CHAPTER 1.2
INFORMATION SOURCES, CODES, AND CHANNELS

Geoffrey C. Orsak, H. Vincent Poor, John B. Thomas

MESSAGE SOURCES

As shown in Fig. 1.1.1, an information source can be considered as emitting a given message u_i from the set $\{U\}$ of possible messages. In general, each message u_i will be represented by a sequence of symbols x_j from the source alphabet $\{X\}$, since the number of possible messages will usually exceed the size M of the source alphabet. Thus sequences of symbols replace the original messages u_i, which need not be considered further. When the source alphabet $\{X\}$ is of finite size M, the source will be called a *finite discrete source*. The problems of concern now are the interrelationships existing between symbols in the generated sequences and the classification of sources according to these interrelationships.

A random or stochastic process x_t, $t \in T$, can be defined as an indexed set of random variables where T is the *parameter set* of the process. If the set T is a sequence, then x_t is a stochastic process with discrete parameter (also called a *random sequence* or *series*). One way to look at the output of a finite discrete source is that it is a discrete-parameter stochastic process with each possible given sequence one of the ensemble members or realizations of the process. Thus the study of information sources can be reduced to a study of random processes.

The simplest case to consider is the *memoryless source*, where the successive symbols obey the same fixed probability law so that the one distribution $p(x_i)$ determines the appearance of each indexed symbol. Such a source is called *stationary*. Let us consider sequences of length n, each member of the sequence being a realization of the random variable x_i with fixed probability distribution $p(x_i)$. Since there are M possible realizations of the random variable and n terms in the sequence, there must be M^n distinct sequences possible of length n. Let the random variable X_i in the jth position be denoted by X_{ij} so that the sequence set (the message set) can be represented by

$$\{U\} = X^n = \{X_{i1}, X_{i2}, \ldots, X_{in}\} \qquad i = 1, 2, \ldots, M \tag{1}$$

The symbol X^n is sometimes used to represent this sequence set and is called the *nth extension of the memoryless source X*. The probability of occurrence of a given message u_i is just the product of the probabilities of occurrence of the individual terms in the sequence so that

$$p\{u_i\} = p(x_{i1})p(x_{i2}) \cdots p\{x_{in}\} \tag{2}$$

Now the entropy for the extended source X^n is

$$H(X^n) = -\sum_{x^n} p\{u_i\} \log p\{u_i\} = nH(X) \tag{3}$$

as expected. Note that, if base 2 logarithms are used, then $H(X)$ has units of bits per symbol, n is symbols per sequence, and $H(X^n)$ is in units of bits per sequence. For a memoryless source, all sequence averages of information measures are obtained by multiplying the corresponding symbol by the number of symbols in the sequence.

MARKOV INFORMATION SOURCE

The memoryless source is not a general enough model in most cases. A constructive way to generalize this model is to assume that the occurrences of a given symbol depends on some number m of immediately preceeding symbols. Thus the information source can be considered to produce an mth-order Markov chain and is called an *mth-order Markov source*.

For an mth-order Markov source, the m symbols preceding a given symbol position are called the *state* s_j of the source at that symbol position. If there are M possible symbols x_i, then the mth-order Markov source will have $M^m = q$ possible states s_j making up the *state set*

$$S = \{s_1, s_2, \ldots, s_q\} \quad q = M^m \tag{4}$$

At a given time corresponding to one symbol position the source will be in a given state s_j. There will exist a probability $p(s_k \mid s_j) = p_{jk}$ that the source will move into another state s_k with the emission of the next symbol. The set of all such conditional probabilities is expressed by the *transition matrix* T, where

$$T = [p_{jk}] = \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1q} \\ p_{21} & p_{22} & \cdots & p_{2q} \\ \cdots & \cdots & \cdots & \cdots \\ p_{q1} & p_{q2} & \cdots & p_{qq} \end{bmatrix} \tag{5}$$

A *Markov matrix* or *stochastic matrix* is any square matrix with nonnegative elements such that the row sums are unity. It is clear that T is such a matrix since

$$\sum_{j=1}^{q} p_{ij} = \sum_{j=1}^{q} p(s_j \mid s_i) = 1 \qquad i = 1, 2, \ldots, q \tag{6}$$

Conversely, any stochastic matrix is a possible transition matrix for a Markov source of order m, where $q = M^m$ is equal to the number of rows or columns of the matrix.

A Markov chain is completely specified by its transition matrix T and by an *initial distribution vector* π giving the probability distribution for the first state occurring. For the memoryless source, the transition matrix reduces to a stochastic matrix where all the rows are identical and are each equal to the initial distribution vector π, which is in turn equal to the vector giving the source alphabet a priori probabilities. Thus, in this case, we have

$$p_{jk} = p(s_k \mid s_j) = p(s_k) = p(x_k) \qquad k = 1, 2, \ldots, M \tag{7}$$

For each state s_i of the source an entropy $H(s_i)$ can be defined by

$$H(s_i) = -\sum_{j=1}^{q} p(s_j \mid s_i) \log p(s_j \mid s_i) = -\sum_{k=1}^{M} p(x_k \mid s_i) \log p(x_k \mid s_i) \tag{8}$$

The source entropy $H(S)$ in information units per symbol is the expected value of $H(s_i)$; that is,

$$H(S) = -\sum_{i=1}^{q} \sum_{j=1}^{q} p(s_i)\, p(s_j \mid s_i) \log p(s_j \mid s_i) = -\sum_{i=1}^{q} \sum_{k=1}^{M} p(s_i)\, p(x_k \mid s_i) \log p(x_k \mid s_i) \tag{9}$$

where $p(s_i) = p_i$ is the *stationary state probability* and is the ith element of the vector **P** defined by

$$\mathbf{P} = [p_1\, p_2 \cdots p_q] \tag{10}$$

It is easy to show, as in Eq. (8), that the source entropy cannot exceed log M, where M is the size of the source alphabet $\{X\}$. For a given source, the ratio of the actual entropy $H(S)$ to the maximum value it can have with the same alphabet is called the *relative entropy* of the source. The *redundancy* η of the source is defined as the positive difference between unity and this relative entropy:

$$\eta = 1 - \frac{H(S)}{\log M} \tag{11}$$

The quantity log M is sometimes called the *capacity* of the alphabet.

NOISELESS CODING

The preceding discussion has emphasized the information source and its properties. We now begin to consider the properties of the communication channel of Fig. 1.1.1. In general, an arbitrary channel will not accept and transmit the sequence of x_i's emitted from an arbitrary source. Instead the channel will accept a sequence of some other elements a_i chosen from a *code alphabet A* of *size D*, where

$$A = \{a_1, a_2, \ldots, a_D\} \tag{12}$$

with D generally smaller than M. The elements a_i of the code alphabet are frequently called *code elements* or *code characters*, while a given sequence of a_i's may be called a *code word*.

The situation is now describable in terms of Fig. 1.1.2, where an encoder E has been added between the source and channel. The process of *coding*, or *encoding*, the source consists of associating with each source symbol x_i a given code word, which is just a given sequence of a_i's. Thus the source emits a sequence of a_i's chosen from the source alphabet A, and the encoder emits a sequence of a_i's chosen from the code alphabet A. It will be assumed in all subsequent discussions that the code words are distinct, i.e., that each code word corresponds to only one source symbol.

