# INFORMATION THEORY AND CODING

V C.S.

# 1 INTRODUCTION TO INFORMATION THEORY

## PREVIOUS YEARS QUESTIONS

### PART-A

**Q.1** *Consider a source x that produces five symbols with $\frac{1}{2},\frac{1}{4},\frac{1}{8},\frac{1}{16},\frac{1}{16}$ probabilities. Determine source entropy H(X).* *[R.T.U. 2017]*

**Ans.** $H(s) = \sum_{i=1}^{5} p_i \log_2 \frac{1}{p_i}$

$$= \frac{1}{2}\log_2(2) + \frac{1}{4}\log_2(4) + \frac{1}{8}\log_2(8) + \frac{1}{16}\log_2(16) + \frac{1}{16}\log_2(16)$$

$= 0.5 + 0.5 + 0.375 + 0.25 + 0.25 = 1.875$ bits/symbol

Information rate, R

$R = r_s H(s)$ bits/sec $= 1000 \times 1.875$ bits/sec

**Q.2** *A continuous signal is band limited to SkHz. The signal is quantized in 8 levels of a PCM system with the probabilities 0.25, 0.2, 0.2, 0.1, 0.1, 0.05, 0.05 and 0.05. Calculate the entropy and rate of information.*

*[Note : Read SkHz = 5 kHz].* *[R.T.U. 2016]*

**Ans.** The signal should be sampled at a frequency $5 \times 2 = 10$ kHz (Sampling theorem). Each sample is then quantized to one of the eight levels. Looking at each quantized level as a message.

We get,

$$H = -(0.25 \log 0.25 + 0.2 \log 0.2 + 0.2 \log 0.2 + 0.1 \log 0.1 + 0.1 \log 0.1 + 0.05 \log 0.05 + 0.05 \log 0.05 + 0.05 \log 0.05)$$

$= 2.74$ bits/message

As the sampling frequency is 10 kHz, the message rate = 10,000 messages/sec. Hence, the rate of information is

$R = rH = 10{,}000 \times 2.74 = 27{,}400$ bits/sec.

**Q.3** *Define Joint Entropy.* *[R.T.U. 2016]*

**Ans. Joint Entropy :** The joint entropy of two discrete random variables X and Y is merely the entropy of their pairing: (X,Y). This implies that if X and Y are independent, then their joint entropy is the sum of their individual entropies. For examples, if (X,Y) represents the position of a chess piece – X the row and Y the column, then the joint entropy of the row of the piece and the column of the piece will be the entropy of the position of the piece.

$$H(X,Y) = E_{X,Y}[-\log p(x,y)]$$

$$= -\sum_{x,y} p(x,y) \log p(x,y)$$

**Q.4** *Define Information Rate.* *[R.T.U. 2016]*

**Ans. Information Rate :** The information rate is represented by R and it is given as,

Information Rate R = rH

Here R is the information rate.

H is the entropy or average information.

**Q.5** ***A high resolution black and white TV picture consists of about $2 \times 10^6$ picture elements and 16 different brightness levels. Pictures are repeated at rate of 32 per sec. All picture elements are assumed to be independent and all levels have equal likelihood of occurance, calculate the average information conveyed by this TV picture service?***

**Ans.** Given

picture element $= 2 \times 10^6$

symbols $= 16$

repeatation rate $= 32$/sec

then $H = \log_2 M$

$= \log_2^{16}$

$= \log_2^{2^4}$, $= 4$ bit/symbols

$r = 2 \times 10^6 \times 32$

$= 64 \times 10^6$ symbols/sec

then Info rat

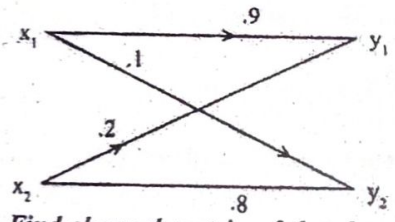
$R = H$

$= 64 \times 10^6 \times 4$ bit/sec

$= 2.56 \times 10^8$ bit/sec

- **Letters** are a more formal method of written communication usually reserved for important messages such as proposals, inquiries, agreements, and recommendations.
- **Presentations** are usually oral and usually include an audiovisual component, like copies of reports, or material prepared in Microsoft PowerPoint or Adobe Flash.
- **Telephone meetings/conference calls** allow for long-distance interaction.
- **Message boards** and **Forums** allow people to instantly post information to a centralized location.
- **Face-to-face meetings** are personal, interactive exchanges that provide the richest communication and are still the preferred method of communication in business.

**Q.7** *Consider a discrete memory less binary channel shown in fig.*




*(i) Find channel matrix of the channel.*
*(ii) Find $P(y_1)$, $P(y_2)$ when $P(x_1) = P(x_2) = .5$*
*(iii) Find $P(x_1, y_2)$ and $P(x_2, y_1)$ when $P(x_1) = P(x_2) = .5$* *[R.T.U. 2017]*

**Ans.(i)** Channel matrix

$$P(y|x) = \begin{bmatrix} P(y_1|x_1) & P(y_2|x_1) \\ P(y_1|x_2) & P(y_2|x_2) \end{bmatrix}$$

$$P(y|x) = \begin{bmatrix} .9 & .1 \\ .2 & .8 \end{bmatrix}$$

**(ii)** $P(y_1)$ and $P(y_2)$ when $P(x_1) = P(x_2) = .5$

$$[P(y)] = [P(x)]\,[P(y/x)] = [.5 \quad .5] \begin{bmatrix} .9 & .1 \\ .2 & .8 \end{bmatrix}$$

$$[P(y)] = [.55 \quad .45]$$

$[P(y_1)] = .55$ and $[P(y_2)] = .45$

**(iii)** $[P(x_1, y_2)] = [P(x)]\,[P(y/x)]$

$$= \begin{bmatrix} .5 & 0 \\ 0 & .5 \end{bmatrix} \begin{bmatrix} .9 & .1 \\ .2 & .8 \end{bmatrix}$$

$$= \begin{bmatrix} .45 & .05 \\ .10 & .40 \end{bmatrix}$$

$P(x_1, y_2) = .05$
$P(x_2, y_1) = .10.$

**Q.8** *Define following terms:*
*(i) Information*
*(ii) Mutual Information* *[R.T.U. 2016]*

**Ans.(i) Information :** The principle of improbability (which is one of the basic principles of the media world)- "if dog bites a man, it's no news, but if a man bites a dog, it's a news" –help us in this regard. The probability of a dog biting a man is quite high, so this is not a news, i.e. very little amount of information is communicated by the message "a dog bites a man". On the other hand, the probability of a man biting a dog is extremely small, so this becomes a news, i.e. quite an amount of information is communicated by the message "a man bites a dog". Thus, we see that there should be some sort of inverse relationship between the probability of an event and the amount of information associated with it. The more the probability of an event, the less is the amount of information associated with it, and vice versa. Thus,

$$I(x_j) = f\left[\frac{1}{p(x_j)}\right]$$

Where $x_j$, is an event with a probability $p(x_j)$, and the amount of information associated with it is $I(x_j)$.

**(ii) Mutual Information :** Mutual information is a quantity that measures a relationship between two random variables that are sampled simultaneously. In particular, it measures how much information is communicated, on average, in one random variable about another. Intuitively, one might ask, how much does one random variable tell me about another.

For example, suppose X represents the roll of a fair 6-sided die, and Y represents whether the roll is even (0 if even, 1 if odd). Clearly, the value of Y tells us something about the value of X and vice versa. That is, these variables share mutual information.

Mutual information measures the amount of information that can be obtained about one random variable by observing another. It is important in communication where it can be used to maximize the amount of information shared between sent and received signals. The mutual information of X relative to Y is given by:

$$I(X;Y) = E_{X,Y}\left[SI(x,y)\right]$$

$$= \sum_{x,y} p(x,y) \log \frac{p(x,y)}{p(x)p(y)}$$

**Ans.(i)** *Refer to Q.7.*

**Ans. (ii)** Mutual Info is given by

$$I(x;y)=\sum_{i=1}^{n}\sum_{j=1}^{m}P(x_i,y_j)\log_2\frac{P(x_i/y_j)}{P(x_i)} \quad \ldots (1)$$

but $$P\left(\frac{x_i}{y_j}\right)=\frac{P(x_i/y_j)}{P(y_j)} \quad \ldots (2)$$

then from eq. (2) to (1)

$$I(x;y)=\sum_{i=1}^{n}\sum_{j=1}^{m}P(x_i,y_j)\log_2\frac{P(x_i/y_j)}{P(x_i)P(x_j)}$$

$$=-\sum_{i=1}^{n}\sum_{j=1}^{m}P(x_i,y_j)\log_2\frac{P(x_j)P(y_j)}{P(x_i,y_j)}$$

$$-I(x;y)=\sum_{i=1}^{n}\sum_{j=1}^{m}P(x_i,y_j)\log_2\frac{P(x_i)P(y_j)}{P(x_i,y_j)} \quad \ldots (3)$$

but we know that

$$\sum_{k=1}^{m}P_k\log_2\left(\frac{q_k}{p_k}\right)\le 0 \quad \ldots (4)$$

then by eq. (3) & (4) we get

$-I(x;y) \le 0$

$$\boxed{I(x;y)\ge 0}$$

**Q.11** ***An analog signal is bandlimited to 2 kHz and sampled at Nyquist rate. The samples are quantized into 4 levels. Each level represents on message. The probabilities of occurrence of these 4 levels (messages) are*** $P_1=P_2=\frac{1}{8}$ ***and*** $P_3=P_4=\frac{3}{8}$.

***Calculate***

***(i) Entropy (H)***

***(ii) Information rate (R)*** *[R.T.U. 2012, 2010]*

**Ans.** Given

$$P_1=\frac{1}{8}\ ,\qquad P_3=\frac{3}{8}$$

$$P_2=\frac{1}{8}\ ,\qquad P_4=\frac{3}{8}$$

**(i)** We know

**Entropy :**

$$H=\sum_{k=1}^{N}P_k\log_2\left(\frac{1}{P_k}\right)$$

$$H=\sum_{k=1}^{4}P_k\log_2\left(\frac{1}{P_k}\right)$$



$$H=P_1\log_2\left(\frac{1}{P_1}\right)+P_2\log_2\left(\frac{1}{P_2}\right)+P_3\log_2\left(\frac{1}{P_3}\right)+P_4\log_2\left(\frac{1}{P_4}\right)$$

$$H=\frac{1}{8}\log_2(8)+\frac{1}{8}\log_2(8)+\frac{3}{8}\log_2\left(\frac{8}{3}\right)+\frac{3}{8}\log_2\left(\frac{8}{3}\right)$$

$\boxed{H=1.8 \text{ bits/message}}$ **Ans.**

**(ii) Information Rate (R) :** We know that $R = rH$

But signal is sampled at Nyquist rate. So the Nyquist rate = 2B sample/sec. and every sample generate are message signal. So message per second

$r$ = 2 (2B) message/sec

= 4B message/sec.

So $R = rH$

$$=(4B).(1.8)\frac{\text{bits}}{\text{message}}\times\frac{\text{message}}{\text{sec.}}$$

$\boxed{R = 7.2 \text{ bits/sec.}}$ **Ans.**

## PART-C

**Q.12** ***(a) Show that for a discrete binding channel:***

***(i) H(X, Y) = H(X/Y) + H(Y)*** *[R.T.U. 2018, Dec. 2013, 2013]*

***(ii) H(X, Y) = H(X) + H(Y)*** *[R.T.U. 2018]*

***(b) Prove the following properties of mutual information :***

***(i) I(X;Y) = H(X) – H(X/Y)*** *[R.T.U. 2018, 2013, 2012, 2010]*

***(ii) I(X;Y) = H(X) + H(Y) – H(X, Y)*** *[R.T.U. 2018]*

***(iii) I(X;Y) = H(X) = H(Y) (for noise free channel)*** *[R.T.U. 2018, 2013]*

**Ans.(a)(i) H(X, Y) = H(X/Y) + H(Y) :** H(X,Y) = H(X/Y) + H(Y) = H(Y/X) + H(X)

$$H(X,Y)=-\sum_{i=1}^{m}\sum_{j=1}^{m}p(X=i,Y=j)\log_2 p(X=i,Y=j)$$

$$=\sum_{i=1}^{m}\sum_{j=1}^{m}p(X=i,Y=j)$$

$$[\log_2 p(X=i)(Y=j/X=i)]$$

$$\left[\begin{matrix}\therefore p(X=i,Y=j)\\ =p(X=i)p(Y=j/X=i)\end{matrix}\right]$$

$$=-\sum_{i=1}^{m}\sum_{j=1}^{m}p(X=i,Y=j)$$

$$\left[\log_2 p(X=i)+\log_2 p(Y=j/X=i)\right]$$

$$[\because \log XY=\log X+\log Y]$$

$$=-\sum_{i=1}^{m}\sum_{j=1}^{m}p(X=i,Y=j)\log_2 p(X=i)$$

$$-\sum_{i=1}^{m}\sum_{j=1}^{m}p(X=i,Y=j)\log_2 p(Y=j/X=i)$$

$$=\sum_{i=1}^{m}\log_2\frac{(X=i)}{(Y=j)}\sum_{j=1}^{m}p(Y=j/X=i)$$

$$-\sum_{i=1}^{m}\sum_{j=1}^{m}p(X=i,Y=j)\log_2 p(Y=j/X=i)$$

$$\left[\because\sum_{j=1}^{m}p(Y=i/X=i)=1\right]$$

$$\left[-\sum_{i=1}^{m}p(X=i)\log_2 p(X=i)\right]$$

$$+\left[-\sum_{i=1}^{m}\sum_{j=1}^{m}p(X=i,Y=j)\log_2 p(Y=j/X=i)\right]$$

$$=H(X)+H(Y/X)$$

Now

$$H(X,Y)=H(Y)+H(X/Y)$$

$$=-\sum_{i=1}^{m}\sum_{j=1}^{m}p(X=i,Y=j)\log_2 p(X=i,Y=j)$$

$$=-\sum_{i=1}^{m}\sum_{j=1}^{m}p(X=i,Y=j)[\log_2 p(Y=j)\,p(X=i/Y=j)]$$

$$[\because p(X=i,Y=j=p(Y=i)p(X=i/Y=j)]$$

$$=-\sum_{i=1}^{m}\sum_{j=1}^{m}p(X=i,Y=j)\log_2 p(Y=j)-\sum_{i=1}^{m}\sum_{j=1}^{m}p(X=i,Y=j)\log_2 p(X=i/Y=j)$$

$$=-\sum_{i=1}^{m}\sum_{j=1}^{m}p(Y=j)\,p(X=i/Y=j)+H(X/Y)\log_2 p(Y=j)$$

$$=\sum_{j=1}^{m}p(Y=j)\log_2 p(Y=j)+H(X/Y)\sum_{i=1}^{m}P(X=i/Y=j)$$

$$\left[\because\sum_{i=1}^{m}p(X=i/Y=j)=1\right]$$

$$=\sum_{j=1}^{m}p(Y=j)\log_2 p(Y=j)+H(X/Y)$$

$$H(X,Y)=H(Y)+H(X/Y)$$

**(ii)** For a very noisy channel (independent), no relation can be established between transmission and receiver, these being independent:

$$\begin{cases} p_{i/j} = p_i \\ q_{j/i} = q_i \end{cases} \quad \ldots (1)$$

It follows that:

$$\begin{cases} H(X/Y) = H(X) \\ H(Y/X) = H(Y) \end{cases} \quad \ldots (2)$$

Also, we know that

$$H(XY) = H(Y) + H(X/Y) \quad \ldots (3)$$
$$I(X;Y) = H(X) - H(X/Y) \quad \ldots (4)$$
$$I(X;Y) = H(Y) - H(Y/X) \quad \ldots (5)$$

From (2), (3), (4), (5) we get

$$H(X, Y) = H(X) + H(Y)$$
$$I(X;Y) = 0$$

**Ans.(b)(i)** Here $H(X/Y)$ is the conditional entropy and it is given as,

$$H(X/Y) = \sum_{i=1}^{n}\sum_{j=1}^{m} P(x_i, y_j) \log_2 \frac{1}{P(x_i/y_j)} \quad \ldots(1)$$

$H(X/Y)$ is the information or uncertainty in X after Y is received. In other words $H(X/Y)$ is the information lost in the noisy channel. It is the average conditional self information.

