Q.2a. Define the time probability. Explain the various properties of probability density function. (8)

Answer:

2.1 Probabilities and ensembles

An ensemble X is a triple $(x, \mathcal{A}_X, \mathcal{P}_X)$, where the *outcome* x is the value of a random variable, which takes on one of a set of possible values, $\mathcal{A}_X = \{a_1, a_2, \ldots, a_i, \ldots, a_I\}$, having probabilities $\mathcal{P}_X = \{p_1, p_2, \ldots, p_I\}$, with $P(x = a_i) = p_i$, $p_i \geq 0$ and $\sum_{a_i \in \mathcal{A}_X} P(x = a_i) = 1$.

The name $\mathcal{A}$ is mnemonic for 'alphabet'. One example of an ensemble is a letter that is randomly selected from an English document. This ensemble is shown in figure 2.1. There are twenty-seven possible letters: a–z, and a space character '–'.

Abbreviations. Briefer notation will sometimes be used. For example, $P(x = a_i)$ may be written as $P(a_i)$ or $P(x)$.

Probability of a subset. If T is a subset of $\mathcal{A}_X$ then:

$$P(T) = P(x \in T) = \sum_{a_i \in T} P(x = a_i). \tag{2.1}$$

For example, if we define V to be vowels from figure 2.1, $V = \{\mathtt{a}, \mathtt{e}, \mathtt{i}, \mathtt{o}, \mathtt{u}\}$, then

$$P(V) = 0.06 + 0.09 + 0.06 + 0.07 + 0.03 = 0.31. \tag{2.2}$$

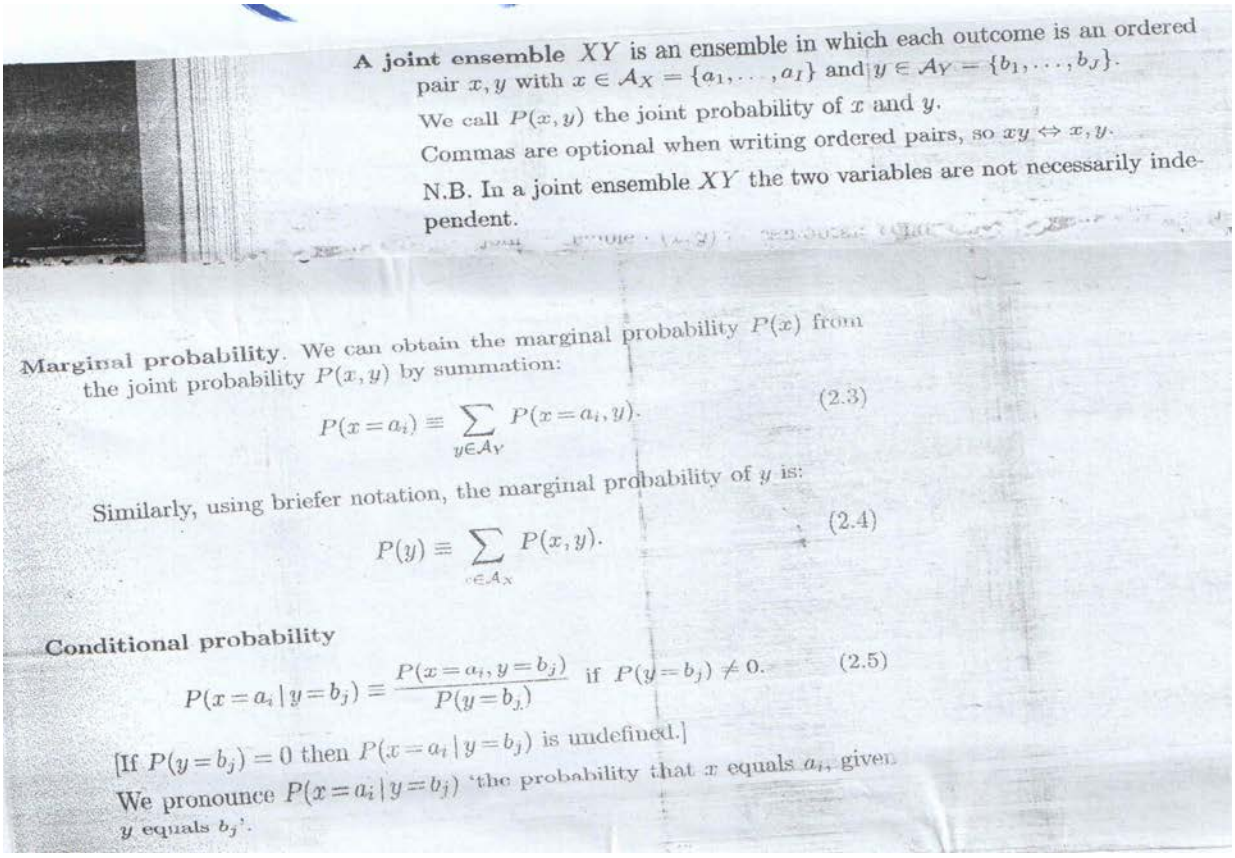
PROPERTIES OF PROBABILITY DENSITY FUNCTION

A continuous random variable X is characterized by a *probability density function* $f_X(x)$, which has the following properties:

1. $f_X(x) \geq 0, \quad -\infty < x < \infty$ (3.35)
2. $\int_{-\infty}^{\infty} f_X(x)\, dx = 1$ (3.36)
3. $P(X \leq a) = F_X(a) = \int_{-\infty}^{a} f_X(x)\, dx$ (3.37)

b. Explain joint probability and conditional probability. (8)

Answer:



A joint ensemble XY is an ensemble in which each outcome is an ordered pair x, y with $x \in \mathcal{A}_X = \{a_1, \ldots, a_I\}$ and $y \in \mathcal{A}_Y = \{b_1, \ldots, b_J\}$.