Even though each code word is required to be distinct, sequences of code words may not have this property. An example is code A of Table 1.2.1, where a source of size 4 has been encoded in binary code with characters 0 and 1. In code A the code words are distinct, but sequences of code words are not. It is clear that such a code is not *uniquely* decipherable. On the other hand, a given sequence of code words taken from code B will correspond to a distinct sequence of source symbols. An examination of code B shows that in no case is a code word formed by adding characters to another word. In other words, no code word is a *prefix* of another. It is clear that this is a *sufficient* (but not necessary) condition for a code to be uniquely decipherable. That it is not necessary can be seen from an examination of codes C and D of Table 1.2.1. These codes are uniquely decipherable even though many of the code words are prefixes of other words. In these cases any sequence of code words can be decoded by subdividing the sequence of 0s and 1s to the left of every 0 for code C and to the right of every 0 for code D. The character 0 is the first (or last) character of every code word and acts as a comma; therefore this type of code is called a *comma code*.

TABLE 1.2.1 Four Binary Coding Schemes

Source symbol	Code A	Code B	Code C	Code D
x_1	0	0	0	0
x_2	1	10	01	10
x_3	00	110	011	110
x_4	11	111	0111	1110

Note: Code A is not uniquely decipherable; codes B, C, and D are uniquely decipherable; codes B and D are instantaneous codes; and codes C and D are comma codes.

In general the channel will require a finite amount of time to transmit each code character. The code words should be as short as possible in order to maximize information transfer per unit time. The average length L of a code is given by

$$L = \sum_{i=1}^{M} n_i p(x_i) \tag{13}$$

where n_i is the length (number of code characters) of the code word for the source symbol x_i and $p(x_i)$ is the probability of occurrence of x_i. Although the average code length cannot be computed unless the set $\{p(x_i)\}$ is given, it is obvious that codes C and D of Table 1.2.1 will have a greater average length than code B unless $p(x_4) = 0$. Comma codes are not optimal with respect to minimum average length.

Let us encode the sequence $x_3x_1x_3x_2$ into codes B, C, and D of Table 1.2.1 as shown below:

Code B:	110011010
Code C:	011001101
Code D:	110011010

Codes B and D are fundamentally different from code C in that codes B and D can be decoded word by word *without examining subsequent code characters* while code C cannot be so treated. Codes B and D are called *instantaneous codes* while code C is noninstantaneous. The instantaneous codes have the property (previously maintained) that no code word is a prefix of another code word.

The aim of noiseless coding is to produce codes with the two properties of (1) *unique decipherability* and (2) *minimum average length* L for a given source S with alphabet X and probability set $\{p(x_i)\}$. Codes which have both these properties will be called *optimal*. It can be shown that if, for a given source S, a code is optimal among instantaneous codes, then it is optimal among all uniquely decipherable codes. Thus it is sufficient to consider instantaneous codes. A *necessary* property of optimal codes is that source symbols with higher probabilities have shorter code words; i.e.,

$$p(x_i) > p(x_j) \Rightarrow n_i \leq n_j \tag{14}$$

The encoding procedure consists of the assignment of a code word to each of the M source symbols. The code word for the source symbol x_i will be of length n_i; that is, it will consist of n_i code elements chosen from the code alphabet of size D. It can be shown that a necessary and sufficient condition for the construction of a uniquely decipherable code is the *Kraft inequality*

$$\sum_{i=1}^{M} D^{-n_i} \leq 1 \tag{15}$$

NOISELESS-CODING THEOREM

It follows from Eq. (15) that the average code length L, given by Eq. (13), satisfies the inequality

$$L \geq H(X)/\log D \tag{16}$$

Equality (and minimum code length) occurs if and only if the source-symbol probabilities obey

$$p(x_i) = D^{-n_i} \qquad i = 1, 2, \ldots, M \tag{17}$$

A code where this equality applies is called *absolutely optimal*. Since an integer number of code elements must be used for each code word, the equality in Eq. (16) does not usually hold; however, by using one more code element, the average code length L can be bounded from above to give

$$H(X)/\log D \leq L \leq H(X)/\log D + 1 \tag{18}$$

This last relationship is frequently called the *noiseless-coding theorem*.

CONSTRUCTION OF NOISELESS CODES

The easiest case to consider occurs when an absolutely optimal code exists; i.e., when the source-symbol probabilities satisfy Eq. (17). Note that code B of Table 1.2.1 is absolutely optimal if $p(x_1) = 1/2$, $p(x_2) = 1/4$, and $p(x_3) = p(x_4) = 1/8$. In such cases, a procedure for realizing the code for arbitrary code-alphabet size ($D \geq 2$) is easily constructed as follows:

1. Arrange the M source symbols in order of decreasing probability.
2. Arrange the D code elements in an arbitrary but fixed order, i.e., $a_1, a_2, \ldots, a_D$.
3. Divide the set of symbols x_i into D groups with equal probabilities of $1/D$ each. This division is always possible if Eq. (17) is satisfied.
4. Assign the element a_1 as the first digit for symbols in the first group, a_2 for the second, and a_i for the ith group.
5. After the first division each of the resulting groups contains a number of symbols equal to D raised to some integral power if Eq. (17) is satisfied.

Thus, a typical group, say group i, contains D^{ki} symbols, where k_i is an integer (which may be zero). This group of symbols can be further subdivided k_i times into D parts of equal probabilities. Each division decides one additional code digit in the sequence. A typical symbol x_i is isolated after q divisions. If it belongs to the i_1 group after the first division, the i_2 group after the second division, and so forth, then the code word for x_i will be $a_{i1}\, a_{i2} \ldots a_{iq}$.

An illustration of the construction of an absolutely optimal code for the case where $D = 3$ is given in Table 1.2.2. This procedure ensures that source symbols with high probabilities will have short code words and vice versa, since a symbol with probability D^{-n_i} will be isolated after n_i divisions and thus will have n_i elements in its code word, as required by Eq. (17).

TABLE 1.2.2 Construction of an Optimal Code; $D = 3$

Source symbols x_i	A priori probabilities $p(x_i)$	Step 1	Step 2	Step 3	Final code		
x_1	$1/3$	1			1		
x_2	$1/9$	0	1		0	1	
x_3	$1/9$	0	0		0	0	
x_4	$1/9$	0	−1		0	−1	
x_5	$1/27$	−1	1	1	−1	1	1
x_6	$1/27$	−1	1	0	−1	1	0
x_7	$1/27$	−1	1	−1	−1	1	−1
x_8	$1/27$	−1	0	1	−1	0	1
x_9	$1/27$	−1	0	0	−1	0	0
x_{10}	$1/27$	−1	0	−1	−1	0	−1
x_{11}	$1/27$	−1	−1	1	−1	−1	1
x_{12}	$1/27$	−1	−1	0	−1	−1	0
x_{13}	$1/27$	−1	−1	−1	−1	−1	−1

Note: Average code length $L = 2$ code elements per symbol: source entropy $H(X) = 2 \log_2 3$ bits per symbol.

$$L = \frac{H(X)}{\log_2 3}$$

TABLE 1.2.3 Construction of Huffman Code; $D = 2$

Source symbols x_i	A priori probabilities $p(x_i)$	Final code	Reduction 1	Step 5	Reduction 2	Step 4	Reduction 3	Step 3	Reduction 4	Step 2	Reduction 5	Step 1
x_1	0.40	0	0.40	0	0.40	0	0.40	0	0.40	0	0.60	1
x_2	0.20	111	0.20	111	0.20	111	0.24	10	0.36	11	0.40	0
x_3	0.12	101	0.12	101	0.16	110	0.20	111	0.24	10		
x_4	0.08	1101	0.12	100	0.12	101	0.16	110				
x_5	0.08	1100	0.08	1101	0.12	100						
x_6	0.08	1001	0.08	1100								
x_7	0.04	1000										

Average code length $L = 1(0.40) + 3(0.20) + 3(0.12) + 4(0.08) + 4(0.08) + 4(0.08) + 4(0.04)$
$= 2.48$ code elements/symbol

The code resulting from the process just discussed is sometimes called the *Shannon-Fano* code. It is apparent that the same encoding procedure can be followed whether or not the source probabilities satisfy Eq. (17). The set of symbols x_i is simply divided into D groups with probabilities as nearly equal as possible. The procedure is sometimes ambiguous, however, and more than one Shannon-Fano code may be possible. The ambiguity arises, of course, in the choice of approximately equiprobable subgroups.