Consider the equation

$$I(X;Y) = \sum_{i=1}^{n}\sum_{j=1}^{m} P(x_i, y_j) \log_2 \frac{P(x_i/y_j)}{P(x_i)}$$

Let us write the above equation as,

$$I(X;Y) = \sum_{i=1}^{n}\sum_{j=1}^{m} P(x_i, y_j) \log_2 \frac{1}{P(x_i)}$$

$$-\sum_{i=1}^{n}\sum_{j=1}^{m} P(x_i, y_j) \log_2 \frac{1}{P(x_i/y_j)}$$

From equation (1), above equation can be written as,

$$I(X;Y) = \sum_{i=1}^{n}\sum_{j=1}^{m} P(x_i, y_j) \log_2 \frac{1}{P(x_i)} - H(X/Y) \quad ..(2)$$

Here let us use the standard probability relation which is given as follows:

$$\sum_{j=1}^{m} P(x_i, y_j) = P(x_i)$$

Hence equation (2) will be,

$$I(X;Y) = \sum_{i=1}^{n} P(x_i) \log_2 \frac{1}{P(x_i)} - H(X/Y)$$

First term of the above equation represents entropy. *i.e.*,

$$H(X) = \sum_{i=1}^{n} P(x_i) \log_2 \frac{1}{P(x_i)} \quad \ldots(3)$$

Since above equation becomes

$$\boxed{I(X;Y) = H(X) - H\left(\frac{X}{Y}\right)}$$

**(ii)** From $H(X, Y) = H(X/Y) + H(Y)$ we know that

$$\therefore \quad H(X/Y) = H(X, Y) - H(Y) \quad \ldots(1)$$

Mutual information is given by

$I(X;Y) = H(X) - H(X/Y)$ *i.e*

Putting for $H(X/Y)$ in above equation from equation (1)

$$I(X;Y) = H(X) + H(Y) - H(X, Y) \quad \ldots(2)$$

Thus the required relation is proved.

**(iii)** In the case of a noiseless channel, *i.e.* no interference or perturbation, the structure of the noise matrix is:

$$P(Y/X) = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & & & \\ 0 & 0 & \cdots & 0 \end{bmatrix} \quad \ldots (1)$$

Having only 0 and 1 as elements; when we transmit the symbol $x_i$ we know with certainty the received symbol. As a result:

$$\begin{cases} H(X/Y) = 0 \\ H(Y/X) = 0 \end{cases} \quad \ldots (2)$$

We also know from $I(X:Y) = H(X) - H(X/Y)$ ... (3)

From (2) and (3) we obtain :

$$I(X;Y) = H(X) = H(Y)$$

---

Q.13 *State and prove source coding theorem.*

*[R.T.U. 2016]*

*OR*

***What is source coding theorem ? State its utility.***

*[R.T.U. 2018]*

---

**Ans. Source Coding Theorem :** Shanon's source coding theorem gives the range of the average code length $\bar{L}$ for a uniquely and instantaneously decodable source code.

The minimum value of $\bar{L}$ lies within the range

$$H(m) \le \bar{L} < H(m) + \epsilon$$

Where $\epsilon$ is a small positive quantity.

**Q.17** ***(a) Consider a DMS with the alphabet $(S_0, S_1, S_2)$ with probabilities $P_0 = \frac{1}{2}$, $P_1 = \frac{1}{4}$, $P_2 = \frac{1}{2}$. Find out the entropy of the original source and second order extension entropy?***

***(b) Two binary channels are connected in cascade as shown in fig.***

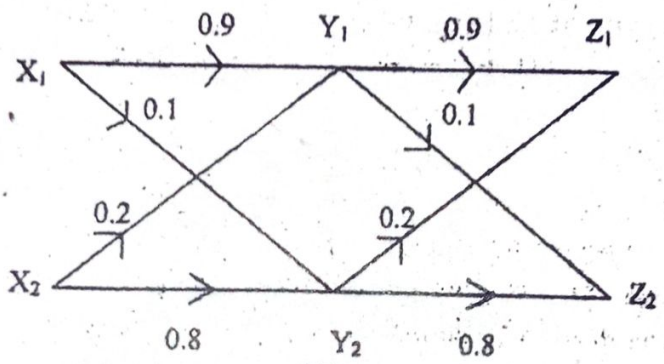



*Fig.*

***(i) Find overall channel matrix and equivalent channel diagram.***

***(ii) Find $P(Z_1)$ and $P(Z_2)$ when $P(X_1) = P(X_2) = 0.5$***

***[R.T.U. 2013]***

---

**Ans.(a)** Given $P_0 = \frac{1}{2}$, $P_1 = \frac{1}{4}$, $P_2 = \frac{1}{2}$

$\therefore$ Entropy of original source $H(\alpha)$

$$H(\alpha) = \sum_{i=0}^{2} P_i \log_2 \frac{1}{P_i}$$

$$= \frac{1}{4}\log_2(4) + \frac{1}{2}\log_2(2) + \frac{1}{2}\log_2(2) = \frac{3}{2}\text{ bits}$$

Now consider second order extension source. Source alphabet $\alpha = (S_0, S_1, S_2)$ consists of three symbols.

[k = no of symbols = 3]

Source alphabet of second order extension source consists of nine symbols

$$k^2 = 3^2 = 9$$

These symbols are

Symbols $\sigma_0\ \sigma_1\ \sigma_2\ \sigma_3\ \sigma_4\ \sigma_5\ \sigma_6\ \sigma_7\ \sigma_8$

$\alpha^2 = (S_0S_1, S_0S_0, S_0S_2, S_1S_0, S_1S_1, S_1S_2, S_2S_0, S_2S_1, S_2S_2)$

Probabilities of these in blocks that consists of 2 symbols are.

$$P(\sigma_i) = \left(\frac{1}{16}, \frac{1}{16}, \frac{1}{8}, \frac{1}{16}, \frac{1}{16}, \frac{1}{8}, \frac{1}{8}, \frac{1}{8}, \frac{1}{4}\right)$$

Entropy of second order extension

$$H(\alpha^2) = \sum_{i=0}^{8} P(\sigma i) \log_2 \left(\frac{1}{P}\right)(\sigma i)$$

$$H(\alpha^2) = \frac{1}{16}\log_2(16) + \frac{1}{16}\log_2(16) + \frac{1}{8}\log_2(8) + \frac{1}{16}\log_2(16) + \frac{1}{16}\log_2(16) + \frac{1}{8}\log_2(8) + \frac{1}{8}\log_2(8) + \frac{1}{8}\log_2(8) + \frac{1}{8}\log_2(8) + \frac{1}{4}\log_2(4)$$

$= 3$ bits

Cross check the ans. by formula

$$H(\alpha^2) = 2H(\alpha)$$

$$3 = 2 \times \frac{3}{2}$$

**Ans. (b)(i)** $[P(Y)] = [P(X)]\, P[P(Y|X)]$ ...(1)

$[P(Z)] = [P(Y)][P(Z|Y)]$ ...(2)

from equation (1) and (2)

$$= [P(X)][P(Y|X)][P(Z|Y)]$$

$$P(Z|X) = [P(Y|X)][P(Z|Y)]$$

$$= \begin{bmatrix} 0.9 & 0.1 \\ 0.2 & 0.8 \end{bmatrix}\begin{bmatrix} 0.9 & 0.1 \\ 0.2 & 0.8 \end{bmatrix}$$

$$= \begin{bmatrix} 0.83 & 0.17 \\ 0.34 & 0.66 \end{bmatrix}$$

$\therefore$ Resultant's Matrix Diagram

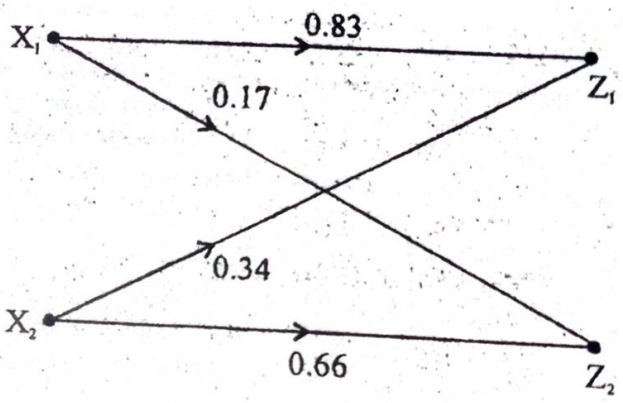



*Fig.*

(ii) $[P(Z)] = [P(X)][P(Z|X)]$

$$= [0.5 \quad 0.5]\begin{bmatrix} 0.83 & 0.17 \\ 0.34 & 0.66 \end{bmatrix}$$

$$= [0.585 \quad 0.415]$$

$P(Z_1) = 0.585$

$P(Z_2) = 0.415$

❑❑❑

# SOURCE CODING SCHEMES FOR DATA COMPACTION 2

## PREVIOUS YEARS QUESTIONS

### PART-A

**Q.1** *Consider a source S = [$S_1$, $S_2$] with probabilities ¾ and ¼ respectively. Obtain Shannon-fono code for source S, its 2nd and 3rd extensions. Calculate efficiency for each case.* *[R.T.U. 2018]*

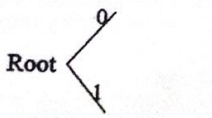
**Ans.** We can determine

$$l_1 = \log 21/S_1 = -0.41$$

$$l_2 = \log 21/S_2 = -2$$

and $l_{max} = -0.41$

Construct a binary tree of depth 1.




The source codewords are

$x_1$ : 0 0

**Q.2** *Apply the Shannon – Fano coding and find code efficiency.*

$[x_i]$ = $[x_1\ x_2\ x_3\ x_4\ x_5\ x_6]$.

$[P(x_i)]$ = [0.30 0.25 0.20 0.12 0.08 0.05]

*[R.T.U. 2016]*

**Ans. Shannon – Fano coding**

| $[x_i]$ | $[P(x_i)]$ | 1st Group | 2nd Group | 3rd Group | Codeword | $L_i$ |
|---|---|---|---|---|---|---|
| $x_1$ | 0.30 | 0 | 0 | | 00 | 2 |
| $x_2$ | 0.25 | 0 | 1 | | 01 | 2 |
| $x_3$ | 0.20 | 1 | 0 | 0 | 100 | 3 |
| $x_4$ | 0.12 | 1 | 0 | 1 | 101 | 3 |
| $x_5$ | 0.08 | 1 | 1 | 0 | 110 | 3 |
| $x_6$ | 0.05 | 1 | 1 | 1 | 111 | 3 |

Entropy

$$H(\zeta) = \sum_{k=0}^{5} P_k \log_2\left(\frac{1}{P_k}\right)$$

$$= 2.36 \text{ bits}$$

Average codeword length,

$$T = \sum_{k=0}^{5} P_k l_k$$

$$= 2.45 \text{ bits}$$

Code efficiency,

$$\eta = \frac{H(\zeta)}{T} \times 100$$

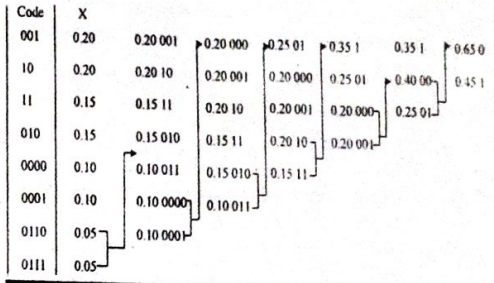
$$= \frac{2.36}{2.45} \times 100$$

$$\boxed{\eta = 96.33\%}$$

**Q.3** *Consider a DMS with source probabilities {20, 20, 15, 15, 10, 10, 05, 05} Determine the Huffman code for this source.* *[R.T.U. 2012]*



**Ans. Huffman Code**




**Q.4** *State Kraft Inequality Theorem.*

**Ans. Kraft Inequality Theorem :** A necessary and sufficient condition for the existence of a binary code with codewords having lenghts $n_1 \le n_2 \le \ldots n_L$ that satisfy the prefix condition is

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

**Q.5** *Define Prefix Code.*

**Ans. Prefix Code :** This is variable length coding algorithm. It assigns binary digits to the messages as per their probabilities of occurrence. Prefix of the codeword means any sequence which is initial part of the codeword. In prefix code, no codeword is the prefix of any other codeword.

### PART-B

**Q.6** *Write short notes on :*
*(i) Noise Free channel*
*(ii) Shannon's theorem*
*[R.T.U 2018, Dec. 2013]*

**Ans. (i) Free Channel :** The channel is called noise free or noiseless if it is both lossless and deterministic – *i.e.*

The channel matrix has only one element in each row and on each column.

It is show in following Fig.

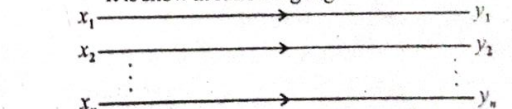



*Fig. : Noiseless or noise free channel*

It is channel matrix coil be like this

$$P(y/x) = \begin{array}{c} \\ x_1 \\ x_2 \\ x_3 \\ x_n \end{array} \begin{array}{c} \begin{array}{cccc} y_1 & y_2 & y_3 & \cdots y_n \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & \cdots \\ 0 & 1 & 0 & \cdots \\ 0 & 0 & 1 & \cdots \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix} \end{array}$$

The binary symmetric channels are also comes under noiseless channels.

$$[P(y/x)] = \begin{bmatrix} 1-P & P \\ P & 1-P \end{bmatrix}$$

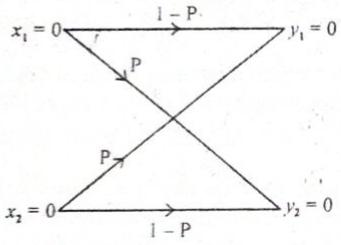



*Fig. Binary symmetric channel*

**(ii) Shannon's Theorem :** The Shannon-Hartley theorem, is of fundamental importance and has two important implications for communication systems engineers.

1. First, it gives us the upper limit that can be reached in the way of reliable data transmission rate over Gaussian channels. Thus, a system designer always tries to optimize his system to have a data rate as close to C given in equation (1), as possible with an acceptable error rate.

2. The second implication of the Shannon-Hartley theorem has to do with the exchange of signal-to-noise ratio for bandwidth. To illustrate this aspect of the theorem, suppose that we want to transmit data at a rate of 10,000 bits/sec. Over a channel having a bandwidth B = 3000 Hz. To transmit data at a rate of 10,000 bits/sec, we need a channel with a capacity of at least 10,000 bits/sec. If the channel capacity is less than the data rate, then errorless transmission is not possible. So, with C = 10,000 bits/sec, we can obtain the (S/N) requirement of the channel as

$$(S/N) = 2^{(C/B)} - 1 = 2^{3.333} - 1 \approx 9$$

The Shannon-Hartley theorem indicates that a noiseless channel has an infinite capacity. However, when noise is present the channel capacity does not approach infinity as the bandwidth is increased because the noise power increases as the bandwidth increases. The channel capacity reaches a finite upper limit with increasing bandwidth if the signal power is fixed. We can calculate



this limit as follows. With $N = \eta B$, where $\eta/2$ is the noise power spectral density, we have

$$C = B\log_2\left(1+\frac{S}{\eta B}\right) = \left(\frac{S}{\eta}\right)\left(\frac{\eta B}{S}\right)\log_2\left(1+\frac{S}{\eta B}\right) \quad ...(1)$$

$$= \frac{S}{\eta}\log_2\left(1+\frac{S}{\eta B}\right)$$

Recalling that $\lim x\to 0\,(1+x)^{1/x} = e$ and letting $x = S/\eta B$ we have

$$\lim_{B\to\infty} C = \frac{S}{\eta}\log_2 = 1.44\frac{S}{\eta}$$

A communication system capable of transmitting information at a rate of $B \log_2 (1 + S/N)$ is called an ideal system. The ideal signalling scheme using noise like signals can convey information at a rate approaching the channel capacity only when $T \to \infty$. Only in the limiting case we have all the conditions satisfied. Under this limiting condition, the ideal system has the following characteristics:

(a) The information rate $\to B \log_2 (1 + S/N)$.

