

## Source Coding

### 4.1 Sampling Theorem

Sampling of the signals is the fundamental operation in digital communication. A continuous time signal is first converted to discrete time signal by sampling process. Also it should be possible to recover or reconstruct the signal completely from its samples.

The sampling theorem state that:

- i- A band limited signal of finite energy, which has no frequency components higher than  $W$  Hz, is completely described by specifying the values of the signal at instant of time separated by  $1/2W$  second and
- ii- A band limited signal of finite energy, which has no frequency components higher than  $W$  Hz, may be completely recovered from the knowledge of its samples taken at the rate of  $2W$  samples per second.

To proof of sampling theorem: Let  $x(t)$  the continuous time signal shown in figure(4.1) below, its band width does not contain any frequency components higher than  $W$  Hz. A sampling function samples this signal regularly at the rate of  $f_s$  sample per second.

Assume an analog waveform,  $x(t)$  with a Fourier transform,  $X(f)$ , which is zero outside the interval  $(-f_m < f < f_m)$ . The sampling of  $x(t)$  can viewed as the product of  $x(t)$  with periodic train of unit impulse function  $x_\delta(t)$  defined as

$$x_\delta(t) = \sum_{n=-\infty}^{\infty} \delta(t - nT_s)$$

The sifting property of unit impulse state that

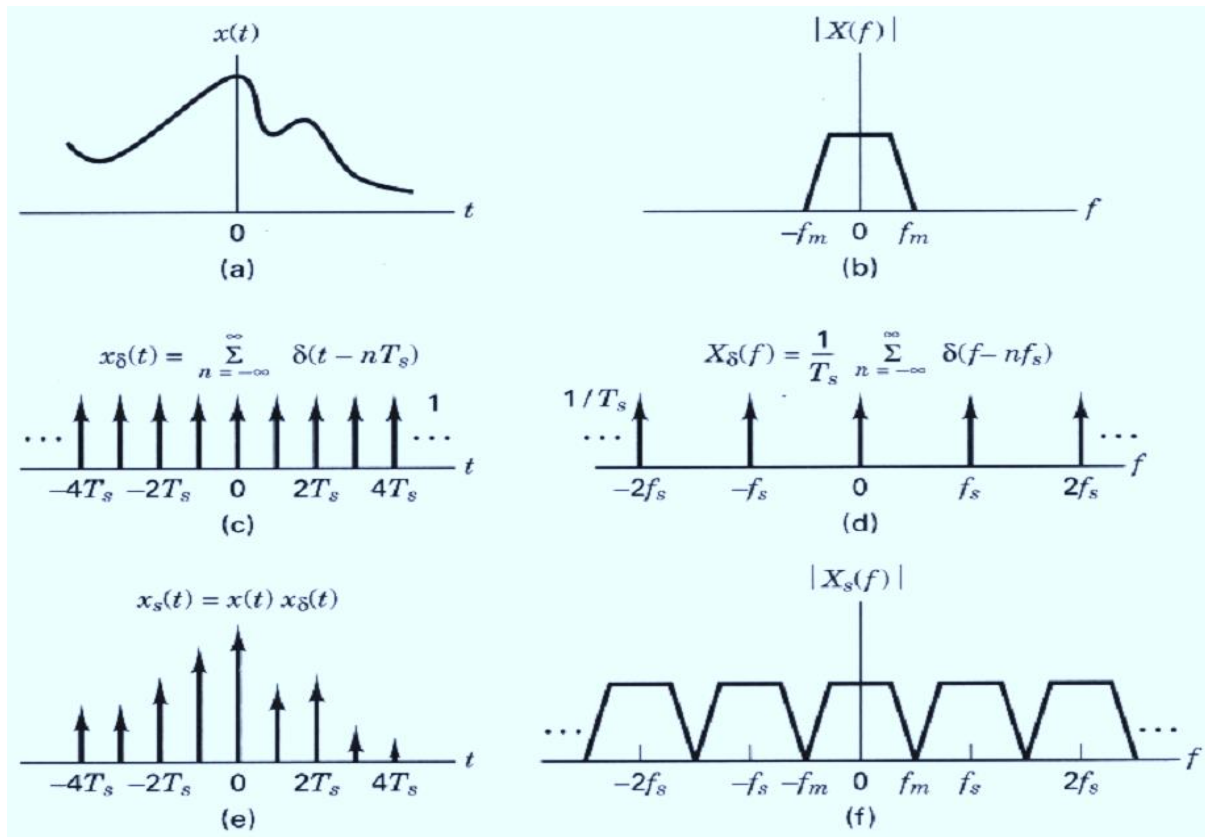
$$x(t)\delta(t - t_0) = x(t_0)\delta(t - t_0)$$

Using this property so that:

$$x_s(t) = x(t)x_\delta(t) = \sum_{n=-\infty}^{\infty} x(t)\delta(t - nT_s) = \sum_{n=-\infty}^{\infty} x(nT_s)\delta(t - nT_s)$$

Notice that the Fourier transform of an impulse train is another impulse train.

$$X_\delta(f) = \frac{1}{T_s} \sum_{n=-\infty}^{\infty} \delta(f - nf_s)$$



figure(4.1): sampling theorem

Convolution with an impulse function simply shifts the original function:

$$X(f) * \delta(f - nf_s)$$

We can now solve for the transform  $X_s(f)$  of the sampled waveform:

$$X(f) * \delta(f - nf_s) = X(f - nf_s)$$

So that

$$X_s(f) = X(f) * X_\delta(f) = X(f) * \left[ \frac{1}{T_s} \sum_{n=-\infty}^{\infty} \delta(f - nf_s) \right] = \frac{1}{T_s} \sum_{n=-\infty}^{\infty} X(f - nf_s)$$

When the sampling rate is chosen  $f_s = 2f_m$  each spectral replicate is separated from each of its neighbors by a frequency band exactly equal to  $f_s$  hertz, and the analog waveform can theoretically be completely recovered from the samples, by the use of filtering. It should be clear that if  $f_s > 2f_m$ , the replications will be move farther apart in frequency making it easier to perform the filtering operation, as in figure (4.2).

When the sampling rate is reduced, such that  $f_s < 2f_m$  as in figure (4.3). The replications will overlap, as shown in figure below, and some information will be lost. This phenomenon is called aliasing.