For the general case where Eq. (17) is not satisfied, a procedure owing to Huffman guarantees an optimal code, i.e., one with minimum average length. This procedure for code alphabet of arbitrary size D is as follows:

1. As before, arrange the M source symbols in order of decreasing probability.
2. As before, arrange the code elements in an arbitrary but fixed order, that is, $a_1, a_2, \ldots, a_D$.
3. Combine (sum) the probabilities of the D least likely symbols and reorder the resulting $M - (D - 1)$ probabilities; this step will be called *reduction* 1. Repeat as often as necessary until there are D ordered probabilities remaining. *Note*: For the binary case ($D = 2$), it will always be possible to accomplish this reduction in $M - 2$ steps. When the size of the code alphabet is arbitrary, the last reduction will result in exactly D ordered probabilities if and only if

$$M = D + n(D - 1)$$

   where n is an integer. If this relationship is not satisfied, *dummy* source symbols with zero probability should be added. The entire encoding procedure is followed as before, and at the end the dummy symbols are thrown away.
4. Start the encoding with the last reduction which consists of exactly D ordered probabilities; assign the element a_1 as the first digit in the code words for all the source symbols associated with the first probability; assign a_2 to the second probability; and a_i to the ith probability.
5. Proceed to the next to the last reduction; this reduction consists of $D + (D - 1)$ ordered probabilities for a net gain of $D - 1$ probabilities. For the D new probabilities, the first code digit has already been assigned and is the same for all of these D probabilities; assign a_1 as the second digit for all source symbols associated with the first of these D new probabilities; assign a_2 as the second digit for the second of these D new probabilities, etc.
6. The encoding procedure terminates after $1 + n(D - 1)$ steps, which is one more than the number of reductions.

As an illustration of the Huffman coding procedure, a binary code is constructed in Table 1.2.3.

CHANNEL CAPACITY

The average mutual information $I(X; Y)$ between an information source and a destination was given by Eqs. (25) and (26) as

$$I(X; Y) = H(Y) - H(Y|X) = H(X) - H(X|Y) \geq 0 \tag{19}$$

The average mutual information depends not only on the statistical characteristics of the channel but also on the distribution $p(x_i)$ of the input alphabet X. If the input distribution is varied until Eq. (19) is a maximum for a given channel, the resulting value of $I(X; Y)$ is called the *channel capacity* C of that channel; i.e.,

$$C = \max_{p(xi)} I(X;\ Y) \tag{20}$$

In general, $H(X)$, $H(Y)$, $H(X\,|\,Y)$, and $H(Y\,|\,X)$ all depend on the input distribution $p(x_i)$. Hence, *in the general case*, it is not a simple matter to maximize Eq. (19) with respect to $p(x_i)$.

All the measures of information that have been considered in this treatment have involved only probability distributions on X and Y. Thus, for the model of Fig. 1.1.1, the joint distribution $p(x_i, y_j)$ is sufficient. Suppose the source [and hence the input distribution $p(x_i)$] is known; then it follows from the usual conditional-probability relationship

$$p(x_i, y_j) = p(x_i)p(y_j\,|\,x_i) \tag{21}$$

that only the distribution $p(y_j\,|\,x_i)$ is needed for $p(x_i\,|\,y_j)$ to be determined. This conditional probability $p(y_j\,|\,x_i)$ can then be taken as a description of the information channel connecting the source X and the destination Y. Thus, a *discrete memoryless channel* can be defined as the probability distribution

$$p(y_j\,|\,x_i) \qquad x_i \in X \text{ and } y_j \in Y \tag{22}$$

or, equivalently, by the *channel matrix* D, where

$$D = [p(y_j\,|\,x_i)] = \begin{bmatrix} p(y_1|x_1) & p(y_2|x_2) & \dots & p(y_N|x_1) \\ p(y_1|x_2) & p(y_2|x_2) & \dots & p(y_N|x_2) \\ \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \\ p(y_1|x_M) & \dots & \dots & p(y_N|x_M) \end{bmatrix} \tag{23}$$

A number of special types of channels are readily distinguished. Some of the simplest and/or most interesting are listed as follows:

(***a***) *Lossless Channel*. Here $H(X\,|\,Y) = 0$ for all input distribution $p(x_i)$, and Eq. (20) becomes

$$C = \max_{p(xi)} H(X) = \log M \tag{24}$$

This maximum is obtained when the x_i are equally likely, so that $p(x_i) = 1/M$ for all i. The channel capacity is equal to the source entropy, and no source information is lost in transmission.

(***b***) *Deterministic Channel*. Here $H(Y\,|\,X) = 0$ for all input distributions $p(x_i)$, and Eq. (20) becomes

$$C = \max_{p(xi)} H(Y) = \log N \tag{25}$$

This maximum is obtained when the y_j are equally likely, so that $p(y_j) = 1/N$ for all j. Each member of the X set is uniquely associated with one, and only one, member of the destination alphabet Y.

(***c***) *Symmetric Channel*. Here the rows of the channel matrix D are identical except for permutations, *and* the columns are identical except for permutations. If D is square, rows and columns are identical except for permutations. In the symmetric channel, the conditional entropy $H(Y\,|\,X)$ is independent of the input distribution $p(x_i)$ and depends only on the channel matrix D. As a consequence, the determination of channel capacity is greatly simplified and can be written

$$C = \log N + \sum_{j=1}^{N} p(y_j\,|\,x_i) \log p(y_j\,|\,x_i) \tag{26}$$

This capacity is obtained when the y_i are equally likely, so that $p(y_j) = 1/N$ for all j.

(*d*) *Binary Symmetric Channel* (*BSC*). This is the special case of a symmetric channel where $M = N = 2$. Here the channel matrix can be written

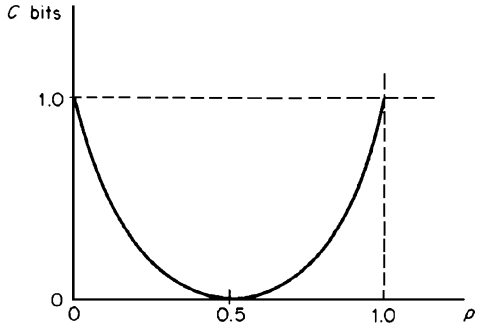

FIGURE 1.2.1 Capacity of the binary symmetric channel.

$$D = \begin{bmatrix} p & 1-p \\ 1-p & p \end{bmatrix} \tag{27}$$

and the channel capacity is

$$C = \log 2 - G(p) \tag{28}$$

where the function $G(p)$ is defined as

$$G(p) = -[p \log p + (1-p) \log (1-p)] \tag{29}$$

This expression is mathematically identical to the entropy of a binary source as given in Eq. (5) and is plotted in Fig. 1.1.3 using base 2 logarithms. For the same base, Eq. (28) is shown as a function of p in Fig. 1.2.1. As expected, the channel capacity is large if p, the probability of correct transmission, is either close to unity or to zero. If $p = 1/2$, there is no statistical evidence which symbol was sent and the channel capacity is zero.