(b) The error rate $\to 0$.

(c) The transmitted and received signals have the characteristics of band limited Gaussian white noise.

(d) As $T \to \infty$, the number of signals $M \to \infty$ and coding delay also tends to $\infty$.

**Q.7** ***Explain Shannon Theorem and Shannon Limit.*** *[R.T.U. 2016]*

**Ans. Shannon's Theorem :** *Refer to Q.6.*

**Shannon Limit :** There exists a limiting value of $E_b/N_0$ below which there can be no error-free communication at any information rate. Using the identity

$$\lim_{x\to 0}(1+x)^{1/x} = e$$

We can calculate the limiting value of $E_b/N_0$ as follows:

Let $$x = \frac{E_b}{N_0}\left(\frac{C}{W}\right)$$

Then, $$\frac{C}{W} = x\log_2(1+x)^{1/x}$$

and $$1 = \frac{E_b}{N_0}\log_2(1+x)^{1/x}$$

In the limit, as $C/W \to 0$, we get

$$\frac{E_b}{N_0} = \frac{1}{\log_2 e} = 0.693$$

Or, in decibels,

$$\frac{E_b}{N_0} = -1.6\text{dB}$$

This value of $E_b/N_0$ is called the Shannon limit.

**Q.8** ***Derive the mathematical expression for channel capacity to transmit the information through it if the channel capacity is :***

$$C = B\log_2\left(1+\frac{S}{N}\right)^{b}/_{s}$$

***[Note : Read B = ω]*** ***[R.T.U. 2016]***

**Ans.** The noise characteristics of channels encountered in practice is generally Gaussian (channels with Gaussian noise characteristic are known as Gaussian channels.) Moreover, the result obtained for a Gaussian channel often provide a lower bound on the performance of a system with the Gaussian channel. Thus, if a particular encoder-decoder is used with a Gaussian channel giving an error probability Pe, then, with a non-Gaussian channel, another encoder –decoder can be designed for which the error probability will be less than Pe. Hence, the study of a Gaussian channel is very important.

For a Gaussian channel,

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}}e^{-x^2/2\sigma^2} \quad ...(1)$$

Hence,

$$H(x) = -\int_{-\infty}^{\infty} p(x)\log p(x)\,dx$$

But

$$-\log p(x) = \log\sqrt{2\pi\sigma^2} + \log e^{x^2/2\sigma^2}$$

From Eq. (1)

Hence,

$$H(x) = \int_{-\infty}^{\infty} p(x)\log\sqrt{2\pi\sigma^2}\,dx + \int_{-\infty}^{\infty} p(x)\log e^{x^2/2\sigma^2}$$

This may be evaluated to yield

$$H(x) = \log\sqrt{2\pi e\sigma^2} \text{ bits/message} \quad ...(2)$$

or $\bar{N} = 2.75$ letters/message

$$H(X) = -\sum_{k=1}^{8} P_k \log P_k$$

$$= -\left[\frac{1}{4}\log\frac{1}{4} + \frac{1}{8}\log\frac{1}{8} + \frac{1}{16}\log\frac{1}{16} + \frac{1}{16}\log\frac{1}{16} + \frac{1}{16}\log\frac{1}{16} + \frac{1}{4}\log\frac{1}{4} + \frac{1}{16}\log\frac{1}{16} + \frac{1}{8}\log\frac{1}{8}\right]$$

= 2.75 bits/message

$\log M = \log 2 = 1$ bits/letter

Hence $$\eta = \frac{H(X)}{\bar{N}\log M}\eta = \frac{2.75}{2.75 \times 1} = 100\%$$

**Q.10** ***The given string S = 0 0 1 2 1 2 1 2 1 0 2 1 0 1 2 1 0 1 2 2 1 0 1 1. Find the encoding and decoding process.*** Lempel Ziv. *[R.T.U. 2013]*

**Ans.** The string S = 001212121021012101221011 is to be encoded. Fig. 1 shows the encoding process.

| 0 | 0 | 1 | 2 | 1 | 2 | 1 | 2 | 1 | 0 | 2 | 1 | 0 | 1 | 2 | 1 | 2 | 1 | 0 | 2 | 1 | 0 | 1 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ⇓ | ⇓ | | ⇓ | ⇓ | ⇓ | | ⇓ | | | | ⇓ | | | | | | ⇓ | | | | | | |

| # | Entry | Phrase | Output | (ternary) |
|---|---|---|---|---|
| 1 | 0 | 0 | 0 0 | (0 0) |
| 2 | 1+1 | 01 | 1 1 | (1 1) |
| 3 | 2 | 2 | 0 2 | (0 2) |
| 4 | 1 | 1 | 0 1 | (00 1) |
| 5 | 3 + 1 | 21 | 3 1 | (10 1) |
| 6 | 5 + 0 | 210 | 5 0 | (12 0) |
| 7 | 6 + 1 | 2101 | 6 1 | (20 1) |
| 8 | 7 + 2 | 21012 | 7 2 | (21 2) |
| 9 | 7 + 1 | 21011 | 7 1 | (21 1) |

*Fig. 1 : Encoding*

(i) In the first step, 0 is encountered and added to the dictionary. The output is 00 because is no match (index 0) and the first non-matching character is 0. The encoder then proceeds to the second position, encountering 0, which is already in the dictionary. The following 1 is not yet in the dictionary, so the encoder adds the string 01 to the dictionary (a reference to the first entry plus the symbol 1) and outputs this pair. The next steps follow the same scheme until the end of the input is reached.

(ii) The decoding process is shown in fig. 2. The decoder receives the reference 00, with the index 0 indicating that a previously unknown symbol (0) needs to be added to the dictionary and to the uncompressed data. The next codeword is 11 resulting in the entry 01 ( a reference to entry 1 plus the symbol 1) being added to the dictionary and the string 01 appended to the uncompressed data. The decoder continues this way until all codewords have been decoded.

| 0 | 0 | 1 | 2 | 1 | 2 | 1 | 2 | 1 | 0 | 2 | 1 | 0 | 1 | 2 | 1 | 0 | 1 | 2 | 2 | 1 | 0 | 1 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ⇓ | ⇓ | | ⇓ | ⇓ | ⇓ | | ⇓ | | | | ⇓ | | | | | | ⇓ | | | | | | |

| # | Entry | Phrase | Output | (ternary) |
|---|---|---|---|---|
| 1 | 0 | 0 | 0 0 | (0 0) |
| 2 | 1+1 | 01 | 1 1 | (1 1) |
| 3 | 2 | 2 | 0 2 | (0 2) |
| 4 | 1 | 1 | 0 1 | (00 1) |
| 5 | 3 + 1 | 21 | 3 1 | (10 1) |
| 6 | 5 + 0 | 210 | 5 0 | (12 0) |
| 7 | 6 + 1 | 2101 | 6 1 | (20 1) |
| 8 | 7 + 2 | 21012 | 7 2 | (21 2) |
| 9 | 7 + 1 | 21011 | 7 1 | (21 1) |

*Fig. 2 : Decoding*

**Q.11** ***Construct Huffman's code to the following set of messages. Also find the efficiency*** $p(x_1) = 0.49$, $p(x_2) = 0.14$, $p(x_3) = 0.14$, $p(x_4) = 0.07$, $p(x_5) = 0.07$, $p(x_6) = 0.04$, $p(x_7) = 0.02$, $p(x_8) = 0.02$, $p(x_9) = 0.01$. *[R.T.U. 2013]*

**Ans.**

| Sym-bol | Proba-bility | SS - I | SS - II | SS - III | SS - IV | SS - V | SS - VI | SS - VII | Code word | Len-gth |
|---|---|---|---|---|---|---|---|---|---|---|
| $x_1$ | 0.49 | 0.49 | 0.49 | 0.49 | 0.49 | 0.49 | 0.49 | 0.51 | 0 | 1 |
| $x_2$ | 0.14 | 0.14 | 0.14 | 0.14 | 0.14 | 0.23 | 0.28 | 0.49 | 100 | 3 |
| $x_3$ | 0.14 | 0.14 | 0.14 | 0.14 | 0.14 | 0.14 | 0.23 | | 101 | 3 |
| $x_4$ | 0.07 | 0.07 | 0.07 | 0.09 | 0.14 | 0.14 | | | 1100 | 4 |
| $x_5$ | 0.07 | 0.07 | 0.07 | 0.07 | 0.09 | | | | 1101 | 4 |
| $x_6$ | 0.04 | 0.04 | 0.05 | 0.07 | | | | | 1110 | 4 |
| $x_7$ | 0.02 | 0.03 | 0.04 | | | | | | 11110 | 5 |
| $x_8$ | 0.02 | 0.02 | | | | | | | 111110 | 6 |
| $x_9$ | 0.01 | | | | | | | | 111111 | 6 |

*Fig.*

Average code word length

$$L = \sum_{k=1}^{K} P_k L_k$$

$$L = (0.49 \times 1) + (0.14 \times 3) + (0.14 \times 3) + (0.07 \times 4) + (0.07 \times 4) + (0.04 \times 4) + (0.02 \times 5) + (0.02 \times 6) + (0.01 \times 6)$$

L = 2.33 bits/symbol

$$H(X) = -\sum_{k=1}^{9} p_k \log_2 p_k$$

$$= -3.32\,[0.49 \log_{10}0.49 + 0.14 \log_{10}0.14 + 0.14 \log_{10}0.14 + 0.07 \log_{10}0.07 + 0.07 \log_{10}0.07 + 0.04 \log_{10}0.04 + 0.02 \log_{10}0.02 + 0.02 \log_{10}0.02 + 0.01 \log_{10}0.01]$$



H(X) = 2.3122 bits/symbol

$$\eta = \frac{H(X)}{L} = \frac{2.3122}{2.33} = 0.992$$

∴ Loading efficiency = 99.2%

∴ R = 1 − η

= 1 − 0.992

∴ R = 0.00763

## PART-C

**Q.12** ***Explain Huffman coding with help of suitable example.*** *[R.T.U. 2018]*

**Ans. Huffman Coding Algorithm with Example**

Huffman coding algorithm was invented by David Huffman in 1952. It is an algorithm which works with integer length codes. A Huffman tree represents Huffman codes for the character that might appear in a text file. Unlike to ASCII or Unicode, Huffman code uses different number of bits to encode letters. If the number of occurrence of any character is more, we use fewer numbers of bits. Huffman coding is a method for the construction of minimum redundancy codes.

Huffman tree can be achieved by using compression technique. Data compression has lot of advantages such as it minimizes cost, time, bandwidth, storage space for transmitting data from one place to another.

In regular text file each character would take up 1 byte (8 bits) i.e. there are 16 characters (including white spaces and punctuations) which normally take up 16 bytes. In the ASCII code there are 256 characters and this leads to the use of 8 bits to represent each character but in any test file we do not have to use all 256 characters. For example, in any English language text, generally the character 'e' appears more than the character 'z'. To achieve compression, we can often use a shorter bit string to represent more frequently occurring characters. We do not have to represent all 256 characters, unless they all appear in the document. The data encoding schemes are broadly categorized in two categories.

**Fixed Length Encoding Scheme**

Fixed length encoding scheme compresses our data by packing it into the minimum number of bits i.e. needed to represent all possible values of our data. The fixed length code can store maximum 224,000 bits data.

$$\text{Efficiency} = \frac{H(z)}{L(z)} = \frac{2.15}{2.28} = 0.943$$

= 94.3%

**(iii) Huffman code :**

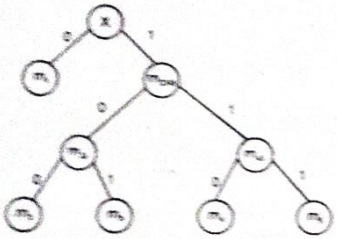


Fig.

| Message | Huffman code |
|---|---|
| $m_1$ | 0 |
| $m_2$ | 100 |
| $m_3$ | 101 |
| $m_4$ | 110 |
| $m_5$ | 111 |

$$H(z) = \sum_{i=1}^{5} p_i \log_2\left(\frac{1}{p_i}\right)$$

= 2.15

$$L(z) = \sum_{i=1}^{5} L(m_i) \cdot P(m_i)$$

$$= 1 \times 0.4 + 3 \times 0.19 + 3 \times 0.16 + 3 \times 0.15 + 3 \times 0.1$$

= 2.2

$$\text{Efficiency} = \frac{H(z)}{L(z)} = \frac{2.15}{2.2} = 0.977$$

= 97.7%

Efficiency of Huffman code is higher for the given set of messages.

**Q.15 *Explain prefix code with the help of an example and define its efficiency.*** *[R.T.U. 2016]*

**Ans. Prefix Code :** *Refer to Q.5.*

Table shows four source symbols, their probabilities and the codewords and the codewords assigned to them by prefix coding.

*Table : Prefix code*

| Source symbol | Probability of occurrence | Prefix code |
|---|---|---|
| $s_0$ | 0.5 | 0 ← Codeword |
| $s_1$ | 0.25 | 1 0 ← Codeword (1 = prefix) |
| $s_2$ | 0.125 | 11 0 ← Codeword (11 = prefix) |
| $s_3$ | 0.125 | 11 1 ← Codeword (11 = prefix) |

In the table, observe that message $s_0$ has codeword '0'. Message $s_1$ has prefix of 1 and codeword of 0. Observe that no codeword is prefix of other codewords.

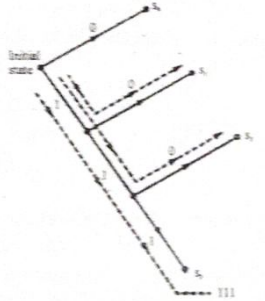



*Fig. 1 : Decision tree for decoding prefix code*

Fig.1 shows the decision tree for decoding the prefix code of table.

As shown in fig.1 the tree has initial state. The decoder always starts from initial state. If the first received bit is '0' then the decoder decides in favour of message $s_0$ and goes to initial state. If the next received bit is 1, then the decoder goes one step down and waits for next bit. If next bit is '0' then decoder decides in favour of message $s_1$(10) and goes to initial stage. This is how the decoder of prefix code works.

**Prefix code efficiency :** Prefix code is a type of code that let you decode the encoded text without special marker. For example, the map {a=0, b=10, c=11} is a prefix code as no marker is needed for decryption of any string. If you have '000101011' then it is clear when one list of bits for character ends and another begins you translate it directly to 'aaabbc'. A counter example, the map {a=0, b=1, c=11} is not prefix code as the string '111' can be translated to 'bbb' or 'bc' i.e., we need some separator marker between list of bits in this code.



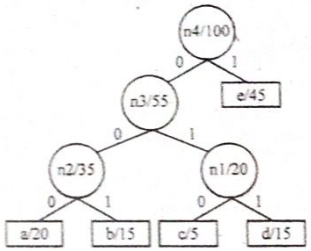
**Coding efficiency :** If the coding is represented by a map from a character c to a list of bits then we will define length (c) be the number of bits representing the character c and define frequency(c) to be the frequency that character c as appear in the text.

The code efficiency is calculated by

$$\sum_{c \in \text{Alphabet}} \text{length}(c) \times \text{frequency}(c)$$

i.e., the weight average of encoding lengths according to their frequencies.

Every Prefix code can also be represented as a binary tree where each edge is marked as '0' or '1' and the leaf are marked with a character so the list of edges to a leaf represent the character's code. Here is a picture that show this idea:




{a = 000, b = 001, c = 010, d = 011, e = 1}

*Fig. 2*

The depth of a leaf is actually the size of a character prefix code and so this representation provide an alternative to the efficient of the code in the language of trees i.e.,

$$B(T) = \sum_{c \in \text{Alphabet}} \text{depth}(c) \times \text{frequency}(c)$$

where depth (c) is the depth of the leaf c in the tree.

