an accumulator with an input equal to the quantized error signal $\tilde{e}_n$. In general, the quantized error signal is scaled by some value, say Δ_1, which is called the *step size*. This equivalent realization is illustrated in Fig. 3-5-6(b). In effect, the encoder shown in Fig. 3-5-6 approximates a waveform $x(t)$ by a linear staircase function. In order for the approximation to be relatively good, the waveform $x(t)$ must change slowly relative to the sampling rate. This requirement implies that the sampling rate must be several (a factor of at least 5) times the Nyquist rate.

At any given sampling rate, the performance of the DM encoder is limited by two types of distortion, as illustrated in Fig. 3-5-7. One is called *slope-overload distortion*. It is due to the use of a step size Δ_1 that is too small to follow portions of the waveform that have a steep slope. The second type of distortion, called *granular noise*, results from using a step size that is too large in parts of the waveform having a small slope. The need to minimize both of these two types of distortion results in conflicting requirements in the selection of the step size Δ_1. One solution is to select Δ_1 to minimize the sum of the mean square values of these two distortions.

Even when Δ_1 is optimized to minimize the total mean square value of the slope-overload distortion and the granular noise, the performance of the DM encoder may still be less than satisfactory. An alternative solution is to employ a variable step size that adapts itself to the short-term characteristics of the source signal. That is, the step size is increased when the waveform has a steep

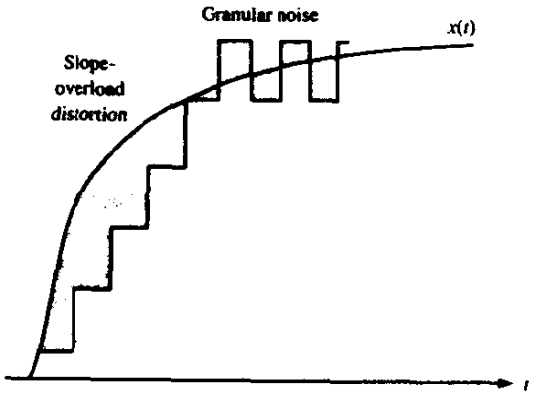


FIGURE 3-5-7 An example of slope overload distortion and granular noise in a delta modulation encoder.

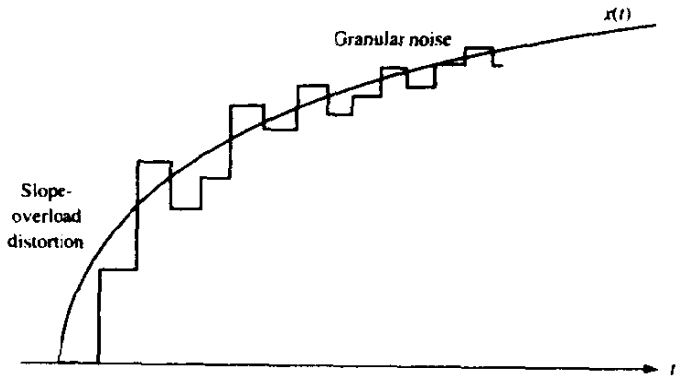


FIGURE 3-5-8 An example of variable-step-size delta modulation encoding.

slope and decreased when the waveform has a relatively small slope. This adaptive characteristic is illustrated in Fig. 3-5-8.

A variety of methods can be used to adaptively set the step size in every iteration. The quantized error sequence $\tilde{e}_n$ provides a good indication of the slope characteristics of the waveform being encoded. When the quantized error $\tilde{e}_n$ is changing signs between successive iterations, this is an indication that the slope of the waveform in that locality is relatively small. On the other hand, when the waveform has a steep slope, successive values of the error $\tilde{e}_n$ are expected to have identical signs. From these observations, it is possible to devise algorithms that decrease or increase the step size depending on successive values of $\tilde{e}_n$. A relatively simple rule devised by Jayant (1970) is to adaptively vary the step size according to the relation

$$\Delta_n = \Delta_{n-1} K^{\tilde{e}_n \tilde{e}_{n-1}}, \qquad n = 1, 2, \ldots$$

where $K \geq 1$ is a constant that is selected to minimize the total distortion. A block diagram of a DM encoder–decoder that incorporates this adaptive algorithm is illustrated in Fig. 3-5-9.

Several other variations of adaptive DM encoding have been investigated and described in the technical literature. A particularly effective and popular technique first proposed by Greefkes (1970) is called *continuously variable slope delta modulation* (CVSD). In CVSD the adaptive step-size parameter may be expressed as

$$\Delta_n = \alpha \Delta_{n-1} + k_1$$

if $\tilde{e}_n$, $\tilde{e}_{n-1}$, and $\tilde{e}_{n-2}$ have the same sign; otherwise,

$$\Delta_n = \alpha \Delta_{n-1} + k_2$$

The parameters α, k_1, and k_2 are selected such that $0 < \alpha < 1$ and $k_1 \gg k_2 > 0$. For more discussion on this and other variations of adaptive DM, the interested reader is referred to the papers by Jayant (1974) and Flanagan *et al.* (1979), which contain extensive references.