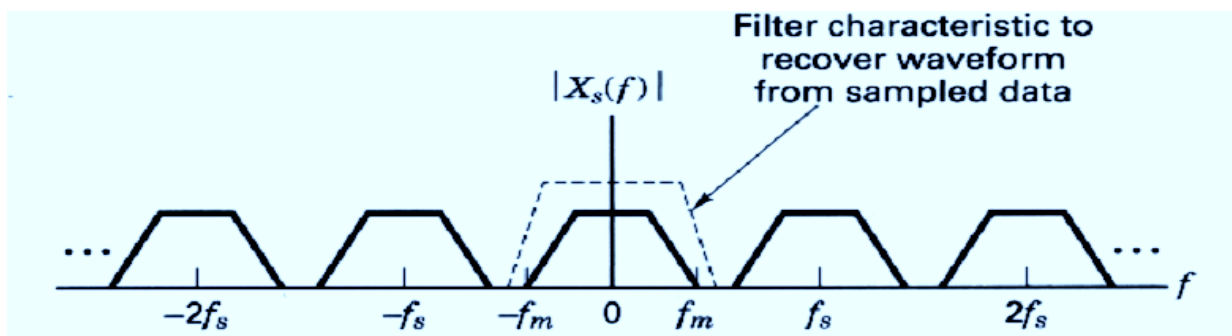


figure (4.2): Sampled spectrum  $f_s > 2f_m$

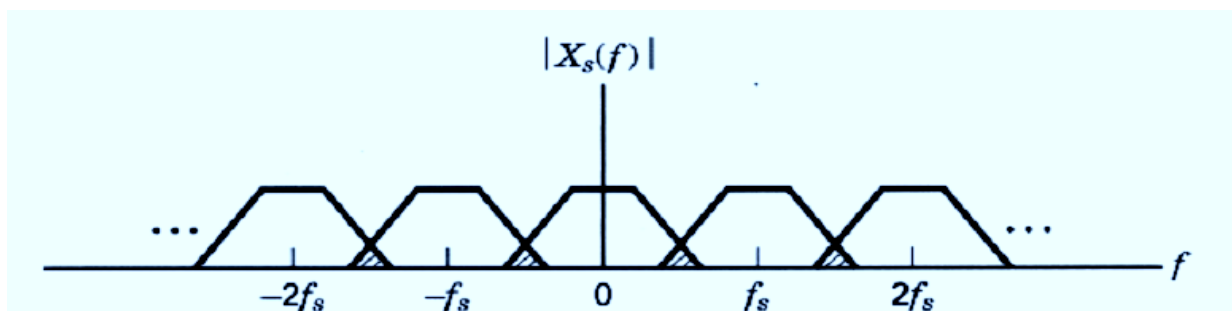


figure (4.3): Sampled spectrum  $f_s < 2f_m$

A bandlimited signal having no spectral components above  $f_m$  hertz can be determined uniquely by values sampled at uniform intervals of  $T_s \leq \frac{1}{2f_m} \text{ sec.}$

The sampling rate is  $f_s = \frac{1}{T_s}$

So that  $f_s \geq 2f_m$ . The sampling rate  $f_s = 2f_m$  is called Nyquist rate.

**Example 4.1:** Find the Nyquist rate and Nyquist interval for the following signals.

i-  $m(t) = \frac{\sin(500\pi t)}{\pi t}$

ii-  $m(t) = \frac{1}{2\pi} \cos(4000\pi t) \cos(1000\pi t)$

**Sol:**

i-  $\omega t = 500\pi t \quad \therefore 2\pi f = 500\pi \rightarrow f = 250\text{Hz}$

Nyquist interval =  $\frac{1}{2f_{\max}} = \frac{1}{2 \times 250} = 2 \text{ msec.}$

Nyquist rate =  $2f_{\max} = 2 \times 250 = 500\text{Hz}$

$$\begin{aligned} \text{ii- } m(t) &= \frac{1}{2\pi} \left[ \frac{1}{2} \{ \cos(4000\pi t - 1000\pi t) + \cos(4000\pi t + 1000\pi t) \} \right] \\ &= \frac{1}{4\pi} \{ \cos(3000\pi t) + \cos(5000\pi t) \} \end{aligned}$$

Then the highest frequency is 2500Hz

Nyquist interval =  $\frac{1}{2f_{\max}} = \frac{1}{2 \times 2500} = 0.2 \text{ msec.}$

Nyquist rate =  $2f_{\max} = 2 \times 2500 = 5000\text{Hz}$

**H. W:** Find the Nyquist interval and Nyquist rate for the following:

i-  $\frac{1}{2\pi} \cos(400\pi t) \cdot \cos(200\pi t)$

ii-  $\frac{1}{\pi} \sin \pi t$

**Example4.2:** A waveform  $[20+20\sin(500t+30^\circ)]$  is to be sampled periodically and reproduced from these sample values. Find maximum allowable time interval between sample values, how many sample values are needed to be stored in order to reproduce 1 sec of this waveform?.

**Sol:**

$$x(t) = 20 + 20 \sin(500t + 30^\circ)$$

$$\omega = 500 \rightarrow 2\pi f = 500 \rightarrow f = 79.58 \text{ Hz}$$

Minimum sampling rate will be twice of the signal frequency:

$$f_{s(\min)} = 2 \times 79.58 = 159.15 \text{ Hz}$$

$$T_{s(\max)} = \frac{1}{f_{s(\min)}} = \frac{1}{159.15} = 6.283 \text{ msec.}$$

$$\text{Number of sample in } 1\text{sec} = \frac{1}{6.283\text{msec}} = 159.16 \approx 160 \text{ sample}$$

## 4.2 Source Coding

An important problem in communications is the efficient representation of data generated by a discrete source. The process by which this representation is accomplished is called source encoding. An efficient source encoder must satisfies two functional requirements:

- i- The code words produced by the encoder are in binary form.
- ii- The source code is uniquely decodable, so that the original source sequence can be reconstructed perfectly from the encoded binary sequence.

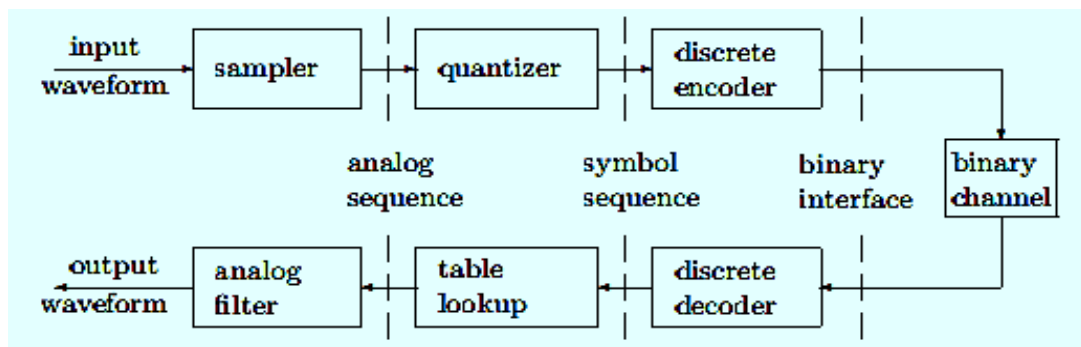


figure (4.4): communications system

The entropy for a source with statistically independent symbols:

$$H(Y) = - \sum_{j=1}^m P(y_j) \log_2 P(y_j)$$

We have:

$$\max[H(Y)] = \log_2 m$$

A code efficiency can therefore be defined as:

$$\eta = \frac{H(Y)}{\max[H(Y)]} \times 100$$

The overall code length,  $L$ , can be defined as the average code word length:

$$L = \sum_{j=1}^m P(x_j) l_j \quad \text{bits/symbol}$$

The code efficiency can be found by:

$$\eta = \frac{H(Y)}{L} \times 100$$

Not that  $\max[H(Y)] \quad \text{bits/symbol} = L \quad \text{bits/codeword}$

#### 4.2.1 Fixed- Length Code Words

If the alphabet  $X$  consists of the 7 symbols  $\{a, b, c, d, e, f, g\}$ , then the following fixed-length code of block length  $L = 3$  could be used.