**Q.16 *(i) State and prove Kraft Inequality theorem.***

***(ii) Consider a DMS with source probabilities {35, 25, 20, 15, 05}***

***(a) Determine the Shannon fano code for this source.***

***(b) Determine the average length $\bar{R}$ of the codewords.***

***(c) What is the efficiency $\eta$ of the code ?*** *[R.T.U. 2012]*

**Ans. (i) Kraft Inequality Theorem : Kraft Inequality Theorem :** A necessary and sufficient condition for the existence of a binary code with codewords having lenghts $n_1 \le n_2 \le \ldots n_L$ that satisfy the prefix condition is

$$\sum_{k=1}^{L} 2^{-n_k} \le 1 \qquad \ldots (1)$$

**Proof :** First we prove the sufficient condition. Consider a binary tree of order (depth) $n = n_L$. This tree has $2^n$ terminal nodes as depicted in Fig. Let us select any code of order $n_1$ as the first codeword $c_1$. Since no codeword is the prefix of any other codeword (the prefix condition), the choice eliminates $2^{n-n_1}$ terminal codes. This process continues until the last codeword is assigned at the terminal node $n = n_L$. Consider the node of order $j < L$. The fraction of number of terminal nodes eliminated is

$$\sum_{k=1}^{j} 2 - n_k < \sum_{k=1}^{j} 2^{-n_k} \le 1 \qquad \ldots (2)$$

Thus, we have been able to construct a prefix code that is embedded in the full tree of $n_L$ nodes. The nodes that are eliminated are depicted by the dotted arrow lines leading on to them in fig.

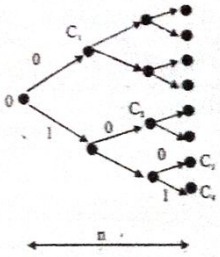



*Fig : A binary tree of order $n_L$*

We now prove the necessary condition. We observe that in the code tree of the order $n = n_L$, the number of terminal nodes eliminated from the total number of $2^n$ terminal nodes is

$$\sum_{k=1}^{L} 2^{n-n_k} \le 2^n. \qquad \ldots (3)$$

This leads to $\sum_{k=1}^{L} 2^{-n_k} \le 1.$ ... (4)

We can easily extend this proof for prefix codes over an alphabet of size M. For the proof we will have to consider an array tree instead of a binary tree. The inequality in this case would become

$$\sum_{k=1}^{L} M^{-n_k} \le 1. \qquad \ldots (5)$$

**Ans. (ii) (a)** By Shannon Fano Code

| P(x) | Code | η |
|---|---|---|
| 0.35 | 0 0 | 2 |
| 0.25 | 0 1 | 2 |
| 0.20 | 1 0 | 2 |
| 0.15 | 1 1 0 | 3 |
| 0.05 | 1 1 1 | 3 |

**(b)** $\bar{R} = \Sigma P_i(x_i).n$

$= 0.35 \times 2 + 0.25 \times 2 + 0.20 \times 2 + 0.15 \times 3 + 0.05 \times 3$

$= 0.70 + 0.50 + 0.40 + 0.45 + 0.15 = 2.20$

**(c)** $\eta = \dfrac{H(X)}{\bar{R}}$

$$H(X) = \sum_{i=1}^{5} P_i(x_i).\log\left[\frac{1}{P(x_i)}\right] = -[0.35 \log (0.35)$$

$+ 0.25 \log (0.25) + 0.20 \log (0.20) + 0.15 \log (0.15)$
$+ 0.05 \log (0.05)]$

$= + [0.15 + 0.15 + 0.13 + 0.12 + 0.06] = 2.02$

$$\eta = \frac{2.02}{2.20} = 92.1\%$$

$\boxed{\eta = 92.1\%}$

---

**Q.17** ***(a) A discrete memory less source has five symbols X1, X2, X3, X4 and X5 with probabilities 0.4, 0.19, 0.16, 0.15 and 0.1 respectively attached to every symbol. Construct Shannon-Fano code for the source and calculate the code efficiency.***

***(b) A channel has the following channel matrix***

$$\left[P\left(\frac{y}{x}\right) = \begin{bmatrix} 1-p & p & 0 \\ 0 & p & 1-p \end{bmatrix}\right]$$

***(i) Draw the channel diagram***

***(ii) If the source has equally likely outputs, compare the probabilities associated with the channel output for p = 0.4 and calculate***

**$H(x), H(y)$ and $H\left(\frac{y}{x}\right)$.** ***[R.T.U. 2011, 2010]***

---

**Ans. (a)** ***Table : To obtain Shannon-Fano code***

| Message | Probability of message | I | II | III | Code word for message | Number of bits per message *i.e.* $n_k$ |
|---|---|---|---|---|---|---|
| $x_1$ | 0.4 | 0 | | | 0 | 1 |
| $x_2$ | 0.19 | 1 | 0 | 0 | 100 | 3 |
| $x_3$ | 0.16 | 1 | 0 | 1 | 101 | 3 |
| $x_4$ | 0.15 | 1 | 1 | 0 | 110 | 3 |
| $x_5$ | 0.1 | 1 | 1 | 1 | 111 | 3 |

The entropy (H) is given as,

$$H = \sum_{k=1}^{M} p_k \log_2\left(\frac{1}{p_k}\right)$$

Here M = 5 and putting the values of probabilities in above equation,

$$H = 0.4 \log_2\left(\frac{1}{0.4}\right) + 0.19 \log_2\left(\frac{1}{0.19}\right)$$

$$+0.16 \log_2\left(\frac{1}{0.16}\right) + 0.15 \log_2\left(\frac{1}{0.15}\right) + 0.1 \log_2\left(\frac{1}{0.1}\right)$$

$= 2.1497$ bits/message

The average number of bits per message $\bar{N}$ is

$$\bar{N} = \sum_{k=1}^{L} p_k n_k$$

Here $p_k$ is the probability of $k^{th}$ message and $n_k$ are number of bits assigned to it. Putting the values in above equation.

$\bar{N} = 0.4(1) + 0.19(3) + 0.16(3) + 0.15(3) + 0.1(3)$
$= 2.2$

The code efficiency is given by equation i.e.

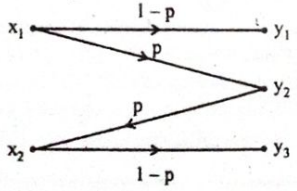
Code efficiency $\eta = \dfrac{H}{\bar{N}}$

$$= \frac{2.1497}{2.2} = 0.977$$

**Ans. (b)** Given

$$p\left(\frac{y}{x}\right) = \begin{matrix} \\ x_1 \\ x_2 \end{matrix}\begin{matrix} \begin{matrix} y_1 & y_2 & y_3 \end{matrix} \\ \begin{bmatrix} 1-p & p & 0 \\ 0 & p & 1-p \end{bmatrix} \end{matrix}$$

**(i)** Channel Diagram








**(ii)** Given that source has equal likely output. Hence

$$p(x_1) = p(x_2) = \frac{1}{2}$$

then the output probabilities are

$$\begin{bmatrix} p(y_1) \\ p(y_2) \\ p(y_3) \end{bmatrix} = [p(x_1)\, p(x_2)] \begin{bmatrix} 1-p & p & 0 \\ 0 & p & 1-p \end{bmatrix}$$

but given that p = 0.4
therefore

$$\begin{bmatrix} p(y_1) \\ p(y_2) \\ p(y_3) \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} 1-0.4 & 0.4 & 0 \\ 0 & 0.4 & 1.04 \end{bmatrix}$$

$$= \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} 0.6 & 0.4 & 0 \\ 0 & 0.4 & 0.6 \end{bmatrix}$$

$$\begin{bmatrix} p(y_1) \\ p(y_2) \\ p(y_3) \end{bmatrix} = \begin{bmatrix} 0.3 \\ 0.4 \\ 0.3 \end{bmatrix}$$

Thus

$$p(y_1) = 0.3$$
$$p(y_2) = 0.4$$
$$p(y_2) = 0.3$$

**(a)** $$H(x) = \sum_{k=1}^{2} p_k \log_2\left(\frac{1}{p_k}\right)$$

$$H(x) = p_1 \log_2\left(\frac{1}{p_1}\right) + p_2 \log_2\left(\frac{1}{p_2}\right)$$

$$H(x) = \frac{1}{2}\log_2(2) + \frac{1}{2}\log_2(2)$$

$H(x) = 1$ bit/message

**(b)** $$H(y) = \sum_{k=1}^{3} p_k \log_2\left(\frac{1}{p_k}\right)$$

$$H(y) = p_1 \log_2\left(\frac{1}{p_1}\right) + p_2 \log_2\left(\frac{1}{p_2}\right) + p_3 \log_3\left(\frac{1}{p_3}\right)$$

$$H(y) = 0.3 \log\left(\frac{1}{0.3}\right) + 0.4 \log\left(\frac{1}{0.4}\right) + 0.3 \log\left(\frac{1}{0.3}\right)$$

$$H(y) = \frac{0.313 + 0.159}{\log_2}$$

$H(y) = 1.56$ bit/message

**(c)** We know that

$$H\left(\frac{y}{x}\right) = H(x_1 y) - H(x)$$

but $$p\left(\frac{y}{x}\right) = \begin{bmatrix} 0.6 & 0.4 & 0 \\ 0. & 0.4 & 0.6 \end{bmatrix}$$

$$H\left(\frac{y}{x}\right) = 0.6 \log_2\left(\frac{1}{0.6}\right) + 0.4 \log_2\left(\frac{1}{0.4}\right) + 0.4 \log_2\left(\frac{1}{0.4}\right)$$

$$+0.6 \log_2\left(\frac{1}{0.6}\right)$$

$$H\left(\frac{y}{x}\right) = \frac{0.266 + 0.318}{0.3010} \text{ bit/message}$$

$H\left(\frac{y}{x}\right) = 1.94$ bit/message

□□□

# LINEAR BLOCK CODE 3

## PREVIOUS YEARS QUESTIONS

### PART-A

**Q.1** ***Define code word.*** *[R.T.U. 2016]*

**Ans.** The channel encoder separates or segments the incoming bit stream (the output of the source encoder) into equal length blocks of L binary digits and maps each L-bit message block into an N-bit code word where N> L and the extra N – L check bits provide the required error protection. There are $M = 2^L$ message and thus $2^L$ code words of length N bits. The channel decoder maps the received N –bit word to the most likely code word and inversely maps the N –bit code word to the corresponding L-bit message.

**Code Word :** The encoded block of N bits is called a code word. It contains message bits and redundant bits.

**Q.2** ***Define block length.*** *[R.T.U. 2016]*

**Ans. Block Length :** The number of bits N after coding is called the block length of the code.

**Q.3** ***Define code rate.*** *[R.T.U. 2016]*

**Ans. Code Rate:** The ratio of message bits (K) and the encoder output bits (N) is called code rate. Code rate is defined by 'r' i.e.,

$$r = \frac{K}{N}$$

We find that $0 < r < 1$.

**Q.4** ***Define channel data rate.*** *[R.T.U. 2016]*

**Ans. Channel Data Rate :** It is the bit rate at the output of encoder. If the bit rate at the input of encoder is $R_s$, then channel data rate will be,

Channel data rate

$$(R_0) = \frac{N}{K} R_S$$

**Q.5** ***What are content errors?***

**Ans. Content Errors :** The content errors are nothing but errors in the contents of a message i.e., a 0 may be received as 1 or vice-versa.

### PART-B

**Q.6** ***Consider a (6, 3) linear block code whose generator matrix is given by***

$$\begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$

*(a) Find the parity check matrix*
*(b) Find the minimum distance of the code.*
*(c) Draw the encoder and syndrome computation circuit.* *[R.T.U. 2017]*

**Ans.(a)**

$$\text{Generator Matrix} = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$

Standard form $G = [I_k | A]$

Parity Check Matrix $= [-A^T | I_{n-k}]$

$$H = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

**(b)** To find minimum distance, we use the property that the minimum distance of a binary linear codes is equal to the smallest number of columns of the parity check matrix that sums up to zero.

Cleary all columns of H are non-zero, and they are all distinct

So, $d \geq 3$

Moreover, we can conclude that $d = 3$

by adding first 3 columns of H :-

$$\begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Hence, minimum distance of code is 3.

**(c) Encoder circuit –**

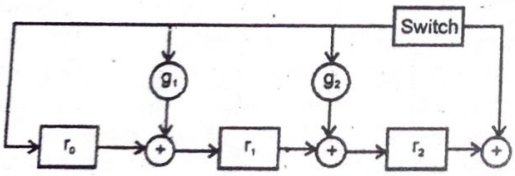



Fig.

Syndrome computation circuit

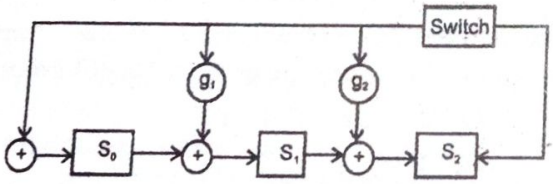



Fig.

**Q.7** ***Differentiate between systematic and non-systematic codes. Give example of each.***

*[R.T.U. 2017]*

**Ans.** In coding theory, a systematic code is any error-correcting code in which the input data is embedded in the encoded output. Conversely, in a non-systematic code the output does not contain the input symbols.

Systematic codes have the advantage that the parity data can simply be appended to the source block, and receivers do not need to recover the original source symbols if received correctly - this is useful for example if error-correction coding is combined with a hash function for quickly determining the correctness of the received source symbols, or in cases where errors occur in erasures and a received symbol is thus always correct. Furthermore, for engineering purposes such as synchronization and monitoring, it is desirable to get reasonable good estimates of the received source symbols without going through the lengthy decoding process which may be carried out at a remote site at a later time.

**Examples:**

- Checksums and hash functions, combined with the input data, can be viewed as systematic error-detecting codes.
- Linear codes are usually implemented as systematic error-correcting codes (e.g., Reed-Solomon codes in CDs).
- Convolutional codes are implemented as either systematic or non-systematic codes. Non-systematic convolutional codes can provide better performance under maximum-likelihood (Viterbi) decoding.

**Q.8** ***Given a (6,3) linear block code with the following parity check matrix H:***

$$H = \begin{bmatrix} 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}$$

***(i) Find the generator matrix G.***

***(ii) Find the code word for data bit 110.***

*[R.T.U. 2016]*

**Ans. (i) To obtain the generator matrix:**

$$H = \left[P^T : I_q\right]_{q \times n}$$

$$H = \left[\begin{array}{ccc:ccc} 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 \end{array}\right]$$

$$P^T = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Hence $$P = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

The generator matrix is given as

$$G = \left[ I_k : P_{kxq} \right]_{kxn}$$

$$\therefore \quad G = \begin{bmatrix} 1 & 0 & 0 & : & 1 & 0 & 1 \\ 0 & 1 & 0 & : & 0 & 1 & 1 \\ 0 & 0 & 1 & : & 1 & 1 & 1 \end{bmatrix}$$

**(ii) To obtain the codeword for data bit 110 :**

$$M = [110]$$

This is (6,3) code. The three check bits can be obtained by equation :

$$C = MP = [110] \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$$= [1 \oplus 0 \oplus 00 \oplus 1 \oplus 01 \oplus 1 \oplus 0]$$

$$= [110]$$

Code vector,

$$X = (m_1, m_2, m_3, c_1, c_2, c_3) = (110\ 110)$$

Code word = (110 110)

---

**Q.9** ***Define minimum distanced $d_{min}$ of Hamming code. Differentiate between Hamming distance and minimum distance. How minimum distance is related to error detection capability?***

*[R.T.U. 2016]*

---

**Ans. Hamming Distance :** The hamming distance between the two code vectors is equal to the number of elements in which they differ. For example, let X = (101) and Y = (110). The two code vectors differ in second and third bits. Therefore hamming distance between X and Y is 'two'. Hamming distance is denoted as d(X,Y) or simply 'd'. i.e.

$$d(X,Y) = d = 2$$

Thus we observe from Fig. that the hamming distance between (100) and (011) is maximum i.e. 3. This is indicated by the vector diagram also.

**Minimum Distance ($d_{min}$) :** It is the smallest hamming distance between the valid code vectors.

Error detection is possible if the received vector is not equal to some other code vector. This shows that the transmission errors in the received code vector should be less than minimum distance $d_{min}$. The table lists some of the requirements of error control capability of the code.