We call $P(x, y)$ the joint probability of x and y.

Commas are optional when writing ordered pairs, so $xy \Leftrightarrow x, y$.

N.B. In a joint ensemble XY the two variables are not necessarily independent.

Marginal probability. We can obtain the marginal probability $P(x)$ from the joint probability $P(x, y)$ by summation:

$$P(x = a_i) \equiv \sum_{y \in \mathcal{A}_Y} P(x = a_i, y). \qquad (2.3)$$

Similarly, using briefer notation, the marginal probability of y is:

$$P(y) \equiv \sum_{x \in \mathcal{A}_X} P(x, y). \qquad (2.4)$$

Conditional probability

$$P(x = a_i \mid y = b_j) \equiv \frac{P(x = a_i, y = b_j)}{P(y = b_j)} \quad \text{if } P(y = b_j) \neq 0. \qquad (2.5)$$

[If $P(y = b_j) = 0$ then $P(x = a_i \mid y = b_j)$ is undefined.]

We pronounce $P(x = a_i \mid y = b_j)$ 'the probability that x equals a_i, given y equals b_j'.

Q.3 a. Show that mean and variance of random variable X having uniform distribution

in the interval [a, b] are $\mu_x = \dfrac{a+b}{2}$ **and** $\sigma_x^2 = (a-b)^2/12$ **(8)**

Answer:

b. Explain the concept of statistical average in random signal theory. (8)

Answer:

Statistical Averages

Having discussed probability and some of its ramificatior
mining the *average* behavior of the outcomes arising in ra

The *expected value* or *mean* of a random variable X

$$\mu_X = E[X] = \int_{-\infty}^{\infty} x f_X(x)\, dx \tag{A1.27}$$

where E denotes the *statistical expectation operator*. That is, the mean μ_X locates the center of gravity of the area under the probability density curve of the random variable X. To interpret the expected value μ_X, we write the integral in the defining Equation (A1.27) as the limit of an approximating sum formulated as follows. Let $\{x_k | k = 0, \pm 1, \pm 2, \ldots\}$ denote a set of uniformly spaced points on the real line:

$$x_k = \left(k + \frac{1}{2}\right)\Delta, \qquad k = 0, \pm 1, \pm 2, \ldots \tag{A1.28}$$

where Δ is the spacing between adjacent points. We may then rewrite Equation (A1.27) as the limiting form of a sum:

$$\begin{aligned} E[X] &= \lim_{\Delta \to 0} \sum_{k=-\infty}^{\infty} \int_{k\Delta}^{(k+1)\Delta} x_k f_X(x)\, dx \\ &= \lim_{\Delta \to 0} \sum_{k=-\infty}^{\infty} x_k P\left(x_k - \frac{\Delta}{2} < X \le x_k + \frac{\Delta}{2}\right) \end{aligned} \tag{A1.29}$$

For a physical interpretation of the sum on the right-hand side of Equation (A1.29), suppose that we make n independent observations of the random variable X. Let $N_n(k)$ denote the number of times that the random variable X falls inside the kth bin:

$$x_k - \frac{\Delta}{2} < X \le x_k + \frac{\Delta}{2}, \qquad k = 0, \pm 1, \pm 2, \ldots$$

Then, as the number of observations, n, is made large, the ratio $N_n(k)/n$ approaches the probability $P(x_k - \Delta/2 < X \le x_k + \Delta/2)$. Accordingly, we may approximate the expected value of the random variable X as

$$\begin{aligned} E[X] &\simeq \sum_{k=-\infty}^{\infty} x_k \left(\frac{N_n(k)}{n}\right) \\ &= \frac{1}{n} \sum_{k=-\infty}^{\infty} x_k N_n(k), \qquad n \text{ large} \end{aligned} \tag{A1.30}$$

We now recognize the quantity on the right-hand side of Equation (A1.30) simply as the *sample average*. The sum is taken over all the values x_k, each of which is weighted by the number of times it occurs; the sum is then divided by the total number of observations to give the sample average. Indeed, Equation (A1.30) provides the basis for computing the expected value $E[X]$.

We next consider a more general situation. Let X denote a random variable, and let $g(X)$ denote a function of X defined on the real line. The quantity obtained by letting the argument of the function $g(X)$ be a random variable is also a random variable, which we denote as

$$Y = g(X) \tag{A1.31}$$

To find the expected value of the random variable Y, we could of course find the probability density function $f_Y(y)$ and then apply the standard formula

$$E[Y] = \int_{-\infty}^{\infty} y f_Y(y)\, dy$$

A simpler procedure, however, is to write

$$E[g(X)] = \int_{-\infty}^{\infty} g(x) f_X(x)\, dx \tag{A1.32}$$

Indeed, Equation (A1.32) may be viewed as generalizing the concept of expected value fo
(X) of a random variable X.

Q4 a. Define the concept of average information content of long independent sequences. (8)

Answer:

1. Average Information Content of Symbols in Long Independence Sequences

Suppose that a source is emitting one of M possible symbols s_0, s_1 ….. s_M in a statically independent sequence

Let p_1, p_2, …….. p_M be the problems of occurrence of the M-symbols resply. suppose further that during a long period of transmission a sequence of N symbols have been generated.

On an average – s_1 will occur NP_1 times

S_2 will occur NP_2 times

:

:

s_i will occur NP_i times

The information content of the i_{th} symbol is $I(s_i) = \log\left(\frac{1}{p_i}\right)$ bits

$\therefore$ P_iN occurrences of s_i contributes an information content of

$$P_iN \cdot I(s_i) = P_iN \cdot \log\left(\frac{1}{p_i}\right) \text{ bits}$$

$\therefore$ Total information content of the message is = Sum of the contribution due to each of M symbols of the source alphabet

i.e., $I_{total} = \sum_{i=1}^{M} NP_i \log\left(\frac{1}{p_i}\right)$ bits

$\therefore$ Average inforamtion content per symbol in given by $H = \frac{I_{total}}{N} = \sum_{i=1}^{M} NP_i \log\left(\frac{1}{p_i}\right)$ bits per symbol ---- IV

This is equation used by Shannon

Average information content per symbol is also called the source entropy.

b. Give the Mark off Model for information sources. (8)

Answer:

Markoff Model for Information Sources

Assumption

A source puts out symbols belonging to a finite alphabet according to certain probabilities depending on preceding symbols as well as the particular symbol in question.