$$C(a) = 000$$

$$C(b) = 001$$

$$C(c) = 010$$

$$C(d) = 011$$

$$C(e) = 100$$

$$C(f) = 101$$

$$C(g) = 110.$$

The encoded output contains  $L$  bits per source symbol. For the above example the source sequence *bad...* would be encoded into 001000011... . Note that the output bits are simply run together (or, more technically, concatenated). This

method is non-probabilistic; it takes no account of whether some symbols occur more frequently than others, and it works robustly regardless of the symbol frequencies.

This is used when the source produces almost equi-probable messages  $p(x_1) \cong p(x_2) \cong p(x_3) \cong \dots \cong p(x_n)$ , then  $l_1 = l_2 = l_3 = \dots = l_n = L_c$  and for binary coding then:

1-  $L_c = \log_2 n$  bit/message if  $n = 2^r$  ( $n = 2, 4, 8, 16, \dots$  and  $r$  is an integer) which gives  $\eta = 100\%$

2-  $L_c = \text{Int}[\log_2 n] + 1$  bits/message if  $n \neq 2^r$  which gives less efficiency

**Example 4.3:** For ten equi-probable messages coded in a fixed length code, find the efficiency.

**Sol:**  $p(x_i) = \frac{1}{10}$  and  $L_c = \text{Int}[\log_2 10] + 1 = 4$  bits

$$\text{and } \eta = \frac{H(X)}{L_c} \times 100\% = \frac{\log_2 10}{4} \times 100\% = 83.048\%$$

**Example 4.4:** For eight equi-probable messages coded in a fixed length code, find the efficiency

**Sol:**  $p(x_i) = \frac{1}{8}$  and  $L_c = \log_2 8 = 3$  bits and  $\eta = \frac{3}{3} \times 100\% = 100\%$

**Example 4.5:** Find the efficiency of a fixed length code used to encode messages obtained from throwing a fair die (a) once, (b) twice, (c) 3 times.

**Sol:**

a- For a fair die, the messages obtained from it are equiprobable with a

probability of  $p(x_i) = \frac{1}{6}$  with  $n = 6$ .

$$L_C = \text{Int}[\log_2 6] + 1 = 3 \text{ bits/message}$$

$$\eta = \frac{H(X)}{L_C} \times 100\% = \frac{\log_2 6}{3} \times 100\% = 86.165\%$$

b- For two throws then the possible messages are  $n = 6 \times 6 = 36$  messages with equal probabilities

$$L_C = \text{Int}[\log_2 36] + 1 = 6 \text{ bits/message} = 6 \text{ bits/2-symbols}$$

$$\text{while } H(X) = \log_2 6 \text{ bits/symbol} \quad \eta = \frac{2 \times H(X)}{L_C} \times 100\% = 86.165\%$$

c- For three throws then the possible messages are  $n = 6 \times 6 \times 6 = 216$  with equal probabilities

$$L_C = \text{Int}[\log_2 216] + 1 = 8 \text{ bits/message} = 8 \text{ bits/3-symbols}$$

$$\text{while } H(X) = \log_2 6 \text{ bits/symbol} \quad \eta = \frac{3 \times H(X)}{L_C} \times 100\% = 96.936\%$$

#### 4.2.2 Variable-Length Code Words

When the source symbols are not equally probable, a more efficient encoding method is to use variable-length code words. For example, a variable-length code for the alphabet  $X = \{a, b, c\}$  and its lengths might be given by

$$C(a) = 0 \quad l(a) = 1$$

$$C(b) = 10 \quad l(b) = 2$$

$$C(c) = 11 \quad l(c) = 2$$

The major property that is usually required from any variable-length code is that of unique decodability. For example, the above code  $C$  for the alphabet  $X = \{a, b, c\}$  is soon shown to be uniquely decodable. However such code is not



uniquely decodable, even though the codewords are all different. If the source decoder observes 01, it cannot determine whether the source emitted (a b) or (c).

**4.2.3 Prefix-free codes:** A **prefix code** is a type of code system (typically a variable-length code) distinguished by its possession of the "prefix property", which requires that there is no code word in the system that is a prefix (initial segment) of any other code word in the system. For example:

$\{a = 0, b = 110, c = 10, d = 111\}$  is a prefix code.

When message probabilities are not equal, then we use variable length codes. The following properties need to be considered when attempting to use variable length codes:

#### 1) Unique decoding:

**Example 4.6:** Consider a 4 alphabet symbols with symbols represented by binary digits as follows:

$A = 0 \quad B = 01 \quad C = 11 \quad D = 00$

If we receive the code word 0011 it is not known whether the transmission was  $DC$  or  $AAC$ . This example is not, therefore, uniquely decodable.

#### 2) Instantaneous decoding:

**Example 4.7:** Consider a 4 alphabet symbols with symbols represented by binary digits as follows:

$A = 0$

$B = 10$

$C = 110$

$D = 111$

This code can be instantaneously decoded since no complete codeword is a prefix of a larger codeword. This is in contrast to the previous example where  $A$  is a prefix of both  $B$  and  $D$ . This example is also a ‘comma code’ as the symbol zero indicates the end of a codeword except for the all ones word whose length is known.

**Example 4.8:** Consider a 4 alphabet symbols with symbols represented by binary digits as follows:

$$A = 0 \qquad B = 01 \qquad C = 011 \qquad D = 111$$

The code is identical to the previous example but the bits are time reversed. It is still uniquely decodable but no longer instantaneous, since early code-words are now prefixes of later ones.