**Table : Error control capabilities**

| Sr. No. | Name of errors detected/corrected | Distance requirement |
|---|---|---|
| 1 | Detect upto 's' errors per word | $d_{min} \geq s+1$ |
| 2 | Correct upto 't' errors per word | $d_{min} \geq 2t+1$ |
| 3 | Correct upto 't' errors and detect $s > t$ errors per word | $d_{min} \geq t+s+1$ |

For the (n,k) block code the minimum distance is given as,

$$d_{min} \leq n - k + 1$$

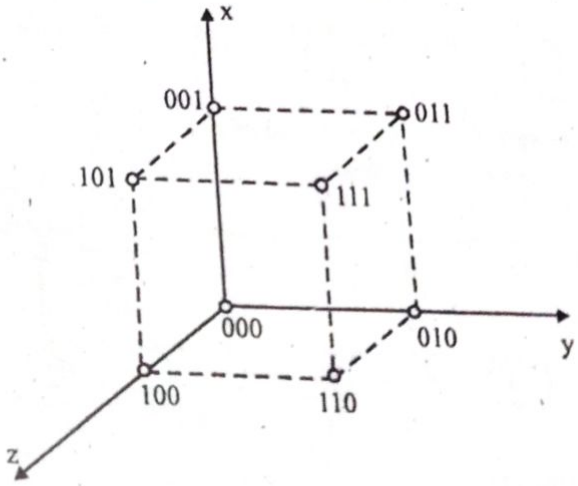



*Fig. Code vectors representing 3-bit code words*

---

**Q.10** ***Given a (7,4) block code generated by (G) below :***

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix}$$

*(i) Find all the code words of the code.*

*(ii) Find H, parity check matrix of the code.*

*[R.T.U. 2016]*

**Ans. (i) Codeword = DG**

| Message (D) | | | | Codeword | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 |
| 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

(ii)
$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix}$$

$$= [I_4 : P]$$

$$P = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix}$$

$$P^T = \begin{bmatrix} 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}$$

$$H = [P^T : I_{7-4}]$$

$$H = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

**Q.11** ***The Parity check matrix of a (7, 4) Hamming code is as under***

$$H = \begin{bmatrix} 1101100 \\ 1110011 \\ 1011001 \end{bmatrix}$$

***Calculate the syndrom vector for single bit errors.*** ***[R.T.U. 2012]***

**Ans.** $n = 7$

$k = 4$

$q = n - k = 3$

Error Pattern for Single Bit Error

| Bit in error | Bits of error (E) | | | | | | (non-zero bit) |
|---|---|---|---|---|---|---|---|
| 1st | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2nd | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 3rd | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 4th | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 5th | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 6th | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 7th | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

**Syndrom Calculation**

$$S = EH^T$$

$$H^T = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}$$

**Syndrom for 1st Bit Error**

$$S = EH^T$$

$$S = [1000000] \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}$$

$$S = [1\ 0\ 1]$$

*i.e.* syndrom vector for 1st bit in error.

Similary syndrom table can be drawn as

| S.No. | Erorr Vector (∈) | Syndrom (s) | Error |
|---|---|---|---|
| 1. | 0 0 0 0 0 0 0 | 0 0 0 | ← 1st row of $H^T$ |
| 2. | 1 0 0 0 0 0 0 | 1 1 1 | ← 2nd row of $H^T$ |
| 3. | 0 1 0 0 0 0 0 | 1 1 0 | ← 3rd row of $H^T$ |
| 4. | 0 0 1 0 0 0 0 | 0 1 1 | ← 4th row of $H^T$ |
| 5. | 0 0 0 1 0 0 0 | 1 0 1 | ← 5th row of $H^T$ |
| 6. | 0 0 0 0 0 1 0 | 1 0 0 | ← 6th row of $H^T$ |
| 7. | 0 0 0 0 0 0 1 | 0 1 0 | ← 7th row of $H^T$ |
| 8. | 0 0 0 0 0 0 0 | 0 1 1 | ← 7th row of $H^T$ |

There is a limitation to parity schemes. A parity bit is only guaranteed to detect an odd number of bit errors. If an even number of bits have errors, the parity bit records the correct number of ones, even though the data is corrupt. Consider the same example as before with an even number of corrupted bits:

| Type of bit parity error | Failed transmission scenario |
|---|---|
| Even parity error in two corrupted bits | Alice wants to transmit: 1001<br>Alice computes even parity value: 1 ^ 0 ^ 0 ^ 1 = 0<br>Alice sends: 10010<br>...TRANSMISSION ERROR...<br>Bob receives: 11011<br>Bob computes overall parity: 1 ^ 1 ^ 0 ^ 1 ^ 1 = 0<br>Bob reports correct transmission though actually incorrect. |

Bob observes even parity, as expected, thereby failing to catch the two bit errors.

**Q.13 Explain the type of errors and classification of codes.** *[R.T.U. 2018, 2017, 2012, 2010]*

**Ans. Type of errors**

The errors intorduced in the transmitted data during their transmission may be categorized as under

(i) Content errors

(ii) Flow integrity errors

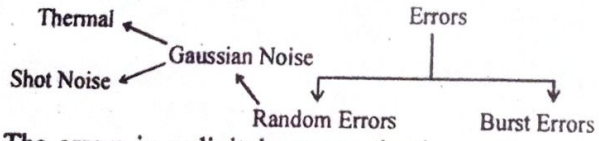
**Content errors:**

The content errors are nothing but errors in the contents of a message i.e., a 0 may be received as 1 or vice-versa.

**Flow integrity errors:**

Flow integrity errors meaning missing blocks of data. It is possible that a data block may be lost in the between as it has been delivered to a wrong destination.

**Types of Errors**




The errors in a digital communication system are caused by noise in the communication channel (Gaussian noise introduce in analog part of common channel).

Random errors due to white Gaussian noise are introduced. Gaussian noise had been our chief concern in designing and evaluating modulators and demodulators. Sources of Gaussian are :

**(a) Thermal Noise :** Due to vibration of individual molecules about their position of equilibrium in a crystal lattice, the conduction electron of metals wander randomly throughout the volume of metal, similarly molecule of an enclosed gas are in constant motion colliding with one another and colliding also with the walls of container. These agitations of molecules are called thermal agitations because they increase with temperature.

**(b) Shot Noise :** Result from a phenomenon associated with flow of current across semiconductor junctions. The charge carriers, electrons or holes enter the junction region from one side, drift or are accumulated at the junction and are collected on other side. The average junction current determines the average interval that elapses between time when two successive carriers enter the junction. The exact interval that elapses is subject to random fluctuations. This randomness give rise to shot noise. As we know that power spectral density of Gaussian noise at receiver input is white Gaussian noise. The transmission errors introduced during a particular interval by white Gaussian noise does not affect the performance of system during subsequent signalling interval.

**(c) Burst Errors :** Which is due to impulse noise by long quite intervals followed by high amplitude noise burst. This type of noise occurs from many natural and man-made causes such as lightning and switching transients. When such noise occurs, it affects more than one symbol or bit and there is usually a dependence of errors in successive transmitted symbols.

Error control schemes for dealing with random errors are random error correcting codes and coding scheme designed to correct burst errors are burst over correcting codes.

**Shot Noise :** Shot noise appears in active devices due to the random behaviour of charge carriers (electrons and holes). In electron tubes, shot noise is generated due to the random emission of electrons from cathodes; in semiconductor devices, it is caused due to the random diffusion of minority carriers or random generation and recombination of electron-hole pairs.

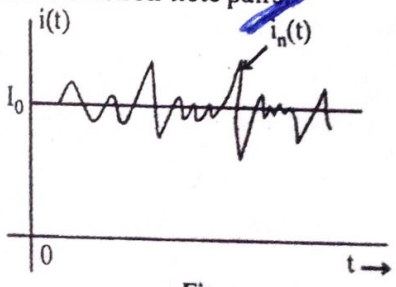



*Fig.*

Current in electron devices (tubes or solid state) flows in the form of discrete pulses, every time a charge carrier moves from one point to the other (e.g., cathode to plate).

The variation of power density spectrum with frequency is shown in Fig.

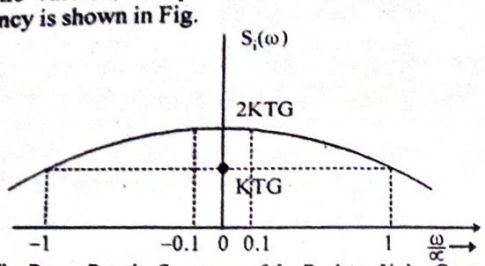



*Fig. Power Density Spectrum of the Resistor Noise Current*

It is obvious from the figure that the spectrum may be considered to be flat for $\frac{\omega}{\alpha} \le 0.1$. The power density spectrum $S_i(\omega)$ for this range of frequency is nearly constant and is given by

$$S_i(\omega) = 2kTG \qquad \ldots(2)$$

The value of $\alpha$ is of the order of $10^{14}$ and hence the frequency corresponding to $\frac{\omega}{\alpha} < 0.1$ is of order of $10^{13}$ Hz. Therefore, the frequency independent expression of $S_i(\omega)$ given by eq. (2) holds up to a frequency range of $10^{13}$Hz. This range covers almost all the practical applications in communication systems. Hence, for all practical purposes, the power density spectrum $S_i(\omega)$ is considered to be independent of frequency.

**Classification of codes :**

The codes are basically classified as under:

**(i) Errors detecting codes** The error detecting codes are capable of may detecting the errors. They cannot connect errors.

**(ii) Error correcting codes**

(1) Block codes (2) Convolution codes

The error correcting codes are capable of detecting as well as correcting the errors. These codes can be classified into block codes and convolution codes or linear and non-linear codes.

For error- free transmission, following codes are used :

(A) Block Codes

(B) Burst and Random Error Correcting Codes

(C) Interleaving

**(A) Block Codes**

**(i) For Error Correction**

1. We compare the performance of system using block codes for error correction with systems (n,k) using no error control coding.
2. Two measures of performance are :
   (a) Problem of incorrectly decoding a message bit.
   (b) Problem of incorrectly decoding a block of message digits.
3. We will do the comparison on the condition that rate of information transmission is same for coded and uncoded systems and both systems are operating with average signal power and noise power spectral density.
4. Coded or uncoded a block of say k message bits must be transmitted in duration of time.

   $$T_W = \frac{k}{r_b}$$

   where $r_b$ = message bit rate
5. $\therefore \quad r_w = \frac{1}{T_W} = \frac{r_b}{k}$

   if system uses an (n, k) block code, then bit rate going into channel

   $$r_c = r_b\left(\frac{n}{k}\right) \text{ or } r_c > r_b$$
6. Now

   $r_b$ = Message bit rate.

   $r_c$ = Channel bit rate.

   $q_c$ = Channel bit error probability for coded system.

   $q_u$ = Channel bit error probability for uncoded system.

   $P_{be}^u$ = Probability of incorrectly decoding a message bit in uncoded system.

   $P_{be}^c$ = Probability of incorrectly decoding a message bit in coded system.

   $P_{we}^u$ = Probability of incorrectly decoding a word of message bits in uncoded system.

   $P_{we}^c$ = Probability of incorrrectly decoding a word if message bits in coded system.
7. Now in uncoded case

   $P_{be}^u = q_u$ and probability that word of k message bit incorrectly received.

   $P_{we}^u = 1 - P$(all k message bits are correctly received)

   $= 1 - (1 - q_u)\,k$ when $kq_u \le 1$

   $= p^4{}_{we} = kq_u$

   since transmission errors are assumed to be independent.
8. In coded system, a word of k message digits will be incorrectly decoded when more than t errors occur in a n-bit codeword since block code is assumed to be able to correct upto t errors.

   Thus

   $P_{we}^c$ = P(t + 1 or more errors in a codeword)

ability of correcting the code comes at the receiving end.

3. They provide accurate transmission of message from one place to other place.
4. They provide good efficiency of message sending.
5. They help in sending correct message to the receiver.

**An another classification of codes are as follows:**

Let us consider the following fuble where a source of size 4 has enceted in binary codes symbol 0 and 1.

**Instantaneous codes:**

A uniquelly decodable code is called an instantaneous code if the end of any codeward is recognizable without examining. Subsequent code symbols. Prefix free codes are sometimes known as instantaneous codes.

**Optimal codes :**

A code is said to be optinal if it is instantaneous and has minimum average L for a given source with a given probability assignment for the source symbol.

---

Q.14 (a) *What is coding efficiency ? Show that the coding efficiency is maximum when P(0) = P (1).* *[R.T.U. 2018, Dec. 2013]*

(b) *Design (n, k) hamming code with a minimum distance of $d_{min}$ = 3 and message length of 4 bits.* *[R.T.U. Dec. 2013]*

---

**Ans.(a) Coding efficiency :** The code efficiency is defined as the ratio of message bits to the number of transimitted bits per blocks.

Let M be the number of symbols in an encoding alphabet. There messsage $[m_1\ m_2\ m_3\ ....\ m_N]$ with the probabilities $[P(m_1), P(m_2) ..... P(m_N)]$.

Let $n_i$ be the number of symbols in the $i^{th}$ message. The average length of the message or the average length per code word is than given by.

$$\tau = \sum_{i=1}^{n} n_i P(n_i) \text{ letters/message ... (1)}$$

I should be minimum to have an efficient transmission coding efficiency, then can be defined as

$$\boxed{\eta = \frac{\tau_{min}}{\tau}}$$

**Prove of coding efficiency is maximum when P(0) = P(1)**

Let H(x) be the entropy of the source in bits/message also let log m be the maximum average information associated with each letter in bits/letter.

Hence, the relation

$$\frac{H(x)}{\log m}$$

having a unit $\frac{\text{bits/message}}{\text{bits/letter}}$ or letter/message, gives the minimum average no. of letters per message $\frac{H(x)}{\log m} = \tau_{min}$

Hence the coding efficiency is

$$M = \frac{\tau_{min}}{L} = \frac{H(x)}{\tau \log m}$$

We know that H(x) will be maximum when symbols are equiproble.

And the coding efficiency will be maximum when H(x) will be maximum. So we can conclude that coding efficiency will be maximum when

$$P(0) = P(1)$$

Let us see an example to prove it

$$[M] = [m_1\ \ m_2]$$

$$P[M] = \left[\frac{1}{2}\ \ \frac{1}{2}\right]$$

$$\text{Efficiency } \eta = \frac{I(x; y)}{C} = \frac{H(x)}{\log_2 M}$$

$$= \left[\frac{\frac{1}{2}\log_2\frac{1}{\frac{1}{2}} + \frac{1}{2}\log_2\frac{1}{\frac{1}{2}}}{\log_2 2}\right] = \left(\frac{1}{1}\right) = 100\%$$

**Ans.(b)**

| ← code word length n = $(2^q - 1)$ → | |
|---|---|
| $K = 2^q - q - 1$ | $q = (n - K)$ |
| ← message bit → | ← parity bits → |

**Code word structure of hamming code**

here since message length is given as 4

$\therefore$ $4 = 2^q - q - 1$

$\Rightarrow$ $q = 3$

1. here, $n = 7 \Rightarrow$ length of code word is 7.
2. Number of message bits K = 4 (given)
3. Number of parity bits : $(n - K) = 3 = q$
4. Minimum distance $d_{min} = 3$
5. Code rate of code efficiency = $\frac{K}{n} = \frac{4}{7}$

**Q.15** ***Consider a (7, 4) block code generated by***

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$

***Find out the error vector and suppose that the received vector R is 1001001.*** *[R.T.U. 2013]*

**Ans.**

**Step : 1** $H = [P^T / I_{n-k}]_{m-k \times n}$

$$H = \underbrace{\left[\begin{array}{cccc} 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \end{array}\right]}_{P^T} \quad \underbrace{\left[\begin{array}{ccc} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{array}\right]}_{I_3}$$

**Step : 2** We have K = 4

n = 7

$\therefore 2^k = 2^4$ codewords for $2^4$ messages (0 0 0 0) ....(1 1 1 1)

**Step : 3** Choose a specific value of D from the 16 combinations for example 1011.

$\therefore$ C = DG

= 1 0 1 1 0 0 1

**Step : 4** Calculate syndrome

$$S = \underset{(1011001)}{CH^T} \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}^T$$

$$= (1011001) \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}$$

**Step : 5** If R = 1 0 0 1 0 0 1 is given find $S = RH^T$

$$= (1001001) \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

= 101

**Step : 6** Compare value of S by $H^T$. Now 101 is equal to third row of $H^T$ $\therefore$ third bit is in error $\therefore$ the transmitted word

C = 1 0 1 1 0 0 1

Error vector E = R – C

E = 0 0 1 0 0 0 0

**Q.16** ***Consider (7, 4) linear code whose generator matrix is***

$$G = \left[\begin{array}{cccc|ccc} 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{array}\right]$$

***(a) Find all the code vectors of this code.***

***(b) Find the parity check matrix for this code.***

***(c) Find the minimum weight of this code.***

***(d) Prove equation $CH^T = 0$.***

*[R. T. U. 2013, 2011; Raj. Univ. 2006, 2002]*

**Ans.** Given (7, 4) linear code

$$G = \left[\begin{array}{cccc|ccc} 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{array}\right]$$

**(a)** Here K = 4, n = 7

K = number of bits per message.