DECISION SCHEMES

A decision scheme or decoding scheme B is a partitioning of the Y set into M disjoint and exhaustive sets $B_1, B_2, \ldots, B_M$ such that when a destination symbol y_k falls into set B_i, it is decided that symbol x_i was sent. Implicit in this definition is a *decision rule* $d(y_j)$, which is a function specifying uniquely a source symbol for each destination symbol. Let $p(e \mid y_j)$ be the probability of error when it is decided that y_j has been received. Then the *total error probability* $p(e)$ is

$$p(e) = \sum_{j=1}^{N} p(y_j) p(e \mid y_j) \tag{30}$$

For a given decision scheme β, the conditional error probability $p(e \mid y_j)$ can be written

$$p(e \mid y_j) = 1 - p[d(y_j) \mid y_j] \tag{31}$$

where $p[d(y_j) \mid y_j]$ is the conditional probability $p(x_i \mid y_j)$ with x_i assigned by the decision rule; i.e., for a given decision scheme $d(y_j) = x_i$. The probability $p(y_j)$ is determined only by the source a priori probability $p(x_i)$ and by the channel matrix $= D\ [p(y_j \mid x_i)]$. Hence, only the term $p(e \mid y_j)$ in Eq. (30) is a function of the decision scheme. Since Eq. (30) is a sum of nonnegative terms, the error probability is a minimum when each summand is a minimum. Thus, the term $p(e \mid y_j)$ should be a minimum for each y_j. It follows from Eq. (31) that the minimum-error scheme is that scheme which assigns a decision rule

$$d(y_j) = x^* \qquad j = 1, 2, \ldots, N \tag{32}$$

where x^* is defined by

$$p(x^* \mid y_j) \geq p(x_i \mid y_j) \qquad i = 1, 2, \ldots, M \tag{33}$$

In other words, each y_j is decoded as the *a posteriori most likely* x_i. This scheme, which minimizes the probability of error $p(e)$, is usually called the *ideal observer*.

The ideal observer is not always a completely satisfactory decision scheme. It suffers from two major disadvantages: (1) For a given channel D, the scheme is defined only for a given input distribution $p(x_i)$. It might be preferable to have a scheme that was insensitive to input distributions. (2) The scheme minimizes average error but does not bound certain errors. For example, some symbols may always be received incorrectly. Despite these disadvantages, the ideal observer is a straightforward scheme which does minimize average error. It is also widely used as a standard with which other decision schemes may be compared.

Consider the special case where the input distribution is $p(x_i) = 1/M$ for all i, so that all x_i are equally likely. Now the conditional likelihood $p(x_i \mid y_j)$ is

$$p(x_i \mid y_j) = \frac{p(x_i)p(y_j \mid x_i)}{p(y_j)} = \frac{p(y_j \mid x_i)}{Mp(y_j)} \tag{34}$$

For a given y_j, that input x_i is chosen which makes $p(y_j \mid x_i)$ a maximum, and the decision rule is

$$d(y_j) = x^\dagger \qquad j = 1, 2, \ldots, N \tag{35}$$

where $x^\dagger$ is defined by

$$p(y_j \mid x^\dagger) \geq p(y_j \mid x_i) \qquad i = 1, 2, \ldots, M \tag{36}$$

The probability of error becomes

$$p(e) = \sum_{j=1}^{N} p(y_j)\left[1 - \frac{p(y_j \mid x^\dagger)}{Mp(y_j)}\right] \tag{37}$$

This decoder is sometimes called the *maximum-likelihood* decoder or decision scheme.

It would appear that a relationship should exist between the error probability $p(e)$ and the channel capacity C. One such relationship is the *Fano bound*, given by

$$H(X \mid Y) \leq G[p(e)] + p(e) \log (M - 1) \tag{38}$$

and relating error probability to channel capacity through Eq. (20). Here $G(\cdot)$ is the function already defined by Eq. (29). The three terms in Eq. (38) can be interpreted as follows:

$H(X \mid Y)$ is the equivocation. It is the average additional information needed at the destination after reception to completely determine the symbol that was sent.

$G[p(e)]$ is the entropy of the binary system with probabilities $p(e)$ and $1 - p(e)$. In other words, it is the average amount of information needed to determine whether the decision rule resulted in an error.

$\log (M - 1)$ is the maximum amount of information needed to determine which among the remaining $M - 1$ symbols was sent if the decision rule was incorrect; this information is needed with probability $p(e)$.

THE NOISY-CODING THEOREM

The concept of channel capacity was discussed earlier. Capacity is a fundamental property of an information channel in the sense that it is possible to transmit information through the channel at any rate less than the channel capacity with arbitrarily small probability of error. This result is called the *noisy-coding theorem* or *Shannon's fundamental theorem for a noisy channel*.

The noisy-coding theorem can be stated more precisely as follows: Consider a discrete memoryless channel with nonzero capacity C; fix two numbers H and ϵ such that

$$0 < H < C \tag{39}$$

and

$$\epsilon > 0 \tag{40}$$

Let us transmit m messages $u_1, u_2, \ldots, u_m$ by code words each of length n binary digits. The positive integer n can be chosen so that

$$m \geq 2^{nH} \tag{41}$$

In addition, at the destination the m sent messages can be associated with a set $V = \{v_1, v_2, \ldots, v_m\}$ of received messages and with a decision rule $d(v_j) = u_j$ such that

$$p[d(v_j) \mid v_j] \geq 1 - \epsilon \tag{42}$$

i.e., decoding can be accomplished with a probability of error that does not exceed ϵ. There is a converse to the noise-coding theorem which states that it is not possible to produce an encoding procedure which allows transmission at a rate greater than channel capacity with arbitrarily small error.

ERROR-CORRECTING CODES

The codes considered earlier were designed for minimum length in the noiseless-transmission case. For noisy channels, the noisy-coding theorem guarantees the existence of a code which will allow transmission at any rate less than channel capacity and with arbitrarily small probability of error; however, the theorem does not provide a constructive procedure to devise such codes. Indeed, it implies that very long sequences of source symbols may have to be considered if reliable transmission at rates near channel capacity are to be obtained. In this section, we consider some of the elementary properties of simple *error-correcting codes*; i.e., codes which can be used to increase reliability in the transmission of information through noisy channels by correcting at least some of the errors that occur so that overall probability of error is reduced.

The discussion will be restricted to the BSC, and the noisy-coding theorem notation will be used. Thus, a source alphabet $X = \{x_1, x_2, \ldots, x_m\}$ of M symbols will be used to form a message set U of m messages u_k, where $U = \{u_1, u_2, \ldots, u_m\}$. Each u_k will consist of a sequence of the x_i's. Each message u_k will be encoded into a sequence of n binary digits for transmission over the BSC. At the destination, there exists a set $V = \{v_1, v_2, \ldots, v_{2n}\}$ of all possible binary sequences of length n. The inequality $m \leq 2^n$ must hold. The problem is to associate with each sent message u_k a received message v_j so that $p(e)$, the overall probability of error, is reduced.