### 4.3 Coding Algorithm

Different types of algorithms that used to encoded the message, some of these are discussed in this chapter

#### 4.3.1 Huffman Code

The Huffman coding algorithm comprises two steps, reduction and splitting. These steps can be summarized as follows:

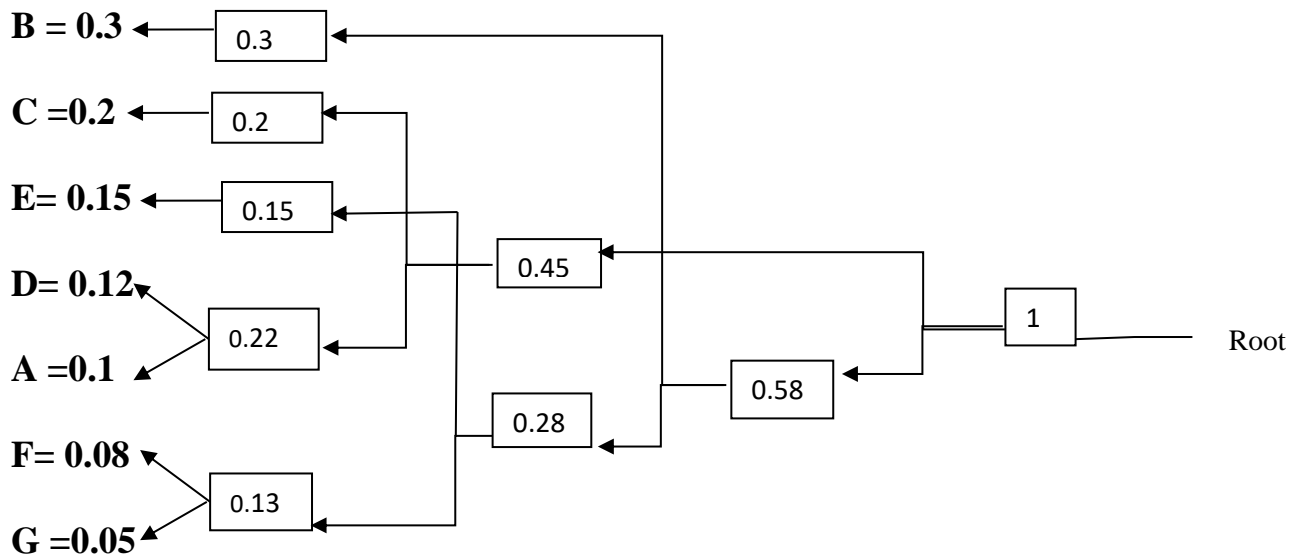
- 1-find the summation of the probabilities if it is not equal 1 then add the any variable to reach the one
- 2-Sort symbols with decreasing probabilities. Assign “0” and “1” to the symbols with the two lowest probabilities
- 3-. Both symbols are combined to a new symbol with the sum of the probabilities. Resort the symbols again with decreasing probabilities.
4. Repeat until the code tree is complete
5. Read out the code words from the back of the tree

6- Determine the efficiency of the cod word  $\eta = \frac{H(i)}{L_a} * 100\%$  note that  $H(x) = -\sum p(i) \log p(i)$  and the average entropy length  $(L_a) = \sum p(i) * l(i)$

**Example 4.9:** implement Huffman code for the source emits message as bellow

A=0.1, B=0.3, C=0.2, D= 0.12, E=0.15, F=0.08, design the code and obtain the entropy & efficiency?

Sol:



symbol	value	Huffman code	Average length( $L_a$ )
<b>B</b>	0.3	00	2
<b>C</b>	0.2	11	2
<b>E</b>	0.15	010	3
<b>D</b>	0.12	100	3
<b>A</b>	0.1	101	3
<b>F</b>	0.08	0110	4
<b>G</b>	0.05	0111	4

$$H(x) = -\sum p(i) \log p(i) = -0.3 \log 1/0.3 - 0.2 \log 1/0.2 - 0.15 \log 1/0.15 -$$

.....


$$= 2.603 \text{ b/s}$$

$$L_a = \sum p(i) * l(i) = 0.3 * 2 + 0.2 * 2 + 0.15 * 3 + 0.12 * 3 + 0.1 * 3 + 0.08 * 4 + 0.05 * 4 = 2.63 \text{ b/s}$$

$$\eta = \frac{H(i)}{L_a} * 100\% = \frac{2.603}{2.63} * 100\% = 98.9\%$$

**Example 4.10:** Design Huffman codes for  $A = \{a_1, a_2, \dots, a_5\}$ , having the probabilities  $\{0.2, 0.4, 0.2, 0.1, 0.1\}$ .

Symbol	Step 1	Step 2	Step 3	Step 4	Codeword
$a_2$	0.4	0.4	0.4	0.6 0	1
$a_1$	0.2	0.2	0.4 } 0	0.4 1	01
$a_3$	0.2	0.2 } 0	0.2 } 1		000
$a_4$	0.1 } 0	0.2 } 1			0010
$a_5$	0.1 } 1				0011



The average code word length:

$$L = 0.4 \times 1 + 0.2 \times 2 + 0.2 \times 3 + 0.1 \times 4 + 0.1 \times 4 = 2.2 \text{ bits/symbol}$$

The source entropy:

$$H(Y) = -[0.4 \ln 0.4 + 2 \times 0.2 \ln 0.2 + 2 \times 0.1 \ln 0.1] / \ln 2 = 2.12193$$

bits/symbol

$$\text{The code efficiency: } \eta = \frac{2.12193}{2.2} \times 100 = 96.45\%$$

It can be design Huffman codes with minimum variance:

Symbol	Step 1	Step 2	Step 3	Step 4	Codeword
$a_2$	0.4	0.4	0.4	0.6 0	00
$a_1$	0.2	0.2	0.4 } 0	0.4 1	10
$a_3$	0.2	0.2 } 0	0.2 } 1		11
$a_4$	0.1 } 0	0.2 } 1			010
$a_5$	0.1 } 1				011

The average code word length is still 2.2 bits/symbol. But variances are different!

**Example 4.11:** Develop the Huffman code for the following set of symbols

Symbol	A	B	C	D	E	F	G	H
Probability	0.1	0.18	0.4	0.05	0.06	0.1	0.07	0.04

**Sol:**

C	0.40	0.40	0.40	0.40	0.40	0.40	0.60	1.0
B	0.18	0.18	0.18	0.19	0.23	0.37	0.40	
A	0.10	0.10	0.13	0.18	0.19	0.23		
F	0.10	0.10	0.10	0.13	0.18			
G	0.07	0.09	0.10	0.10				
E	0.06	0.07	0.09					
D	0.05	0.06						
H	0.04							

So we obtain the following codes

Symbol	A	B	C	D	E	F	G	H
Probability	0.1	0.18	0.4	0.05	0.06	0.1	0.07	0.04
Codeword	011	001	1	00010	0101	0000	0100	00011
$l_i$	3	3	1	5	4	4	4	5

$$H(X) = -\sum_{i=1}^8 p(x_i) \log_2 p(x_i) = 2.552 \text{ bits/symbol}$$

$$L_C = \sum_{i=1}^8 l_i p(x_i) = 2.61 \text{ bits/symbol}$$

$$\eta = \frac{H(X)}{L_C} \times 100\% = 97.778\%$$

**H.W:** develop the Huffman code for  $p(x)=[0.4, 0.25, 0.15, 0.1, 0.07, 0.03]$  then find the coding efficiency?

### 4.3.2 Shannon- Fano Code ( Fano code)

In Shannon–Fano coding, the symbols are arranged in order from most probable to least probable, and then divided into two sets whose total probabilities are as close as possible to being equal. All symbols then have the first digits of their codes assigned; symbols in the first set receive "0" and symbols in the second set receive "1". As long as any sets with more than one member remain, the same process is repeated on those sets, to determine successive digits of their codes.