There are 16 possible combinations of 4-bits ranging from 0000 - 1111. For each combination, the code word is formed by

C = DG

$$C = \underset{D}{[1\ 1\ 1\ 1]} \overset{G}{\left[\begin{array}{cccc|ccc} 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{array}\right]} \overset{G}{}$$

C = 1 1 1 1 1 1 1

Similarly, for other combinations, codewords can be formed.

(b) Parity check matrix

$H = [P^T / I_{n-k}]$

$$H = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

(c) Minimum weight

First construct all codewords by C = DG, these are shown in given table.

Table

| | | Weight |
|---|---|---|
| 0 0 0 0 | 0 0 0 | 0 |
| 0 0 0 1 | 0 1 1 | 3 |
| 0 0 1 0 | 1 1 0 | 3 |
| 0 0 1 1 | 1 0 1 | 4 |
| 0 1 0 0 | 1 1 1 | 4 |
| 0 1 0 1 | 1 0 0 | 3 |
| 0 1 1 0 | 0 0 1 | 3 |
| 0 1 1 1 | 0 1 0 | 4 |
| 1 0 0 0 | 1 0 1 | 3 |
| 1 0 0 1 | 0 1 1 | 4 |
| 1 0 1 1 | 0 0 0 | 3 |
| 1 1 0 0 | 0 1 0 | 3 |
| 1 1 0 1 | 0 0 1 | 4 |
| 1 1 1 0 | 1 0 0 | 4 |
| 1 1 1 1 | 1 1 1 | 7 |

**(d)** Equation - $CH^T = 0$

Choose any C = 0 0 1 0 1 1 1

$$H^T = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$ Now $CH^T = [0\ 0\ 1\ 0\ 1\ 1\ 1]$

$$\begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = [000]$$ **Hence proved.**

**Q.17** *The parity check matrix of a particular (7, 4) linear block code is given by*

$$H = \begin{bmatrix} 1110100 \\ 1101010 \\ 1011001 \end{bmatrix}$$

*(a) Find the generator matrix G .*

*(b) List all the code vectors.*

*(c) What is the minimum distance between the code vectors ?*

*(d) How many errors can be detected and how many can be corrected ?* *[R.T.U. 2012]*

**Ans.** $H = \begin{bmatrix} 1 & 1 & 1 & 0 & : & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & : & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & : & 0 & 0 & 1 \end{bmatrix}$ ... (1)

$H = [P^T : I_3]$ ...(2)

Comparing matrix (1) & (2)

$$P^T = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \end{bmatrix}$$

$$P = \left[P^T\right]^T$$

$$= \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}_{4\times 3}$$

**(a) Generator Matrix (G)**

$$G = [I_K : P_{k \times q}]_{k \times n}$$

but $k = 4,\ q = 7,\ n = 7$

$$G = [I_4 : P_{4 \times 3}]_{4 \times 7}$$

so $$G = \begin{bmatrix} 1 & 0 & 0 & 0 & : & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & : & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & : & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & : & 0 & 1 & 1 \end{bmatrix}_{4\times 7}$$

**(b) Code Word**

$$C = MP$$

$$[C_1\ C_2\ C_3], = [M_1, M_2, M_3, M_4] \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

$$C_1 = M_1 \oplus M_2 \oplus M_3$$

$$C_2 = M_1 \oplus M_2 \oplus M_4$$

$$C_3 = M_1 \oplus M_3 \oplus M_4$$

**Code table**

5.3.2/5-20.

**(c) Minimum distance:** It is equal to minimum weight of any non-zero code vector. So from the table

$$d_{min} = [w(x)]_{min}$$

$$\boxed{d_{min} = 3}$$

**(d) Error Detection & Correction**

$$d_{min} \geq s + 1$$

$$3 \geq s + 1$$

$$s \leq 2$$

two errors will be detected.

$$d_{min} \geq 2t + 1$$

$$3 \geq 2t + 1$$

$$t \leq 1$$

only one error will be corrected.

# CYCLIC CODE 4

## PREVIOUS YEARS QUESTIONS

### PART-A

**Q.1** ***The intersection of cyclic codes is cyclic. Find the generator polynomial of $C_1 \cap C_2$.*** *[R.T.U. 2018]*

**Ans.** The generator polynomial of $C_1 \cap C_2$ is $g(x) = \text{lcm}(g_1(x), g_2(x))$.

Every codeword in the intersection of two cyclic codes is divisible by both generator polynomials and therefore by their least common multiple.

Conversely, every multiple of the least common multiple belongs to both codes, hence to their intersection. When $g_1(x)$ and $g_2(x)$ are relatively prime, their least common multiple is their product. In this case, the generator polynomial of the intersection of two cyclic codes is $g_1(x)g_2(x)$.

**Q.2** ***The following polynomial f(x) and g(x) are defined over GF(3).***
***f(x) = 2 + x + x² + 2x⁴***
***g(x) = 1 + 2x² + 2x⁴ + x⁵***
***Calculate addition and multiplication of the above two polynomials.*** *[R.T.U. 2013]*

**Ans.** $f(x) + g(x) = (2 + 1) + x + (1 + 2)x^2 + (2 + 2)x^4 + x^5 = x + x^4 + x^5$

$$f(x).g(x) = (2 + x + x^2 + 2x^4)(1 + 2x^2 + 2x^4 + x^5)$$
$$= 2 + x + (1 + 2.2)x^2 + 2x^3 + (2 + 2 + 2.2)x^4 + (2 + 2)x^5 + (1 + 2 + 1)x^6 + x^7 + 2.2x^8 + 2x^9$$
$$= 2 + x + (1 + 1)x^2 + 2x^3 + (2 + 2 + 1)x^4 + (2 + 2)x^5 + (1 + 2 + 1)x^6 + x^7 + x^8 + 2x^9$$
$$= 2 + x + 2x^2 + 2x^3 + 2x^4 + x^5 + x^6 + x^7 + x^8 + 2x^9$$

**Q.3** *How RS code can be organized? Explain in short.*

**Ans.** RS code is organized on the basis of groups bits. Such groups of bits are referred to as symbols.

**Q.4** *Write one disadvantage of cyclic codes.*

**Ans.** The error detection in cyclic codes is simpler but error correction is little complicated since the combinational logic circuits in error detector are complex.

**Q.5** *Define parity-check polynomial.*

**Ans. Parity-Check Polynomial :** It is a polynomial that can be found as the remainder polynomial.

**Q.6** *What do you mean by cyclic codes.*

**Ans. Cyclic Codes:** It has the property that a cyclic shift of one codeword of the code forms another codeword.

### PART-B

**Q.7** *Design a (4, 2) LBC :*
*(i) Find the generator matrix for code vector set*
*(ii) Find the parity check matrix*
*(iii) Make an enconding ckt.*
*(iv) Draw the encoding ckt.*
*(v) Draw the syndrome calculation ckt.* *[R.T.U. 2018]*



**Ans.(i)** Generator matrix of a (4, 2)

1011
1101

**(ii)**
0100
1001
0011

**(iii)**

| Inputs | | | | Outputs | |
|---|---|---|---|---|---|
| $D_3$ | $D_2$ | $D_1$ | $D_0$ | $Q_1$ | $Q_0$ |
| 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 0 | 0 | x | x |

**(iv)**

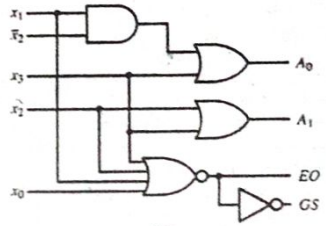



Fig.

**(v)**

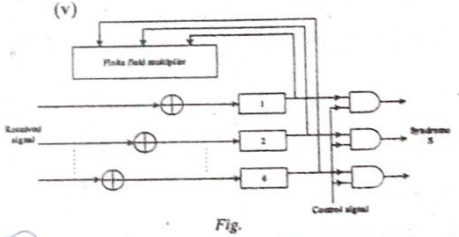



Fig.

**Q.8** *Write short notes on Cyclic codes.* *[R.T.U. 2017]*
*OR*
*What are the cyclic codes? Write the advantages and disadvantages of cyclic codes.* *[R.T.U. 2016]*

**Ans. Cyclic Codes :** Cyclic code has the property that a cyclic shift of one codeword of the code forms another codeword.

Meaning of cyclic shift is explained from figure i.e. n bit word instead of being written out horizontally is written around a circle. Starting at any point A the 7-bit word encountered by a clockwise rotation is 1101001; starting at some other arbitrary point say B we would read 0111010. The two words are related such that one is derived from other by cyclic shift. There are seven possible starting plans as shown in fig. Order in which the words are generated depends on the direction, clockwise or counterclockwise of the shift, but the end result of the resultant collection of words is not affected by the shift direction.

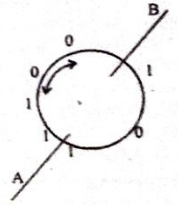



Fig.

A procedure for generating an (n, k) cyclic code is the following :

The bits of the uncoded word $\bar{A} = (A_0 A_1 .. A_{k-1})$ are written as the polynomial.

$$A(x) = A_0 \oplus A_1 x \oplus A_2 x^2 \oplus ... \oplus A_{k-1} X^{k-1} \quad ...(1)$$

The bits of the coded word $\bar{T} = [T_0 T_1 ... T_{n-1}]$ are written as the coefficients of the polynomial.

$$T(x) = T_0 \oplus T_1 x \oplus T_2 x^2 \oplus ... \oplus T_{n-1} X^{n-1} \quad ...(2)$$

We next form the "generating" polynomial g(x) of degree $r = n - k$.

$$g(x) = 1 \oplus g_1 x \oplus g_2 x^2 \oplus ... \oplus g_{r-1} x^{r-1} \oplus x^r \quad ...(3)$$

and we determine the values of the coefficient $g_1, g_2, ... g_{r-1}$ from the condition that g(x) be a factor of the polynomial

$$f(x) = x^n \oplus 1 \quad ...(4)$$

where n is the number of bits in the codeword. Finally, when g(x) is determined, T(x) is found from the equation-

$$T(x) = g(x) A(x) \quad ...(5)$$

As an example of the application of this procedure, let us generate a (7, 4) code since n = 7

$$f(x) = x^7 \oplus 1 \quad ... (6)$$

It can be verified that factors of f(x) are

$$f_1(x) = \lambda_1(x) . \lambda_2(x) . \lambda_3(x)$$
$$= (1 \oplus x)(1 \oplus x \oplus x^3)(1 \oplus x^2 \oplus x^3) \quad ...( 7)$$

To generate a code with n = 7 bits, T(x) in equation (2) must be a polynomial of degree $n - 1 = 6$.

**Advantages and Disadvantages of Cyclic Codes**

As we have seen that cyclic codes are the subclass of linear block codes, they have some advantages over noncyclic block codes as given below-

(iii) RS code is able to correct errors in t symbols

$$t = \frac{r}{2}$$

(iv) Code Rate $= \frac{k}{n} = R_c$

(v) and no. of correctable bits $B = \frac{m}{t}$

(vi) RS code is not effective code for correcting Random errors.

It can correct only half parity symbols making a total codeword length of n = k +r. So it gives trade off in error correcting capacity than other codes as the code rate does not depend on the parity symbols. So independency on the parity symbol of the code rate gives another trade off of RS code.

**Q.11** ***Design an encoder for (7, 4) BCC generated by $g(x) = 1 + x + x^3$ and verify its operation using message vector 0101.*** *[R.T.U. 2013]*

**Ans.** Here $g_0 = 1$

$g_1 = 1 \rightarrow$ closed path

$g_2 = 0 \rightarrow$ open path

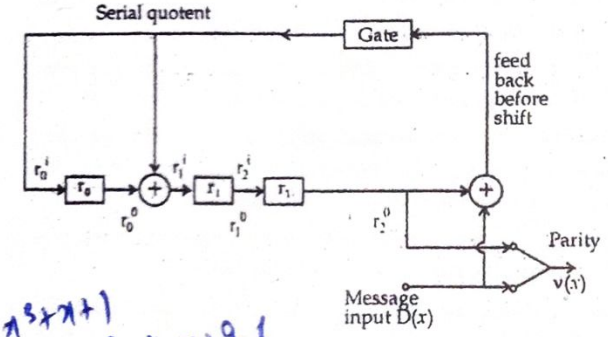



*Fig.*

Equation for $r_0^i = r_2^0 + d$

$r_1^i = r_0^0 + d$

$r_2^i = r_1^0$

| Input bit | Register Inputs | | | Register Outputs | | |
|---|---|---|---|---|---|---|
| d | $r_0^i$ | $r_1^i$ | $r_2^i$ | $r_0^0$ | $r_1^0$ | $r_2^0$ |
| - | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 shift 1 | 1 | 1 | 0 |
| 0 | 0 | 0 | 1 shift 2 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 shift 3 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 shift 4 | 1 | 1 | 0 |

The code vector for (0101) is (1100101).

**Q.12** ***A (15, 5) linear cyclic code has a generator polynomial $g(x) = 1 + x + x^2 + x^4 + x^5 + x^8 + x^{10}$***
***(i) Draw block diagram of an encoder and syndrome calculator for this code.***
***(ii) Find the code polynomial for the message polynomial.***
***$D(x) = 1 + x^2 + x^4$ (in a systematic - form).*** *[R.T.U. 2013]*

**Ans.** Given (15, 5) LBC

$$g(x) = 1 + x + x^2 + x^4 + x^5 + x^8 + x^{10}$$

**(i) Block Diagram of Encoder**

Here n – k = 10

∴ no of shift register is = 10 from $r_0 - r_q$ to calculate 10

Check bits

Here $g_0 = 1$ $g_1 = 1$ $g_2 = 1$ $g_3 = 0$ $g_4 = 1$ $g_5 = 1$

$g_6 = 0$ $g_7 = 0$ $g_8 = 1$ $g_9 = 0$ $g_{10} = 1$

According to these valves encoder is designed.