- **Define a random process**

A statistical model of a system that produces a sequence of symbols stated above is and which is governed by a set of probs. is known as a random process.

Therefore, we may consider a discrete source as a random process

and

the converse is also true.

i.e. A random process that produces a discrete sequence of symbols chosen from a finite set may be considered as a discrete source.

- **Can you give an example of such a source?**
- **What is a discrete stationary Markoff process?**

Provides a statistical model for the symbol sequences emitted by a discrete source.

General description of the model can be given as below:

1. At the beginning of each symbol interval, the source will be in the one of 'n' possible states 1, 2, ….. n

   Where 'n' is defined as

   $$n \leq (M)^m$$

   M = no of symbol / letters in the alphabet of a discrete stationery source,

   m = source is emitting a symbol sequence with a residual influence lasting 'm' symbols.

   i.e. m: represents the order of the source.

   m = 2 means a 2^{nd} order source

   m = 1 means a first order source.

The source changes state once during each symbol interval from say i to j. The proby of this transition is P_{ij}. P_{ij} depends only on the initial state i and the final state j but does not depend on the states during any of the preceeding symbol intervals.

2. When the source changes state from i to j it emits a symbol.

   Symbol emitted depends on the initial state i and the transition i → j.

3. Let $s_1, s_2, \ldots s_M$ be the symbols of the alphabet, and let $x_1, x_2, x_3, \ldots x_k, \ldots$ be a sequence of random variables, where x_k represents the k^{th} symbol in a sequence emitted by the source.

   Then, the probability that the k^{th} symbol emitted is s_q will depend on the previous symbols $x_1, x_2, x_3, \ldots, x_{k-1}$ emitted by the source.

   i.e., $P(X_k = s_q / x_1, x_2, \ldots, x_{k-1})$

4. The residual influence of

   $x_1, x_2, \ldots, x_{k-1}$ on x_k is represented by the state of the system at the beginning of the k^{th} symbol interval.

   i.e. $P(x_k = s_q / x_1, x_2, \ldots, x_{k-1}) = P(x_k = s_q / S_k)$

When S_k in a discrete random variable representing the state of the system at the beginning of the k^{th} interval.

Term 'states' is used to remember past history or residual influence in the same context as the use of state variables in system theory / states in sequential logic circuits.

Q.5 a. Derive the mathematical formula of Self Information of the event $X=x_i$ of random variable X with possible outcomes x_i, i= 1, 2, 3, ..., n . Give the various properties of information. (8)

Answer:

Definition 1.1 Consider a discrete random variable X with the possible outcomes x_i, $i = 1, 2, ..., n$. The **Self-Information** of the event $X = x_i$ is defined as

$$I(x_i) = \log\left(\frac{1}{P(x_i)}\right) = -\log P(x_i). \quad (1.1)$$

We note that a high probability event conveys less information than a low probability event. For an event with $P(x_i) = 1$, $I(x_i) = 0$. Since a lower probability implies a higher degree of uncertainty (and *vice-versa*), a random variable with a higher degree of uncertainty contains more information. We will use this correlation between uncertainty and information for physical interpretation throughout this chapter.

The units of $I(x_i)$ are determined by the base of the logarithm, which is usually selected as 2 or e. When the base is 2, the units are in **bits** and when the base is e, the units are in nats (natural units). Since $0 \le P(x_i) \le 1$, $I(x_i) \ge 0$, *i.e.*, the self-information is *non-negative*. The following two examples illustrate why a logarithmic measure of information is appropriate.

This definition exhibits the following important properties that are intuitively satisfying:

1.
$$I(s_k) = 0 \quad \text{for } p_k = 1 \quad (9.5)$$

Obviously, if we are absolutely *certain* of the outcome of an event, even before it occurs, there is *no* information gained.

2.
$$I(s_k) \ge 0 \quad \text{for } 0 \le p_k \le 1 \quad (9.6)$$

That is to say, the occurrence of an event $S = s_k$ either provides some or no information, but never brings about a *loss* of information.

3.
$$I(s_k) > I(s_i) \quad \text{for } p_k < p_i \quad (9.7)$$

That is, the less probable an event is, the more information we gain when it occurs.

4. $I(s_k s_i) = I(s_k) + I(s_i)$ if s_k and s_i are statistically independent.

b. Define the term entropy. Compute the formula of entropy with its properties. (8)

Answer:

$$I(s_k) = \log_2\left(\frac{1}{p_k}\right) = -\log_2 p_k \quad \text{for } k = 0, 1, \ldots, K-1 \tag{9.8}$$

When $p_k = 1/2$, we have $I(s_k) = 1$ bit. Hence, *one bit is the amount of information that we gain when one of two possible and equally likely (i.e., equiprobable) events occurs.*

Note that the information $I(s_k)$ is positive, since the logarithm of a number less than one, such as a probability, is negative.