#### Algorithm:

The algorithm used for generating Shannon-Fano codes is as follows:

- 1) For a given list of symbols, develop a corresponding list of probabilities so that each symbol's relative probability is known.
- 2) List the symbols in the order of decreasing probability.

- 3) Divide the symbols into two groups so that each group has equal probability.
- 4) Assign a value 0 to first group and a value 1 to second group.
- 5) Repeat steps 3 and 4, each time partitioning the sets with nearly equal probabilities as possible until further partitioning is not possible.

**Example 4.12:** The five symbols which have the following frequency and probabilities, design suitable Shannon-Fano binary code. Calculate average code length, source entropy and efficiency.

Symbol	count	Probabilities	Binary codes	Length
<b>A</b>	15	0.385	00	2
<b>B</b>	7	0.1795	01	2
<b>C</b>	6	0.154	10	2
<b>D</b>	6	0.154	110	3
<b>E</b>	5	0.128	111	3

**Sol:** The average code word length:

$$L = \sum_{j=1}^m P(x_j) l_j$$

$$L = 2 \times 0.385 + 2 \times 0.1793 + 2 \times 0.154 + 3 \times 0.154 + 3 \times 0.128$$

$$= 2.28 \text{ bits/symbol}$$

The source entropy is:  $H(Y) = -\sum_{j=1}^m P(y_j) \log_2 P(y_j)$

$$H(Y) = -[0.385 \ln 0.385 + 0.1793 \ln 0.1793 + 2 \times 0.154 \ln 0.154 + 0.128 \ln 0.128] / \ln 2 = 2.18567 \text{ bits/symbol}$$

The code efficiency:

$$\eta = \frac{H(Y)}{L} \times 100 = \frac{2.18567}{2.28} \times 100 = 95.86\%$$

**Example 4.13:** Develop the Shannon -Fano code for the following set of messages,  $p(x) = [0.35 \ 0.2 \ 0.15 \ 0.12 \ 0.1 \ 0.08]$  then find the code efficiency.

**Sol:**

$x_i$	$p(x_i)$	Code			$l_i$
$x_1$	0.35	0	0		2
$x_2$	0.2	0	1		2
$x_3$	0.15	1	0	0	3
$x_4$	0.12	1	0	1	3
$x_5$	0.10	1	1	0	3
$x_6$	0.08	1	1	1	3

$$L_C = \sum_{i=1}^6 l_i p(x_i) = 2.45 \text{ bits/symbol}$$

$$H(X) = -\sum_{i=1}^6 p(x_i) \log_2 p(x_i) = 2.396 \text{ bits/symbol}$$

$$\eta = \frac{H(X)}{L_C} \times 100\% = 97.796\%$$



**Example 4.14:** Repeat the previous example using with  $r = 3$

**Sol:**

$x_i$	$p(x_i)$	Code		$l_i$
$x_1$	0.35	0		1
$x_2$	0.2	1	0	2
$x_3$	0.15	1	1	2
$x_4$	0.12	2	0	2
$x_5$	0.10	2	1	2
$x_6$	0.08	2	2	2

$$L_C = \sum_{i=1}^6 l_i p(x_i) = 1.65 \quad \text{ternary unit/symbol}$$

$$H(X) = -\sum_{i=1}^6 p(x_i) \log_3 p(x_i) = 1.512 \quad \text{ternary unit/symbol}$$

$$\eta = \frac{H(X)}{L_C} \times 100\% = 91.636\%$$

**Example 4.15:** The source of information A generates the symbols {a,b,c,d,e,f,g,h} with the corresponding probabilities {0.15,0.14, 0.3,0.1, 0.12,0.08,0.06, 0.05}. Encoding the source symbols using binary Shannon-Fano encoder give the efficiency?

**Sol:**

Symbol	prob.					CW	$I_k$ / bit
c	0.3	0	0			00	2
a	0.15	0	1			01	2
b	0.14	1	0	0		100	3
e	0.12	1	0	1		101	3
d	0.1	1	1	0	0	1100	4
f	0.08	1	1	0	1	1101	4
g	0.06	1	1	1	0	1110	4
h	0.05	1	1	1	1	1111	4

$$H=2.78\text{b/s}, \quad L_a=2.84 \text{ b/s}, \quad \eta = 97.8\%$$

**Example 4.16:** The five symbols A,B,C,D and E which have the following frequency and probabilities [0.385, 0.1795, 0.154, 0.154, 0.128], design suitable Shannon-Fano binary code. Calculate average code length, source entropy and efficiency.

**Sol:**

Symbol	count	Probabilities	Binary codes	Length
A	15	0.385	00	2
B	7	0.1795	01	2
C	6	0.154	10	2
D	6	0.154	110	3
E	5	0.128	111	3

The average code word length:

$$L = \sum_{j=1}^m P(x_j)l_j$$

$$L = 2 \times 0.385 + 2 \times 0.1793 + 2 \times 0.154 + 3 \times 0.154 + 3 \times 0.128 \\ = 2.28 \text{ bits/symbol}$$

The source entropy is:  $H(Y) = -\sum_{j=1}^m P(y_j) \log_2 P(y_j)$

$$H(Y) = -[0.385 \ln 0.385 + 0.1793 \ln 0.1793 + 2 \times 0.154 \ln 0.154 + \\ 0.128 \ln 0.128] / \ln 2 = 2.18567 \text{ bits/symbol}$$

$$\text{The code efficiency: } \eta = \frac{H(Y)}{L} \times 100 = \frac{2.18567}{2.28} \times 100 = 95.86\%$$

**Example 4.17:** Develop the Shannon - Fano code for the following set of messages,  $p(x) = [0.35 \ 0.2 \ 0.15 \ 0.12 \ 0.1 \ 0.08]$  then find the code efficiency.

**Sol:**

$x_i$	$p(x_i)$	Code			$l_i$
$x_1$	0.35	0	0		2
$x_2$	0.2	0	1		2
$x_3$	0.15	1	0	0	3
$x_4$	0.12	1	0	1	3
$x_5$	0.10	1	1	0	3
$x_6$	0.08	1	1	1	3

$$L_C = \sum_{i=1}^6 l_i p(x_i) = 2.45 \text{ bits/symbol}$$

$$H(X) = -\sum_{i=1}^6 p(x_i) \log_2 p(x_i) = 2.396 \text{ bits/symbol}$$

$$\eta = \frac{H(X)}{L_C} \times 100\% = 97.796\%$$

**Example 4.18:** Repeat the previous example using with  $r = 3$

**Sol:**

$x_i$	$p(x_i)$	Code		$l_i$
$x_1$	0.35	0		1
$x_2$	0.2	1	0	2
$x_3$	0.15	1	1	2
$x_4$	0.12	2	0	2
$x_5$	0.10	2	1	2
$x_6$	0.08	2	2	2

$$L_C = \sum_{i=1}^6 l_i p(x_i) = 1.65 \text{ ternary unit/symbol}$$

$$H(X) = -\sum_{i=1}^6 p(x_i) \log_3 p(x_i) = 1.512 \text{ ternary unit/symbol}$$

$$\eta = \frac{H(X)}{L_C} \times 100\% = 91.636\%$$

### 4.3.3 Shannon Code

For messages  $x_1, x_2, x_3, \dots, x_n$  with probabilities  $p(x_1), p(x_2), p(x_3), \dots, p(x_n)$  then:

$$1) l_i = -\log_2 p(x_i) \quad \text{if } p(x_i) = \left(\frac{1}{2}\right)^r \quad \left\{\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \dots\right\}$$

$$2) l_i = \text{Int}[-\log_2 p(x_i)] + 1 \quad \text{if } p(x_i) \neq \left(\frac{1}{2}\right)^r$$