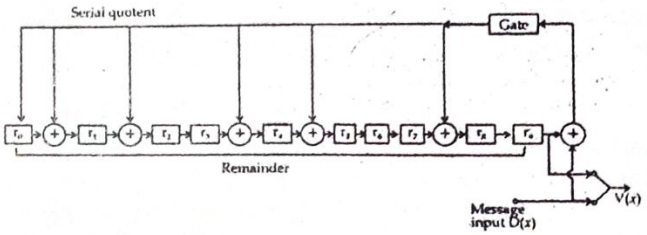



*Fig. 1*

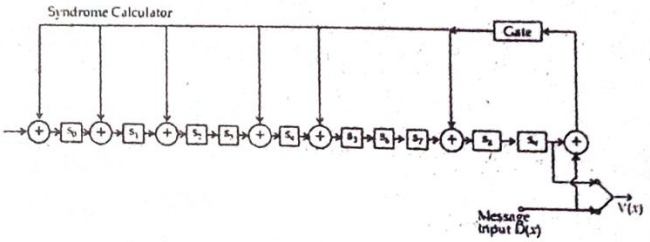



*Fig. 2*

(ii) Code Polynomial for Message Poly $D(x) = 1 + x^2 + x^4$ (in systematic form)

$\frac{x^{n-k}D(x)}{g(x)}$ Its remainder gives the values of parity check poly.

$$\frac{x^{15-5}D(x)}{g(x)}$$

$$\Rightarrow \frac{x^{10}(1+x^2+x^4)}{1+x+x^2+x^4+x^5+x^8+x^{10}}$$

$$1+x+x^2+x^4+x^5+x^8+x^{10} \overline{)\, x^{10}+x^{12}+x^{14}} \quad x^4+1$$

$$x^4+x^5+x^6+x^8+x^9+x^{12}+x^{14}$$

$$x^4+x^5+x^6+x^8+x^9+x^{10}$$

$$1+x+x^2+x^4+x^5+x^8+x^{10}$$

$$r(x) = 1 + x + x^2 + x^6 + x^9$$

$$\therefore \quad V = \underbrace{1110001001}_{r} \quad \underbrace{10101}_{+} \quad \underbrace{)}_{D=15}$$

**Q.13** ***Let C be a (7, 4) cyclie code with g (x) = 1 + x + $x^3$. Find the generator matrix G for C and also find the codeword for d = (1010) ?*** *[R.T.U. 2012]*

**Ans.** Since n = 7, k = 4, we have

$g(x) = 1 + x + x^3 \leftrightarrow 1101000$

$xg(x) = x + x^2 + x^4 \leftrightarrow 0110100$

$x^2g(x) = x^2 + x^3 + x^5 \leftrightarrow 0011010$

$x^3 g(x) = x^3 + x^4 + x^6 \leftrightarrow 0001101$

Then we have

$$G = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix} \quad \ldots (1)$$

For $d = [1\ 0\ 1\ 0]$

$$c = dG = [1\ 0\ 1\ 0] \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix} = [1\ 1\ 1\ 0\ 0\ 1\ 0]$$

So code word is 1110010.

## PART-C

**Q.14** ***(a) What do you understand by parity – check polynomial? Explain it in correspondence with generator polynomial.***

***(b) The generator polynomial of a (6,3) cyclic code is $g(x) = 1 + x^2$. Find all the code words of the code.*** *[R.T.U. 2016]*

**Ans.(a) Parity-Check Polynomial :** The parity-check polynomial is a polynomial that can be found as the remainder polynomial.

When the message polynomial, shifted by (n – k) times, is divided by the generator polynomial g(x).

For an (n, k) cyclic code, the generator g(x) must divide $(x^n - 1)$ and the quotient $h(x) = (x^n - 1)/g(x)$ is called the parity-check polynomial. For any codeword c(x), it follows that h(x) satisfies

$$h(x)\, c(x) \bmod x^n - 1 = 0$$

Since h(x) is given by dividing $x^n - 1$ by g(x), one can prove this statement by observing that c(x) = m(x) g(x) for some m(x).

**Explaination :**

$$h(x)\, c(x) = m(x)\, g(x)\, h(x) = m(x)\, (x^n - 1).$$

Since $x^n - 1$ divides h(x) c(x), the remainder is zero.

The parity-check polynomial $h(x) = h_0 + h_1x + h_2x^2 + \ldots h_k x^k$ of an (n, k) code has cyclic parity-check matrix of the form

$$H = \begin{bmatrix} h_k & h_{k-1} & \cdots & h_0 & 0 & 0 & 0 \\ 0 & h_k & h_{k-1} & \cdots & h_0 & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & 0 & h_k & \cdots & h_1 & h_0 \end{bmatrix}$$

In this case, one can use the fact that $g(x)h(x) = x^n - 1$ to verify that $GH^T = 0$.

**Ans.(b)** (6,3) cyclic code ⇒ no. of message bits = 3

No. of check bits = 6 – 3 = 3

$g(x) = 1 + x^2$

| Message | D(x) | V(x) | V |
|---|---|---|---|
| 0 0 1 | 0 | 0 | 0 0 0 0 0 0 |
| 0 0 1 | 1 | $1+x^2$ | 1 0 1 0 0 0 |
| 0 1 0 | x | $x+x^2$ | 0 1 1 0 0 0 |
| 0 1 1 | 1+ x | $1 + x + x^2 + x^3$ | 1 1 1 1 0 0 |
| 1 0 0 | $x^2$ | $x^2 + x^4$ | 0 0 1 0 1 0 |
| 1 0 1 | $1 + x^2$ | $1 + x^2 + x^2 + x^4 = 1 + x^4$ | 1 0 0 0 1 0 |
| 1 1 0 | $x + x^2$ | $x + x^2 + x^3 + x^4$ | 0 1 1 1 1 0 |
| 1 1 1 | $1 + x + x^2$ | $1 + x^2 + x + x^3 + x^2 + x^4 = 1 + x + x^3 + x^4$ | 1 1 0 1 1 0 |

$$D(x) = (D_0)1 + (D_1)x + (D_2)x^2$$

Code polynomial,

$$V(x) = D(x)\, g(x)$$

$$V(x) = (V_0)1 + (V_1)\, x + (V_2)x^2 + (V_3)x^3 + (V_4)x^4 + (V_5)x^5$$

$$V = [V_0\ V_1\ V_2\ V_3\ V_4\ V_5]$$

**Q.15** ***What is Galois field? Explain properties of Galois fields.*** *[R.T.U. 2013]*

***OR***

***Explain the construction of Galois Field (GF) along with its basic properties.*** *[R.T.U. 2016]*

**Ans. Galois Field :** A field can have a finite number m of elements in A. In this case, the field is called m degree finite field. The minimum number of elements is 2, namely the neutral elements of the two operations, so with the additive and multiplicative notations: 0 and 1.In this case, the second group contains a single element, the unit element 1. The operation tables for both elements are in $Z_2$:

| $\oplus$ | 0 | 1 |
|---|---|---|
| 0 | 0 | 1 |
| 1 | 1 | 0 |

| $\otimes$ | 0 | 1 |
|---|---|---|
| 0 | 0 | 0 |
| 1 | 0 | 1 |

This is the binary field, noted with **GF**(2), a very used one in digital processing. If p is a prime number, $Z_p$ is a field, because {1, 2,…., p – 1} form a group with modulo p multiplication.

So the set {1, 2, …., p – 1} forms a field related to modulo p addition and multiplication.

This field is called *prime field* and is noted by **GF**(p). There is a generalisation which says that, for each positive integer m, we should extend the previous field into a field with $p^m$ elements, called the extension of the field **GF**(p), noted by **GF**($p^m$).

Finite fields are also called *Galois fields*, which justify the initials of the notation **GF** (Galois Field).

**Field characteristic:** We consider the finite field with q elements **GF**(q), where q is a natural number. If 1 is the neutral element for addition, be the summations :

$$\sum_{i=1}^{1} 1 = 1, \sum_{i=1}^{2} 1 = 1+1 = 2, \ldots, \sum_{i=1}^{k} 1 = 1+1+k+1$$

As the field is closed with respect to addition, these summations must be elements of the field.

The field having a finite number of elements, these summations cannot all be distinct, so they must repeat somewhere; there are two integers m and n (m < n), so that

$$\sum_{i=1}^{m} 1 = \sum_{i=1}^{n} 1 \Rightarrow \sum_{i=1}^{n-m} 1 = 0$$

There is the smallest integer λ so that $\sum_{i=1}^{\lambda} 1 = 0$.

This integer is called the characteristic of the field **GF**(q).

The characteristic of the binary field **GF**(2) is 2, because the smallest λ for which

$$\sum_{i=1}^{\lambda} 1 = 0 \text{ is 2, meaning } 1 + 1 = 0$$

The characteristic of the prime field **GF**(p) is p. It results that

1 the characteristic of a finite field is a prime number

2 for n, m < λ, $\sum_{i=1}^{n} 1 \neq \sum_{i=1}^{m} 1$

The summations: $1, \sum_{i=1}^{2} 1, \sum_{i=1}^{3} 1, \ldots, \sum_{i=1}^{\lambda-1} 1, \sum_{i=1}^{\lambda} 1 = 0$: are λ distinct elements in **GF**(q), which form a field with λ elements **GF**(λ), called *subfield* of **GF**(q). Subsequently, any finite field **GF**(q) of characteristic λ contains a subfield with λ elements and it can be shown that if $q \neq \lambda$ then q is an exponent of λ .

**Order of an element :** We proceed a similar manner for multiplication: if a is a non zero element in **GF**(q), the smallest positive integer, n, so that $a^n = 1$ gives the order of the element a.

This means that a, $a^2$, ….., $a^n = 1$ are all distinct, so they form a multiplicative group in **GF**(q).

A group is called *cyclic group* if it contains an element whose successive exponents should give all the elements of the group. If in the multiplicative group, there are q–1 elements, we have $a^{q-1} = 1$ for any element, so the order n of the group divides q–1.

In a finite field **GF**(q) an element a is called *primitive element* if its order is q–1. The exponents of such an element generate all the non zero elements of **GF**(q). Any finite field has a primitive element.

**Example :** Let us consider the field **GF**(5), we have

$2^1 = 2$, $2^2 = 4$, $2^3 = 3$, $2^4 = 1$ so 2 is primitive
$3^1 = 3$, $3^2 = 4$, $3^3 = 2$, $3^4 = 1$ so 3 is primitive
$4^1 = 4$, $4^2 = 1$, so 4 is not primitive.

---

**Q.16** ***Consider a (7, 4) cylic code with generator polynomial $g(x) = 1 + x + x^3$ and let data word d = (1010).***

***(a) Find corresponding systematic codeword.***

***(b) Find all the cyclic binary code of block length.***

***(c) Find the minimum distance of each code.***

***[R.T.U. 2012]***

---

**Ans.** Given

$g(x) = 1 + x + x^3$

$n = 7$

$k = 4$

$q = n - k = 7 - 4 = 3$

$2^4 = 16$ message vector

# CONVOLUTIONAL CODE 5

## PREVIOUS YEARS QUESTIONS

### PART-A

**Q.1** *Define Code Tree.* *[R.T.U. 2018, 2016]*

**Ans. Code Tree**

(a) Code tree indicate flow of the coded signal along the nodes of tree.
(b) Code tree is lengthy way of representing coding process.
(c) Decoding is very simple using code tree.
(d) It repeats after no. of stages used in encoder.
(e) It is complex to implement in programming.

**Q.2** *Define Trellis.* *[R.T.U. 2016]*

**Ans. Trellis :**

(a) It indicates transitions from current to next state.
(b) It is shorter or compact way of representing coding process.
(c) Decoding is little complex using trellis diagram.
(d) It repeats in every stage in steady state, it has only one stage.
(e) It is simpler to implement in programming.

**Q.3** *Define Constraint Length.* *[R.T.U. 2016]*

**Ans. Constraint Length :** Constraint length of a convolutional code is defined as the number of shifts over which a single information bit can influence the encoder output. The constraint lengths of the encoder form a vector whose length is the number of inputs in the encoder diagram.

**Q.4** ***Design an encoder for the (7, 4) binary cyclic code generated by g (x) 1+ x+ x³ and verify its operation using the message vectors (1001) and (1011).*** *[R. T. U. Dec.2013]*

**Ans.** Given $g(x) = 1 + x + x^3$

The given generator polynomial can be written as.

$$g(x) = 1 + x + 0.x^2 + x^3$$

$$g_1 = 1$$

$$g_2 = 0$$

(Comparing with equation $x^3 + g_2x^2 + g_1x + 1$)

Encoder for generator polynomial $g(x) = 1 + x + x^3$ is given by figure

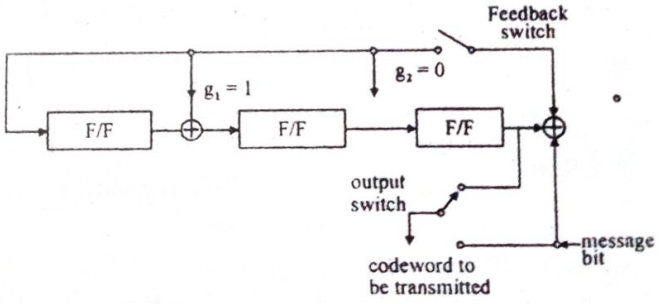



*Fig.*

**Q.5** ***For a (7, 4) cyclic code, the received vector Z (X) is 1110101 and g (x) = 1 + x + x³. Draw the syndrome calculation ckt and correct the single error in the received vector.*** *[R.T.U.Dec. 2013]*

**Ans.** Given generator polynomial is

$$g(\lambda) = 1 + x + x^3$$

$$g(x) = x^3 + 0\,x^2 + 1.x + 1 \quad \text{... (i)}$$

We know general form of generator polynomial is given by:

$$g(x) = x^3 + g_2x^2 + g_1x + 1 \quad \text{... (ii)}$$

Comparing (i) & (ii) we get

$$g_2 = 0;\ g_1 = 1$$

The syndrome calculator is shown in fig.

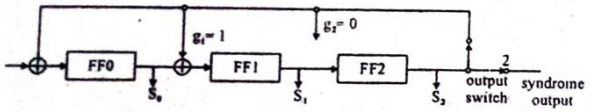



*Fig. : Syndrome calculator for polynomial $g(x) = 1 + x + x^3$*

## PART-B

**Q.6** *Explain the coding and decoding in the convolution code.* *[R.T.U. 2017]*

*OR*

*Decoding Probability of Convolution code.* *[R.T.U. 2018]*

**Ans.** In telecommunication, a convolutional code is a type of error-correcting code that generates parity symbols via the sliding application of a boolean polynomial function to a data stream. The sliding application represents the 'convolution' of the encoder over the data, which gives rise to the term 'convolutional coding.' The sliding nature of the convolutional codes facilitates trellis decoding using a time-invariant trellis. Time invariant trellis decoding allows convolutional codes to be maximum-likelihood soft-decision decoded with reasonable complexity.

Convolutional codes are often characterized by the base code rate and the depth (or memory) of the encoder [n,k,K]. The base code rate is typically given as n/k, where n is the input data rate and k is the output symbol rate. The depth is often called the "constraint length" 'K', where the output is a function of the current input as well as the previous K-1 inputs. The depth may also be given as the number of memory elements 'v' in the polynomial or the maximum possible number of states of the encoder (typically 2^v).

Convolutional codes are often described as continuous. However, it may also be said that convolutional codes have arbitrary block length, rather than being continuous, since most real-world convolutional encoding is performed on blocks of data. Convolutionally encoded block codes typically employ termination. The arbitrary block length of convolutional codes can also be contrasted to classic block codes, which generally have fixed block lengths that are determined by algebraic properties.

The code rate of a convolutional code is commonly modified via symbol puncturing. For example, a convolutional code with a 'mother' code rate n/k=1/2 may be punctured to a higher rate of, for example, 7/8 simply by not transmitting a portion of code symbols. The performance of a punctured convolutional code generally scales well with the amount of parity transmitted. The ability to perform economical soft decision decoding on convolutional codes, as well as the block length and code rate flexibility of convolutional codes, makes them very popular for digital communications.

**Decoding in Convolution code:**

Several algorithms exist for decoding convolutional codes. For relatively small values of k, the Viterbi algorithm is universally used as it provides maximum likelihood performance and is highly parallelizable. Viterbi decoders are thus easy to implement in VLSI hardware and in software on CPUs with SIMD instruction sets.

Longer constraint length codes are more practically decoded with any of several sequential decoding algorithms, of which the Fano algorithm is the best known. Unlike Viterbi decoding, sequential decoding is not maximum likelihood but its complexity increases only slightly with constraint length, allowing the use of strong, long-constraint-length codes. Such codes were used in the Pioneer program of the early 1970s to Jupiter and Saturn, but gave way to shorter, Viterbi-decoded codes, usually concatenated with large Reed-Solomon error correction codes that steepen the overall bit-error-rate curve and produce extremely low residual undetected error rates.

Both Viterbi and sequential decoding algorithms return hard decisions: the bits that form the most likely codeword. An approximate confidence measure can be added to each bit by use of the Soft output Viterbi algorithm. Maximum a posteriori(MAP) soft decisions for each bit can be obtained by use of the BCJR algorithm.

**Q.7** *A convolutional code is given by :*

$g_1 = [1\ 0\ 0],\ g_2 = [1\ 0\ 1],\ g_3 = [1\ 1\ 1]$

*(i) Draw the encoder corresponding to this code.*

*(ii) Draw the state – transition diagram for this code.*

*(iii) Draw the trellis diagram for this code.*

*[R.T.U. 2016]*



**Ans.**

(i)

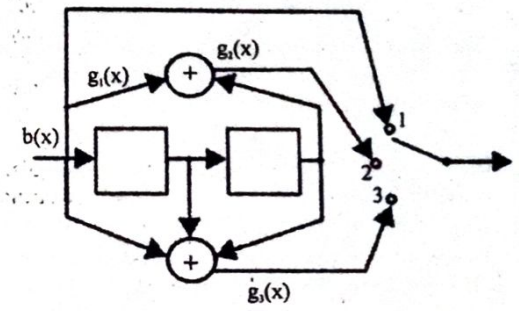



*Fig.1*

(ii)

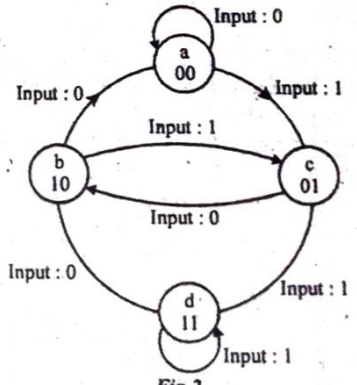



*Fig.2*

(iii) nth moment (n+1)st moment

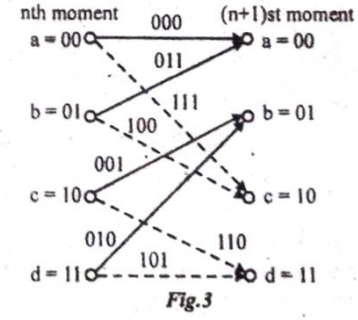



*Fig.3*

**Q.8** *Define Brust error.* *[R. T. U. Dec.2013]*

**Ans. Burst-Error Detection and Correction**

A Burst of length is defined as a sequence of digits in which the first digit and the $b^{th}$ digit are in error, with the b-2 digit in between either in error or received correctly. For example an error vector e = 0010010100 has a burst length of 6.

In usual transmission, we transmit one row after another. In the interlaced case, we transmit columns (of $\lambda$ elements) in sequence. When all the 15(n) columns are transmitted, we repeat the procedure for the next $\lambda$ code words to be transmitted.

To explain the error correcting capability of this code, we observe that the decoder will first remove the interlacing and regroup the received digits as $x_1, x_2 \ldots x_{15}, y_1, y_2 \ldots y_{15}, z_1, z_2 \ldots z_{15}$. Suppose that shaded digits in fig.(a) were in error.

Because the code is a two-error correcting code, two or less errors in each row will be corrected. Hence all the errors in fig.(a) are correctable. In general, if the original (n, k) code is t-error correcting, the interlaced code can correct any combination of t bursts of length $\lambda$ or less.

**Q.9** ***Write short notes on Trellies codes.***

***[R.T.U. Dec. 2013, 2013]***

**Ans. Trellis codes :** *Refer to Q.2.*

**Trellis diagram**

In shift register $M_2$ $M_1$, will indicate the state of encoder

So, let $M_2 M_1 = 0\ 0$ State *a*
$M_2 M_1 = 0\ 1$ State *b*
$M_2 M_1 = 1\ 0$ State *c*
$M_2 M_1 = 1\ 1$ State *d*

Then state transition table is

| | Current State ($M_2$ $M_1$) | Input ($M$) | O/P | | | Next State ($M_1$ M) |
|---|---|---|---|---|---|---|
| | | | $X_1$ | $X_2$ | $X_3$ | |
| (a) | 0 0 | 0 | 0 | 0 | 0 | 0 0 (a) |
| | 0 0 | 1 | 1 | 1 | 1 | 0 1 (b) |
| (b) | 0 1 | 0 | 0 | 1 | 0 | 1 0 (c) |
| | 0 1 | 1 | 1 | 0 | 1 | 1 1 (d) |
| (c) | 1 0 | 0 | 0 | 1 | 1 | 0 0 (a) |
| | 1 0 | 1 | 1 | 0 | 0 | 0 1 (b) |
| (d) | 1 1 | 0 | 0 | 0 | 1 | 1 0 (c) |
| | 1 1 | 1 | 1 | 1 | 0 | 1 1 (d) |

**Trellis diagram**

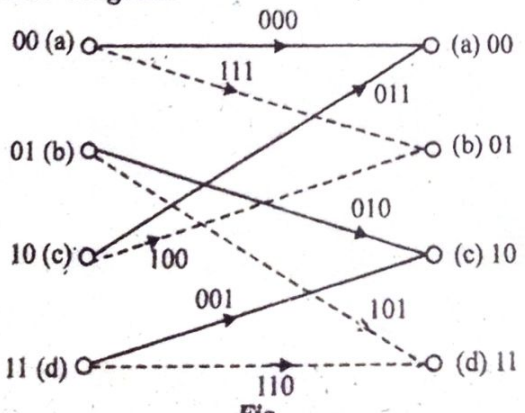



*Fig.*

Where Dotted line ⇒ 9/P M = 0
Solid line ⇒ 9/P M = 1

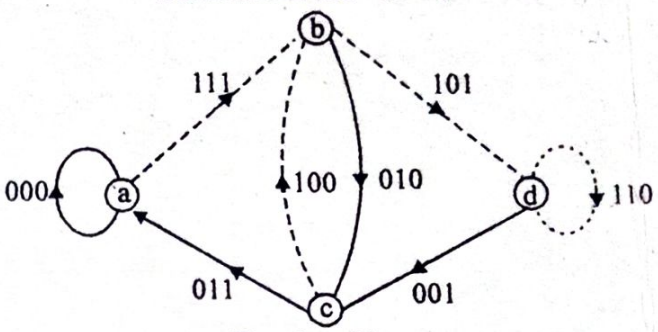



*Fig. : State Diagram*

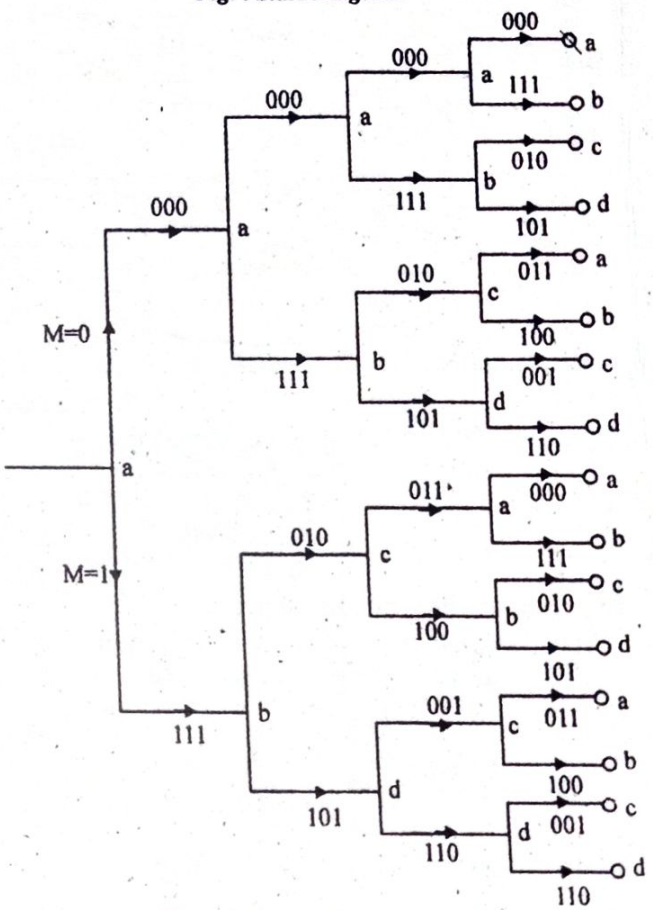



*Fig. : Code tree for given encoder*

Q.10 ***Initially consider that the register contains all zeroes. What will be the code sequence if the input data sequence is 100110?***

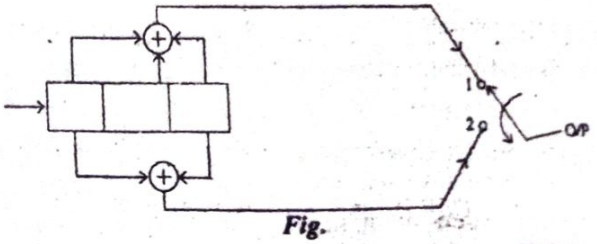



*Fig.*

***[R.T.U. 2013]***

*Table : Decoding procedure for the coder shown in fig. 2*

| | $c_1^{(m)}$ | $c_1^{(c)}$ | $c_2^{(m)}$ | $c_2^{(c)}$ | $c_3^{(m)}$ | $c_3^{(c)}$ | $c_4^{(m)}$ | $c_4^{(c)}$ | $c_5^{(m)}$ | $c_5^{(c)}$ | $c_6^{(m)}$ | $c_6^{(c)}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $s_1$ | X | X | | | | | | | | | | |
| $s_2$ | | | X | X | | | | | | | | |
| $s_3$ | X | | | | X | X | | | | | | |
| $s_4$ | | | X | | | | X | X | | | | |
| $s_5$ | X | | | | X | | | | X | X | | |
| $s_6$ | | | X | | | | X | | | | X | X |
| $s_7$ | | | | | X | | | | X | | | |
| $s_8$ | | | | | | | X | | | | X | |

In short we can say :

If all input data sequences are equally likely, a decoder that chooses $\hat{c}$ if

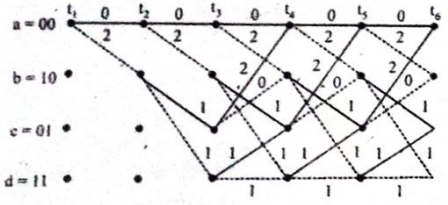



*Fig. 2*

$$P(r|\hat{c}) = \max p(rc_i) \text{ for all } c_i$$

where r is the received sequence and $c_i$ is one of the possible transmitted sequences, is called the maximum *likelihood decoder*. The conditional probabilities $P(rc_i)$ are called the *likelihood functions*. Note that for the BSC (binary symmetrical channel) the maximum likelihood decoder reduces to a minimum distance decoder. The rule of the minimum distance decoding is as follows : Choose $\hat{c}$ that minimizes the Hamming distance between the received sequence r and the transmitted sequence c.

**Q.12 *Describe Viterbi Algorithm.*** *[R.T.U. 2018]*

***OR***

***Explain the viterbi algorithm.*** *[R.T.U. 2017, 2016]*

***OR***

***Write short note on Viterbi Decoding.*** *[R.T.U.Dec. 2013]*

**Ans. Viterbi Algorithm :**

1. We consider all possible paths through the coder from starting point to the end point. Each possible input bit sequence generates its own path.
2. For such path we determine the corresponding sequence of coder output bits and compare each of these output sequences with actual received sequence.
3. If received sequence is exactly, identical to a sequence corresponding to some particular path through coder, then we shall assume that corresponding input sequence is the one corresponding to some particular path.

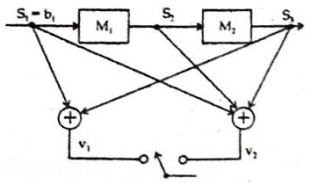



*Fig. 1*

4. If we find no exact corresponding, then we shall assume that input sequence to be one whose path generates the fewer bit discrepancies when compared to received sequence.
5. Now to illustrate viterbi algorithm let us use encoder.

Here $v_1 = S_1 \oplus S_3$

$v_2 = S_1 \oplus S_2 \oplus S_3$

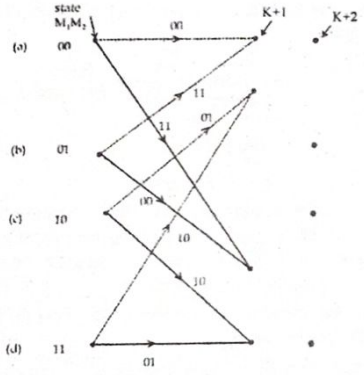



*Fig. 2*

6. Now initially encoder is clear

$M_1M_2 = 00$

Let there be presented at encoder a sequence of five information bits and let it be the corresponding sequence $V_{1R}V_{2R}$ bits are

$V_{1R}V_{2R} = \underline{10\ 00\ 10\ 00\ 00}$

7. Now trellis diagram

Here from state (a) is 00 if 0 is read, then received bits are 00 and if 1 is read then, received bits are 11. In either case $V_{1R}V_{2R} \neq 10$.



8. Now firstly without reference to received sequence let us trace the possible paths through encoder state as shown in trellis.
9. Starting from (a) in clock interval k = 1, a 0 will cause an output = 00 and will carry encoder in state (a). a 1 will generate output 11 and will carry encoder to state (c).
10. The number of discrepancies in each clock cycle between the bits associated with rallis in trellis diagram and actual received bits is shown in fig. 2
11. Thus if starting state is (a) at K = 1, a0 output generate an output = 00. Since, actual output is 10, the number of bit discrepancies = 1.
12. In next interval if input = 0 should again yield output = 00 and since the corresponding set of received bit is also 00 ∴ number discrepancies = 0. The cumulative discrepancies shown in circles.

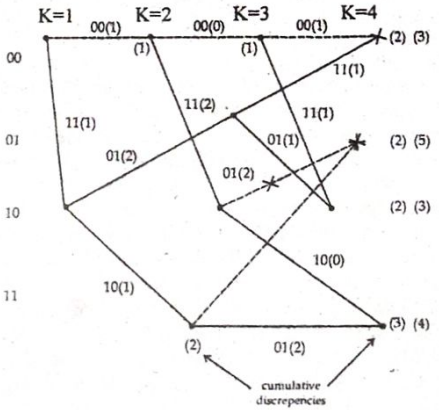



*Fig. 3*

To reach state (c) at K = 3 we go from (a) at K = 1 to (a) K = 2 and then from (a) at K = 2 to (c) at K = 3. In first path discrepancy = 1

In second path discrepancy = 2

∴ cumulative = 3

13. Suppose that we have to move from state (a) at K = 1 to state (a) K = L. How let us assume that path from state (b) at K = 4 to state (a) at K = L is fixed.

∴ We have to choose minimum discrepancy path from state (a) at K = 1 to state (b) at K = 4.

14. From trellis diagram we can notice that there are two paths to reach at K = 4 state (b) and cumulative discrepancies is written. Here we choose path with minimum discrepancies of (2) ∴ another path is discarded, the discarded path is shown by X.

∴ No paths surviving = no of states.

15. Surviving paths can be redrawn as shown : K = 1, K = 2, K = 3, K = 4

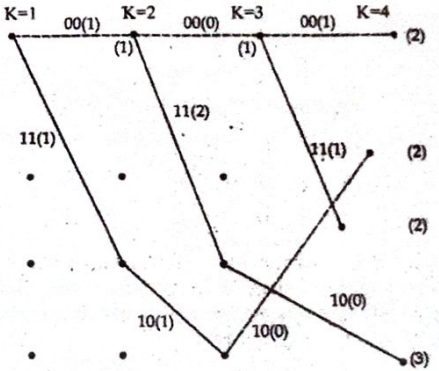



*Fig. 4*

16. Here we notice for output sequence of path through trellis corresponding to input bit stream consisting of all '0' yields minimum no of discrepancies. With such as result we would then decide that input sequence was all zero's, the received sequence as is readily verified should be all 0's.
17. Because of noise of received sequence = 10 00 10 00 00, the coder would have correct two errors.
18. As the number of paths are reduced ∴ memory required is reduced.

**Q.13 *Explain the operation of any convolutional encoder with the help of block diagram.*** *[R.T.U. 2016]*

**Ans. Encoder for Convolutional Code :** An encoder for a convolution code is shown in Fig. In this case.

K = no. of shift registers

= 3

v = no. of modulo-2 adders

= no. of bits in the code –block

= 3