The amount of information $I(s_k)$ produced by the source during an arbitrary signaling interval depends on the symbol s_k emitted by the source at that time. Indeed, $I(s_k)$ is a discrete random variable that takes on the values $I(s_0), I(s_1), \ldots, I(s_{K-1})$ with probabilities $p_0, p_1, \ldots, p_{K-1}$ respectively. The mean of $I(s_k)$ over the source alphabet $\mathcal{S}$ is given by

$$H(\mathcal{S}) = E[I(s_k)] = \sum_{k=0}^{K-1} p_k I(s_k) = \sum_{k=0}^{K-1} p_k \log_2\left(\frac{1}{p_k}\right) \tag{9.9}$$

The important quantity $H(\mathcal{S})$ is called the *entropy*[3] of a discrete memoryless source with source alphabet $\mathcal{S}$. It is a measure of the *average information content per source symbol.* Note that the entropy $H(\mathcal{S})$ depends only on the probabilities of the symbols in the alphabet $\mathcal{S}$ of the source. Thus the symbol $\mathcal{S}$ in $H(\mathcal{S})$ is not an argument of a function but rather a label for a source.

(contd. 5b

CHAPTER 9 ■ FUNDAMENTAL LIMITS IN INFORMATION THEORY

■ SOME PROPERTIES OF ENTROPY

Consider a discrete memoryless source whose mathematical model is defined by Equations (9.1) and (9.2). The entropy $H(\mathcal{S})$ of such a source is bounded as follows:

$$0 \le H(\mathcal{S}) \le \log_2 K \tag{9.10}$$

where K is the *radix* (number of symbols) of the alphabet $\mathcal{S}$ of the source. Furthermore, we may make two statements:

1. $H(\mathcal{S}) = 0$, if and only if the probability $p_k = 1$ for some k, and the remaining probabilities in the set are all zero; this lower bound on entropy corresponds to *no uncertainty.*
2. $H(\mathcal{S}) = \log_2 K$, if and only if $p_k = 1/K$ for all k (i.e., all the symbols in the alphabet $\mathcal{S}$ are *equiprobable*); this upper bound on entropy corresponds to *maximum uncertainty.*

Q.6 a. Explain discrete memory less channel in detail. (8)

Answer:

9.5 Discrete Memoryless Channels

Up to this point in the chapter, we have been preoccupied with discrete memoryless sources responsible for information generation. We next consider the issue of information transmission, with particular emphasis on reliability. We start the discussion by considering a discrete memoryless channel, the counterpart of a discrete memoryless source.

A *discrete memoryless channel* is a statistical model with an input X and an output Y that is a *noisy* version of X; both X and Y are random variables. Every unit of time, the

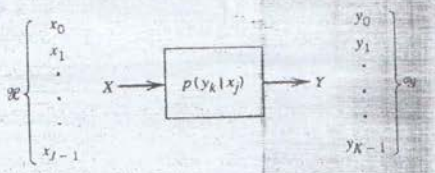


FIGURE 9.7 Discrete memoryless channel.

channel accepts an input symbol X selected from an alphabet $\mathcal{X}$ and, in response, it emits an output symbol Y from an alphabet $\mathcal{Y}$. The channel is said to be "discrete" when both of the alphabets $\mathcal{X}$ and $\mathcal{Y}$ have *finite* sizes. It is said to be "memoryless" when the current output symbol depends *only* on the current input symbol and *not* any of the previous ones.

Figure 9.7 depicts a view of a discrete memoryless channel. The channel is described in terms of an *input alphabet*

$$\mathcal{X} = \{x_0, x_1, \ldots, x_{J-1}\}, \tag{9.31}$$

an *output alphabet*,

$$\mathcal{Y} = \{y_0, y_1, \ldots, y_{K-1}\}, \tag{9.32}$$

and a set of *transition probabilities*

$$p(y_k|x_j) = P(Y = y_k|X = x_j) \quad \text{for all } j \text{ and } k \tag{9.33}$$

Naturally, we have

$$0 \leq p(y_k|x_j) \leq 1 \quad \text{for all } j \text{ and } k \tag{9.34}$$

Also, the input alphabet $\mathcal{X}$ and output alphabet $\mathcal{Y}$ need not have the same size. For example, in channel coding, the size K of the output alphabet $\mathcal{Y}$ may be larger than the size J of the input alphabet $\mathcal{X}$; thus, $K \geq J$. On the other hand, we may have a situation in which the channel emits the same symbol when either one of two input symbols is sent, in which case we have $K \leq J$.

A convenient way of describing a discrete memoryless channel is to arrange the various transition probabilities of the channel in the form of a matrix as follows:

$$\mathbf{P} = \begin{bmatrix} p(y_0|x_0) & p(y_1|x_0) & \cdots & p(y_{K-1}|x_0) \\ p(y_0|x_1) & p(y_1|x_1) & \cdots & p(y_{K-1}|x_1) \\ \vdots & \vdots & & \vdots \\ p(y_0|x_{J-1}) & p(y_1|x_{J-1}) & \cdots & p(y_{K-1}|x_{J-1}) \end{bmatrix} \tag{9.35}$$

The J-by-K matrix $\mathbf{P}$ is called the *channel matrix*, or *transition matrix*. Note that each *row* of the channel matrix $\mathbf{P}$ corresponds to a *fixed channel input*, whereas each column of the matrix corresponds to a *fixed channel output*. Note also that a fundamental property of the channel matrix $\mathbf{P}$, as defined here, is that the sum of the elements along any row of the matrix is always equal to one; that is,

Suppose now that the inputs to a discrete memoryless channel are selected according to the *probability distribution* $\{p(x_j), j = 0, 1, \ldots, J-1\}$. In other words, the event that the channel input $X = x_j$ occurs with probability

$$p(x_j) = P(X = x_j) \quad \text{for } j = 0, 1, \ldots, J-1 \tag{9.37}$$

Having specified the random variable X denoting the channel input, we may now specify the second random variable Y denoting the channel output. The *joint probability distribution* of the random variables X and Y is given by

$$\begin{aligned} p(x_j, y_k) &= P(X = x_j, Y = y_k) \\ &= P(Y = y_k | X = x_j)P(X = x_j) \\ &= p(y_k | x_j)p(x_j) \end{aligned} \tag{9.38}$$