Also define 
$$F_i = \sum_{k=1}^{i-1} p(x_k) \quad 1 \geq \omega_i \geq 0$$

then the codeword of  $x_i$  is the binary equivalent of  $F_i$  consisting of  $l_i$  bits.

$$C_i = (F_i)_2^{l_i}$$

where  $C_i$  is the binary equivalent of  $F_i$  up to  $l_i$  bits. In encoding, messages must be arranged in a decreasing order of probabilities. The steps of the

**Shannon algorithm can summarize in these points**

- 1- List the symbols in the order of decreasing probability.
- 2- Compute the index  $bi$  which gives the number of bits necessary for encoding the message according to:  $\log \frac{1}{p_i} \leq bi \leq 1 + \log \frac{1}{p_i}$
- 3- Evaluate the vector  $F_i$  for the accumulation of the probability :  $F_i = \sum_{k=1}^{i-1} p_k$
- 4- Simulate the value of  $F_i$  in to binary for  $C_i = (F_i)_2^{bi}$

**Example 4.19:** a communication source emits the following symbol with the prob.

A=0.3, B=0.2, C=0.15, D=0.12, E=0.1, F=0.08 & G=0.05

**Sol:**

symbol	Index i	Prob Pi	No.of b/m(bi)	Fi	Code Ci
A	1	0.3	2	0	00
B	2	0.2	3	0.3	010
C	3	0.15	3	0.5	100
D	4	0.12	4	0.65	1010
E	5	0.1	4	0.77	1100
F	6	0.08	4	0.87	1101
G	7	0.05	5	0.95	11110

1- Do  $\log \frac{1}{p_i} \leq b_i \leq 1 + \log \frac{1}{p_i}$  for all value

$$A=0.3 \rightarrow \log \frac{1}{0.3} \leq b_i \leq 1 + \log \frac{1}{0.3} \dots \quad \Rightarrow b_1 \leq 2.7 \quad b_1 \Rightarrow$$

$$B=0.2 \Rightarrow \log \frac{1}{0.2} \leq b_i \leq 1 + \log \frac{1}{0.2} \Rightarrow 2.3 \leq b_2 \leq 3.3 \Rightarrow b_2=3 \text{ And so on...}$$

$$2- F_i = \sum_{k=1}^{i-1} p_k \Rightarrow F_1=0, F_2=\sum_{k=1}^{2-1} p_k = \sum_{k=1}^1 p_1=0.3,$$

$$F_3 = p_1 + p_2 = 0.3 + 0.2 = 0.5,$$

$$F_4 = p_1 + p_2 + p_3 = 0.3 + 0.2 + 0.15 = 0.65$$

$$F_5 = p_1 + p_2 + p_3 + p_4 = 0.3 + 0.2 + 0.15 + 0.12 = 0.77$$

$$F_6 = p_1 + p_2 + p_3 + p_4 + p_5 = 0.77 + 0.1 = 0.87$$

$$F_7 = p_1 + p_2 + p_3 + p_4 + p_5 + p_6 = 0.87 + 0.08 = 0.95$$

3-the binary code number:  $C_i = (F_i)_2^{b_i}$

$$C_1 = (F_1)_2^{b_1} = (0)_2^2 = 00$$

$$C_2 = (F_2)_2^{b_2} = (0.3)_2^3 = 010$$

$$C_3 = (F_3)_2^{b_3} = (0.5)_2^3 = 100$$

$$C_4 = (F_4)_2^{b_4} = (0.65)_2^4 = 1010$$

$$C_5 = (F_5)_2^{b_5} = (0.77)_2^4 = 1100$$

$$C_6 = 1101, C_7 = 11110$$

4-entropy length  $(L_c) = \sum p(i) * b(i)$

$$b(i) = 0.3*2 + 0.2*3 + 0.15*3 + 0.12*4 + 0.1*4 + 0.08*4 + 0.05*5 = 3.1 \text{ bit/symbol}$$

$$5-H(i) = -\sum p(i) \log p(i) = -0.3 \log 1/0.3 - 0.2 \log 1/0.2 - 0.15 \log 1/0.15 - 0.12 \log 1/0.12 - 0.1 \log 1/0.1 - 0.08 \log 1/0.08 - 0.05 \log 1/0.05 = 2.603 \text{ b/s}$$

$$6\text{-efficiency } \eta = \frac{H(i)}{L_c} * 100\% = 2.603/3.1 * 100\% = 83.96\%$$

**Example 4.20:** Develop the Shannon code for the following set of messages,

$$p(x) = [0.3 \quad 0.2 \quad 0.15 \quad 0.12 \quad 0.1 \quad 0.08 \quad 0.05]$$

then find: (a) Code efficiency, (b)  $p(0)$  at the encoder output.

**Sol:**

To find  $C_1$

$$0 \times 2 = 0 \quad 0 \quad \downarrow$$

To find  $C_2$

$$\begin{array}{l} 0.3 \times 2 = 0.6 \quad 0 \\ 0.6 \times 2 = 1.2 \quad 1 \end{array} \quad \downarrow$$

To find  $C_3$

$$\begin{array}{l} 0.5 \times 2 = 1.0 \quad 1 \\ 0.0 \times 2 = 0.0 \quad 0 \end{array} \quad \downarrow$$

To find  $C_4$ 

$$\begin{array}{rcl}
 0.65 \times 2 = 1.3 & 1 & \\
 0.30 \times 2 = 0.6 & 0 & \\
 0.60 \times 2 = 1.2 & 1 & \\
 0.20 \times 2 = 0.4 & 0 & \downarrow
 \end{array}$$

To find  $C_5$ 

$$\begin{array}{rcl}
 0.77 \times 2 = 1.54 & 1 & \\
 0.54 \times 2 = 1.08 & 1 & \\
 0.08 \times 2 = 0.16 & 0 & \\
 0.16 \times 2 = 0.32 & 0 & \downarrow
 \end{array}$$