In the discussion of the noisy-coding theorem, a decoding scheme was used that examined the received message v_j and identified it with the sent message u_k, which differed from it in the least number of binary digits. In all the discussions here it will be assumed that this decoder is used. Let us define the *Hamming distance* $d(v_j, v_k)$ between two binary sequences v_j and v_k of length n as the number of digits in which v_j and v_k disagree. Thus, if the distance between two sequences is zero, the two sequences are identical. It is easily seen that this distance measure has the following four elementary properties:

$$d(v_j, v_k) \geq 0 \text{ with equality if and only if } v_j = v_k \tag{43}$$

$$d(v_j, v_k) = d(v_k, v_j) \tag{44}$$

$$d(v_j, v_l) \leq d(v_j, v_k) + d(v_k, v_l) \tag{45}$$

$$d(v_j, v_k) \leq n \tag{46}$$

The decoder we use is a *minimum-distance* decoder. As mentioned earlier, the ideal-observer decoding scheme is a minimum-distance scheme for the BSC.

It is intuitively apparent that the sent messages should be represented by code words that all have the greatest possible distances between them. Let us investigate this matter in more detail by considering all binary sequences of length $n = 3$; there are $2^n = 2^3 = 8$ such sequences, viz.,

000	001	011	111
	010	110	
	100	101	

It is convenient to represent these as the eight corners of a unit cube, as shown in Fig. 1.2.2*a*, where the x axis corresponds to the first digit, the y axis to the second, and the z axis to the third. Although direct pictorial representation is not possible, it is clear that binary sequences of length n greater than 3 can be considered as the corners of the corresponding n-cube.

Suppose that all eight binary sequences are used as code words to encode a source. If any binary digit is changed in transmission, an error will result at the destination since the sent message will be interpreted incorrectly as one of the three possible messages that differ in one code digit from the sent message. This situation is illustrated in Fig. 1.2.2 for the code words 000 and 111. A change of one digit in each of these code words produces one of three possible other code words.

Figure 1.2.2 suggests that only two code words, say 000 and 111, should be used. The distance between these two words, or any other two words on opposite corners of the cube, is 3. If only one digit is changed in the transmission of each of these two code words, they can be correctly distinguished at the destination by a minimum-distance decoder. If two digits are changed in each word in transmission, it will not be possible to make this distinction.

This reasoning can be extended to sequences containing more than three binary digits. For any $n \geq 3$, single errors in each code word can be corrected. If double errors are to be corrected without fail, there must be at least two code words with a minimum distance between them of 5; thus, for this case, binary code words of length 5 or greater must be used.

Note that the error-correcting properties of a code depend on the distance $d(v_j, v_k)$ between the code words. Specifically, single errors can be corrected if all code words employed are at least a distance of 3 apart, double errors if the words are at a distance of 5 or more from each other, and, in general, q-fold errors can be corrected if

$$d(v_j, v_k) \geq 2q + 1 \qquad j \neq k \tag{47}$$

Errors involving less than q digits per code word can also be corrected if Eq. (63) is satisfied. If the distance between two code words is $2q$, there will always be a group of binary sequences which are in the middle, i.e., a distance q from *each* of the two words. Thus, by the proper choice of code words, q-fold errors can be *detected* but not corrected if

$$d(v_j, v_k) = 2q \qquad j \neq k \tag{48}$$

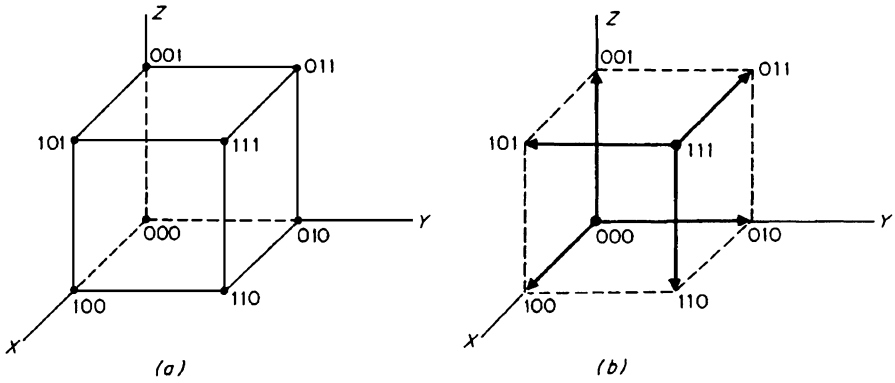

FIGURE 1.2.2 Representation of binary sequences as the corners of an n-cube, $n = 3$; (*a*) the eight binary sequences of length 3; (*b*) shift in sequences 000 and 111 from a single error.

TABLE 1.2.4 Parity-Check Code for Single-Error Detection

Message digits	Check digit	Word	Message digit	Check digit	Word
000	0	0000	110	0	1100
100	1	1001	101	0	1010
010	1	0101	011	0	0110
001	1	0011	111	1	1111

Now consider the maximum number of code words r that can be selected from the set of 2^n possible binary sequences of length n to form a code that will correct all single, double, . . . , q-fold errors. In the example of Fig. 1.2.2, the number of code words selected was 2. In fact, it can be shown that there is no single-error-correcting code for $n = 3, 4$ containing more than two words. Suppose we consider a given code consisting of the words . . . , u_k, u_j, ... All binary sequences of distance q or less from u_k must "belong" to u_k, and to u_k only, if q-fold errors are to be corrected. Thus, associated with u_k are all binary sequences of distance 0, 1, 2, . . . , q from u_k. The number of such sequences is given by

$$\binom{n}{0} + \binom{n}{1} + \binom{n}{2} + \cdots + \binom{n}{q} = \sum_{i=0}^{q} \binom{n}{i} \tag{49}$$

Since there are r of the code words, the total number of sequences associated with all the code words is

$$r \sum_{i=0}^{q} \binom{n}{i}$$

This number can be no larger than 2^n, the total number of distinct binary sequences of length n. Therefore the following inequality must hold:

$$r \sum_{i=0}^{q} \binom{n}{i} \leq 2^n \qquad \text{or} \qquad r \leq \frac{2^n}{\sum_{i=0}^{q} \binom{n}{i}} \tag{50}$$

This is a *necessary* upper bound on the number of code words that can be used to correct all errors up to and including q-fold errors. It can be shown that it is not *sufficient*.

Consider the eight possible distinct binary sequences of length 3. Suppose we add one binary digit to each sequence in such a way that the total number of 1s in the sequence is *even* (or odd, if you wish). The result is shown in Table 1.2.4. Note that all the word sequences of length 4 differ from each other by a distance of at least 2. In accordance with Eq. (48), it should be possible now to *detect single errors* in all eight sequences. The detection method is straightforward. At the receiver, count the number of 1s in the sequence; if the number is odd, a single error (or, more precisely, an odd number of errors) has occurred; if the number is even, no error (or an even number of errors) has occurred. This particular scheme is a good one if only single errors are likely to occur and if detection only (rather than correction) is desired. Such is often the case, for example, in closed digital systems such as computers. The added digit is called a *parity-check digit*, and the scheme is a very simple example of a *parity-check code*.