The *marginal probability distribution* of the output random variable Y is obtained by averaging out the dependence of $p(x_j, y_k)$ on x_j, as shown by

$$\begin{aligned} p(y_k) &= P(Y = y_k) \\ &= \sum_{j=0}^{J-1} P(Y = y_k | X = x_j)P(X = x_j) \\ &= \sum_{j=0}^{J-1} p(y_k | x_j)p(x_j) \qquad \text{for } k = 0, 1, \ldots, K-1 \end{aligned}$$

The probabilities $p(x_j)$ for $j = 0, 1, \ldots, J-1$, are known as the *a priori probabilities* of the various input symbols. Equation (9.39) states that if we are given the input *a priori* probabilities $p(x_j)$ and the channel matrix [i.e., the matrix of transition probabilities $p(y_k|x_j)$], then we may calculate the probabilities of the various output symbols, the $p(y_k)$.

b. Show that $H(X,Y) = H(X) + H(Y|X)$
$= H(Y) + H(X|Y)$ **(8)**

Answer:

$$\begin{aligned} H(X, Y) &= H(X|Y) + H(Y) \\ &= H(Y|X) + H(X) \end{aligned} \tag{4.29}$$

where

$$H(Y|X) = -\sum_{i=1}^{M}\sum_{j=1}^{M} P(X = i, Y = j) \log_2(P(Y = j|X = i))$$

For a BSC, $P(X = i|Y = i)$ $(i = 0, 1)$ measures the uncertainty about the transmitted bit based on the received bit. The uncertainty is minimum when $P(X = i|Y = i) = 1$ for $i = 0, 1$, that is, an errorless channel. The uncertainty is maximum when $P(X = i|Y = i) = \frac{1}{2}$ for $i = 0, 1$. If we define the uncertainty as $-\log_2[P(X = i|Y = i)]$, then we have one bit of uncertainty when the output is independent of the input. When we have one bit of uncertainty associated with each received bit, the received value of the bit does not convey any information!

The conditional entropy $H(X|Y)$ is an *average measure of uncertainty*

about X when we know Y. In one extreme we can have Y and X related in a one-to-one manner such as $Y = X$. For this case, there is no uncertainty about X when we know Y; $P(X = i|Y = j) = \delta_{ij}$, where δ_{ij} is the delta function that is 0 for $i \neq j$ and 1 for $i = j$. We can easily verify that $H(X|Y) = 0$ when $Y = X$. In the context of a communication channel $Y = X$ represents an errorless channel, and there is no uncertainty about the input when the output is known. Alternatively, we can say that no information is lost in the channel since the output is uniquely related to the input. As another example, let us consider a communication channel that is so noisy that the output is statistically independent of the input. In this case we can easily verify that $H(X, Y) = H(X) + H(Y)$, and $H(X|Y) = H(X)$, that is, Y does not contain any information about X.

Q.7 a. Elaborate the channel capacity theorem for discrete memory less channel in detail. (8)

Answer:

A channel for which the noise and the received signal are as described in Equations (9.84) and (9.85) is called a *discrete-time, memoryless Gaussian channel*. It is modeled as in Figure 9.13. To make meaningful statements about the channel, however, we have to assign a *cost* to each channel input. Typically, the transmitter is *power limited*; it is therefore reasonable to define the cost as

$$E[X_k^2] = P, \quad k = 1, 2, \ldots, K \tag{9.86}$$

where P is the *average transmitted power*. The *power-limited Gaussian channel* described herein is of not only theoretical but also practical importance in that it models many communication channels, including line-of-sight radio and satellite links.

The *information capacity* of the channel is defined as the maximum of the mutual information between the channel input X_k and the channel output Y_k over all distributions on the input X_k that satisfy the power constraint of Equation (9.86). Let $I(X_k; Y_k)$ denote

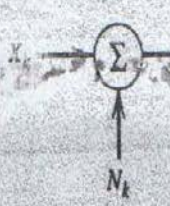


FIGURE 9.13 Model of discrete-time, memoryless Gaussian channel.

the mutual information between X_k and Y_k. We may then define the information capacity of the channel as

$$C = \max_{f_{X_k}(x)} \{I(X_k; Y_k) : E[X_k^2] = P\} \tag{9.87}$$

where the maximization is performed with respect to $f_{X_k}(x)$, the probability density function of X_k.

The mutual information $I(X_k; Y_k)$ can be expressed in one of the two equivalent forms shown in Equation (9.81). For the purpose at hand, we use the second line of this equation and so write

$$I(X_k; Y_k) = h(Y_k) - h(Y_k | X_k) \tag{9.88}$$

Since X_k and N_k are independent random variables, and their sum equals Y_k, as in Equation (9.84), we find that the conditional differential entropy of Y_k, given X_k, is equal to the differential entropy of N_k (see Problem 9.28):

$$h(Y_k | X_k) = h(N_k) \tag{9.89}$$

Hence, we may rewrite Equation (9.88) as

$$I(X_k; Y_k) = h(Y_k) - h(N_k) \tag{9.90}$$

Since $h(N_k)$ is independent of the distribution of X_k, maximizing $I(X_k; Y_k)$ in accordance with Equation (9.87) requires maximizing $h(Y_k)$, the differential entropy of sample Y_k of the received signal. For $h(Y_k)$ to be maximum, Y_k has to be a Gaussian random variable (see Example 9.8). That is, the samples of the received signal represent a noiselike process. Next, we observe that since N_k is Gaussian by assumption, the sample X_k of the transmitted signal must be Gaussian too. We may therefore state that the maximization specified in Equation (9.87) is attained by choosing the samples of the transmitted signal from a noiselike process of average power P. Correspondingly, we may reformulate Equation (9.87) as

$$C = I(X_k; Y_k) : X_k \text{ Gaussian}, \quad E[X_k^2] = P \tag{9.91}$$

where the mutual information $I(X_k; Y_k)$ is defined in accordance with Equation (9.90).