$x_i$	$p(x_i)$	$l_i$	$F_i$	$C_i$	$0_i$
$x_1$	0.3	2	0	00	2
$x_2$	0.2	3	0.3	010	2
$x_3$	0.15	3	0.5	100	2
$x_4$	0.12	4	0.65	1010	2
$x_5$	0.10	4	0.77	1100	2
$x_6$	0.08	4	0.87	1101	1
$x_7$	0.05	5	0.95	11110	1

(a) To find the code efficiency, we have

$$L_C = \sum_{i=1}^7 l_i p(x_i) = 3.1 \text{ bits/message.}$$

$$H(X) = -\sum_{i=1}^7 p(x_i) \log_2 p(x_i) = 2.6029 \text{ bits/message.}$$

$$\eta = \frac{H(X)}{L_C} \times 100\% = 83.965\%$$

(b)  $p(0)$  at the encoder output is



$$p(0) = \frac{\sum_{i=1}^7 0_i p(x_i)}{L_C} = \frac{0.6 + 0.4 + 0.3 + 0.24 + 0.2 + 0.08 + 0.05}{3.1}$$

$$p(0) = 0.603$$

**Example 4.21:** Repeat the previous example using ternary coding.

**Sol:**

$$1) l_i = -\log_3 p(x_i) \quad \text{if } p(x_i) = \left(\frac{1}{3}\right)^r \quad \left\{\frac{1}{3}, \frac{1}{9}, \frac{1}{27}, \dots\right\}$$

$$2) l_i = \text{Int}[-\log_3 p(x_i)] + 1 \quad \text{if } p(x_i) \neq \left(\frac{1}{3}\right)^r \quad \text{and} \quad C_i = (F_i)_3^{l_i}$$

$x_i$	$p(x_i)$	$l_i$	$F_i$	$C_i$	$0_i$
$x_1$	0.3	2	0	00	2
$x_2$	0.2	2	0.3	02	1
$x_3$	0.15	2	0.5	11	0
$x_4$	0.12	2	0.65	12	0
$x_5$	0.10	3	0.77	202	1
$x_6$	0.08	3	0.87	212	0
$x_7$	0.05	3	0.95	221	0

To find  $C_1$

$$0 \times 3 = 0 \quad 0 \downarrow$$

To find  $C_2$

$$0.3 \times 3 = 0.9 \quad 0 \downarrow$$

To find  $C_3$

$$0.5 \times 3 = 1.5 \quad 1 \downarrow$$

To find  $C_4$

$$0.65 \times 3 = 1.95 \quad 1 \downarrow$$

$$0.95 \times 3 = 2.85 \quad 2$$

To find  $C_5$

$$0.77 \times 3 = 2.31 \quad 2 \downarrow$$

$$0.31 \times 3 = 0.93 \quad 0 \downarrow$$

(a) To find the code efficiency, we have

$$L_c = \sum_{i=1}^7 l_i p(x_i) = 2.23 \text{ ternary unit/message.}$$

$$H(X) = -\sum_{i=1}^7 p(x_i) \log_3 p(x_i) = 1.642 \text{ ternary unit/message.}$$

$$\eta = \frac{H(X)}{L_c} \times 100\% = 73.632\%$$

(b)  $p(0)$  at the encoder output is

$$p(0) = \frac{\sum_{i=1}^7 0_i p(x_i)}{L_c} = \frac{0.6 + 0.2 + 0.1}{2.23}$$

$$p(0) = 0.404$$

### Note:

The condition that the number of symbols  $n$  so that we can decode them using  $r$  Huffman coding is  $\frac{n-r}{r-1}$  must be an integer value, otherwise, add a redundant symbols with a probabilities equal to zero so that the condition is satisfied.

## 4.4 Data Compression

In computer science and information theory, data compression, source coding, or bit-rate reduction involves encoding information using fewer bits than the original representation. Compression can be either lossy or lossless.

**4.4.1 Lossless Data Compression Algorithms:** usually exploit statistical redundancy to represent data more concisely without losing information, so that the process is reversible. Lossless compression is possible because most real-

world data has statistical redundancy. For example, an image may have areas of color that do not change over several pixels.

**4.4.2 Lossy Data Compression:** is the converse of lossless data compression. In these schemes, some loss of information is acceptable. Dropping nonessential detail from the data source can save storage space. There is a corresponding trade-off between preserving information and reducing size.

#### 4.4.3 Run-Length Encoding (RLE)

Run-Length Encoding is a very simple lossless data compression technique that replaces runs of two or more of the same character with a number which represents the length of the run, followed by the original character; single characters are coded as runs of 1. RLE is useful for highly-redundant data, indexed images with many pixels of the same color in a row.

**Example 4.22:** Input: AAABBCCCCDEEEEEEEAAAAAAAAAAAAAAAAAAAA  
AAAAAAAAAAAAAAAAAAAAAAAAAAAA Output: 3A2B4C1D6E38A

The input message to RLE encoder is a variable while the output code word is fixed, unlike Huffman code where the input is fixed while the output is varied.

**Example 4.23:** Consider these repeated pixels values in an image ... 0 0 0 0 0 0  
0 0 0 0 0 0 5 5 5 5 0 0 0 0 0 0 0 0 We could represent them more efficiently as  
(12, 0)(4, 5)(8, 0)

24 bytes reduced to 6 which gives a compression ratio of  $24/6 = 4:1$ .

**Example 4.24:** Original Sequence (1 Row): 111122233333311112222 can be encoded as: (4,1),(3,2),(6,3),(4,1),(4,2). 21 bytes reduced to 10 gives a compression ratio of  $21/10 = 2.1:1$ .

**Example 4.25:** Original Sequence (1 Row): HHHHHHHUFFFFFFFFFFFFFFFFF can be encoded as: (7,H),(1,U),(14,F) . 22 bytes reduced to 6 gives a compression ratio of  $22/6 = 11:3$ .

**Savings Ratio:** the savings ratio is related to the compression ratio and is a measure of the amount of redundancy between two representations (compressed and uncompressed).

Let:  $N_1$  = the total number of bytes required to store an uncompressed (raw) source image.

$N_2$  = the total number of bytes required to store the compressed data.

The compression ratio  $C_r$  is then defined as:

$$C_r = \frac{N_1}{N_2}$$

- ▶ Larger compression ratios indicate more effective compression
- ▶ Smaller compression ratios indicate less effective compression
- ▶ Compression ratios less than one indicate that the uncompressed representation has high degree of irregularity.

The saving ratio  $S_r$  is then defined as :

$$S_r = \frac{(N_1 - N_2)}{N_1}$$

- ▶ Higher saving ratio indicate more effective compression while negative ratios are possible and indicate that the compressed image has larger memory size than the original.

**Example 4.26:** a 5 Megabyte image is compressed into a 1 Megabyte image, the savings ratio is defined as  $(5-1)/5$  or  $4/5$  or 80%.

This ratio indicates that 80% of the uncompressed data has been eliminated in the compressed encoding.