PARITY-CHECK CODES

More generally, in parity-check codes, the encoded sequence consists of n binary digits of which only $k < n$ are *information digits* while the remaining $l = n - k$ digits are used for error detection and correction and are called *check digits* or *parity checks*. The example of Table 1.2.4 is a single-error-detecting code, but, in general, q-fold

errors can be detected and/or corrected. As the number of errors to be detected and/or corrected increases, the number l of check digits must increase. Thus, for fixed word length n, the number of information digits $k = n - l$ will decrease as more and more errors are to be detected and/or corrected. Also the total number of words in the code cannot exceed the right side of Eq. (50) or the number 2^k.

Parity-check codes are relatively easy to implement. The simple example given of a single-error-detecting code requires only that the number of 1s in each code word be counted. In this light, it is of considerable importance to note that these codes satisfy the noisy-coding theorem. In other words, it is possible to encode a source by parity-check coding for transmission over a BSC at a rate approaching channel capacity and with arbitrarily small probability of error. Then, from Eq. (41), we have

$$2^{nH} = 2^k \tag{51}$$

or H, the rate of transmission, is given by

$$H = k/n \tag{52}$$

As $n \to \infty$, the probability of error $p(e)$ approaches zero. Thus, in a certain sense, it is sufficient to limit a study of error-correcting codes to the parity-check codes.

As an example of a parity-check code, consider the simplest nondegenerate case where l, the number of check digits, is 2 and k, the number of information digits, is 1. This system is capable of single-error detection and correction, as we have already decided from geometric considerations. Since $l + k = 3$, each encoded word will be three digits long. Let us denote this word by $a_1a_2a_3$, where each a_i is either 0 or 1. Let a_1 represent the information digit and a_2 and a_3 represent the check digits.

Checking for errors is done by forming two independent equations from the three a_i, each equation being of the form of a modulo-2 sum, i.e., of the form

$$a_i \oplus a_j = \begin{cases} 0 & a_i = a_j \\ 1 & a_i \neq a_j \end{cases}$$

Take the two independent equations to be

$$a_2 \oplus a_3 = 0 \quad \text{and} \quad a_1 \oplus a_3 = 0$$

for an *even-parity* check. For an odd-parity check, let the right sides of both of these equations be unity. If these two equations are to be satisfied, the only possible code words that can be sent are 000 and 111. The other six words of length 3 violate one or both of the equations.

Now suppose that 000 is sent and 100 is received. A solution of the two independent equations gives, for the received word,

$$a_2 \oplus a_3 = 0 \oplus 0 = 0$$
$$a_1 \oplus a_3 = 1 \oplus 0 = 1$$

The check yields the binary check number 1, indicating that the error is in the first digit a_1, as indeed it is. If 111 is sent and 101 received, then

$$a_2 \oplus a_3 = 0 \oplus 1 = 1$$
$$a_1 \oplus a_3 = 1 \oplus 1 = 0$$

and the binary check number is 10, or 2, indicating that the error is in a_2.

In the general case, a set of l independent linear equations is set up in order to derive a binary checking number whose value indicates the position of the error in the binary word. If more than one error is to be detected and corrected, the number l of check digits must increase, as discussed previously.

In the example just treated, the l check digits were used only to check the k information digits immediately preceding them. Such a code is called a *block code*, since all the information digits and all the check digits are contained in the block (code word) of length $n = k + l$. In some encoding procedures, the l check digits may also be used to check information digits appearing in preceding words. Such codes are called *convolutional* or

recurrent codes. A parity-check code (either block or convolutional) where the word length is n and the number of information digits is k is usually called an (n, k) *code*.

OTHER ERROR-DETECTING AND ERROR-CORRECTING CODES

Unfortunately, a general treatment of error-detecting and error-correcting codes requires that the code structure be cast in a relatively sophisticated mathematical form. The commonest procedure is to identify the code letters with the elements of a finite (algebraic) field. The code words are then taken to form a vector subspace of n-tuples over the field. Such codes are called *linear codes* or, sometimes, *group codes*. Both the block codes and the convolutional codes mentioned in the previous paragraph fall in this category.

An additional constraint often imposed on linear codes is that they be *cyclic*. Let a code word a be represented by

$$a = (a_0, a_1, a_2, \ldots, a_{n-1})$$

Then the ith *cyclic permutation* a^{-i} is given by $a^i = (a_i, a_{i+1}, \ldots, a_{n-1}, a_0, a_1, \ldots, a_{i-1})$. A linear code is cyclic if, and only if, for every word a in the code, there is also a word a^i in the code. The permutations need not be distinct and, in fact, generally will not be. The eight code words

$$0000 \quad 0110 \quad 1001 \quad 1010$$

$$0011 \quad 1100 \quad 0101 \quad 1111$$

constitute a cyclic set. Included in the cyclic codes are some of those most commonly encountered such as the Bose and Ray-Chaudhuri (BCH) codes and shortened Reed-Muller codes.

CONTINUOUS-AMPLITUDE CHANNELS

The preceding discussion has concerned discrete message distributions and channels. Further, it has been assumed, either implicitly or explicitly, that the time parameter is discrete, i.e., that a certain number of messages, symbols, code digits, and so forth, are transmitted per unit time. Thus, we have been concerned with *discrete-amplitude*, *discrete-time* channels and with messages which can be modeled as *discrete random processes* with *discrete parameter*. There are three other possibilities, depending on whether the process amplitude and the time parameter have discrete or continuous distributions.

We now consider the *continuous-amplitude*, *discrete-time* channel, where the input messages can be modeled as *continuous random processes* with *discrete parameter*. It will be shown later that continuous-time cases of engineering interest can be treated by techniques which amount to the replacement of the continuous parameter by a discrete parameter. The most straightforward method involves the application of the sampling theorem to band-limited processes. In this case the process is sampled at equispaced intervals of length $1/2W$, where W is the highest frequency of the process. Thus the continuous parameter t is replaced by the discrete parameter $t_k = k/2W$, $k = \cdots, -1, 0, 1, \ldots$.

Let us restrict our attention for the moment to continuous-amplitude, discrete-time situations. The discrete density $p(x_i)$, $i = 1, 2, \ldots, M$, of the source-message set is replaced by the continuous density $f_x(x)$, where, in general, $-\infty < x < \infty$, although the range of x may be restricted in particular cases. In the same way, other discrete densities are replaced by continuous densities. For example, the destination distribution $p(y_j)$, $j = 1, 2, \ldots, N$, becomes $f_y(y)$, and the joint distribution $p(x_i, y_j)$ will be called $f_2(x, y)$.

In analogy with the discrete-amplitude case [Eq. (4)], the entropy of a continuous distribution $f_x(x)$ can be defined as

$$H(X) = -\int_{-\infty}^{\infty} f_x(x) \log f_x(x)\, dx \tag{53}$$

This definition is not completely satisfactory because of some of the properties of this new $H(X)$. For example, it can be negative and it can depend on the coordinate system used to represent the message.

Joint and conditional entropies can also be defined in exact analogy to the discrete case discussed in Chap. 1.1. If the joint density $f_2(x, y)$ exists, then the joint entropy $H(X, Y)$ is given by

$$H(X, Y) = -\int_{-\infty}^{\infty}\int f_2(x, y)\log f_2(x, y)\, dx\, dy \tag{54}$$

and the conditional entropies $H(X|Y)$ and $H(Y|X)$ are

$$H(X|Y) = -\int_{-\infty}^{\infty}\int f_2(x,y)\log\frac{f_2(x,y)}{f_y(y)}\, dx\, dy \tag{55}$$

and

$$H(X|Y) = -\int_{-\infty}^{\infty}\int f_2(x,y)\log\frac{f_2(x,y)}{f_y(y)}\, dx\, dy \tag{56}$$

where

$$f_x(x) = \int_{-\infty}^{\infty} f_2(x,y)\, dy \qquad \text{and} \qquad f_y(y) = \int_{-\infty}^{\infty} f_2(x,y)\, dx$$

The average mutual information follows from Eq. (15) and is

$$I(X; Y) = -\int_{-\infty}^{\infty}\int f_2(x,y)\log\frac{f_x(x)f_y(y)}{f_2(x,y)}\, dx\, dy \tag{57}$$

Although the entropy of a continuous distribution can be negative, positive, or zero, the average mutual information $I(X; Y) \geq 0$ with equality when x and y are statistically independent, i.e., when $f_2(x, y) = f_x(x)f_y(y)$.

MAXIMIZATION OF ENTROPY OF CONTINUOUS DISTRIBUTIONS

The entropy of a discrete distribution is a maximum when the distribution is uniform, i.e., when all outcomes are equally likely. In the continuous case, the entropy depends on the coordinate system, and it is possible to maximize this entropy subject to various constraints on the associated density function.

The Maximization of H(X) for a Fixed Variance of x. Maximizing $H(X)$ subject to the constraint that

$$\int_{-\infty}^{\infty} x^2 f_x(x)\, dx = \sigma^2 \tag{58}$$

yields the gaussian density

$$f_x(x) = (1/\sqrt{2\pi\sigma})e^{-x2/2\sigma2} \qquad -\infty < x < \infty \tag{59}$$

Thus, for fixed variance, the normal distribution has the largest entropy. The entropy in this case is

$$H(X) = {}^1/_2 \ln 2\pi\sigma^2 + {}^1/_2 \ln e = {}^1/_2 \ln 2\pi e\sigma^2 \tag{60}$$

This last result will be of considerable use later. For convenience, the natural logarithm has been used, and the units of H are nats.

The Maximization of H(X) for a Limited Peak Value of x. In this case, the single constraint is

$$\int_{-M}^{M} f_x(x)\ dx = 1 \tag{61}$$

One obtains the uniform distribution

$$f_x(x) = \begin{cases} 1/2M & |x| \le M \\ 0 & |x| > M \end{cases}$$

and, the associated entropy is

$$H(X) = -\int_{-M}^{M} \frac{1}{2M} \log \frac{1}{2M}\, dx = \log 2M \tag{62}$$

The Maximization of H(X) for x Limited to Nonnegative Values and a Given Average Value. The constraints

$$\int_{0}^{\infty} f_x(x)\ dx = 1$$

and

$$\int_{0}^{\infty} x f_x(x)\, dx = \mu \tag{63}$$

lead to the *exponential distribution*

$$f_x(x) = \begin{cases} 0 & x < 0 \\ (1/\mu)e^{-(x/\mu)} & x \ge 0 \end{cases}$$

The entropy associated with this distribution is

$$H(X) = \ln \mu + 1 = \ln \mu e \tag{64}$$

GAUSSIAN SIGNALS AND CHANNELS

Let us assume that the source symbol x and the destination symbol y are jointly gaussian, i.e., that the joint density $f_2(x, y)$ is

$$f_2(x,y) = \frac{1}{2\pi\sigma_x\sigma_y\sqrt{1-\rho^2}} \exp\left\{-\frac{1}{2(1-\rho^2)}\left[\left(\frac{x}{\sigma_x}\right)^2 - 2\rho\frac{xy}{\sigma_x\sigma_y} + \left(\frac{y}{\sigma_y}\right)^2\right]\right\} \tag{65}$$

where σ_x^2 and σ_y^2 are the variances of x and y, respectively, and ρ is the correlation coefficient given by

$$\rho = \frac{E\{xy\}}{\sigma_x\sigma_y} \tag{66}$$

The univariate densities of x and y are given, of course, by

$$f_x(x) = \frac{1}{\sqrt{2\pi}\sigma_x} \exp\left[-\frac{1}{2}\left(\frac{x}{\sigma_x}\right)^2\right] \quad -\infty < x < \infty \tag{67}$$

and

$$f_y(y) = \frac{1}{\sqrt{2\pi}\sigma_y} \exp\left[-\frac{1}{2}\left(\frac{y}{\sigma_y}\right)^2\right] \quad -\infty < y < \infty \tag{68}$$

In this case we have

$$I(X; Y) = -\,{}^1/_2 \ln (1 - \rho^2) \tag{69}$$

Thus the average mutual information in two jointly gaussian random variables is a function only of the correlation coefficient ρ and varies from zero to infinity since $-1 \le \rho \le 1$.

The noise entropy $H(Y|X)$ can be written

$$H(Y|X) = H(Y) - I(X; Y) = {}^1/_2 \ln 2\pi e\sigma_y^2 (1 - \rho^2) \tag{70}$$

Suppose that x and y are jointly gaussian as a result of independent zero-mean gaussian noise n being added in the channel to the gaussian input x, so that

$$y = x + n \tag{71}$$

In this case the correlation coefficient ρ becomes

$$\rho = \frac{E\{x^2 + nx\}}{\sigma_x\sigma_y} = \frac{\sigma_x^2}{\sigma_x\sigma_y} = \frac{\sigma_x}{\sigma_y} \tag{72}$$

and the noise entropy is

$$H(Y|X) = {}^1/_2 \ln 2\pi e\sigma_n^2 \tag{73}$$

where σ_n^2 is the noise variance given by

$$\sigma_n^2 = E\{n^2\} = \sigma_y^2 - \sigma_x^2 \tag{74}$$

In this situation, Eq. (69) can be rewritten as

$$I(X; Y) = {}^1/_2 \ln (1 + \sigma_x^2 / \sigma_n^2) \tag{75}$$

It is conventional to define the signal power as $S_p = \sigma_x^2$ and the noise power as $N_p = \sigma_n^2$ and to rewrite this last expression as

$$I(X; Y) = {}^1/_2 \ln (1 + S_p / N_p) \tag{76}$$

where S_p/N_p is the signal-to-noise power ratio.

Channel capacity C for the continuous-amplitude, discrete-time channel is

$$C = \max_{f_X(x)} I(X;\ Y) = \max_{f_X(x)} [H(Y) - H(Y|X)] \tag{77}$$

Suppose the channel consists of an additive noise that is a sequence of independent gaussian random variables n each with zero mean and variance σ_n^2. In this case the conditional probability $f(y/x)$ at each time instant is normal with variance σ_n^2 and mean equal to the particular realization of X. The noise entropy $H(Y|X)$ is given by Eq. (73), and Eq. (77) becomes

$$C = \max_{f_X(x)} [H(Y)] - \tfrac{1}{2} \ln 2\pi e\sigma_n^2 \tag{78}$$

If the input power is fixed at σ_x^2 then the output power is fixed at $\sigma_y^2 = \sigma_x^2 + \sigma_n^2$ and $H(Y)$ is a maximum if $Y = X + N$ is a sequence of independent gaussian random variables. The value of $H(Y)$ is

$$H(Y) = \tfrac{1}{2} \ln 2\pi e\left(\sigma_x^2 + \sigma_n^2\right)$$

and the channel capacity becomes

$$C = \tfrac{1}{2} \ln\left(1 + \sigma_x^2/\sigma_n^2\right) = \tfrac{1}{2} \ln\left(1 + S_p/N_p\right) \tag{79}$$

where S_p/N_p is the signal-to-noise power ratio. Note that the input X is a sequence of independent gaussian random variables and this last equation is identical to Eq. (76). Thus, for additive independent gaussian noise and an input power limitation, the discrete-time continuous-amplitude channel has a capacity given by Eq. (79). This capacity is realized when the input is an independent sequence of independent, identically distributed gaussian random variables.

BAND-LIMITED TRANSMISSION AND THE SAMPLING THEOREM

In this section, messages will be considered which can be modeled as continuous random processes $x(t)$ with continuous parameter t. The channels which transmit these messages will be called *amplitude-continuous*, *time-continuous* channels. Specifically attention will be restricted to signals (random processes) $x(t)$, which are *strictly band-limited*.

Suppose a given arbitrary (deterministic) signal $f(t)$ is available for all time. Is it necessary to know the amplitude of the signal for every value of time in order to characterize it uniquely? In other words, can $f(t)$ be represented (and reconstructed) from some set of *sample values* or *samples* . . . , $f(t)$, $f(t_0)$, $f(t_1)$, . . . ? Surprisingly enough, it turns out that, under certain fairly reasonable conditions, a signal can be represented exactly by samples spaced relatively far apart. The reasonable conditions are that the signal be *strictly band-limited*.

A (real) signal $f(t)$ will be called *strictly band-limited* $(-2\pi W, 2\pi W)$ if its Fourier transform $F(\omega)$ has the property

$$F(\omega) = 0 \qquad |\omega| > 2\pi W \tag{80}$$

Such a signal can be represented in terms of its sample taken at the *Nyquist sampling times,* $t_k = \dfrac{k}{2W}$ $k = 0, \pm 1, \ldots$ via the *sampling representation*

$$f(t) = \sum_{k=-\infty}^{\infty} f\left(\frac{k}{2W}\right) \frac{\sin(2\pi Wt - k\pi)}{2\pi Wt - k\pi} \tag{81}$$

This expression is sometimes called the *Cardinal series* or *Shannon's sampling theorem*. It relates the discrete time domain $\{k/2W\}$ with sample values $f(k/2W)$ to the continuous time domain $\{t\}$ of the function $f(t)$.

The interpolation function

$$k(t) = (\sin 2\pi Wt) / 2\pi Wt \tag{82}$$

has a Fourier transform $K(\omega)$ given by

$$K(\omega) = \begin{cases} 1/4\pi W & |\omega| < 2\pi W \\ 0 & |\omega| > 2\pi W \end{cases} \tag{83}$$

Also the shifted functions $k(t - k/2W)$ has the Fourier transform

$$\mathfrak{F}\{k(t - k/2W)\} = K(\omega)e^{jwk/2W} \tag{84}$$

Therefore, each term on the right side of Eq. (81) is a time function which is strictly band-limited $(-2\pi W, 2\pi W)$. Note also that

$$k\left(t - \frac{k}{2W}\right) = \frac{\sin(2Wt - k\pi)}{2\pi Wt - k\pi} = \begin{cases} 1 & t = t_k = k\pi/2W \\ 0 & t = t_n, \quad n \neq k \end{cases} \tag{85}$$

Thus, this sampling function $k(t - k/2W)$ is zero at all Nyquist instants except t_k, where it equals unity.

Suppose that a function $h(t)$ is not strictly band-limited to at least $(-2\pi W, 2\pi W)$ rad/s and an attempt is made to reconstruct the function using Eq. (81) with sample values spaced $1/2W \cdot$ s apart. It is apparent that the reconstructed signal [which is strictly band-limited $(-2\pi W, 2\pi W)$, as already mentioned] will differ from the original. Moreover, a given set of sample values $\{f(k/2W)\}$ could have been obtained from a whole class of different signals. Thus, it should be emphasized that the reconstruction of Eq. (81) is unambiguous only for signals strictly band-limited to at least $(-2\pi W, 2\pi W)$ rad/s. The set of different possible signals with the same set of sample values $\{f(k/2W)\}$ is called the *aliases* of the band-limited signal $f(t)$.

Let us now consider a signal (random process) $X(t)$ with *autocorrelation function* given by

$$R_x(\tau) = E\{X(t)X(t + \tau)\} \tag{86}$$

and *power spectral density*

$$\varphi_x(\omega) = \int_{-\infty}^{\infty} R_x(r)e^{-jwr}\,dr \tag{87}$$

which is just the Fourier transform of $R_x(\tau)$. The process will be assumed to have zero mean and to be strictly *band-limited* $(-2\pi W, 2\pi W)$ in the sense that the power special density $\phi_x(\omega)$ vanishes outside this interval; i.e.,

$$\varphi_x(\omega) = 0 \qquad |\omega| > 2\pi W \tag{88}$$

It has been noted that a deterministic signal $f(t)$ band-limited $(-2\pi W, 2\pi W)$ admits the *sampling representation* of Eq. (81). It can also be shown that the random process $X(t)$ admits the same expansions; i.e.,

$$X(t) = \sum_{k=-\infty}^{\infty} X\left(\frac{k}{2W}\right)\frac{\sin(2\pi Wt - k\pi)}{2\pi Wt - k\pi} \tag{89}$$

The right side of this expression is a random variable for each value of t. The infinite sum means that

$$\lim_{N\to\infty} E\{|X(t)-X_N(t)|^2\}=0$$

where

$$X_N(t)=\sum_{k=-N}^{N} X\left(\frac{k}{2W}\right)\frac{\sin(2\pi Wt-k\pi)}{2\pi Wt-k\pi}$$

Thus, the process $X(t)$ with continuous time parameter t can be represented by the process $X(k/2W)$, $k=\ldots,-2,-1,0,1,2,\ldots$, with discrete time parameter $t_k=k/2W$. For band-limited signals or channels it is sufficient, therefore, to consider the discrete-time case and to relate the results to continuous time through Eq. (89).

Suppose the continuous-time process $X(t)$ has a spectrum $\varphi_x(\omega)$ which is *flat and band-limited* so that

$$\varphi_x(\omega)=\begin{cases} N_0 & |\omega| \le 2\pi W \\ 0 & |\omega| > 2\pi W \end{cases} \tag{90}$$

Then the autocorrelation function passes through zero at intervals of $1/2W$ so that

$$R_x(k/2W)=0 \qquad k=\ldots,-2,-1,1,2,\ldots \tag{91}$$

Thus, samples spaced $k/2W$ apart are *uncorrelated if the power spectral density is flat and band-limited* $(-2\pi W, 2\pi W)$. *If the process is gaussian, the samples are independent.* This implies that continuous-time band-limited $(-2\pi W, 2\pi W)$ gaussian channels, where the noise has a flat spectrum, have a capacity C given by Eq. (79) as

$$C=\tfrac{1}{2}\ln(1+S_p/N_p) \qquad \text{(nats/sample)} \tag{92}$$

Here N_p is the variance of the additive, flat, band-limited gaussian noise and S_p is $R_x(0)$, the fixed variance of the input signal. The units of Eq. (92) are on a per sample basis. Since there are $2W$ samples per unit time, the capacity C' per unit time can be written as

$$C'=W\ln(1+S_p/N_p) \qquad \text{(nats/s)} \tag{93}$$

The ideas developed thus far in this section have been somewhat abstract notions involving information sources and channels, channel capacity, and the various coding theorems. We now look more closely at conventional channels. Many aspects of these topics fall into the area often called *modulation theory*.