For the evaluation of the information capacity C, we proceed in three stages:

1. The variance of sample Y_k of the received signal equals $P + \sigma^2$. Hence, the use of Equation (9.76) yields the differential entropy of Y_k as

$$h(Y_k) = \tfrac{1}{2} \log_2[2\pi e(P + \sigma^2)] \tag{9.92}$$

2. The variance of the noise sample N_k equals σ^2. Hence, the use of Equation (9.76) yields the differential entropy of N_k as

$$h(N_k) = \tfrac{1}{2} \log_2(2\pi e\sigma^2) \tag{9.93}$$

3. Substituting Equations (9.92) and (9.93) into Equation (9.90) and recognizing the definition of information capacity given in Equation (9.91), we get the desired result:

$$C = \frac{1}{2} \log_2\left(1 + \frac{P}{\sigma^2}\right) \text{ bits per transmission} \tag{9.94}$$

With the channel used K times for the transmission of K samples of the process $X(t)$ in T seconds, we find that the information capacity per unit time is (K/T) times the result

given in Equation (9.94). The number K equals $2BT$, as in Equation (9.83). Accordingly, we may express the information capacity in the equivalent form:

$$C = B \log_2\left(1 + \frac{P}{N_0 B}\right) \text{ bits per second} \qquad (9.95)$$

where we have used Equation (9.85) for the noise variance σ^2.

Based on the formula of Equation (9.95), we may now state Shannon's third (and most famous) theorem, the *information capacity theorem*,[10] as follows:

The information capacity of a continuous channel of bandwidth B hertz, perturbed by additive white Gaussian noise of power spectral density $N_0/2$ and limited in bandwidth to B, is given by

$$C = B \log_2\left(1 + \frac{P}{N_0 B}\right) \text{ bits per second}$$

where P is the average transmitted power.

The information capacity theorem is one of the most remarkable results of information theory for, in a single formula, it highlights most vividly the interplay among three key system parameters: channel bandwidth, average transmitted power (or, equivalently, average received signal power), and noise power spectral density at the channel output. The dependence of information capacity C on channel bandwidth B is *linear*, whereas its dependence on signal-to-noise ratio P/N_0B is *logarithmic*. Accordingly, *it is easier to increase the information capacity of a communication channel by expanding its bandwidth than increasing the transmitted power for a prescribed noise variance.*

The theorem implies that, for given average transmitted power P and channel bandwidth B, we can transmit information at the rate of C bits per second, as defined in Equation (9.95), with arbitrarily small probability of error by employing sufficiently complex encoding systems. It is not possible to transmit at a rate higher than C bits per second by any encoding system without a definite probability of error. Hence, the channel capacity theorem defines the *fundamental limit* on the rate of error-free transmission for a power-limited, band-limited Gaussian channel. To approach this limit, however, the transmitted signal must have statistical properties approximating those of white Gaussian noise.

b. Give the concept of Differential entropy and mutual information for continuous ensembles. (8)

Answer:

Definition 1 (Differential entropy) *The differential entropy $h(X)$ of a continuous random variable X with pdf $f(X)$ is defined as*

$$h(X) = -\int_{\mathcal{S}} f(x) \log f(x) dx,$$

where $\mathcal{S}$ is the support region of the random variable.

Example

$$X \sim U(0, a), \quad h(X) = -\int_0^a \frac{1}{a} \log \frac{1}{a} dx = \log a.$$

Example. If $X \sim N(0, \sigma^2)$ with pdf $\phi(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2}{2\sigma^2}}$, then

$$\begin{aligned} h_a(x) &= -\int \phi(x) \log_a \phi(x) dx \\ &= -\int \phi(x) \left(\log_a \frac{1}{\sqrt{2\pi\sigma^2}} - \frac{x^2}{2\sigma^2} \log_a e \right) dx \\ &= \frac{1}{2} \log_a (2\pi\sigma^2) + \frac{\log_a e}{2\sigma^2} E_\phi[X^2] = \frac{1}{2} \log_a (2\pi e \sigma^2) \quad \square \end{aligned}$$

Definition 7 (Mutual information) *The mutual information $I(X;Y)$ between two random variables with joint density $f(x,y)$ is defined as*

$$I(X;Y) = \int f(x,y) \log \frac{f(x,y)}{f(x)f(y)} dxdy$$

Example

Let $(X, Y) \sim N(0, \mathbf{K})$ where

$$\mathbf{K} = \begin{bmatrix} \sigma^2 & \rho\sigma^2 \\ \rho\sigma^2 & \sigma^2 \end{bmatrix}.$$

Then $h(X) = h(Y) = \frac{1}{2} \log(2\pi e)\sigma^2$ and

$$h(X,Y) = \frac{1}{2} \log(2\pi e)^2 |\mathbf{K}| = \frac{1}{2} \log(2\pi e)^2 \sigma^4 (1-\rho^2).$$

Therefore,

$$I(X;Y) = h(X) + h(Y) - h(X;Y) = -\frac{1}{2} \log(1-\rho^2).$$

Q.8 a. If g(x) is a polynomial of degree (x-k) and is a factor of x^n+1 then g(x) generates an (x, k) cyclic code in which the code polynomial V(x) for a data vector $D=(d_0,d_1,.....d_{x-1})$ is generated by $V(x)=D(x)g(x)$ (8)

Answer:

This theorem can be proved as follows. Consider k polynomials $g(x)$, $xg(x)$, $x^2g(x), \ldots, x^{k-1}g(x)$, which all have degree $n-1$ or less. Now, any linear combination of these polynomials of the form

$$\begin{aligned} V(x) &= d_0g(x) + d_1xg(x) + \cdots + d_{k-1}x^{k-1}g(x) \\ &= D(x)g(x) \end{aligned}$$

is a polynomial of degree $n-1$ or less and is a multiple of $g(x)$. There are a total of 2^k such polynomials corresponding to the 2^k data vectors and the code vectors corresponding to the 2^k polynomials form a linear (n, k) code. To prove that this code is cyclic, let $V(x) = v_0 + v_1x + \cdots + v_{n-1}x^{n-1}$ be a code polynomial in this code. Consider

$$\begin{aligned} xV(x) &= v_0x + v_1x^2 + \cdots + v_{n-1}x^n \\ &= v_{n-1}(x^n + 1) + (v_{n-1} + v_0x + \cdots + v_{n-2}x^{n-1}) \\ &= v_{n-1}(x^n + 1) + V^{(1)}(x) \end{aligned}$$

where $V^{(1)}(x)$ is a cyclic shift of $V(x)$. Since $xV(x)$ and x^n+1 are both divisible by $g(x)$, $V^{(1)}(x)$ must be divisible by $g(x)$. Thus $V^{(1)}(x)$ is a multiple of $g(x)$ and can be expressed as a linear combination of $g(x)$, $xg(x), \ldots, x^{k-1}g(x)$. This says that $V^{(1)}(x)$ is also a code polynomial. Hence from the definition of cyclic codes, it follows that the linear code generated by $g(x), xg(x), \ldots, x^{k-1}g(x)$ is an (n, k) cyclic code.

The polynomial $g(x)$ is called the *generator polynomial* of the cyclic code. Given the generator polynomial $g(x)$ of a cyclic code, the code can be put into a systematic form as

$$V = \underbrace{(r_0, r_1, r_2 \ldots, r_{n-k-1}}_{\substack{n-k \text{ parity} \\ \text{check bits}}} \underbrace{d_0, d_1, \ldots, d_{k-1})}_{k \text{ message bits}} \qquad (9.20)$$

where

$$r(x) = r_0 + r_1x + r_2x^2 + \cdots + r_{n-k-1}x^{n-k-1}$$

is the *parity check polynomial* for the message polynomial $D(x)$. The parity check polynomial $r(x)$ is the remainder from dividing $x^{n-k}D(x)$ by $g(x)$:

$$x^{n-k}D(x) = q(x)g(x) + r(x) \qquad (9.21)$$

where $q(x)$ and $r(x)$ are the quotient and remainder, respectively. The code polynomial $V(x)$ is given by

$$V(x) = r(x) + x^{n-k}D(x) \qquad (9.22)$$

b. Design a linear block code with a minimum distance of three and a message block size of 8 bits R. (8)

Answer:

Q.9 a. Explain the special classes of cyclic codes:

(i) BCH

(ii) Burst & random error correcting codes (8)

Answer:

(i) BCH

Optimum design of error-correcting codes consists of designing a code with the smallest block size (n) for a given size of the message block (k) and for a desirable value of the minimum distance (d_{min}) for the code. Or, for a given code length (n) and efficiency (k/n), we might want to design codes with the largest possible values for d_{min}. That is, we might want to design codes with the best error-correcting capabilities. BCH codes, as a class, are the most extensive and powerful error-correcting cyclic codes known.

Decoding algorithms for BCH codes can be implemented with a reasonable amount of equipment. A detailed mathematical description of BCH codes would require extensive use of modern algebra. To discuss modern algebra is beyond the scope of this book and hence we will not include a mathematical description of BCH codes in this book. However, we will state the following properties of the BCH code that illustrate the power of this code:

For any positive integer m and t ($t < 2^{m-1}$) there exists a BCH code with the following parameters:

$$\text{Block length: } n = 2^{m-1}$$
$$\text{Number of parity check bits: } n - k \leq mt$$
$$\text{Minimum distance: } d_{\min} \geq 2t + 1 \qquad (9.26)$$

Several iterative procedures for decoding BCH codes are available. Many of these procedures can be programmed on a general purpose digital computer. In many practical applications, digital computers form an integral part of data communication networks. In such systems software implementation of decoding algorithms has several advantages over hardware implementation. The reader may find descriptions of BCH codes in the references listed at the end of this chapter.

Majority Logic Decodable Codes. These codes form a smaller sub-class of cyclic codes than do the BCH codes. Also, they are slightly inferior to BCH codes in terms of error-correcting capabilities for most interesting values of code length and efficiency. The main advantage of majority logic decodable codes is that the decoding operation can be implemented using simple circuits. The decoder for these codes has the form shown in Figure 9.6 with the combinatorial portion consisting of modulo-2 adders and a few layers of majority gates. An example is shown in Figure 9.7 for a (7, 4) majority logic decodable code. This code has $d_{\min} = 3$ and the reader can verify (using the decoding steps given below Figure 9.6) that the decoder can correct single errors.

Several classes of cyclic codes have been found recently that could be decoded using layers of majority gates. The construction procedures and the derivation of decoding rules are based on properties of finite geometries. Interested readers may refer to the books listed at the end of this chapter.

Shortened Cyclic Codes. The cyclic codes we have considered so far have generator polynomials that are divisors of $x^n + 1$. In general, the polynomial $x^n + 1$ has relatively few divisors and as a result there are usually very few cyclic codes of a given length. To circumvent this difficulty and to increase the number of pairs (n, k) for which useful codes can be constructed, cyclic codes are often used in shortened form. In the shortened form the last i information digits are always taken to be zeros (i.e., the last i bits of the codeword are padded with zeros). These bits are not transmitted; the decoder for the original cyclic code can decode the shortened code words simply by padding the received $(n - i)$-tuple with i zeros.