## Problems

- 1-Find the Nyquist rate and Nyquist interval for the following signals.
- 2-A waveform  $[20+20\sin(500t+30^\circ)]$  is to be sampled periodically and reproduced from these sample values. Find maximum allowable time interval between sample values, how many sample values are needed to be stored in order to reproduce 1 sec of this waveform?
- 3-Find the efficiency of a fixed length code used to encode messages obtained from throwing a fair die (a) once, (b) twice, (c) 3 times
- 4-Develop the Shannon code for the following set of messages,  
 $p(x)=[0.3 \ 0.2 \ 0.15 \ 0.12 \ 0.1 \ 0.08 \ 0.05]$

then find: 1- Code efficiency, 2-  $p(0)$  at the encoder output.

- 5- The five symbols A,B,C,D and E which have the following frequency and probabilities[0.385, 0.1795, 0.154, 0.154, 0.128], design suitable Shannon-Fano binary code. Calculate average code length, source entropy and efficiency.

- 6- Develop variable length code by Shannon-Fano code for the message  
 [F F F A B C B B F E E G G E D F G G F G]

- 1-Find code table , code efficiency,  $P(0)$ ,  $P(1)$ , and draw code rate tree.
- 2-Use code table to encode the text[ F F F A B C D E].

- 7- Develop variable length code by Shannon-Fano code for the message.

x	$X_1$	$X_2$	$X_3$	$X_4$	$X_5$	$X_6$
$P(x)$	0.04	0.01	0.5	0.24	0.11	0.11

1. Find code table , code efficiency,  $P(0)$ ,  $P(1)$ , and draw code rate tree.
2. Encode the data [  $X_4 X_3 X_4 X_3 X_1 X_1 X_2$ ]

3. Dcode the received binary [ 1011011110001010].

8-Develop the Shannon - Fano code for the following set of messages,  
 $p(x)=[0.35 \ 0.2 \ 0.15 \ 0.12 \ 0.1 \ 0.08]$  then find the code efficiency.

9- Develop the Shannon code for the following set of messages,

$p(x)=[0.3 \ 0.2 \ 0.15 \ 0.12 \ 0.1 \ 0.08 \ 0.05]$  then find:

(a) Code efficiency,

(b)  $p(0)$  at the encoder output.

10- Apply the Shannon -Fano coding and find the code efficiency

$[x] = [x_1 \ x_2 \ x_3 \ x_4 \ x_5 \ x_6 \ x_7]$

$[P] = [.4 \ .2 \ .12 \ .08 \ .08 \ .08 \ .04]$

11- A source produces the following messages  $P(x) = [0.18 \ 0.17 \ 0.14 \ 0.12 \ 0.11 \ 0.08]$  at a rate of 1500 message /sec. if these are coded using Shannon – Fano code then transmitted through a 1.5 KHZ AWGN channel. Find code efficiency and the minimum theoretical SNR required at the channel.

12- Design Huffman codes for  $A = \{a_1, a_2, \dots, a_5\}$ , having the probabilities  $\{0.2, 0.4, 0.2, 0.1, 0.1\}$ .

13- Develop the Huffman code for the following set of symbols

Symbol	A	B	C	D	E	F	G	H
Probability	0.1	0.18	0.4	0.05	0.06	0.1	0.07	0.04

14- Having the text [A A B B A A B A A A C C B C B B D A A A]

i-What is the minimum codeword using Shannon-Fano. What is code efficiency and  $P(0), P(1)$ , then draw code tree.

ii-What is the minimum codeword using Huffman, What is code efficiency and  $P(0), P(1)$ , then draw code tree.

15- A computer executes four instruction that are designed by the code word (00,01,10,11). Assuming that the instruction are used independently with probabilities  $(1/2, 1/8, 1/8, 1/4)$  calculate the percentage by which the number of bits used for the instruction may be reduced by the use of an optimum source code. Constructing Huffman code to realize the reduction.

16- A source outputs consists of nine equally likely message. Encoding the source output using both binary Shannon-fano and Huffman codes. Compute the efficiency of both of the resulting codes and compare the results.

17- A source code produces the set message of:  $p(x)=[0.5 \ 0.25 \ 0.15 \ 0.1]$ . Using Shannon method to find code efficiency and the probability of ones  $p(1)$  at the output of encoder. Find the code efficiency of fixed code for sixteen equiprobable message.

18- A source emit 3 equiprobable symbols randomly & independently.

- Find the efficiency & redundancy of a ternary Huffman code for 1 symbol / message.
- Repeat (a) for binary Huffman code.
- Repeat (a) for binary Huffman code with 2 symbol / message.

19- a discrete source produces the symbols

$$P(x) = [0.31 \ 0.17 \ 0.15 \ 0.1 \ 0.05 \ 0.07 \ 0.06 \ 0.04 \ 0.02]$$

These are encoded using ternary Shannon – fano code, then transmitted through a discrete ternary symmetric channel having a transition matrix.

$$P(y/x) = \begin{bmatrix} 0.8 & 0.1 & 0.1 \\ 0.1 & 0.8 & 0.1 \\ 0.1 & 0.1 & 0.8 \end{bmatrix}$$

Find coding efficiency , channel redundancy and information losses?

20- A source produces the symbols with prob.

$$P(x) = [0.4 \ 0.2 \ 0.12 \ 0.08 \ 0.08 \ 0.04 \ 0.04 \ 0.04]$$

These are encoded using Shannon – fano code then the source encoder o/p is transmitted through a BSC having BER of 0.1. Find.

- Coding efficiency
- Receiver entropy
- Information lost in the channel
- Channel efficiency

21- For the following message “AFBBDBCEACDFDBDDEAEF” develop huffman coding and find

- Code redundancy.
- $p(0)$
- $p(1)$

22- An information source produces a sequence of independent symbols having the following probabilities

symbol	$S_1$	$S_2$	$S_3$	$S_4$	$S_5$	$S_6$	$S_7$
probabilities	$\frac{1}{3}$	$\frac{1}{27}$	$\frac{1}{3}$	$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{27}$	$\frac{1}{27}$

Construct binary code using Huffman encoding procedure and find its efficiency.

23- A source emits an independent sequence of symbols A, B, C, D and E with the probabilities  $\frac{1}{4}, \frac{1}{8}, \frac{1}{8}, \frac{3}{16}$  and  $\frac{5}{16}$  respectively. Find the Shannon code and efficiency.



24- Consider a DMS with three symbols  $x_i$ ,  $i=1,2,3$  and their respective probabilities  $p_1=0.5$ ,  $p_2=0.3$  and  $p_3=0.2$ . Encode the source symbols using the Huffman encoding algorithm and compute the efficiency of the code suggested

25- Apply the Huffman coding procedure for the following message ensemble

$$[x] = [x_1 \quad x_2 \quad x_3 \quad x_4 \quad x_5 \quad x_6 \quad x_7]$$

$$[p] = [0.4 \quad 0.2 \quad 0.12 \quad 0.08 \quad 0.08 \quad 0.08 \quad 0.04]$$