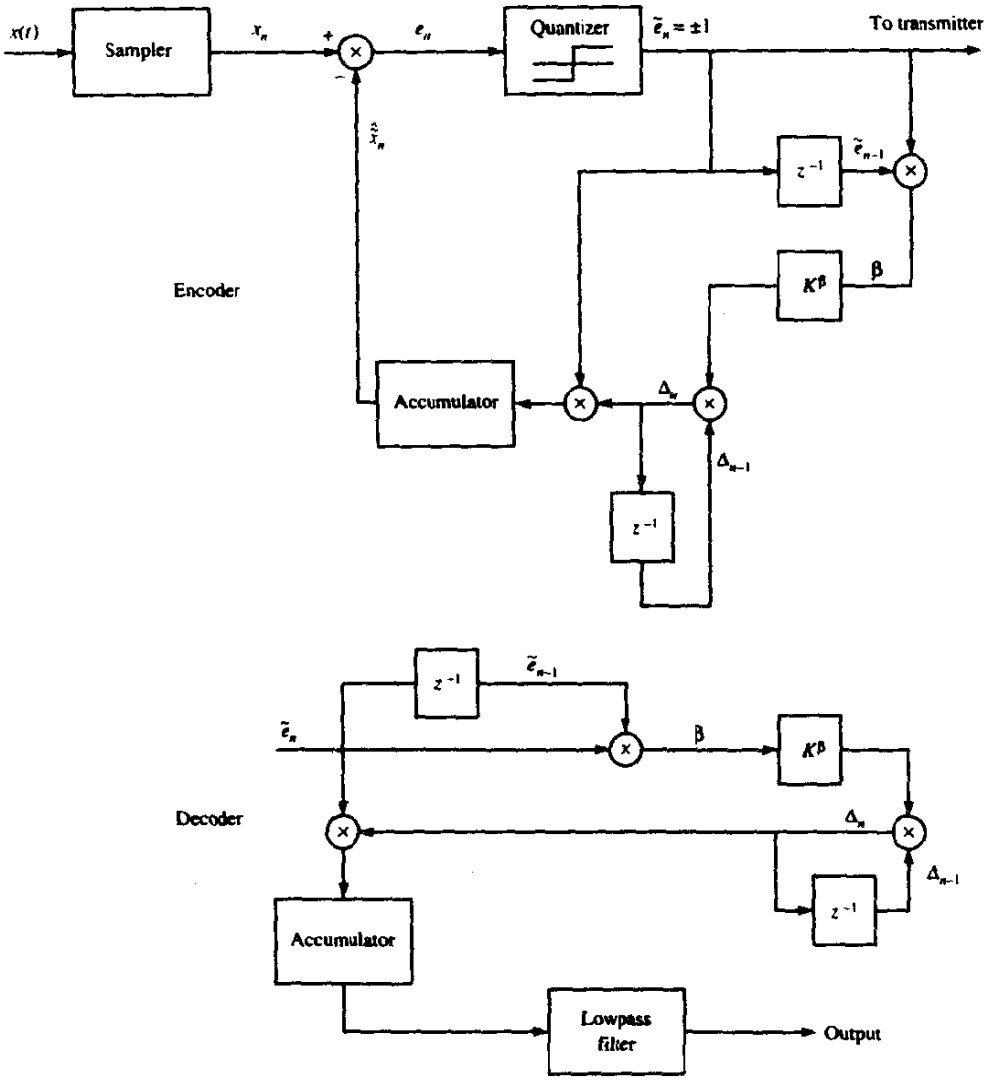


FIGURE 3-5-9 An example of a delta modulation system with adaptive step size.

PCM, DPCM, adaptive PCM, and adaptive DPCM and DM are all source encoding techniques that attempt to faithfully represent the output waveform from the source. The following class of waveform encoding methods is based on a spectral decomposition of the source signal.

3-5-2 Spectral Waveform Coding

In this section, we briefly describe waveform coding methods that filter the source output signal into a number of frequency bands or subbands and separately encode the signal in each subband. The waveform encoding may be

performed either on the time-domain waveforms in each subband or on the frequency-domain representation of the corresponding time-domain waveform in each subband.

Subband Coding In subband coding (SBC) of speech and image signals, the signal is divided into a small number of subbands and the time waveform in each subband is encoded separately. In speech coding, for example, the lower-frequency bands contain most of the spectral energy in voiced speech. In addition, quantization noise is more noticeable to the ear in the lower-frequency bands. Consequently, more bits are used for the lower-band signals and fewer are used for the higher-frequency bands.

Filter design is particularly important in achieving good performance in SBC. In practice, quadrature–mirror filters (QMFs) are generally used because they yield an alias-free response due to their perfect reconstruction property (see Vaidyanathan, 1993). By using QMFs in subband coding, the lower-frequency band is repeatedly subdivided by factors of two, thus creating octave-band filters. The output of each QMF filter is decimated by a factor of two, in order to reduce the sampling rate. For example, suppose that the bandwidth of a speech signal extends to 3200 Hz. The first pair of QMFs divides the spectrum into the low (0–1600 Hz) and high (1600–3200 Hz) bands. Then, the low band is split into low (0–800 Hz) and high (800–1600 Hz) bands by the use of another pair of QMFs. A third subdivision by another pair of QMFs can split the 0–800 Hz band into low (0–400 Hz) and high (400–800 Hz) bands. Thus, with three pairs of QMFs, we have obtained signals in the frequency bands 0–400, 400–800, 800–1600 and 1600–3200 Hz. The time-domain signal in each subband may now be encoded with different precision. In practice, adaptive PCM has been used for waveform encoding of the signal in each subband.

Adaptive Transform Coding In adaptive transform coding (ATC), the source signal is sampled and subdivided into frames of N_f samples, and the data in each frame is transformed into the spectral domain for coding and transmission. At the source decoder, each frame of spectral samples is transformed back into the time domain and the signal is synthesized from the time-domain samples and passed through a D/A converter. To achieve coding efficiency, we assign more bits to the more important spectral coefficients and fewer bits to the less important spectral coefficients. In addition, by designing an adaptive allocation in the assignment of the total number of bits to the spectral coefficients, we can adapt to possibly changing statistics of the source signal.

An objective in selecting the transformation from the time domain to the frequency domain is to achieve uncorrelated spectral samples. In this sense, the Karhunen–Loéve transform (KLT) is optimal in that it yields spectral values that are uncorrelated, but the KLT is generally difficult to compute (see

Wintz, 1972). The DFT and the *discrete cosine transform* (DCT) are viable alternatives, although they are suboptimum. Of these two, the DCT yields good performance compared with the KLT, and is generally used in practice (see Campanella and Robinson, 1971; Zelinsky and Noll, 1977).

In speech coding using ATC, it is possible to attain communication-quality speech at a rate of about 9600 bits/s.

3-5-3 Model-Based Source Coding

In contrast to the waveform encoding methods described above, model-based source coding represents a completely different approach. In this, the source is modeled as a linear system (filter) that, when excited by an appropriate input signal, results in the observed source output. Instead of transmitting the samples of the source waveform to the receiver, the parameters of the linear system are transmitted along with an appropriate excitation signal. If the number of parameters is sufficiently small, the model-based methods provide a large compression of the data.

The most widely used model-based coding method is called *linear predictive coding* (LPC). In this, the sampled sequence, denoted by x_n, $n = 0, 1, \ldots, N-1$, is assumed to have been generated by an all-pole (discrete-time) filter having the transfer function

$$H(z) = \frac{G}{1 - \sum_{k=1}^{p} a_k z^{-k}} \tag{3-5-18}$$

Appropriate excitation functions are an impulse, a sequence of impulses, or a sequence of white noise with unit variance. In any case, suppose that the input sequence is denoted by v_n, $n = 0, 1, 2, \ldots$. Then the output sequence of the all-pole model satisfies the difference equation

$$x_n = \sum_{k=1}^{p} a_k x_{n-k} + G v_n, \qquad n = 0, 1, 2, \ldots \tag{3-5-19}$$

In general, the observed source output x_n, $n = 0, 1, 2, \ldots, N-1$, does not satisfy the difference equation (3-5-19), but only its model does. If the input is a white-noise sequence or an impulse, we may form an estimate (or prediction) of x_n by the weighted linear combination

$$\hat{x}_n = \sum_{k=1}^{p} a_k x_{n-k}, \qquad n > 0 \tag{3-5-20}$$

The difference between x_n and $\hat{x}_n$, namely,

$$\begin{aligned} e_n &= x_n - \hat{x}_n \\ &= x_n - \sum_{k=1}^{p} a_k x_{n-k} \end{aligned} \tag{3-5-21}$$

represents the error between the observed value x_n and the estimated (predicted) value $\hat{x}_n$. The filter coefficients $\{a_k\}$ can be selected to minimize the mean square value of this error.

Suppose for the moment that the input $\{v_n\}$ is a white-noise sequence. Then, the filter output x_n is a random sequence and so is the difference $e_n = x_n - \hat{x}_n$. The ensemble average of the squared error is

$$\begin{aligned}\mathscr{E}_p &= E(e_n^2) \\ &= E\left[\left(x_n - \sum_{k=1}^{p} a_k x_{n-k}\right)^2\right] \\ &= \phi(0) - 2\sum_{k=1}^{p} a_k \phi(k) + \sum_{k=1}^{p}\sum_{m=1}^{p} a_k a_m \phi(k-m)\end{aligned} \tag{3-5-22}$$

where $\phi(m)$ is the autocorrelation function of the sequence x_n, $n = 0, 1, \ldots, N-1$. But $\mathscr{E}_p$ is identical to the MSE given by (3-5-8) for a predictor used in DPCM. Consequently, minimization of $\mathscr{E}_p$ in (3-5-22) yields the set of normal equations given previously by (3-5-9). To completely specify the filter $H(z)$, we must also determine the filter gain G. From (3-5-19), we have

$$E[(Gv_n)^2] = G^2 E(v_n^2) = G^2 = E\left[\left(x_n - \sum_{k=1}^{p} a_k x_{n-k}\right)^2\right] = \mathscr{E}_p \tag{3-5-23}$$

where $\mathscr{E}_p$ is the residual MSE obtained from (3-5-22) by substituting the optimum prediction coefficients, which result from the solution of (3-5-9). With this substitution, the expression for $\mathscr{E}_p$ and, hence, G^2 simplifies to

$$\mathscr{E}_p = G^2 = \phi(0) - \sum_{k=1}^{p} a_k \phi(k) \tag{3-5-24}$$

In practice, we do not usually know *a priori* the true autocorrelation function of the source output. Hence, in place of $\phi(n)$, we substitute an estimate $\hat{\phi}(n)$ as given by (3-5-10), which is obtained from the set of samples x_n, $n = 0, 1, \ldots, N-1$, emitted by the source.

As indicated previously, the Levinson–Durbin algorithm derived in Appendix A may be used to solve for the predictor coefficients $\{a_k\}$ recursively, beginning with a first-order predictor and iterating the order of the predictor up to order p. The recursive equations for the $\{a_k\}$ may be expressed as

$$\begin{aligned}a_{ii} &= \frac{\hat{\phi}(i) - \sum_{k=1}^{i-1} a_{i-1,k}\hat{\phi}(i-k)}{\hat{\mathscr{E}}_{i-1}} \qquad i = 2, 3, \ldots, p \\ a_{ik} &= a_{i-1,k} - a_{ii}a_{i-1,i-k}, \qquad 1 \le k \le i-1 \\ \hat{\mathscr{E}}_i &= (1 - a_{ii})\hat{\mathscr{E}}_{i-1} \\ a_{11} &= \frac{\hat{\phi}(1)}{\hat{\phi}(0)}, \qquad \hat{\mathscr{E}}_0 = \hat{\phi}(0)\end{aligned} \tag{3-5-25}$$

where a_{ik}, $k = 1, 2, \ldots, i$, are the coefficients of the ith-order predictor. The desired coefficients for the predictor of order p are

$$a_k \equiv a_{pk}, \quad k = 1, 2, \ldots, p \tag{3-5-26}$$

and the residual MSE is

$$\hat{\mathcal{E}} = G^2 = \hat{\phi}(0) - \sum_{k=1}^{p} a_k \hat{\phi}(k)$$

$$= \hat{\phi}(0) \prod_{i=1}^{p} (1 - a_{ii}^2) \tag{3-5-27}$$

We observe that the recursive relations in (3-5-25) give us not only the coefficients of the predictor for order p, but also the predictor coefficients of all orders less than p.

The residual MSE $\hat{\mathcal{E}}_i$, $i = 1, 2, \ldots, p$, forms a monotone decreasing sequence, i.e. $\hat{\mathcal{E}}_p \leq \hat{\mathcal{E}}_{p-1} \leq \ldots \leq \hat{\mathcal{E}}_1 \leq \hat{\mathcal{E}}_0$, and the prediction coefficients a_{ii} satisfy the condition

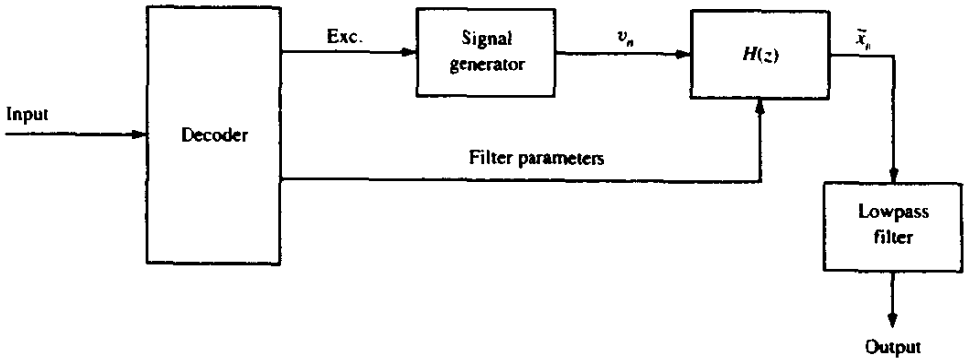
$$|a_{ii}| < 1, \quad i = 1, 2, \ldots, p \tag{3-5-28}$$

This condition is necessary and sufficient for all the poles of $H(z)$ to be inside the unit circle. Thus (3-5-28) ensures that the model is stable.

LPC has been successfully used in the modeling of a speech source. In this case, the coefficients a_{ii}, $i = 1, 2, \ldots, p$, are called *reflection coefficients* as a consequence of their correspondence to the reflection coefficients in the acoustic tube model of the vocal tract (see Rabiner and Schafer, 1978; Deller *et al.*, 1993).

Once the predictor coefficients and the gain G have been estimated from the source output $\{x_n\}$, each parameter is coded into a sequence of binary digits and transmitted to the receiver. Source decoding or waveform synthesis may be accomplished at the receiver as illustrated in Fig. 3-5-10. The signal generator is used to produce the excitation function $\{v_n\}$, which is scaled by G

FIGURE 3-5-10 Block diagram of a waveform synthesizer (source decoder) for an LPC system.



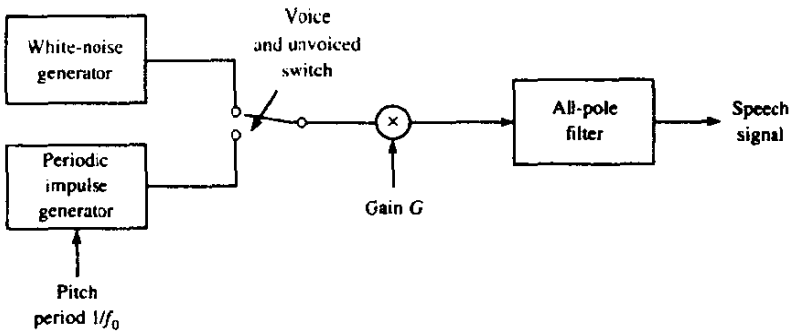


FIGURE 3-5-11 Block diagram model of the generation of a speech signal.

to produce the desired input to the all-pole filter model $H(z)$ synthesized from the received prediction coefficients. The analog signal may be reconstructed by passing the output sequence from $H(z)$ through an analog filter that basically performs the function of interpolating the signal between sample points. In this realization of the waveform synthesizer, the excitation function and the gain parameter must be transmitted along with the prediction coefficients to the receiver.

When the source output is stationary, the filter parameters need to be determined only once. However, the statistics of most sources encountered in practice are at best quasistationary. Under these circumstances, it is necessary to periodically obtain new estimates of the filter coefficients, the gain G, and the type of excitation function, and to transmit these estimates to the receiver.

Example 3-5-1

The block diagram shown in Fig. 3-5-11 illustrates a model for a speech source. There are two mutually exclusive excitation functions to model voiced and unvoiced speech sounds. On a short-time basis, voiced speech is periodic with a fundamental frequency f_0 or a pitch period $1/f_0$ that depends on the speaker. Thus voiced speech is generated by exciting an all-pole filter model of the vocal tract by a periodic impulse train with a period equal to the desired pitch period. Unvoiced speech sounds are generated by exciting the all-pole filter model by the output of a random-noise generator. The speech encoder at the transmitter must determine the proper excitation function, the pitch period for voiced speech, the gain parameter G, and the prediction coefficients. These parameters are encoded into binary digits and transmitted to the receiver. Typically, the voiced and unvoiced information requires 1 bit, the pitch period is adequately represented by 6 bits, and the gain parameter may be represented by 5 bits after its dynamic range is compressed logarithmically. The prediction coefficients require 8–10 bits/coefficient for adequate representation (see Rabiner and Schafer, 1978). The reason for such high accuracy is that relatively small changes in

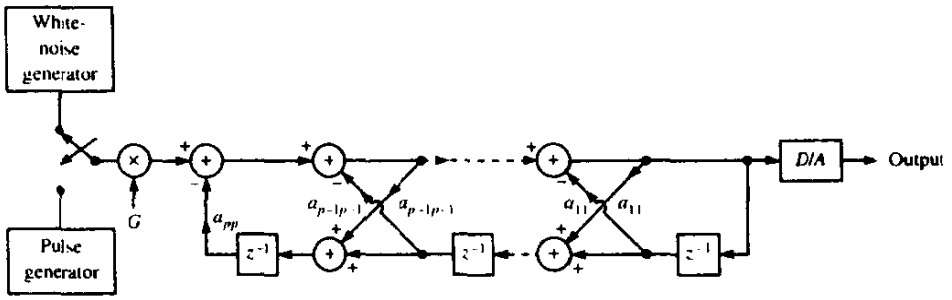


FIGURE 3-5-12 All-pole lattice filter for synthesizing the speech signal.

the prediction coefficients result in a large change in the pole positions of the filter model $H(z)$. The accuracy requirements may be lessened by transmitting the reflection coefficients a_{ii}, which have a smaller dynamic range. These are adequately represented by 6 bits. Thus, for a predictor of order $p = 10$ [five poles in $H(z)$], the total number of bits is 72. Due to the quasistationary nature of the speech signal, the linear system model must be changed periodically, typically once every 15–30 ms. Consequently, the bit rate from the source encoder is in the range 4800–2400 bit/s.

When the reflection coefficients are transmitted to the decoder, it is not necessary to recompute the prediction coefficients in order to realize the speech synthesizer. Instead, the synthesis is performed by realizing a lattice filter, shown in Fig. 3-5-12, which utilizes the reflection coefficients directly and which is equivalent to the linear prediction filter.

The linear all-pole filter model, for which the filter coefficients are estimated via linear prediction, is by far the simplest linear model for a source. A more general source model is a linear filter that contains both poles and zeros. In a pole–zero model, the source output x_n satisfies the difference equation

$$x_n = \sum_{k=1}^{p} a_k x_{n-k} + \sum_{k=0}^{q} b_k v_{n-k}$$

where v_n is the input excitation sequence. The problem now is to estimate the filter parameters $\{a_k\}$ and $\{b_k\}$ from the data x_i, $i = 0, 1, \ldots, N-1$, emitted by the source. However, the MSE criterion applied to the minimization of the error $e_n = x_n - \hat{x}_n$, where $\hat{x}_n$ is an estimate of x_n, results in a set of nonlinear equations for the parameters $\{a_k\}$ and $\{b_k\}$. Consequently, the evaluation of the $\{a_k\}$ and $\{b_k\}$ becomes tedious and difficult mathematically. To avoid having to solve the nonlinear equations, a number of suboptimum methods have been devised for pole–zero modeling. A discussion of these techniques would lead us too far afield, however.

LPC as described above forms the basis for more complex model-based source encoding methods. When applied to speech coding, the model-based

methods are generally called *vocoders* (for voice coders). In addition to the conventional LPC vocoder described above, other types of vocoders that have been implemented include the residual excited LPC (RELP) vocoder, the multipulse LPC vocoder, the code-excited LPC (CELP) vocoder, and the vector-sum-excited LPC (VSELP) vocoder. The CELP and VSELP vocoders employ vector-quantized excitation codebooks to achieve communication quality speech at low bit rates.

Before concluding this section, we consider the application of waveform encoding and LPC to the encoding of speech signals and compare the bit rates of these coding techniques.

Encoding Methods Applied to Speech Signals The transmission of speech signals over telephone lines, radio channels, and satellite channels constitutes by far the largest part of our daily communications. It is understandable, therefore, that over the past three decades more research has been performed on speech encoding than on any other type of information-bearing signal. In fact, all the encoding techniques described in this section have been applied to the encoding of speech signals. It is appropriate, therefore, to compare the efficiency of these methods in terms of the bit rate required to transmit the speech signal.

The speech signal is assumed to be band-limited to the frequency range 200–3200 Hz and sampled at a nominal rate of 8000 samples/s for all encoders except DM, where the sampling rate is f_s identical to the bit rate. For an LPC encoder, the parameters given in Example 3-5-1 are assumed.

Table 3-5-2 summarizes the main characteristics of the encoding methods described in this section and the required bit rate. In terms of the quality of the speech signal synthesized at the receiver from the (error-free) binary sequence, all the waveform encoding methods (PCM, DPCM, ADPCM, DM, ADM) provide telephone (toll) quality speech. In other words, a listener would have difficulty discerning the difference between the digitized speech and the analog speech waveform. ADPCM and ADM are particularly efficient waveform encoding techniques. With CVSD, it is possible to operate down to 9600 bits/s

TABLE 3-5-2 ENCODING TECHNIQUES APPLIED TO SPEECH SIGNALS

Encoding method	Quantizer	Coder	Transmission rate (bits/s)
PCM	Linear	12 bits	96 000
Log PCM	Logarithamic	7–8 bits	56 000–64 000
DPCM	Logarithmic	4–6 bits	32 000–48 000
ADPCM	Adaptive	3–4 bits	24 000–32 000
DM	Binary	1 bit	32 000–64 000
ADM	Adaptive binary	1 bit	16 000–32 000
LPC			2400–4800

with some noticeable waveform distortion. In fact, at rates below 16 000 bits/s, the distortion produced by waveform encoders increases significantly. Consequently, these techniques are not used below 9600 bits/s.

For rates below 9600 bits/s, encoding techniques, such as LPC, that are based on linear models of the source are usually employed. The synthesized speech obtained from this class of encoding techniques is intelligible. However, the speech signal has a synthetic quality and there is noticeable distortion.

3-6 BIBLIOGRAPHICAL NOTES AND REFERENCES

Source coding has been an area of intense research activity since the publication of Shannon's classic papers in 1948 and the paper by Huffman (1952). Over the years, major advances have been made in the development of highly efficient source data compression algorithms. Of particular significance is the research on universal source coding and universal quantization published by Ziv (1985), Ziv and Lempel (1977, 1978), Davisson (1973), Gray (1975), and Davisson *et al.* (1981).

Treatments of rate distortion theory are found in the books by Gallager (1968). Berger (1971), Viterbi and Omura (1979), Blahut (1987) and Gray (1990).

Much work has been done over the past several decades on speech encoding methods. Our treatment provides an overview of this important topic. A more comprehensive treatment is given in the books by Rabiner and Schafer (1978), Jayant and Noll (1984), and Deller *et al.* (1993). In addition to these texts, there have been special issues of the *IEEE Transactions on Communications* (April 1979 and April 1982) and, more recently, the *IEEE Journal on Selected Areas in Communications* (February 1988) devoted to speech encoding. We should also mention the publication by IEEE Press of a book containing reprints of published papers on waveform quantization and coding, edited by Jayant (1976).

Over the past decade, we have also seen a number of important developments in vector quantization. Our treatment of this topic was based on the tutorial paper by Makhoul *et al.* (1985). A comprehensive treatment of vector quantization and signal compression is provided in the book by Gersho and Gray (1992).

PROBLEMS

3-1 Consider the joint experiment described in Problem 2-1 with the given joint probabilities $P(A_i, B_j)$. Suppose we observe the outcomes A_i, $i = 1, 2, 3, 4$ of experiment A.

a Determine the mutual information $I(B_j; A_i)$ for $j = 1, 2, 3$ and $i = 1, 2, 3, 4$, in bits.

b Determine the average mutual information $I(B; A)$.

3-2 Suppose the outcomes B_j, $j = 1, 2, 3$, in Problem 3-1 represent the three possible output letters from the DMS. Determine the entropy of the source.

3-3 Prove that $\ln u \leq u - 1$ and also demonstrate the validity of this inequality by plotting $\ln u$ and $u - 1$ on the same graph.

3-4 X and Y are two discrete random variables with probabilities

$$P(X = x, Y = y) \equiv P(x, y)$$

Show that $I(X; Y) \geq 0$, with equality if and only if X and Y are statistically independent.

[*Hint:* Use the inequality $\ln u < u - 1$, for $0 < u < 1$, to show that $-I(X; Y) \leq 0$.]

3-5 The output of a DMS consists of the possible letters $x_1, x_2, \ldots, x_n$, which occur with probabilities $p_1, p_2, \ldots, p_n$, respectively. Prove that the entropy $H(X)$ of the source is at most $\log n$.

3-6 Determine the differential entropy $H(X)$ of the uniformly distributed random variable X with pdf

$$p(x) = \begin{cases} a^{-1} & (0 \leq x \leq a) \\ 0 & (\text{otherwise}) \end{cases}$$

for the following three cases:

a $a = 1$;

b $a = 4$;

c $a = \frac{1}{4}$.

Observe from these results that $H(X)$ is not an absolute measure, but only a relative measure of randomness.

3-7 A DMS has an alphabet of eight letters, x_i, $i = 1, 2, \ldots, 8$, with probabilities 0.25, 0.20, 0.15, 0.12, 0.10, 0.08, 0.05, and 0.05.

a Use the Huffman encoding procedure to determine a binary code for the source output.

b Determine the average number $\bar{R}$ of binary digits per source letter.

c Determine the entropy of the source and compare it with $\bar{R}$.

3-8 A DMS has an alphabet of five letters, x_i, $i = 1, 2, \ldots, 5$, each occurring with probability $\frac{1}{5}$. Evaluate the efficiency of a fixed-length binary code in which

a each letter is encoded separately into a binary sequence;

b two letters at a time are encoded into a binary sequence;

c three letters at a time are encoded into a binary sequence.

3-9 Recall (3-2-6):

$$I(x_i; y_j) = I(x_i) - I(x_i \mid y_j)$$

Prove that

a $I(x_i; y_j) = I(y_j) - I(y_j \mid x_i)$;

b $I(x_i; y_j) = I(x_i) + I(y_j) - I(x_i y_j)$, where $I(x_i y_j) = -\log P(x_i, y_j)$.

3-10 Let X be a geometrically distributed random variable; that is,

$$p(X = k) = p(1 - p)^{k-1}, \quad k = 1, 2, 3, \ldots$$

a Find the entropy of X.

b Knowing that $X > K$, where K is a positive integer, what is the entropy of X?

3-11 Let X and Y denote two jointly distributed discrete valued random variables.

a Show that

$$H(X) = -\sum_{x,y} P(x, y) \log P(x)$$

$$H(Y) = -\sum_{x,y} P(x, y) \log P(y)$$

b Use the above result to show that

$$H(X, Y) \leq H(X) + H(Y)$$

When does equality hold?

c Show that

$$H(X \mid Y) \leq H(X)$$

with equality if and only if X and Y are independent.

3-12 Two binary random variables X and Y are distributed according to the joint distributions $p(X = Y = 0) = p(X = 0, Y = 1) = p(X = Y = 1) = \frac{1}{3}$. Compute $H(X)$, $H(Y)$, $H(X \mid Y)$, $H(Y \mid X)$, and $H(X, Y)$.

3-13 A Markov process is a process with one-step memory, i.e., a process such that

$$p(x_n \mid x_{n-1}, x_{n-2}, x_{n-3}, \ldots) = p(x_n \mid x_{n-1})$$

for all n. Show that, for a stationary Markov process, the entropy rate is given by $H(X_n \mid X_{n-1})$.

3-14 Let $Y = g(X)$, where g denotes a deterministic function. Show that, in general, $H(Y) \leq H(X)$. When does equality hold?

3-15 Show that $I(X; Y) = H(X) + H(Y) - H(XY)$.

3-16 Show that, for statistically independent events,

$$H(X_1 X_2 \cdots X_n) = \sum_{i=1}^{n} H(X_i)$$

3-17 For a noiseless channel, show that $H(X \mid Y) = 0$.

3-18 Show that

$$I(X_3; X_2 \mid X_1) = H(X_3 \mid X_1) - H(X_3 \mid X_1 X_2)$$

and that

$$H(X_3 \mid X_1) \geq H(X_3 \mid X_1 X_2)$$

3-19 Let X be a random variable with pdf $p_X(x)$ and let $Y = aX + b$ be a linear transformation of X, where a and b are two constants. Determine the differential entropy $H(Y)$ in terms of $H(X)$.

3-20 The outputs x_1, x_2, and x_3 of a DMS with corresponding probabilities $p_1 = 0.45$, $p_2 = 0.35$, and $p_3 = 0.20$ are transformed by the linear transformation $Y = aX + b$, where a and b are constants. Determine the entropy $H(Y)$ and comment on what effect the transformation has had on the entropy of X.

3-21 The optimum four-level nonuniform quantizer for a gaussian-distributed signal amplitude results in the four levels a_1, a_2, a_3, and a_4, with corresponding probabilities of occurrence $p_1 = p_2 = 0.3365$ and $p_3 = p_4 = 0.1635$.

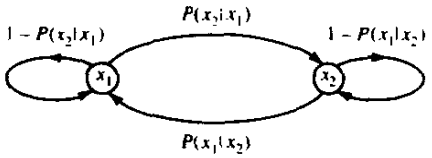


FIGURE P3-22

a Design a Huffman code that encodes a single level at a time and determine the average bit rate.

b Design a Huffman code that encodes two output levels at a time and determine the average bit rate.

c What is the minimum rate obtained by encoding J output levels at a time as $J \to \infty$?

3-22 A first-order Markov source is characterized by the state probabilities $P(x_i)$, $i = 1, 2, \ldots, L$, and the transition probabilities $P(x_k \mid x_i)$, $k = 1, 2, \ldots, L$, and $k \neq i$. The entropy of the Markov source is

$$H(X) = \sum_{k=1}^{L} P(x_k) H(X \mid x_k)$$

where $H(X \mid x_k)$ is the entropy conditioned on the source being in state x_k.

Determine the entropy of the binary, first-order Markov source shown in Fig. P3-22, which has the transition probabilities $P(x_2 \mid x_1) = 0.2$ and $P(x_1 \mid x_2) = 0.3$. [Note that the conditional entropies $H(X \mid x_1)$ and $H(X \mid x_2)$ are given by the binary entropy functions $H[P(x_2 \mid x_1)]$ and $H[P(x_1 \mid x_2)]$, respectively.] How does the entropy of the Markov source compare with the entropy of a binary DMS with the same output letter probabilities $P(x_1)$ and $P(x_2)$?

3-23 A memoryless source has the alphabet $\mathcal{A} = \{-5, -3, -1, 0, 1, 3, 5\}$, with corresponding probabilities $\{0.05, 0.1, 0.1, 0.15, 0.05, 0.25, 0.3\}$.

a Find the entropy of the source.

b Assuming that the source is quantized according to the quantization rule

$$q(-5) = q(-3) = 4$$
$$q(-1) = q(0) = q(1) = 0$$
$$q(3) = q(5) = 4$$

find the entropy of the quantized source.

3-24 Design a *ternary* Huffman code, using 0, 1, and 2 as letters, for a source with output alphabet probabilities given by $\{0.05, 0.1, 0.15, 0.17, 0.18, 0.22, 0.13\}$. What is the resulting average codeword length? Compare the average codeword length with the entropy of the source. (In what base would you compute the logarithms in the expression for the entropy for a meaningful comparison?)

3-25 Find the Lempel–Ziv source code for the binary source sequence

0001001000000110000100000001000000101000010000001101010000000 1100

Recover the original sequence back from the Lempel–Ziv source code.

[*Hint:* You require two passes of the binary sequence to decide on the size of the dictionary.]

3-26 Find the differential entropy of the continuous random variable X in the following cases:

a X is an exponential random variable with parameter $\lambda > 0$, i.e.,

$$f_X(x) = \begin{cases} \lambda^{-1} e^{-x/\lambda} & (x > 0) \\ 0 & \text{(otherwise)} \end{cases}$$

b X is a Laplacian random variable with parameter $\lambda > 0$, i.e.,

$$f_X(x) = \frac{1}{2\lambda} e^{-|x|/\lambda}$$

c X is a triangular random variable with parameter $\lambda > 0$, i.e.,

$$f_X(x) = \begin{cases} (x + \lambda)/\lambda^2 & (-\lambda \leq x \leq 0) \\ (-x + \lambda)/\lambda^2 & (0 < x \leq \lambda) \\ 0 & \text{(otherwise)} \end{cases}$$

3-27 It can be shown that the rate-distortion function for a Laplacian source, $f_X(x) = (2\lambda)^{-1} e^{-|x|/\lambda}$ with an absolute value of error-distortion measure $d(x, \hat{x}) = |x - \hat{x}|$ is given by

$$R(D) = \begin{cases} \log(\lambda/D) & (0 \leq D \leq \lambda) \\ 0 & (D > \lambda) \end{cases}$$

(see Berger, 1971).

a How many bits per sample are required to represent the outputs of this source with an average distortion not exceeding $\frac{1}{2}\lambda$?

b Plot $R(D)$ for three different values of λ and discuss the effect of changes in λ on these plots.

3-28 It can be shown that if X is a zero-mean continuous random variable with variance σ^2, its rate distortion function, subject to squared error distortion measure, satisfies the lower and upper bounds given by the inequalities

$$h(X) - \tfrac{1}{2} \log 2\pi e D \leq R(D) \leq \tfrac{1}{2} \log \tfrac{1}{2}\sigma^2$$

where $h(X)$ denotes the differential entropy of the random variable X (see Cover and Thomas, 1991).

a Show that, for a Gaussian random variable, the lower and upper bounds coincide.

b Plot the lower and upper bounds for a Laplacian source with $\sigma = 1$.

c Plot the lower and upper bounds for a triangular source with $\sigma = 1$.

3-29 A stationary random process has an autocorrelation function given by $R_X = \frac{1}{2}A^2 e^{-|\tau|} \cos 2\pi f_0 \tau$ and it is known that the random process never exceeds 6 in magnitude. Assuming $A = 6$, how many quantization levels are required to guarantee a signal-to-quantization noise ratio of at least 60 dB?

3-30 An additive white gaussian noise channel has the output $Y = X + G$, where X is the channel input and G is the noise with probability density function

$$p(n) = \frac{1}{\sqrt{2\pi}\,\sigma_n} = e^{-n^2/2\sigma_n^2}$$

If X is a white gaussian input with $E(X) = 0$ and $E(X^2) = \sigma_x^2$, determine

a the conditional differential entropy $H(X \mid G)$;

b the average mutual information $I(X; Y)$.

3-31 A DMS has an alphabet of eight letters, x_i, $i = 1, 2, \ldots, 8$, with probabilities

given in Problem 3-7. Use the Huffman encoding procedure to determine a ternary code (using symbols 0, 1, and 2) for encoding the source output.

[*Hint:* Add a symbol x_9 with probability $p_9 = 0$, and group three symbols at a time.]

3-32 Determine whether there exists a binary code with code word lengths $(n_1, n_2, n_3, n_4) = (1, 2, 2, 3)$ that satisfy the prefix condition.

3-33 Consider a binary block code with 2^n code words of the same length n. Show that the Kraft inequality is satisfied for such a code.

3-34 Show that the entropy of an n-dimensional gaussian vector $\mathbf{X} = [x_1\ x_2\ \ldots\ x_n]$ with zero mean and covariance matrix $\mathbf{M}$ is

$$H(\mathbf{X}) = \tfrac{1}{2}\log_2 (2\pi e)^n\, |\boldsymbol{M}|$$

3-35 Consider a DMS with output bits $(0, 1)$ that are equiprobable. Define the distortion measure as $D = P_e$, where P_e is the probability of error in transmitting the binary symbols to the user over a BSC. Then the rate distortion function is (Berger, 1971)

$$R(D) = 1 + D\log_2 D + (1 - D)\log_2 (1 - D), \qquad 0 \leq D = P_e \leq \tfrac{1}{2}$$

Plot $R(D)$ for $0 \leq D \leq \frac{1}{2}$.

3-36 Evaluate the rate distortion function for an M-ary symmetric channel where $D = P_M$ and

$$R(D) = \log_2 M + D\log_2 D + (1 - D)\log_2 \frac{1 - D}{M - 1}$$

for $M = 2, 4, 8$, and 16. P_M is the probability of error.

3-37 Consider the use of the weighted mean-square-error (MSE) distortion measure defined as

$$d_w(\mathbf{X}, \tilde{\mathbf{X}}) = (\mathbf{X} - \tilde{\mathbf{X}})'\mathbf{W}(\mathbf{X} - \tilde{\mathbf{X}})$$

where $\mathbf{W}$ is a symmetric, positive-definite wieghting matrix. By factorizing $\mathbf{W}$ as $\mathbf{W} = \mathbf{P}'\mathbf{P}$, show that $d_w(\mathbf{X}, \tilde{\mathbf{X}})$ is equivalent to an unweighted MSE distortion measure $d_2(\mathbf{X}', \tilde{\mathbf{X}}')$ involving transformed vectors $\mathbf{X}'$ and $\tilde{\mathbf{X}}'$.

3-38 Consider a stationary stochastic signal sequence $\{X(n)\}$ with zero mean and autocorrelation sequence

$$\phi(n) = \begin{cases} 1 & (n = 0) \\ \frac{1}{2} & (n = \pm 1) \\ 0 & (\text{otherwise}) \end{cases}$$

a Determine the prediction coefficient of the first-order minimum MSE predictor for $\{X(n)\}$ given by

$$\hat{x}(n) = a_1 x(n - 1)$$

and the corresponding minimum mean square error $\mathscr{E}_1$.

b Repeat (a) for the second-order predictor

$$\hat{x}(n) = a_1 x(n - 1) + a_2 x(n - 2)$$

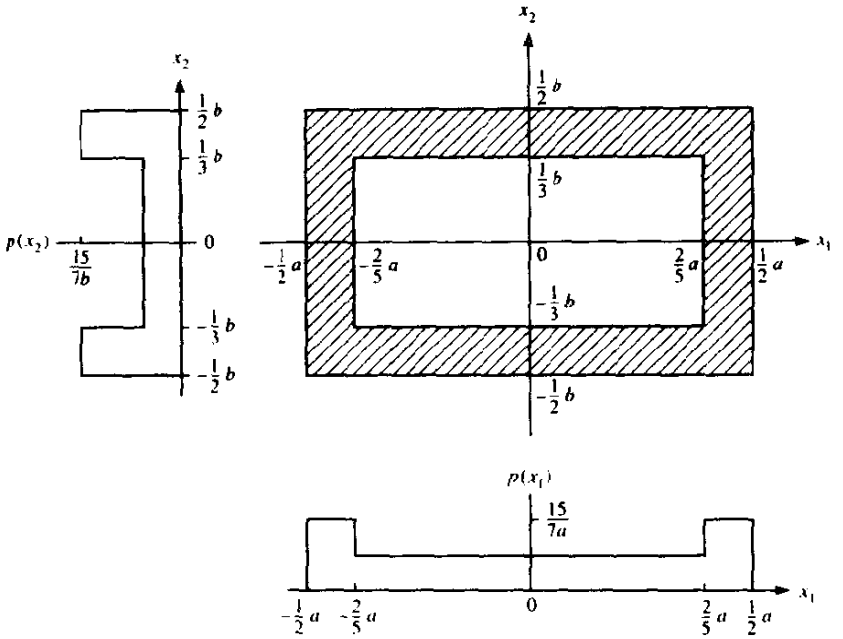


FIGURE P3-39

3-39 Consider the encoding of the random variables x_1 and x_2 that are characterized by the joint pdf $p(x_1, x_2)$ given by

$$p(x_1, x_2) = \begin{cases} 15/7ab & (x_1, x_2 \in C) \\ 0 & (\text{otherwise}) \end{cases}$$

as shown in Fig. P3-39. Evaluate the bit rates required for uniform quantization of x_1 and x_2 separately (scalar quantization) and combined (vector) quantization of (x_1, x_2). Determine the difference in bit rate when $a = 4b$.

3-40 Consider the encoding of two random variables X and Y that are uniformly distributed on the region between two squares as shown in Fig. P3-40.

a Find $f_X(x)$ and $f_Y(y)$.

b Assume that each of the random variables X and Y are quantized using four level uniform quantizers. What is the resulting distortion? What is the resulting number of bits per (X, Y) pair?

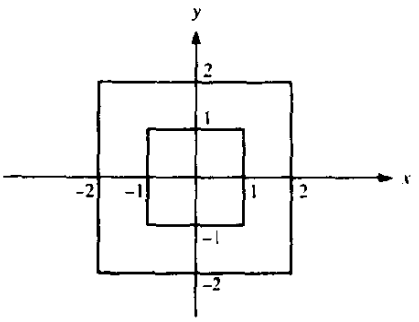


FIGURE P3-40

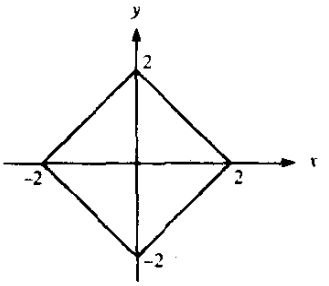


FIGURE P3-41

c Now assume that instead of scalar quantizers for X and Y, we employ a vector quantizer to achieve the same level of distortion as in (b). What is the resulting number of bits per source output pair (X, Y)?

3-41 Two random variables X and Y are uniformly distributed on the square shown in Fig. P3-41.

a Find $f_X(x)$ and $f_Y(y)$.

b Assume that each of the random variables X and Y are quantized using four level uniform quantizers. What is the resulting distortion? What is the resulting number of bits per (X, Y) pair?

c Now assume that, instead of scalar quantizers for X and Y, we employ a vector quantizer with the same number of bits per source output pair (X, Y) as in (b). What is the resulting distortion for this vector quantizer?