Hence, given an (n, k) cyclic code, it is always possible to construct an $(n - i, k - i)$ shortened cyclic code. The shortened cyclic code is a subset of

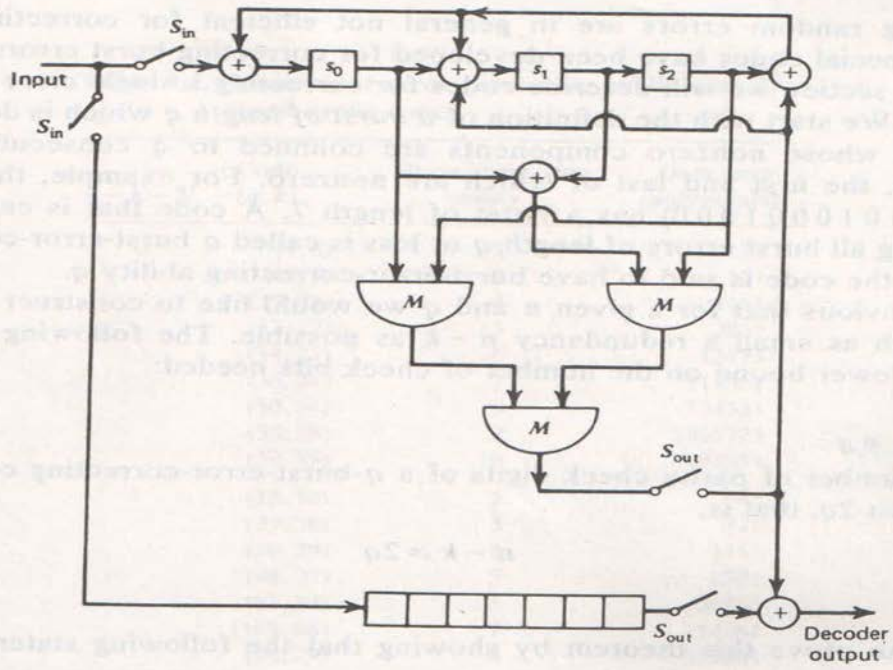


Figure 9.7 Decoder for a (7, 4) majority logic code. ($g(x) = 1 + x + x^3$). M–majority gate.

the cyclic code from which it was derived and hence its minimum distance and error correcting ability is at least as great as that of the original code. The encoding operation, syndrome calculation, and error correction procedures for shortened cyclic codes are identical to the procedures described for cyclic codes. As a result, shortened cyclic codes inherit nearly all of the implementation advantages and much of the mathematical structure of cyclic codes.

(ii)

9.5 BURST- AND RANDOM-ERROR-CORRECTING CODES

In the preceding sections we dealt with the design of codes to correct specific well-defined classes of error patterns. We treated the problems of correcting random errors, and burst errors, separately. Unfortunately, in most practical systems, errors occur neither independently, at random, nor in well-defined bursts. Consequently, random-error-correcting codes or single-burst-error-correcting codes will be either inefficient or inadequate for combating a mixture of random and burst errors. For channels in which both types of errors occur, it is better to design codes capable of correcting random errors and/or single or multiple bursts.

Several methods of constructing codes for the correction of random and burst errors have been proposed. The most effective method uses the interlacing technique. Given an (n, k) cyclic code, it is possible to construct a $(\lambda n, \lambda k)$ cyclic *interlaced code* by simply arranging λ code vectors of the original code into λ rows of a rectangular array and transmitting them column by column. The resulting code is called an interlaced code with an interlacing degree λ.

In an interlaced code, a burst of length λ or less will affect no more than one digit in each row since the transmission is done on a column by column fashion. If the original code (whose code words are rows of the two-dimensional array) can correct single errors, then the interlaced code can correct single bursts of length λ or less. If the original code can correct, say, t errors ($t > 1$), then the interlaced code can correct any combination of t bursts of length λ or less. The performance of the $(\lambda n, \lambda k)$ interleaved cyclic code against purely random errors is identical to that of the (n, k) cyclic code from which it was generated. The following example illustrates the concepts of interleaving.

Consider a (15, 7) BCH code generated by $g(x) = x^8 + x^4 + x^2 + x + 1$. This code has a minimum distance 5; and hence it is double-error-correcting. We can construct a (75, 35) interleaved code with $\lambda = 5$ with a burst-error-correcting ability of 10. The arrangement of codewords in an interleaved fashion is shown in Table 9.6. A 35-bit message block is divided into five 7-bit message blocks, and five codewords of length 15 bits are generated using $g(x)$ given above. These codewords are arranged as five rows of a 5×15 matrix. The columns of the matrix are transmitted in the order indicated in Table 9.6 as a 75-bit-long code vector.

To illustrate the burst- and random-error-correcting capabilities of this code, assume that errors have occurred in bit positions 5, 37 through 43, and 69. The decoder operates on the rows of the Table 9.6. Each row has a maximum of two errors, and the (15,7) BCH code from which the rows of the table are obtained is capable of correcting up to two errors per row. Hence, the error pattern shown in Table 9.6 can be corrected. The reader may think of the isolated errors in bits 5 and 69 as random errors, and the cluster of errors in bit positions 37 to 43 as a burst error.

Table 9.6

Each row is a 15 bit code word.

Five code words

1	6		36	41			66	71
2	7		37	42			67	72
3	8	...	38	43	.	...	68	73
4	9		39	44			69	74
5	10		40	45			70	75

While operating on the rows of the code array may be an obvious way to encode and decode an interlaced code, this is generally not the simplest implementation. The simplest implementation results from the property that if the original code is cyclic, then the interlaced code is also cyclic. Further, the generator polynomial for the interlaced code is $g(x^{\lambda})$, where $g(x)$ is the generator polynomial for the original code. Thus, encoding and decoding can be accomplished using shift registers. The decoder for the interlaced code can be derived from the decoder for the original code by replacing each shift register stage of the original decoder by λ stages without changing other connections. This allows the decoder to look at successive rows of the code array on successive decoder cycles. Also, if the decoder for the original code was simple, then the decoder for the interlaced code will also be simple. Thus, the interlacing technique is an effective tool for deriving long powerful codes from short optimal codes.

b. Decode the given sequence 1101 011001 of a convolutional code with a code rate r = ½ and constraint length k = 3 using Viterbi decoding algorithm.(8)

Answer: