

LECTURE NOTES
IN
INFORMATION THEORY AND
CODING

III B.Tech I Sem

Prepared by:

Mr. M. Ravikishore

Assistant Professor

**DEPARTMENT OF ELECTRONICS AND COMMUNICATION
ENGINEERING**



**ANNAMACHARYA INSTITUTE OF
TECHNOLOGY AND SCIENCES**

(AN AUTONOMOUS INSTITUTION)

(Approved by AICTE, NEWDELHI & Affiliated to J.N.T.U.A, Anantapuramu)

(Accredited by NAAC(A-Grade), Bangalore & NBA)

New Boyanapalli, Rajampet, Kadapa (Dist.), A.P-516126

ANNAMACHARYA INSTITUTE OF TECHNOLOGY AND SCIENCES RAJAMPET

(An Autonomous Institution)

R23

Title of the Course: Information Theory & Coding
Category: PEC-I
Couse Code: 23A045BT
Branch/es: ECE
Year III
Semester: I Semester

Lecture Hours	Tutorial Hours	Practice Hours	Credits
3	-	-	3

Course Objectives:

- 1 To provide an insight into the concept of information in the context of communication theory.
- 2 To implement various source coding algorithms and analyse their performance.
- 3 To gain knowledge about techniques for error detection and error correction.
- 4 To design cyclic codes and decoding of Cyclic codes.
- 5 To get familiar with design of convolutional codes.

Course Outcomes:

At the end of the course, the student will be able to

1. Explain the concepts of information in the context of communication theory.
2. Implement various source coding algorithms and analyze their performance.
3. Implementation and Design linear block codes.
4. Analyze Encoding and Decoding of cyclic codes.
5. Design of convolutional code and Viterbi algorithm.

Unit 1 Information Theory 9

Introduction, Definition of Entropy, Conditional Entropy, Relative Entropy, Basic Properties of Entropy, Mutual Information, Information Inequalities, Problem solving, Prefix-free Code, Coding a single Random Variable, Prefix, Free Code, Kraft Inequality.

Unit 2 Source Coding 10

Encoding of the Source output, Shannon's Encoding algorithm, Shannon's Fano Encoding Algorithm, Huffman Coding, Universal Source Coding: Lempel-Ziv Algorithm, Lempel-Ziv Welch Algorithm, LZW encoding & decoding, Channel Capacity of Different Channels- BSC, BEC.

Unit 3 Error Control Coding 12

Introduction to Error Control Codes, Types of Codes, Linear Block Codes: Introduction to Linear Block Codes, Matrix Description of Linear Block Codes, Error Detection and Correction capabilities of Linear Block Codes, The Hamming Distance, Encoding Block Codes, Syndrome and Error Detection, Decoding of Block Codes.

Unit 4 Binary Cyclic Codes**12**

Introduction, Algebraic Structure of Cyclic Codes, Generator and Parity Check Matrices for Cyclic codes, Encoding of Cyclic codes using Shift Register, Syndrome Calculation, Decoding of Cyclic Codes, Introduction to BCH Codes.

Unit 5 Convolutional Codes**10**

Introduction, Time domain and Transform Domain approach, Encoder for Convolution Code, State diagram, Tree Diagram and Trellis Diagram, The Viterbi Decoding algorithm, Turbo Codes.

Prescribed Textbooks:

- 1 Simon Haykin, Communication Systems, John Wiley, 4th Edition, 2010
2. Digital and Analog Communication Systems, K Sam Shanmugam, John Wiley India Pvt.Ltd.1996.

Reference Books:

- 1 Shu Lin, Daniel J. Costello Jr., Error Control Coding, Pearson, Second Edition, 2013.
2. Herbert Taub, Donald L Shilling, Goutam Saha, Principles of Communication Systems, 4th Edition, McGraw Hill, 2017.
3. Simon Haykin, Communication Systems, John Wiley, 4th Edition, 2010.

Information Theory & Coding

Information Theory

It was developed by Claude Shannon in 1948. It is a functional frame work for understanding, analysing, designing Communication system.

* Information theory is a branch of probability theory which can be used to study the Communication system.

By using information theory:

1. Efficient data transmission
2. Error free transmission
3. Secured data
4. Maximise the usage of channel

Consider an event x_j and the information associated with the event is $I(x_j)$ and probability of an event is $P(x_j)$

The relation between probability of an event and the amount of information associated with it is inversely proportional.

$$I(x_j) \propto \frac{1}{P(x_j)}$$

$$I(x_j) = -\log_2 \left[\frac{1}{P(x_j)} \right]$$

Consider two events x_j, y_k then information associated with the event is $I(x_j, y_k)$

$$I(x_j, y_k) \propto \frac{1}{P(x_j, y_k)}$$

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

If x_j, y_k are independent

$$P(x_j, y_k) = P(x_j) \cdot P(y_k)$$

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

$$I(x_j, y_k) = \log\left[\frac{1}{P(x_j)P(y_k)}\right]$$

The function f on RHS (8) in the above eqn must be a function which converts multiplication into addition.

$$I(x_j, y_k) = \log \frac{1}{P(x_j)} + \log \frac{1}{P(y_k)}$$

$$I(x_j, y_k) = I(x_j) + I(y_k)$$

$$\text{where } I(x_j) = \log \frac{1}{P(x_j)} = -\log P(x_j)$$

$$I(y_k) = \log \frac{1}{P(y_k)} = -\log P(y_k)$$

} self information

The units of Information:

If the base of logarithm is 2, e, 10 then the units of information is bits, nats and decits of Hartley.

Base : Units of Information

2 : bits

e

nats

10

decits (Hartley).

Entropy (H):

Let M different messages be $m_1, m_2, \dots, m_M$ with their respective probability values $P_1, P_2, \dots, P_M$. Assume that in a long time interval L messages have been generated.

The total information is given by

$$I_t = P_1 L \log \frac{1}{P_1} + P_2 L \log \frac{1}{P_2} + \dots + P_M L \log \frac{1}{P_M}$$

$$\text{Entropy (H)} = \frac{I_t}{L}$$

$$= P_1 \log \frac{1}{P_1} + P_2 \log \frac{1}{P_2} + \dots + P_M \log \frac{1}{P_M}$$

$$H = \sum_{j=1}^M P_j \log \frac{1}{P_j} \quad (31) = \sum_{j=1}^M P_j \log \frac{1}{P_j}$$

Average information per message is called an "Entropy".

Properties of Entropy:

1. If there is single possible message.

$$H = P_1 \log \frac{1}{P_1} \quad \text{let } P_1 = 1$$

$$[\because \log 1 = 0]$$

$$H = -P_1 \log P_1$$

$$H = 1 \log 1$$

$$H = 0$$

Single message does not carry (or) convey any information.

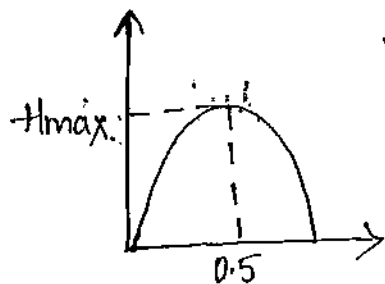
2. Let there is only one message out of M messages having probability 1 and all other zero.

$$H = P_1 \log \frac{1}{P_1} + P_2 \log \frac{1}{P_2} + \dots + P_M \log \frac{1}{P_M}$$

$$= 1 \log 1 + \lim_{P \rightarrow 0} \left[P_2 \log \frac{1}{P_2} + P_3 \log \frac{1}{P_3} + \dots + P_M \log \frac{1}{P_M} \right]$$

$$H = 0$$

∴ Entropy is maximum at 0.5



Rate of Information (R):

It is a measure of how much information is transmitted per unit time.
It is measured in bits/sec.

$$R = r \cdot H$$

where, r = Symbol rate

H = Entropy

Q1) An event has 6 possible outcomes with probabilities $P_1 = 1/2$, $P_2 = 1/4$, $P_3 = 1/8$, $P_4 = 1/16$, $P_5 = 1/32$, $P_6 = 1/32$. Find the Entropy of the system and also find the rate of information if there are 16 outcomes per second.

$$\text{Entropy (H)} = P_1 \log \frac{1}{P_1} + P_2 \log \frac{1}{P_2} + \dots + P_M \log \frac{1}{P_M}$$

$$= \sum_{j=1}^M P_j \log P_j$$

Given $M=6$.

$$= -[P_1 \log P_1 + P_2 \log P_2 + P_3 \log P_3 + P_4 \log P_4 + P_5 \log P_5 + P_6 \log P_6]$$

$$= -\left[\frac{1}{2} \log \frac{1}{2} + \frac{1}{4} \log \frac{1}{4} + \frac{1}{8} \log \frac{1}{8} + \frac{1}{16} \log \frac{1}{16} + \frac{1}{32} \log \frac{1}{32} + \frac{1}{32} \log \frac{1}{32}\right]$$

$$= 0.5254 - 0.5644 = 0.583$$

$$H = 1.93 \text{ bits/Message} \quad | \quad 1.26 \text{ bits/Message}$$

$$R = rH$$

$$= (16)(1.26)$$

$$R = 20.16 \text{ bits/sec}$$

Joint Entropy and Conditional Entropy:

Consider two events $[X] = [x_1, x_2, x_3, \dots, x_m]$

$$[Y] = [y_1, y_2, y_3, \dots, y_n]$$

Joint Entropy:

$$[X, Y] = \begin{bmatrix} x_1 y_1 & x_1 y_2 & \dots & x_1 y_n \\ x_2 y_1 & x_2 y_2 & \dots & x_2 y_n \\ \vdots & \vdots & \ddots & \vdots \\ x_m y_1 & x_m y_2 & \dots & x_m y_n \end{bmatrix}$$

We have three sets of complete probability schemes.

$$P(X) = P(x_j)$$

$$P(Y) = P(y_k)$$

$$P(X, Y) = P(x_j, y_k)$$

Corresponding entropies are

$$P(X) = H(X)$$

$$P(Y) = H(Y)$$

$$P(X, Y) = H(X, Y)$$

$$H(X) = \sum_{j=1}^m P(x_j) \log \frac{1}{P(x_j)} \quad (1) \quad - \sum_{j=1}^m P(x_j) \log P(x_j)$$

$$H(Y) = \sum_{k=1}^N P(y_k) \log \frac{1}{P(y_k)} \quad (2) \quad - \sum_{k=1}^N P(y_k) \log P(y_k)$$

$$\text{Joint Entropy: } H(X, Y) = \sum_{j=1}^m \sum_{k=1}^N P(x_j, y_k) \log \frac{1}{P(x_j, y_k)} \quad (3)$$

$$= - \sum_{j=1}^M \sum_{k=1}^N P(x_j, y_k) \log P(x_j, y_k)$$

Conditional Entropy:

Conditional Entropy are $H(X/Y)$ and $H(Y/X)$

$$H(X/Y) = - \sum_{j=1}^M \sum_{k=1}^N P(x_j, y_k) \log \frac{1}{P(x_j/y_k)}$$

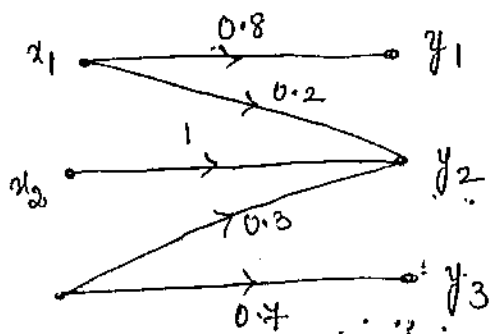
$$H(Y/X) = - \sum_{j=1}^M \sum_{k=1}^N P(x_j, y_k) \cdot \log \frac{1}{P(y_k/x_j)}$$

Relation between joint Entropy and Conditional Entropy:

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

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

Q. A discrete source transmits message x_1, x_2 and x_3 with probabilities 0.3, 0.4 and 0.3. The source is connected to the channel as shown in figure. Calculate all the entropies.



$$P(X) \quad P(Y) \quad P(X, Y) \quad P(X/Y) \quad P(Y/X)$$

$$H(X) \quad H(Y) \quad H(X, Y) \quad H(X/Y) \quad H(Y/X)$$

$X = \text{row}, Y = \text{column}$

The figure shows the Conditional probability matrix

$$P(Y/X) = \begin{matrix} & \begin{matrix} y_1 & y_2 & y_3 \end{matrix} \\ \begin{matrix} x_1 \\ x_2 \\ x_3 \end{matrix} & \begin{bmatrix} 0.8 & 0.2 & 0 \\ 0 & 1 & 0 \\ 0 & 0.3 & 0.7 \end{bmatrix} \end{matrix}$$

$\{P(Y/X)\}$ is taken from
o/p by i/p P's given in

$$P(X) = [0.3, 0.4, 0.3]$$

The joint probability matrix is

$$P(Y/X) = \frac{P(X,Y)}{P(X)}$$

$$P(X,Y) = P(Y/X) \cdot P(X)$$

The joint probability matrix $P(X,Y)$ can be obtained by multiplying rows of $P(Y/X)$ by $P(X_1)$, $P(X_2)$ & $P(X_3)$.

Entire 1st row is multiplied with $P(X_1) = 0.3$

Entire 2nd row is multiplied with $P(X_2) = 0.4$

Entire 3rd row is multiplied with $P(X_3) = 0.3$

$$P(X,Y) = \begin{bmatrix} 0.8 & 0.2 & 0 \\ 0 & 1 & 0 \\ 0 & 0.3 & 0.7 \end{bmatrix} \cdot [0.3, 0.4, 0.3]$$

$$P(Y) = \begin{bmatrix} 0.24 & 0.06 & 0 \\ 0 & 0.4 & 0 \\ 0 & 0.09 & 0.21 \end{bmatrix}$$

$\downarrow$ $\downarrow$ $\downarrow$
 $P(Y_1)$ $P(Y_2)$ $P(Y_3)$

addition of rows gives $P(X)$, addition of column gives $P(Y)$ in joint probability.

$$P(Y) = [0.24, 0.55, 0.21]$$

$$P(X/Y) = \frac{P(X,Y)}{P(Y)} = \begin{bmatrix} 0.24 & 0.06 & 0 \\ 0 & 0.4 & 0 \\ 0 & 0.09 & 0.21 \end{bmatrix}$$

$[0.24, 0.55, 0.21]$

Entire 1st Column is divided by $P(Y_1) = 0.24$

" 2nd " " " " " $P(Y_2) = 0.55$

" 3rd " " " " " $P(Y_3) = 0.21$

$P(x/y)$ can be obtained by dividing Columns of $P(x,y)$ by $P(y)$.

$$P(x/y) = \begin{bmatrix} 0.109 & 0 & 0 \\ 0 & 0.427 & 0 \\ 0 & 0.164 & 1 \end{bmatrix}$$

Entropies:

$$\begin{aligned} H(x) &= - \left[\sum_{j=1}^M P(x_j) \log P(x_j) \right] \\ &= - [P(x_1) \log P(x_1) + P(x_2) \log P(x_2) + P(x_3) \log P(x_3)] \\ &= - [0.3 \log 0.3 + 0.4 \log 0.4 + 0.3 \log 0.3] \\ &= 0.472 / \log 2 \\ &= 1.546 \text{ bits / Message} \end{aligned}$$

$$\begin{aligned} H(y) &= - \left[\sum_{k=1}^N P(y_k) \log P(y_k) \right] \\ &= - [P(y_1) \log P(y_1) + P(y_2) \log P(y_2) + P(y_3) \log P(y_3)] \\ &= - [0.24 \log 0.24 + 0.55 \log 0.55 + 0.21 \log 0.21] \\ &= 0.433 / \log 2 \\ &= 1.441 \text{ bits / message} \end{aligned}$$

$$\begin{aligned} H(x,y) &= - \left[\sum_{j=1}^M \sum_{k=1}^N P(x_j, y_k) \log P(x_j, y_k) \right] \\ &= - [P(x_1, y_1) \log P(x_1, y_1) + P(x_1, y_2) \log P(x_1, y_2) + P(x_1, y_3) \log P(x_1, y_3) + \\ &\quad P(x_2, y_1) \log P(x_2, y_1) + P(x_2, y_2) \log P(x_2, y_2) + P(x_2, y_3) \log P(x_2, y_3) + \\ &\quad P(x_3, y_1) \log P(x_3, y_1) + P(x_3, y_2) \log P(x_3, y_2) + P(x_3, y_3) \log P(x_3, y_3)] \\ &= - [0.24 \log 0.24 + 0.06 \log 0.06 + 0 + 0 + 0.4 \log 0.4 + 0 + 0 + 0.09 \log 0.09 + \\ &\quad 0.21 \log 0.21] \\ &= \frac{0.617}{\log(2)} = 2.051 \text{ bits / Message.} \end{aligned}$$

$$H(X/Y) = \sum_{j=1}^M \sum_{k=1}^N P(x_j, y_k) \log \frac{1}{P(x_j/y_k)}$$

$$= 0.24 \log 1 + 0.06 \log \frac{1}{0.109} + 0 + 0 + 0.4 \log \frac{1}{0.727} + 0 + 0 +$$

$$0.09 \log \frac{1}{0.164} + 0.21 \log 1$$

$$= -[0.24 \log 1 + 0.06 \log 0.109 + 0.4 \log 0.727 + 0.09 \log 0.164 + 0.21 \log 1]$$

$$= \frac{0.183}{\log 2} = 0.610 \text{ bits/message}$$

$$H(Y/X) = \sum_{j=1}^M \sum_{k=1}^N P(x_j, y_k) \log \frac{1}{P(y_k/x_j)}$$

$$= -[0.24 \log 0.8 + 0.06 \log 0.2 + 0.4 \log 1 + 0.09 \log 0.3 + 0.21 \log 0.7]$$

$$= \frac{0.1444}{\log 2} = 0.480 \text{ bits/message}$$

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

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

$$H(X, Y) = H(X, Y) - H(Y)$$

$$H(X, Y) = H(X, Y) - H(X)$$

$$H(X, Y) = 2.051 - 1.441 = 0.61 \text{ bits/message}$$

$$H(Y, X) = 2.051 - 1.570 = 0.481 \text{ bits/message}$$

$$H(X) = 1.570 \text{ bits/Message} \quad H(Y) = 1.441 \text{ bits/Message}$$

$$H(X, Y) = 2.051 \text{ bits/Msg} \quad H(X/Y) = 0.610 \text{ bits/Msg} \quad H(Y/X) = 0.480 \text{ bits/Msg}$$

Q3. A transmitter has an alphabet of 4 letters $[x_1, x_2, x_3, x_4]$ and the receiver has an alphabet of 3 letters $[y_1, y_2, y_3]$. The joint probability matrix is $P(X, Y) = \begin{bmatrix} 0.3 & 0.05 & 0 \\ 0 & 0.25 & 0 \\ 0 & 0.15 & 0.05 \\ 0 & 0.05 & 0.15 \end{bmatrix}$. Calculate all the Entropies.

Given,

$$P(X, Y) = \begin{matrix} & y_1 & y_2 & y_3 \\ \begin{matrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{matrix} & \begin{bmatrix} 0.3 & 0.05 & 0 \\ 0 & 0.25 & 0 \\ 0 & 0.15 & 0.05 \\ 0 & 0.05 & 0.15 \end{bmatrix} \end{matrix}$$

" $P(X), P(Y), P(X, Y), P(X/Y), P(Y/X)$

$$P(X) = [0.35, 0.25, 0.2, 0.2]$$

$$P(Y) = [0.3, 0.5, 0.2]$$

$$P(X/Y) = \frac{P(X, Y)}{P(Y)}$$

Each Column of $P(X, Y)$ divide with y_1, y_2, y_3

$$= \begin{bmatrix} 0.3 & 0.05 & 0 \\ 0 & 0.25 & 0 \\ 0 & 0.15 & 0.05 \\ 0 & 0.05 & 0.15 \end{bmatrix}$$

$\begin{array}{ccc} 0.3 & 0.5 & 0.2 \end{array}$

$$P(X/Y) P(Y) = \begin{bmatrix} 0.3 & 0.05 & 0 \\ 0 & 0.25 & 0 \\ 0 & 0.3 & 0.25 \\ 0 & 0.1 & 0.45 \end{bmatrix}$$

$$P(Y/X) = \frac{P(X, Y)}{P(X)}$$

Each row of $P(X, Y)$ is divided by x_1, x_2, x_3, x_4

$$= \begin{bmatrix} 0.3 & 0.05 & 0 \\ 0 & 0.25 & 0 \\ 0 & 0.15 & 0.05 \\ 0 & 0.05 & 0.15 \end{bmatrix} \begin{array}{l} / 0.35 \\ / 0.25 \\ / 0.2 \\ / 0.2 \end{array}$$

$$P(Y/X) = \begin{bmatrix} 0.85 & 0.14 & 0 \\ 0 & 1 & 0 \\ 0 & 0.45 & 0.25 \\ 0 & 0.25 & 0.45 \end{bmatrix}$$

Entropies (H): $H(X)$, $H(Y)$, $H(X,Y)$, $H(X/Y)$, $H(Y/X)$

$$H(X) = - \sum_{j=1}^N P_j \log P_j \quad M=4$$

$$= -[P_1 \log P_1(x) + P_2(x) \log P_2(x) + P_3(x) \log P_3(x) + P_4(x) \log P_4(x)]$$

$$= -[0.35 \log 0.35 + 0.25 \log 0.25 + 0.2 \log 0.2 + 0.2 \log 0.2]$$

$$= 0.589$$

$$= \frac{0.589}{\log 2}$$

$$= 1.958 \text{ bits/Message}$$

$$H(Y) = - \sum_{k=1}^N P_k \log P_k \quad N=3$$

$$= -[P_1(y) \log P_1(y) + P_2(y) \log P_2(y) + P_3(y) \log P_3(y)]$$

$$= -[0.3 \log 0.3 + 0.5 \log 0.5 + 0.2 \log 0.2]$$

$$= \frac{0.444}{\log 2}$$

$$= 1.485 \text{ bits/Message}$$

$$H(X,Y) = - \sum_{j=1}^M \sum_{k=1}^N P(X,Y) \log P(X,Y)$$

$$= -[0.3 \log 0.3 + 0.05 \log 0.05 + 0 + 0 + 0.25 \log 0.25 + 0 + 0 + 0.15 \log 0.15 + 0.05 \log 0.05 + 0 + 0.05 \log 0.05 + 0.15 \log 0.15]$$

$$= 0.749$$

$$= \frac{0.749}{\log 2}$$

$$= 2.490 \text{ bits/Message}$$

$$H(X/Y) = - \sum_{j=1}^M \sum_{k=1}^N P(x_j, y_k) \log P(y_k/x_j)$$

$$\begin{aligned} H(X/Y) &= H(X, Y) - H(Y) \\ &= 2.490 - 1.485 \\ &= 1.005 \text{ bits/message} \end{aligned}$$

$$\begin{aligned} H(Y/X) &= H(X, Y) - H(X) \\ &= 2.490 - 1.958 \\ &= 0.532 \text{ bits/message} \end{aligned}$$

Mutual Information:

It is defined as the amount of information transferred when x_i is transmitted and y_j is received.

It is represented by $I(x_i, y_j) = \log \frac{P(x_i/y_j)}{P(x_i)}$ bits

The average mutual information is represented as $I(X, Y)$ and it is given by

$$I(X, Y) = \sum_{i=1}^M \sum_{j=1}^N P(x_i, y_j) I(x_i, y_j)$$

$$I(X, Y) = \sum_{i=1}^M \sum_{j=1}^N P(x_i, y_j) \log \frac{P(x_i/y_j)}{P(x_i)}$$

Properties of Mutual Information:

1. The mutual information of the channel is symmetric

Condition for Symmetry $I(X, Y) = I(Y, X)$

P2b It can be represented in terms of Entropies

$$I(X, Y) = H(X) - H(X/Y)$$

$$I(X, Y) = H(Y) - H(Y/X)$$

P3b The mutual information is always positive -

$$I(X, Y) > 0$$

P4b It is related to Joint Entropy $H(X, Y)$ by

$$I(X, Y) = H(X) + H(Y) - H(X, Y)$$

Q1b Find the mutual information of the channel whose joint probability matrix is given by $P(X, Y) = \begin{bmatrix} 0.25 & 0.25 \\ 0.15 & 0.15 \\ 0.1 & 0.1 \end{bmatrix}$

$$I(X, Y) = H(X) + H(Y) - H(X, Y)$$

$$P(X, Y) = \begin{matrix} & \begin{matrix} y_1 & y_2 \end{matrix} \\ \begin{matrix} x_1 \\ x_2 \\ x_3 \end{matrix} & \begin{bmatrix} 0.25 & 0.25 \\ 0.15 & 0.15 \\ 0.1 & 0.1 \end{bmatrix} \end{matrix}$$

$$P(X) = [0.5, 0.3, 0.2]$$

$$P(Y) = [0.5, 0.5]$$

Entropies:

$$H(X) = - \sum_{j=1}^N P(x_j) \log P(x_j)$$

$$= - [0.5 \log 0.5 + 0.3 \log 0.3 + 0.2 \log 0.2]$$

$$= 0.44$$

$$= \frac{0.44}{\log 2}$$

$$= 1.48 \text{ bits/Message}$$

$$H(Y) = - \sum_{k=1}^N P(Y_k) \log P(Y_k)$$

$$= -[0.5 \log 0.5 + 0.5 \log 0.5]$$

$$= 0.30$$

$$= \frac{0.30}{\log 2} = 1 \text{ bit/message}$$

$$H(X, Y) = - \sum_{j=1}^M \sum_{k=1}^N P(X_j, Y_k) \log P(X_j, Y_k)$$

$$= -[(0.5)(0.5) \log (0.25) + 0.25 \log 0.25 + 0.15 \log 0.15 + 0.15 \log 0.15 + 0.1 \log 0.1 + 0.1 \log 0.1]$$

$$= 0.44$$

$$= \frac{0.44}{\log 2}$$

$$= 2.48 \text{ bits/message}$$

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

$$= 2.48 - 1$$

$$= 1.48 \text{ bits/message}$$

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

$$= 2.48 - 1.48$$

$$= 1 \text{ bit/message}$$

Mutual Information:

$$I(X, Y) = H(X) + H(Y) - H(X, Y)$$

$$= 1 + 1.48 - 2.48$$

$$= 0.$$

Q2b for a channel whose matrix is given by

$$P(Y/X) = \begin{bmatrix} 0.6 & 0.2 & 0.6 \\ 0.2 & 0.6 & 0.2 \\ 0.2 & 0.2 & 0.6 \end{bmatrix}$$

find $I(X, Y)$

$$P(X) = [1/3, 1/3, 1/3]$$

$$P(Y) = [1/3, 1/3, 1/3]$$

$$I(X, Y) = H(X) + H(Y) - H(X, Y)$$

$$P(Y/X) = \frac{P(X,Y)}{P(X)}$$

$$P(X,Y) = P(Y/X) \cdot P(X)$$

$$= \begin{bmatrix} 0.6 & 0.2 & 0.6 \\ 0.2 & 0.6 & 0.2 \\ 0.2 & 0.2 & 0.6 \end{bmatrix} \cdot P(X)$$

$$P(X,Y) = \begin{bmatrix} 0.2 & 0.06 & 0.2 \\ 0.06 & 0.2 & 0.06 \\ 0.06 & 0.06 & 0.2 \end{bmatrix}$$

$$H(X) = - \sum_{j=1}^M P(X_j) \log P(X_j)$$

$$= - \left[\frac{1}{3} \log\left(\frac{1}{3}\right) + \frac{1}{3} \log\left(\frac{1}{3}\right) + \frac{1}{3} \log\left(\frac{1}{3}\right) \right]$$

$$= 0.477$$

$$= \frac{0.477}{\log 2}$$

$$H(X) = 1.58 \text{ bits/message}$$

$$H(Y) = - \sum_{k=1}^N P(Y_k) \log P(Y_k)$$

$$= - \left[\frac{1}{3} \log\left(\frac{1}{3}\right) + \frac{1}{3} \log\left(\frac{1}{3}\right) + \frac{1}{3} \log\left(\frac{1}{3}\right) \right]$$

$$H(Y) = 1.58 \text{ bits/message}$$

$$H(X,Y) = - \sum_{j=1}^M \sum_{k=1}^N P(X_j, Y_k) \log P(X_j, Y_k)$$

$$= - \left[0.2 \log 0.2 + 0.06 \log 0.06 + 0.2 \log 0.2 + 0.06 \log 0.06 + 0.2 \log 0.2 + 0.06 \log 0.06 + 0.06 \log 0.06 + 0.06 \log 0.06 + 0.2 \log 0.2 \right]$$

$$= 0.925$$

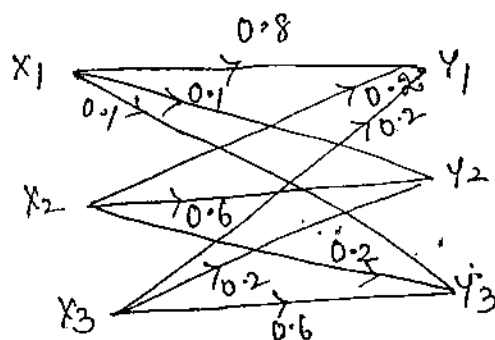
$$= \frac{0.925}{\log(2)} = 3.075 \text{ bits/message}$$

Mutual Information:

$$\begin{aligned} I(X;Y) &= H(X) + H(Y) - H(X,Y) \\ &= 1.58 + 1.58 - 3.07 \\ &= 0.09 \end{aligned}$$

Q3b Consider that a source emit messages x_1, x_2, x_3 and y_1, y_2, y_3 with joint probability $p(x,y)$ has shown the fig

- (i) Calculate all the entropy
(ii) Calculate average mutual information.



$$P(X,Y) = \begin{bmatrix} 0.8 & 0.1 & 0.1 \\ 0.2 & 0.6 & 0.2 \\ 0.2 & 0.2 & 0.6 \end{bmatrix}$$

$$P(X) = [1, 1, 1]$$

$$P(Y) = [1.2, 0.9, 0.9]$$

$$P(X|Y) = \frac{P(X,Y)}{P(Y)} = \begin{bmatrix} 0.8/1.2 & 0.1/0.9 & 0.1/0.9 \\ 0.2/1.2 & 0.6/0.9 & 0.2/0.9 \\ 0.2/1.2 & 0.2/0.9 & 0.6/0.9 \end{bmatrix}$$

$$= \begin{bmatrix} 0.6 & 0.1 & 0.1 \\ 0.16 & 0.6 & 0.2 \\ 0.16 & 0.2 & 0.6 \end{bmatrix}$$

$$P(Y/X) = \frac{P(X,Y)}{P(X)} = \begin{bmatrix} 0.8/1 & 0.1/1 & 0.1/1 \\ 0.2/1 & 0.6/1 & 0.2/1 \\ 0.2/1 & 0.2/1 & 0.6/1 \end{bmatrix} = \begin{bmatrix} 0.8 & 0.1 & 0.1 \\ 0.2 & 0.6 & 0.2 \\ 0.2 & 0.2 & 0.6 \end{bmatrix}$$

Entropies:

$$H(X) = -\sum_{j=1}^M P(x_j) \log P(x_j)$$

$$= -[1 \log 1 + 1 \log 1 + 1 \log 1] = 0 = 0/\log 2 = 0$$

$$H(Y) = -\sum_{k=1}^N P(y_k) \log P(y_k)$$

$$= -[1.2 \log 1.2 + 0.9 \log 0.9 + 0.9 \log 0.9] = -0.01265$$

$$= -0.042 \text{ bits/message}$$

$$H(X,Y) = -\sum_{j=1}^M \sum_{k=1}^N P(x_j, y_k) \log P(x_j, y_k)$$

$$= -[0.8 \log 0.8 + 0.1 \log 0.1 + 0.1 \log 0.1 + 0.2 \log 0.2 + 0.6 \log 0.6 + 0.2 \log 0.2 + 0.2 \log 0.2 + 0.6 \log 0.6]$$

$$= 1.02$$

$$= \frac{1.02}{\log 2} = 3.66 \text{ bits/message}$$

$$H(X|Y) = H(X,Y) - H(Y) = 3.702 \text{ bits/message}$$

$$H(Y|X) = H(X,Y) - H(X) = 3.66 \text{ bits/message}$$

average mutual information:

$$I(X,Y) = H(X) + H(Y) - H(X,Y)$$

$$= 0 - 0.042 - 3.66$$

$$= -3.702$$

Q4) Consider the two sources emit messages x_1, x_2, x_3 and y_1, y_2, y_3 with joint probabilities $P(x, y)$ shown in figure.

i) Calculate all entropies.

ii) Calculate average mutual information.

$$P(x, y) = \begin{bmatrix} 3/40 & 1/40 & 1/40 \\ 1/20 & 3/20 & 1/20 \\ 1/8 & 1/8 & 3/8 \end{bmatrix}$$

$$P(x, y) = \begin{bmatrix} y_1 & y_2 & y_3 \\ \begin{bmatrix} 3/40 & 1/40 & 1/40 \\ 1/20 & 3/20 & 1/20 \\ 1/8 & 1/8 & 3/8 \end{bmatrix} \begin{matrix} x_1 \\ x_2 \\ x_3 \end{matrix} \end{bmatrix}$$

$$P(x) = [0.125, 0.25, 0.625]$$

$$P(y) = [0.25, 0.3, 0.45]$$

$$P(y/x) = \frac{P(x, y)}{P(x)}$$

Entropy:

$$H(x) = -[0.125 \log 0.125 + 0.25 \log 0.25 + 0.625 \log 0.625]$$

$$= 0.39$$

$$= \frac{0.39}{\log 2} = 1.29 \text{ bits/message}$$

$$H(y) = -\sum_{k=1}^N P(y_k) \log P(y_k)$$

$$= -[0.25 \log 0.25 + 0.3 \log 0.3 + 0.45 \log 0.45]$$

$$= 0.46$$

$$= \frac{0.46}{\log 2} = 1.53 \text{ bits/message}$$

$$\begin{aligned}
 H(X, Y) &= - \sum_{j=1}^M \sum_{k=1}^N P(x_j, y_k) \log P(x_j, y_k) \\
 &= - \left[\frac{3}{40} \log \frac{3}{40} + \frac{1}{40} \log \frac{1}{40} + \frac{1}{40} \log \frac{1}{40} + \frac{1}{20} \log \frac{1}{20} + \frac{3}{20} \log \frac{1}{20} + \right. \\
 &\quad \left. \frac{1}{20} \log \frac{1}{20} + \frac{1}{8} \log \frac{1}{8} + \frac{1}{8} \log \frac{1}{8} + \frac{3}{8} \log \frac{3}{8} \right] \\
 &= 0.4982 + 0.385 = 0.8832 \quad \text{OR} \\
 &= \frac{0.8832}{\log 2} = 2.93 \text{ bits / message} \quad 2.6685
 \end{aligned}$$

$$\begin{aligned}
 H(X, Y) &= H(X/Y) + H(Y) & H(X, Y) &= H(X/Y) + H(Y) \\
 H(X/Y) &= H(X, Y) - H(Y) & H(Y/X) &= H(X, Y) - H(X) \\
 &= 2.93 - 1.53 & &= 2.93 - 1.29 \\
 &= 1.4 \text{ bits/message} & &= 1.64 \text{ bits/message}
 \end{aligned}$$

Mutual Information:

$$\begin{aligned}
 I(X, Y) &= H(X) + H(Y) - H(X, Y) \\
 &= 0.11 + 0.38 - 0.16
 \end{aligned}$$

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

$$H(X, Y) = - \sum_{j=1}^M \sum_{k=1}^N P(x_j, y_k) \log P(x_j, y_k) \rightarrow (1)$$

$$\left. \begin{aligned} P(x_j) &= \sum_{k=1}^N P(x_j, y_k) \\ P(y_k) &= \sum_{j=1}^M P(x_j, y_k) \end{aligned} \right\}$$

from probability theory,

$$P(A/B) = \frac{P(A, B)}{P(B)} \Rightarrow P(A, B) = P(A/B) \cdot P(B)$$

$$P(x_j/y_k) = \frac{P(x_j, y_k)}{P(y_k)} \Rightarrow P(x_j, y_k) = P(x_j/y_k) \cdot P(y_k) \rightarrow (2)$$

Substitute eq (2) in eq (1) [in log term].

$$H(X, Y) = - \sum_{j=1}^M \sum_{k=1}^N P(x_j, y_k) \log [P(\cancel{x_j}) P(y_k)]$$

$$= - \sum_{j=1}^M \sum_{k=1}^N P(x_j, y_k) [\log (P(x_j/y_k)) + \log P(y_k)]$$

$$= - \sum_{j=1}^M \sum_{k=1}^N P(x_j, y_k) \log P(x_j/y_k) - \sum_{j=1}^M \sum_{k=1}^N P(x_j, y_k) \log P(y_k)$$

$$= H(X/Y) - \underbrace{\sum_{k=1}^N \sum_{j=1}^M P(x_j, y_k) \log P(y_k)}_{\text{term to be simplified}}$$

$$= H(X/Y) - \sum_{k=1}^N P(y_k) \log P(y_k)$$

$$= H(X/Y) + \left[- \sum_{k=1}^N P(y_k) \log P(y_k) \right]$$

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

$$\boxed{\therefore H(X, Y) = H(X/Y) + H(Y)}$$

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

$$H(X, Y) = - \sum_{j=1}^M \sum_{k=1}^N P(x_j, y_k) \log P(x_j, y_k) \rightarrow (1)$$

$$P(B/A) = \frac{P(A \cap B)}{P(A)}$$

$$P(y_k/x_j) = \frac{P(x_j, y_k)}{P(x_j)} \Rightarrow P(x_j, y_k) = P(y_k/x_j) \cdot P(x_j) \rightarrow (2)$$

Substitute eq (2) in eq (1) [in log term]

$$H(X, Y) = - \sum_{j=1}^M \sum_{k=1}^N P(x_j, y_k) \log [P(y_k/x_j) \cdot P(x_j)]$$

$$= - \sum_{j=1}^M \sum_{k=1}^n P(x_j, y_k) [\log P(y_k/x_j) + \log P(x_j)]$$

$$= - \sum_{j=1}^M \sum_{k=1}^n P(x_j, y_k) \log P(y_k/x_j) - \sum_{j=1}^M \sum_{k=1}^n P(x_j, y_k) \log P(x_j)$$

$$= H(Y/X) - \sum_{j=1}^M \underbrace{\sum_{k=1}^n P(x_j, y_k)}_{P(x_j)} \log P(x_j)$$

$$= H(Y/X) - \sum_{j=1}^M P(x_j) \log P(x_j)$$

$$= H(Y/X) + \left[- \sum_{j=1}^M P(x_j) \log P(x_j) \right]$$

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

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

Information Inequalities:

It plays a fundamental role in communication theory, specially in the analysis of information concepts like channel capacity and Mutual Information.

Jensen's Inequality:

Let $f(x)$ be a convex function, and x be a random variable then

$$f(E[x]) \leq E[f(x)]$$

if $f(x)$ a concave function, the inequality is reverse

$$f(E(x)) \geq E[f(x)]$$

1. Verify Jensen's Inequality for convex function. $f(x) = x^2$,
 the random variable x takes the values 1 & 3 with
 probabilities 0.3 and 0.7.

(1) Compute $E(x)$:

$$E(x) = 1 \times 0.3 + 3 \times 0.7$$

$$= 2.4$$

(2) Compute $f[E(x)]$:

$$f(x) = x^2$$

$$f[E(x)] = f(2.4) = (2.4)^2$$

$$= 5.76$$

(3) Compute $E[f(x)]$:

$$0.3(1)^2 + 0.7(3)^2$$

$$= 6.6$$

$$0.3 f(x) + 0.7 f(x)$$

$$= 0.3 x^2 + 0.7 x^2$$

$$= 0.3(1)^2 + 0.7(3)^2$$

$$\therefore f[E(x)] \leq E[f(x)]$$

$$5.76 \leq 6.6$$

2. Verify Jensen's Inequality for Convex function $f(x) = e^x$
 the random variable x takes the values 0 and 1 with
 probability 0.6 and 0.4

(1) Compute $E(x)$:

$$E(x) = 0(0.6) + 1(0.4)$$

$$= 0.4$$

(2) Compute $f[E(x)]$:

$$f(x) = e^x$$

$$f(0.4)$$

$$f[E(n)] = f(0.4) = e^{0.4} \\ = 1.49$$

(3) Compute $E[f(n)]$:

$$0.6f(n) + 0.4f(n) \\ = 0.6e^x + 0.4e^x \\ = 0.6e^0 + 0.4e^1 \\ = 1.68$$

$$f[E(n)] \leq E[f(n)]$$

$$1.49 \leq 1.68$$

Ex. $f(x) = \log x$, x takes the values 1 and 4 with equal probabilities. [Concave function]
 $\frac{1}{2}, \frac{1}{2} \Rightarrow 0.5, 0.5$

(1) Compute $E(x)$:

$$E(x) = 1 \times 0.5 + 4(0.5) \\ = 2.5$$

(2) Compute $f[E(n)]$:

$$f(n) = \log n \\ f[E(n)] = f(2.5) = \log(2.5) \\ = 0.394$$

(3) Compute $E[f(n)]$:

$$0.5f(n) + 0.5f(n) \\ = 0.5 \log 1 + 0.5 \log 4 \\ = 0.5 \log 1 + 0.5 \log 4 \\ = 0.301$$

$$f[E(n)] \not\geq E[f(n)]$$

$$0.394 \geq 0.301$$

Jensen's inequality
 is not for convex function
 It is a Concave function.

Fano's Inequality:

It is widely used in Communication system to analyze the reliable transmission or prediction.

$$H(X/Y) \leq H(P_e) + P_e \log(|X|-1)$$

where $H(X/Y)$ is Conditional entropy.

$$H(P_e) = -P_e \log P_e - (1-P_e) \log (1-P_e)$$

∴ Consider four possible messages $P(x_1) = P(x_2) = P(x_3) = P(x_4) = \frac{1}{4}$
with error probability $P_e = 0.25$.

$$H(X) = -\sum_{i=1}^m P(x_i) \log P(x_i)$$

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

$$= 0.602$$

$$= 2 \text{ bits/message}$$

$$H(P_e) = -0.25 \log (0.25) - (1-0.25) \log (1-0.25)$$

$$= \frac{0.244}{\log 2}$$

$$= 0.8112$$

$X = \text{no. of messages}$

$$P_e \log(|X|-1) = 0.25 \log[4-1]$$

$$= 0.25 \log(3)$$

$$= \frac{0.1192}{\log 2} = 0.395$$

$$\therefore H(X/Y) \leq 0.8112 + 0.395 = 1.2062 \text{ bits/message}$$

Ex: $P(a) = P(b) = P(c) = \frac{1}{3}$ with $P_e = 0.2$

$$H(X) = - \sum_{i=1}^m P(x_i) \log P(x_i)$$

$$= - \left[\frac{1}{3} \log \frac{1}{3} + \frac{1}{3} \log \frac{1}{3} + \frac{1}{3} \log \frac{1}{3} \right]$$

$$= 0.4771$$

$$= 1.58 \text{ bits/message}$$

$$H(P_e) = - 0.2 \log 0.2 - (1-0.2) \log (1-0.2)$$

$$= 0.721 \text{ bits/message}$$

$$P_e \log (1/x - 1) = 0.2 \log (1/3 - 1)$$

$$= 0.2 \log 2 = \frac{0.0602}{\log 2}$$

$$= 0.2$$

$$H(X/Y) \leq 0.4771 + 0.2 = 0.6771 \text{ bits/message}$$

Prefix Code:

Prefix code is also known as prefix free code (or) instantaneous code. It is a variable length code where no code word is a prefix of any other code word.

Ex:

$$\left. \begin{array}{l} a = 0 \\ b = 110 \\ c = 10 \\ d = 111 \end{array} \right\}$$

Ex: - Non-prefix code:

$$\begin{array}{l} a = 0 \\ b = 01 \\ c = 10 \end{array}$$

In this case '0' is a prefix of '01' so, it is not a prefix code.

Coding a Single random Variable;

It refers to representing the possible outcomes of a discrete random variable using binary code words that minimizes the average code word length.

Ideally approaching the Entropy of the Variable

Ex:

Consider a random variable $[X] = [x_1, x_2, x_3, x_4, x_5, x_6]$

$$[P] = [0.4, 0.2, 0.15, 0.1, 0.1, 0.05]$$

Length n(i)	Message	Probability	Reduction-1	Reduction-2	Reduction-3
1	x_1	0.4	0	0.4	0
3	x_2	0.2	111	0.25	10
3	x_3	0.15	110	0.2	111
3	x_4	0.1	100	0.15	110
4	x_5	0.1	1011	0.1	101
4	x_6	0.05	1010	0.05	100

Reduction-4
 $\rightarrow 0.6$ 1
 0.4 0

$$\begin{aligned} \text{Average length } (L) &= \sum_{i=1}^n P_i n(i) \\ &= 0.4(1) + 0.2(3) + 0.15(3) + 0.1(3) + 0.1(4) + 0.05(4) \\ &= 2.35 \end{aligned}$$

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

$$= -[0.4 \log 0.4 + 0.2 \log 0.2 + 0.15 \log 0.15 + 0.1 \log 0.1 + 0.05 \log 0.05]$$

$$= 0.6876$$

$$= 2.284$$

Average length is approaching the Entropy.

Kraft Inequality:

It provides necessary and sufficient condition for the existence of prefix code. Kraft Inequality states that

$$\sum_{i=1}^n D^{-l_i} \leq 1$$

$$D=2$$

$$\sum_{i=1}^n D^{-l_i} \leq 1$$

Ex:

Q. Design a prefix code for 3 symbols with lengths $l_1=1$, $l_2=2$, and $l_3=3$.

$$D=2$$

$$P=3$$

$$\sum_{i=1}^3 D^{-l_i} \leq 1 \Rightarrow \sum_{i=1}^3 2^{-l_i} \leq 1 \Rightarrow 2^{-1} + 2^{-2} + 2^{-3} \leq 1$$

$$\Rightarrow 2^{-1} + 2^{-2} + 2^{-3} \leq 1$$

~~2~~

$\Rightarrow 0.875 \leq 1$ satisfies Kraft inequality and

Prefix code exists.

Q. $l_1=1$, $l_2=1$, $l_3=3$

$$P=3$$

$$\sum_{i=1}^3 D^{-l_i} \leq 1 \Rightarrow \sum_{i=1}^3 2^{-l_i} \leq 1 \Rightarrow 2^{-1} + 2^{-1} + 2^{-3} \leq 1$$

$$\Rightarrow 2^{-1} + 2^{-1} + 2^{-3} \leq 1$$

$$\Rightarrow \frac{1}{2} + \frac{1}{2} + \frac{1}{8} \leq 1$$

$$\Rightarrow 1.5 \geq 1$$

Kraft inequality is not satisfied ~~though~~ ^{No} Prefix code exists.

UNIT-2 :

Source Coding:

Shannon - Fano Code:

(Information source) → encoder → ...

Procedure:

1. The messages are written in the order of non increasing (decreasing) probabilities.
2. The messages are partitioned into equal equiprobable subsets X_1 and X_2 .
3. A 0 is assigned to 1st subset X_1 and one (1) is assigned to X_2 .
4. The same procedure is continued until each subset contains only one message.

The efficiency is calculated as

$$\eta = \frac{H(X)}{\bar{L} \log_2 M}$$

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

$$\text{Average Length } (\bar{L}) = \sum_{i=1}^m P_i n(i)$$

$$M = 2$$

Qr Apply Huffman Coding for the following message

$$[X] = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 \end{bmatrix}$$

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

Compress
data,
removes
redundant bits
add bits
Error
detection
Error
Correction

Length (n)	Message	Probability	Reduction-1	Reduction-2
1	x_1	0.4 0	0.4 00	0.4 0
3	x_2	0.2 111	0.2 111	0.2 111
3	x_3	0.12 101	0.12 101	0.12 101
4	x_4	0.08 1101	0.08 1101	0.08 1101
4	x_5	0.08 1100	0.08 1100	0.08 1100
4	x_6	0.08 1001	0.08 1001	0.08 1001
4	x_7	0.04 1000	0.04 1000	0.04 1000

0.8 [] MoI
0.8 [] 1000
11. Reduction - 1 defect has 100
0.12 ka 101 hai aur 1 red 1 n
ke pahle 200 hai aur 200 ke 1

Reduction - 3

0.4 0

→ 0.24 10

0.2 111

0.16 110

Reduction - 4

0.4 0

→ 0.36 11

0.24 10

Reduction - 5

0.6 11

0.4 10

Reduction - 6

→

Assign -1 to 1st probability, design-0 to 2nd probability, in every reduction as last value.

by reduction as last value
is a probab. & is stat. to
be used in the next step.
The number of steps
to place from each state
is lower than the number of

6. The first part of the book is devoted to the study of the properties of the function $f(x)$ which is defined by the equation $f(x) = \sum_{n=0}^{\infty} \frac{x^n}{n!}$. It is shown that this function is entire and that its Taylor series converges to the function for all values of x . The second part of the book is devoted to the study of the properties of the function $f(x)$ which is defined by the equation $f(x) = \sum_{n=0}^{\infty} \frac{x^n}{n!}$. It is shown that this function is entire and that its Taylor series converges to the function for all values of x .

$$\text{Average length } \bar{L} = \sum_{i=1}^n p_i n(i)$$

$$= 0.4(1) + 0.2(3) + 0.12(3) + 0.08(4) + 0.08(4) + 0.08(4) + 0.04(4)$$

$$\bar{L} = 2.48$$

$$H(X) = - \sum_{k=1}^m p_k \log p_k$$

$$= - [0.4 \log 0.4 + 0.2 \log 0.2 + 0.12 \log 0.12 + 0.08 \log 0.08 + 0.08 \log 0.08 + 0.08 \log 0.08 + 0.04 \log 0.04]$$

$$= 0.728 = \frac{0.728}{\log 2} = 2.41$$

$$\eta = \frac{H(X)}{\bar{L} \log_2 2}$$

$$[\because \log_2 2 = 1]$$

$$\eta = \frac{2.41}{2.48(1)}$$

$$\eta = 0.975$$

$$\therefore \eta = 97\%$$

Q2b Apply Huffman Coding for the following message.

$$1b \quad [X] = [x_1 \ x_2 \ x_3 \ x_4 \ x_5 \ x_6]$$

$$[P] = \left[\overset{=0.33}{1/3}, \overset{=0.03}{1/24}, \overset{=0.03}{1/3}, \overset{=0.1}{1/4}, \overset{=0.1}{1/9}, \overset{=0.1}{1/24} \right]$$

$$2b \quad [X] = [x_1 \ x_2 \ x_3 \ x_4] \ x_5$$

$$[P] = [0.15 \ 0.10 \ 0.35 \ 0.30 \ 0.10]$$

l_i	length(m_i)	Message	Probability	Reduction-1			Reduction-2	
1		x_1	$\frac{1}{3}$ 0.3	0	0.3	0	0.3	0
2		x_2	$\frac{1}{3}$ 0.3	11	0.3	11	0.3	11
3		x_3	$\frac{1}{4}$ 0.1	100	0.1	100	0.16	101
4		x_4	$\frac{1}{4}$ 0.1	1011	0.1		0.1	100
5		x_5	$\frac{1}{8}$ 0.03	10101	0.06			
5		x_6	$\frac{1}{8}$ 0.03	10100				

Reduction-3		Reduction-4	
0.3	0	0.56	1
0.3	11	0.3	0
→ 0.26	10		

$$\begin{aligned}
 \text{Average length } (L) &= \sum_{i=1}^n p_i l_i \\
 &= \frac{1}{3}(1) + \frac{1}{3}(2) + \frac{1}{4}(3) + \frac{1}{4}(4) + \frac{1}{24}(5) + \frac{1}{24}(5) \\
 &= 2.14
 \end{aligned}$$

$$\begin{aligned}
 H(X) &= -\sum_{k=1}^n p_k \log p_k \\
 &= -\left[\frac{1}{3} \log \frac{1}{3} + \frac{1}{3} \log \frac{1}{3} + \frac{1}{4} \log \frac{1}{4} + \frac{1}{4} \log \frac{1}{4} + \frac{1}{24} \log \frac{1}{24} + \frac{1}{24} \log \frac{1}{24} \right] \\
 &= 0.636 \\
 &= \frac{0.636}{\log 2} = 2.09
 \end{aligned}$$

$$\eta = \frac{H(X)}{L \log_2 2}$$

$$= \frac{2.09}{2.14} = 0.976 \Rightarrow 97.6\%$$

no	length	Message	Probability	Reduction-1	Reduction-2
1		x_3	0.35	0	0.35
2		x_4	0.30	10	0
3		x_1	0.15	110	0.35
4		x_2	0.10	111	0.30
4		x_5	0.10	1110	11
					10

Reduction-3
 → 0.65 1
 0.35 0.

$$\begin{aligned} \text{Average Length} &= \sum_{i=1}^n p_n(i) \\ &= 0.35(1) + 0.30(2) + 0.15(3) + 0.10(4) + 0.10(4) \\ &= 2.2 \end{aligned}$$

$$\begin{aligned} \text{Entropy } H(X) &= - \sum_{k=1}^m P_k \log P_k \\ &= - [0.35 \log 0.35 + 0.30 \log 0.30 + 0.15 \log 0.15 + 0.10 \log 0.10 + 0.10 \log 0.10] \\ &= 0.64 \\ &= \frac{0.64}{\log 2} \\ &= 2.12 \end{aligned}$$

$$\begin{aligned} \eta &= \frac{H(X)}{\sum \log_2 2} = \frac{2.12}{2.2(1)} \\ &= 0.963 \\ &= 96.3\% \end{aligned}$$

Q. Obtain Huffman Coding procedure for the ^{code} word

COMMITTEE

Q. The probability of 'C' = $\frac{1}{9} = 0.11$

The probability of 'O' = $\frac{1}{9} = 0.11$

The probability of 'M' = $\frac{2}{9} = 0.22$

The probability of 'I' = $\frac{1}{9} = 0.11$

The probability of 'T' = $\frac{2}{9} = 0.22$

The probability of 'E' = $\frac{2}{9} = 0.22$

[Symbols] = [C O M I T E]

[P] = [0.11 0.11 0.22 0.11 0.22 0.22]

length(n_i)	Message	Probability	Reduction-1	Reduction-2
2	M	0.22 10	0.22 10	0.33 11
2	T	0.22 01	0.22 01	0.22 10
2	E	0.22 00	0.22 00	0.22 01
2	C	0.11 110	0.11 110	0.22 00
3	O	0.11 1111	0.11 1111	
4	I	0.11 1110	0.11 1110	

Reduction-3

→ 0.44 0
0.33 11
0.22 10

Reduction-4

→ 0.55 1
0.44 0

$$\begin{aligned}
 \text{Average length } (\bar{L}) &= \sum_{i=1}^n P_i l_i \\
 &= 2(0.22) + 2(0.22) + 2(0.22) + 3(0.11) + 4(0.11) + 4(0.11) \\
 &= 2.53
 \end{aligned}$$

$$\begin{aligned}
 \text{Entropy} &= - \sum_{k=1}^m P_k \log P_k \\
 &= - [0.22 \log 0.22 + 0.22 \log 0.22 + 0.22 \log 0.22 + 0.11 \log 0.11 + 0.11 \log 0.11 + 0.11 \log 0.11] \\
 &= \frac{0.45}{\log 2} \\
 &= 2.491
 \end{aligned}$$

$$\begin{aligned}
 \eta &= \frac{H(X)}{\bar{L} \log_2 2} \\
 &= \frac{2.49}{2.53(1)} \\
 &= 0.984
 \end{aligned}$$

$$\therefore \eta = 98\% \Rightarrow 98.4\%$$

Q4) Find the Huffman Coding procedure for the code word.

ENGINEER

The probability of 'E' = $\frac{3}{8} = 0.375$

The probability of 'N' = $\frac{2}{8} = 0.25$

The probability of 'G' = $\frac{1}{8} = 0.125$

The probability of 'I' = $\frac{1}{8} = 0.125$

The probability of 'R' = $\frac{1}{8} = 0.125$

$$[Symbol] = [E \quad N \quad G \quad I \quad R]$$

$$[Probability] = [0.34 \quad 0.25 \quad 0.12 \quad 0.12 \quad 0.12]$$

Length (n _i)	Symbol	Probability	Reduction-1	Reduction-2
1	E	0.34	0.34 0	0.34 0
2	N	0.25	0.25 10	0.36 11 0.25 10
3	G	0.12	0.24 110	0.24 111
4	I	0.12	0.12 1110	0.12 1110
4	R	0.12	0.12 1110	0.12 1110

Reduction-3

→ 0.61 1

0.34 0

$$\begin{aligned} \text{Average length } (\bar{L}) &= \sum_{i=1}^n P_i n(i) \\ &= 1(0.34) + 2(0.25) + 3(0.12) + 4(0.12) + 4(0.12) \\ &= 2.19 \end{aligned}$$

$$\begin{aligned} \text{Entropy } H(x) &= -\sum_{K=1}^n P_K \log P_K \\ &= -[0.34 \log 0.34 + 0.25 \log 0.25 + 0.12 \log 0.12 + 0.12 \log 0.12 + 0.12 \log 0.12] \\ &= \frac{0.64}{\log 2} = 2.12 \end{aligned}$$

$$\eta = \frac{H(x)}{\bar{L} \log_2 2} = \frac{2.12}{2.19(1)} = 0.968$$

$$\% \eta = 96.8\%$$

Lempel-Ziv Codes:

Huffman and Shannon-Fano coding techniques are variable length codes. The drawback of these codes are

1. They require prior knowledge of symbol probabilities.
2. They can be applied for discrete memory less sources.

Lempel-Ziv Coding is a fixed length loss source coding technique and it is used for data compression.

Q. Procedure:

1. Partition the data vector into variable length blocks. These are called phrases.
2. A sequence of n bits or letters generated by source can be coded by parsing them into groups that are shortest subsequences not encountered earlier.

Ex:

Encode and decode the following sequence using Lempel-Ziv algorithm.

101011011010101011

Given data sequence: 1010110110101011

Sub sequence: 1, 0, 10, 11, 01, 101, 010, 1011

Numerical position: 1 2 3 4 5 6 7 8

Encoding Table:

Numerical Position	Dictionary location	Dictionary contents	Binary Encoded Blocks	
			Binary equation of: prefix, memory location	Innovative bit (Last bit)
1	001	No prefix	000	1
2	010	No prefix	000	0
3	011	010	001	0
4	100	011	001	1
5	101	100	010	1
6	110	101	011	1
7	111	110	101	0
8	-	111	110	1

Encoded Sequence: 0001, 0000, 0010, 0011, 0101, 0111, 1010, 1101

Decoding Table:

Memory location

Decoded Sequence

000

001

010

011

100

101

110

111

NULL

1

0

10

11

01

101

010

1011

Encode and decode the following sequence using Lempel-Ziv algorithm. 0101110010100101, let A^1 be A^0 as A^0

Given data sequence: 0 1 0 1 1 1 0 0 1 0 1 0 0 1 0 1

Sub sequence: 0, 1, 01, 11, 00, 10, 100, 101

Numerical positions: 1 2 3 4 5 6 7 8

Encoding Table:

Numerical position	Dictionary location	Dictionary Contents	Binary Encoded blocks	
			Binary equivalent of prefix memory location	Innovative bit
1	001	0	000	0
2	010	1	000	1
3	011	01	001	1
4	100	11	010	1
5	101	00	001	0
6	110	10	010	0
7	111	100	110	0
8	—	101	110	1

Encoded sequence: 0000, 0001, 0011, 0101, 0010,
0100, 1100, 1101

Decoding Table:

Memory Location	Decoded Sequence
	NULL
000	0
001	1
010	01
011	11
100	00
101	10
110	01
111	101

Lempel-Ziv-Welch Algorithm (LZW):

It is an advanced version of Lempel-Ziv algorithm. It is a lossless data compression algorithm that replaces repeating sequences of characters with codes.

It builds a dictionary of strings dynamically as it compresses data, replacing longer sequences with shorter codes.

This results in smaller file sizes. It is widely used in GIF image (Graphic Interchange Format).

Ex:

Encode and decode the following string using LZW Algorithm: ABB CB CAB ABC AAB CAA B

Given data sequence: ABB CB CAB ABC AAB CAA B

1 2 3 4 5 6 7
A, B, BC, BCA, BA, BCAB, BCAB

Compression Table:

o/p	Dictionary	
	Index	String
(0, A)	1	A
(0, B)	2	B
(1, C)	3	BC
(3, A)	4	BCA
(2, A)	5	BA
(4, A)	6	BCAA
(6, B)	7	BCAAB

The Compressed data is:

(0, A) (0, B) (2, C) (3, A) (2, A) (4, A) (6, B)

DeCompression: To decode the above data we need to

(0, A) (0, B) (2, C) (3, A) (2, A) (4, A) (6, B)

o/p DATA: A, B, BC, BCA, BA, BCAA, BCAAB.

Q. A, B, A, B, A, B, A

1 2 3 4 5
A, B, AB, ABA, BA

Compression Table:

o/p	Dictionary	
	Index	string
(0, A)	1	A
(0, B)	2	B
(1, B)	3	AB
(3, A)	4	ABA
(2, A)	5	BA

o/p is (Index, incremented
if
last number
if no index then
zero

The compressed data is: (0, A) (0, B) (1, B) (3, A) (2, A)

Decompression:

o/p DATA: A, B, AB, ABA, BA

Shannon's Encoding Algorithm:

Procedure:

1. Arrange the symbols in decreasing probability.
2. Compute the sequence

$$\alpha_1 = 0$$

$$\alpha_2 = p_1$$

$$\alpha_3 = p_1 + p_2$$

$$\vdots$$

$$\alpha_q = 1$$

3. Determine the smallest integer value

$$2^{l_i} \geq \frac{1}{p_i}$$

where $i = 1, 2, \dots, q$

4. Expand the decimal number α_i in binary form upto l_i places neglecting expansion beyond l_i places.

5. Remove the binary code points to get the desired code.

$$\eta_c = \frac{H(S)}{L}$$

$$H(S) = - \sum_{i=1}^n P_i \log P_i$$

$$L = \sum_{i=1}^n P_i l_i$$

Redundancy $\boxed{R_{re} = 1 - \eta_c}$

A source emits sequence of symbols from an alphabet consisting of symbols A, B, C, D and E with probabilities.

$\frac{4}{16}, \frac{3}{16}, \frac{2}{16}, \frac{3}{16}, \frac{5}{16}$ respectively - find the Shannon code for each symbol and efficiency of coding scheme

(1)

$$\frac{5}{16} \geq \frac{4}{16} \geq \frac{3}{16} \geq \frac{3}{16} \geq \frac{2}{16}$$

E A B B C

(2)

$$\alpha_1 = 0$$

$$\alpha_2 = P_1 = \frac{5}{16}$$

$$\alpha_3 = P_1 + P_2 = \frac{5}{16} + \frac{4}{16} = \frac{9}{16}$$

$$\alpha_4 = P_1 + P_2 + P_3 = \frac{5}{16} + \frac{4}{16} + \frac{3}{16} = \frac{12}{16}$$

$$\alpha_5 = P_1 + P_2 + P_3 + P_4 = \frac{5}{16} + \frac{4}{16} + \frac{3}{16} + \frac{2}{16} = \frac{14}{16}$$

$$\alpha_6 = P_1 + P_2 + P_3 + P_4 + P_5 = \frac{5}{16} + \frac{4}{16} + \frac{3}{16} + \frac{2}{16} + \frac{2}{16} = \frac{16}{16} = 1$$

$\therefore \alpha_6 = 1$

(3)

$$2^{l_i} \geq \frac{1}{p_i} \quad \forall i=1,2,3,4,5$$

$$\text{for } i=1, \quad 2^{l_1} \geq \frac{1}{p_1} \Rightarrow 2^{l_1} \geq \frac{16}{5} = 3.2, \quad l_1 = 2$$

$$i=2, \quad 2^{l_2} \geq \frac{1}{p_2} \Rightarrow 2^{l_2} \geq \frac{16}{4} = 4, \quad l_2 = 2$$

$$i=3, \quad 2^{l_3} \geq \frac{1}{p_3} \Rightarrow 2^{l_3} \geq \frac{16}{3} = 5.33, \quad l_3 = 3$$

$$i=4, \quad 2^{l_4} \geq \frac{1}{p_4} \Rightarrow 2^{l_4} \geq \frac{16}{2} = 8, \quad l_4 = 3$$

$$i=5, \quad 2^{l_5} \geq \frac{1}{p_5} \Rightarrow 2^{l_5} \geq \frac{16}{2} = 8, \quad l_5 = 3$$

(4)

$$\alpha_1 = 0$$

$$l_1 = 2$$

$$\text{code} = 00$$

$$\alpha_2 = (0.375)_{10}, \quad l_2 = 2$$

$$\text{code} = 01$$

$$\alpha_3 = (0.5625)_{10}, \quad l_3 = 3$$

$$\text{code} = 100$$

$$\alpha_4 = (0.75)_{10}, \quad l_4 = 3$$

$$\text{code} = 110$$

$$\alpha_5 = (0.875)_{10}, \quad l_5 = 3$$

$$\text{code} = 111$$

$$0.3125 \times 2 = 0.625$$

$$= 0.1$$

$$0.625 \times 2 = 1.25 = 1$$

$$0.25 \times 2 = 0.5 = 0$$

$$0.5 \times 2 = 1 = 1$$

$$0.5625 \times 2 = 1.125 = 1$$

$$0.125 \times 2 = 0.25 = 0$$

$$0.25 \times 2 = 0.5 = 0$$

$$0.5 \times 2 = 1 = 1$$

$$(0.1001)$$

$$0.75 \times 2 = 1.5 = 1$$

$$0.5 \times 2 = 1 = 1$$

$$(0.75)_{10} = (0.1110)$$

$$0.875 \times 2 = 1.75 = 1$$

$$0.75 \times 2 = 1.5 = 1$$

$$0.5 \times 2 = 1$$

$$(0.111)_2$$

Symbol	Probability	Length (l_i)	Code
E	5/16	2	00
A	4/16	2	01
D	3/16	3	100
B	2/16	3	110
C	2/16	3	111

$$H(S) = - \sum_{i=1}^n P_i \log_2 P_i$$

$$= - \left[\frac{5}{16} \log_2 \frac{5}{16} + \frac{4}{16} \log_2 \frac{4}{16} + \frac{3}{16} \log_2 \frac{3}{16} + \frac{2}{16} \log_2 \frac{2}{16} + \frac{2}{16} \log_2 \frac{2}{16} \right]$$

$$= 0.64$$

$$= 2.22$$

$$L = \sum_{i=1}^n P_i L_i$$

$$= \frac{5}{16}(2) + \frac{4}{16}(2) + \frac{3}{16}(3) + \frac{2}{16}(3) + \frac{2}{16}(3)$$

$$= 2.43$$

$$\eta = \frac{H(S)}{L} = \frac{H(S)}{L}$$

$$= \frac{2.22}{2.43}$$

$$= 0.9135$$

$$= 91.35\%$$

$$R_{uc} = 1 - \eta$$

$$= 1 - 0.9135$$

$$= 0.0864$$

Apply Shannon Encoding Algorithm for the following messages

$$[S] = [s_1 \quad s_2 \quad s_3]$$

$$[P] = [0.5 \quad 0.3 \quad 0.2]$$

find the Code efficiency and Redundancy.

$$(1) \quad 0.5 > 0.3 > 0.2$$

$$s_1 \quad s_2 \quad s_3$$

$$(2) \quad \alpha_1 = 0$$

$$\alpha_2 = p_2 = 0.5$$

$$\alpha_3 = p_1 + p_2 = 0.5 + 0.3 = 0.8$$

$$\alpha_4 = p_1 + p_2 + p_3 = 0.5 + 0.3 + 0.2 = 1$$

$$(3) \quad 2^{l_i} \geq \frac{1}{p_i} \quad \forall i=1,2,3$$

$$\text{for } i=1, \quad 2^{l_1} \geq \frac{1}{0.5} = 2, \quad l_1 = 1$$

$$\text{for } i=2, \quad 2^{l_2} \geq \frac{1}{0.3} = 3.33, \quad l_2 = 2$$

$$\text{for } i=3, \quad 2^{l_3} \geq \frac{1}{0.2} = 5, \quad l_3 = 3$$

$$(4) \quad \alpha_1 = 0, \quad l_1 = 1, \quad \text{code} = 0$$

$$\alpha_2 = (0.5)_{10} = (0.10)_2, \quad l_2 = 2, \quad \text{code} = 10$$

$$\alpha_3 = (0.8)_{10} = (0.1100)_2, \quad l_3 = 3, \quad \text{code} = 110$$

Symbol	Probability	Length (l_i)	Code
s_1	0.5	1	0
s_2	0.3	2	10
s_3	0.2	3	110

$$0.5 \times 2 = 1 \rightarrow (0.10)_2$$

$$\begin{aligned} 0.8 \times 2 &= 1.6 \rightarrow 1 \\ 0.6 \times 2 &= 1.2 \rightarrow 1 \\ 0.2 \times 2 &= 0.4 \rightarrow 0 \\ 0.4 \times 2 &= 0.8 \rightarrow 0 \\ &= (0.1100)_2 \end{aligned}$$

$$H(S) = - \sum_{i=1}^3 p_i \log p_i$$

$$= -[0.5 \log 0.5 + 0.3 \log 0.3 + 0.2 \log 0.2]$$

$$= 0.44 = \frac{0.44}{\log 2} = \frac{Ans}{\log 2}$$

$$= 1.46 \quad (\text{or}) \quad 1.48$$

$$C = \sum_{i=1}^n P_i L_i$$

$$= 0.5(1) + 0.3(2) + 0.2(3)$$

$$= 1.4$$

$$\eta = \frac{H(S)}{C}$$

$$= \frac{1.46}{1.4}$$

$$= 0.8588$$

$$\% \eta = 85.88\%$$

$$= \frac{1.48}{1.4}$$

$$= 0.8735$$

$$\% \eta = 87.35\%$$

$$R_{nc} = 1 - \eta_c$$

$$= 1 - 0.8588$$

$$R_{nc} = 0.1412 //$$

$$= 1 - \eta_c$$

$$= 1 - 0.8735$$

$$= 0.1265$$

Channel Capacity of different channels:

A maximum mutual information is called channel capacity.

$$\text{Channel Capacity } C = \max I(X, Y)$$

$$\text{channel efficiency (\%) or } \eta = \frac{\text{actual Trans Information}}{\text{Maximum Trans Information}}$$

$$\eta = \frac{I(X, Y)}{\max I(X, Y)}$$

$$\boxed{\eta = \frac{I(X, Y)}{C}}$$

$$\text{Redundancy} = 1 - \eta$$

$$= 1 - \frac{I(X;Y)}{C}$$

$$R = \frac{C - I(X;Y)}{C}$$

Symmetric channel:

In symmetric channel the rows and columns are identical

$$i) P(Y/X) = \begin{bmatrix} 1/2 & 1/4 & 1/4 \\ 1/4 & 1/2 & 1/4 \\ 1/4 & 1/4 & 1/2 \end{bmatrix}$$

$$\left[\begin{array}{l} \text{Symmetric} \\ A^T = A \end{array} \right]$$

this is a symmetric channel, as the rows and columns are identical.

$$ii) P(Y/X) = \begin{bmatrix} 1/2 & 1/4 & 1/4 \\ 1/4 & 1/4 & 1/2 \\ 1/2 & 1/4 & 1/4 \end{bmatrix}$$

$$\begin{aligned} H(X) + \\ H(Y) - \\ H(X;Y) \end{aligned}$$

This is not a symmetric matrix.

$$H(X) + H(Y) - \underline{\underline{[H(X) + H(Y)]}}$$

For a symmetric channel,

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

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

$$H(Y/X_j) = A$$

$$= H(Y) - A \underbrace{\sum_{j=1}^m P(X_j)}_{=1}$$

$$I(x, y) = H(y) - A$$

$$\begin{aligned} \text{Channel Capacity} &\leq \max [I(x, y)] \\ &= \max [H(y) - A] \\ &= \max [H(y)] - A \end{aligned}$$

$$C = \log n - A$$

$$\text{where } \log n = \max [H(y)]$$

Binary Symmetric channel (BSC):

The most important case of symmetric channel is the Binary Symmetric channel. In this case $m = n = 2$.

$$P(y/x) = \begin{bmatrix} p & 1-p \\ 1-p & p \end{bmatrix} = \begin{matrix} y_1 & y_2 \\ x_1 & \begin{bmatrix} p & q \\ q & p \end{bmatrix} \\ x_2 & \end{matrix}$$

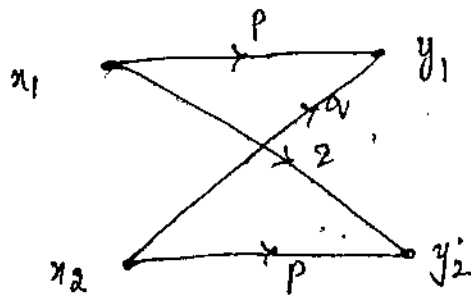


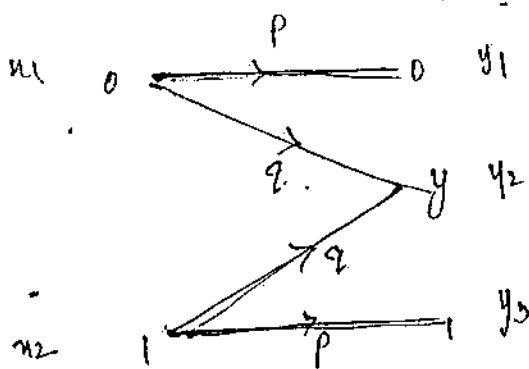
Fig: Practical representation of BSC
Graphical.

Binary Erasure channel (BEC):

It has two inputs 0, 1 and three outputs 0, ϵ , 1.
- If 0, and 1 are transmitted and they are received as 0, ϵ , 1.

Transmitter Receiver

$(0, 1)$ $(0, y_1, 1)$



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

Fig: Graphical representation of BEC.

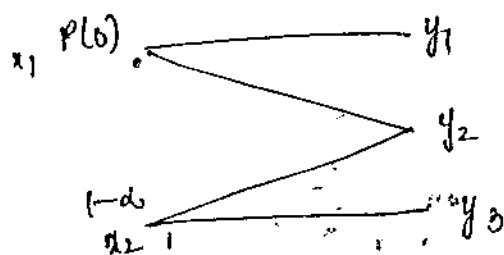
$$p(y|x) = \begin{matrix} & y_1 & y_2 & y_3 \\ \begin{matrix} x_1 \\ x_2 \end{matrix} & \begin{bmatrix} p & q & 0 \\ 0 & q & p \end{bmatrix} \end{matrix}$$

det $p(0) = \alpha$
 $p(1) = 1 - \alpha$
 can transmitted

$$x_1 = \alpha$$

$$\alpha_1 = \alpha$$

$$x_2 = 1 - \alpha$$



$$H(x) = -[\alpha \log \alpha + (1 - \alpha) \log (1 - \alpha)]$$

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

$$p(x,y) = p(y|x) \cdot p(x)$$

$$p(y|x) = \begin{matrix} & y_1 & y_2 & y_3 \\ \begin{matrix} x_1 \\ x_2 \end{matrix} & \begin{bmatrix} p & q & 0 \\ 0 & q & p \end{bmatrix} \end{matrix}$$

$$P(X, Y) = \begin{bmatrix} (p)(1-d) & (q)(1-d) & (0)(1-d) \\ (0)(1-d) & q(1-d) & p(1-d) \end{bmatrix}$$

or: rows of $P(X)$
 is multiplied by
 each column of $P(Y/X)$

$$P(X, Y) = \begin{bmatrix} y_1 & y_2 & y_3 \\ p d & q d & 0 \\ 0 & (1-d)q & (1-d)p \end{bmatrix}$$

$$P(X/Y) = \frac{P(X, Y)}{P(Y)} \quad , \quad P(Y) = [p d + q d + q, (1-d)p]$$

$$= [p d + q, (1-d)p]$$

$$P(X/Y) = \begin{bmatrix} \frac{p d}{p d + q} & \frac{q d}{p d + q} & \frac{0}{p d + q} \\ \frac{0}{(1-d)p} & \frac{(1-d)q}{(1-d)p} & \frac{(1-d)p}{(1-d)p} \end{bmatrix}$$

Each column is divided by each element of 1

$$P(X/Y) = \begin{bmatrix} 1 & d & 0 \\ 0 & (1-d) & 1 \end{bmatrix}$$

$$H(X/Y) = - \sum_{j=1}^m \sum_{k=1}^n P(x_j, y_k) \log \frac{1}{P(x_j/y_k)}$$

$$= - \sum_{j=1}^m \sum_{k=1}^n P(x_j, y_k) \log P(x_j/y_k)$$

$$= - [p d \log 1 + q d \log d + 0 \log 0 + 0 \log 0 + (1-d)q \log (1-d) + (1-d)p \log 1]$$

[$\because \log 1 = 0$]

$$= - [q d \log d + (1-d)q \log (1-d)]$$

$$= -q [d \log d + (1-d) \log (1-d)]$$

$$= p \{-[a \log a + (1-a) \log (1-a)]\}$$

$$H(X/Y) = p H(X).$$

The mutual information is,

$$I(X,Y) = H(X) - H(X/Y)$$

$$= H(X) - p H(X)$$

$$I(X,Y) = H(X) [1-p]$$

$$I(X,Y) = H(X) p$$

Channel Capacity: $C = \max I(X,Y)$

$$= \max [p H(X)]$$

$$= p \max [H(X)]$$

$$= p$$

Since max of $H(X) = 1$
 $\max [H(X)] = 1$

UNIT III

Linear Block Codes

Introduction

Coding theory is concerned with the transmission of data across noisy channels and the recovery of corrupted messages. It has found widespread applications in electrical engineering, digital communication, mathematics and computer science. The transmission of the data over the channel depends upon two parameters. They are transmitted power and channel bandwidth. The power spectral density of channel noise and these two parameters determine signal to noise power ratio.

The signal to noise power ratio determine the probability of error of the modulation scheme. Errors are introduced in the data when it passes through the channel. The channel noise interferes the signal. The signal power is reduced. For the given signal to noise ratio, the error probability can be reduced further by using coding techniques. The coding techniques also reduce signal to noise power ratio for fixed probability of error.

Principle of block coding

For the block of k message bits, $(n-k)$ parity bits or check bits are added. Hence the total bits at the output of channel encoder are ' n '. Such codes are called (n,k) block codes. Figure illustrates this concept.

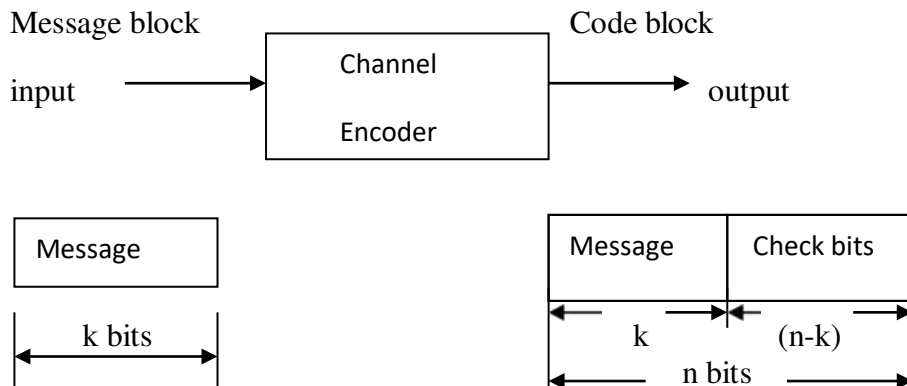


Figure: Functional block diagram of block coder

Types are

Systematic codes:

In the systematic block code, the message bits appear at the beginning of the code word. The message appears first and then check bits are transmitted in a block. This type of code is called systematic code.

Nonsystematic codes:

In the nonsystematic block code it is not possible to identify the message bits and check bits. They are mixed in the block.

Consider the binary codes and all the transmitted digits are binary.

Linear Block Codes

A code is linear if the sum of any two code vectors produces another code vector. This shows that any code vector can be expressed as a linear combination of other code vectors. Consider that the particular code vector consists of $m_1, m_2, m_3, \dots, m_k$ message bits and $c_1, c_2, c_3, \dots, c_q$ check bits. Then this code vector can be written as,

$$X = (m_1, m_2, m_3, \dots, m_k, c_1, c_2, c_3, \dots, c_q)$$

Here $q = n - k$

Where q are the number of redundant bits added by the encoder.

Code vector can also be written as

$$X = (M/C)$$

Where $M = k$ -bit message vector

$C = q$ -bit check vector

The main aim of linear block code is to generate check bits and this check bits are mainly used for error detection and correction.

Example :

The (7, 4) linear code has the following matrix as a generator matrix

$$\mathbf{G} = \begin{bmatrix} \mathbf{g}_0 \\ \mathbf{g}_1 \\ \mathbf{g}_2 \\ \mathbf{g}_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{aligned} \text{If } u = (1 \ 1 \ 0 \ 1) \text{ is the message to be encoded, its corresponding code word would be} \\ \mathbf{v} &= 1 \cdot \mathbf{g}_0 + 1 \cdot \mathbf{g}_1 + 0 \cdot \mathbf{g}_2 + 1 \cdot \mathbf{g}_3 \\ &= (1101000) + (0110100) + (1010001) \\ &= (0001101) \end{aligned}$$

A linear systematic (n, k) code is completely specified by $k \times n$ matrix G of the following form

$$\mathbf{G} = \begin{bmatrix} \mathbf{g}_0 \\ \mathbf{g}_1 \\ \mathbf{g}_2 \\ \vdots \\ \mathbf{g}_{k-1} \end{bmatrix} = \begin{array}{c|c} \begin{array}{cccccc} \longleftarrow & & \text{P matrix} & & \longrightarrow & \end{array} & \begin{array}{c} \longleftarrow k \times k \text{ identity matrix} \longrightarrow \end{array} \\ \begin{bmatrix} p_{00} & p_{01} & \cdot & \cdot & \cdot & p_{0,n-k-1} \\ p_{10} & p_{11} & \cdot & \cdot & \cdot & p_{1,n-k-1} \\ p_{20} & p_{21} & \cdot & \cdot & \cdot & p_{2,n-k-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ p_{k-1,0} & p_{k-1,1} & \cdot & \cdot & \cdot & p_{k-1,n-k-1} \end{bmatrix} & \begin{bmatrix} 1 & 0 & 0 & \cdot & \cdot & \cdot & 0 \\ 0 & 1 & 0 & \cdot & \cdot & \cdot & 0 \\ 0 & 0 & 1 & \cdot & \cdot & \cdot & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \end{array}$$

where $p_{ii} = 0$ or 1

Let $\mathbf{u} = (u_0, u_1, \dots, u_{k-1})$ be the message to be encoded. The corresponding code word is

$$\begin{aligned} \mathbf{v} &= (v_0, v_1, v_2, \dots, v_{n-1}) \\ &= (\mathbf{u}_0, \mathbf{u}_1, \dots, \mathbf{u}_{k-1}) \cdot \mathbf{G} \end{aligned}$$

The components of $\mathbf{v}$ are

$$v_{n-k+i} = u_i \quad \text{for } 0 \leq i < k$$

$$v_j = u_0 p_{0j} + u_1 p_{1j} + \dots + u_{k-1} p_{k-1,j} \quad \text{for } 0 \leq j < n-k$$

The $n - k$ equations given by above equation are called parity-check equations of the code

Example for Codeword

The matrix $\mathbf{G}$ given by

$$\mathbf{G} = \begin{bmatrix} \mathbf{g}_0 \\ \mathbf{g}_1 \\ \mathbf{g}_2 \\ \mathbf{g}_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Let $\mathbf{u} = (u_0, u_1, u_2, u_3)$ be the message to be encoded and $\mathbf{v} = (v_0, v_1, v_2, v_3, v_4, v_5, v_6)$ be the corresponding code word

Solution :

$$\mathbf{v} = \mathbf{u} \cdot \mathbf{G} = (u_0, u_1, u_2, u_3) \cdot \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

By matrix multiplication, the digits of the code word $\mathbf{v}$ can be determined.

$$v_6 = u_3$$

$$v_5 = u_2$$

$$v_4 = u_1$$

$$v_3 = u_0$$

$$v_2 = u_1 + u_2 + u_3$$

$$v_1 = u_0 + u_1 + u_2$$

$$v_0 = u_0 + u_2 + u_3$$

The code word corresponding to the message (1 0 1 1) is (1 0 0 1 0 1 1)

If the generator matrix of an (n, k) linear code is in systematic form, the parity-check matrix may take the following form

$$\mathbf{H} = \begin{bmatrix} \mathbf{I}_{n-k} & \mathbf{P}^T \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & \cdot & \cdot & \cdot & 0 & p_{00} & p_{10} & \cdot & \cdot & \cdot & p_{k-1,0} \\ 0 & 1 & 0 & \cdot & \cdot & \cdot & 0 & p_{01} & p_{11} & \cdot & \cdot & \cdot & p_{k-1,1} \\ 0 & 0 & 1 & \cdot & \cdot & \cdot & 0 & p_{02} & p_{12} & \cdot & \cdot & \cdot & p_{k-1,2} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & \cdot & \cdot & \cdot & 1 & p_{0,n-k-1} & p_{1,n-k-1} & \cdot & \cdot & \cdot & p_{k-1,n-k-1} \end{bmatrix}$$

Encoding circuit for a linear systematic (n,k) code is shown below.

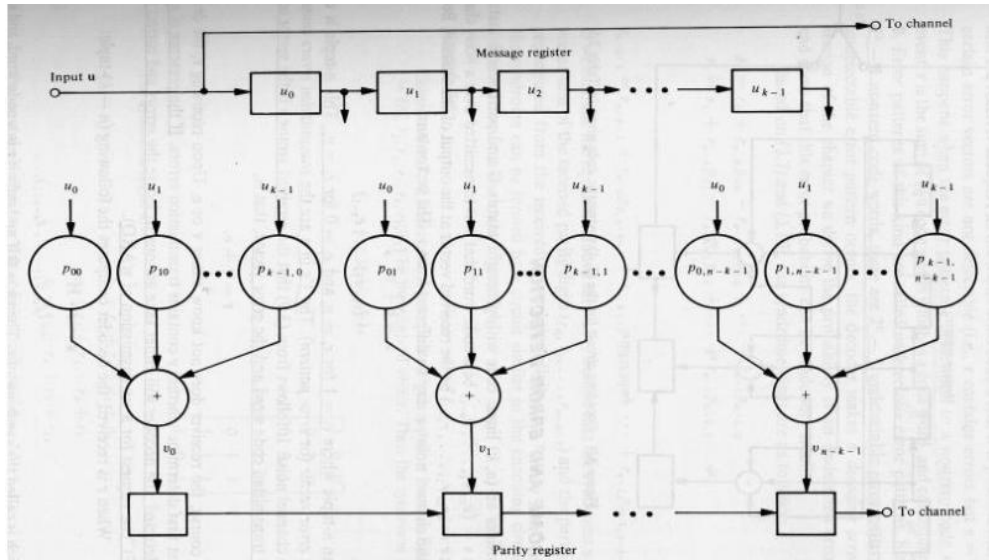


Figure: Encoding Circuit

For the block of $k=4$ message bits, $(n-k)$ parity bits or check bits are added. Hence the total bits at the output of channel encoder are $n=7$. The encoding circuit for $(7, 4)$ systematic code is shown below.

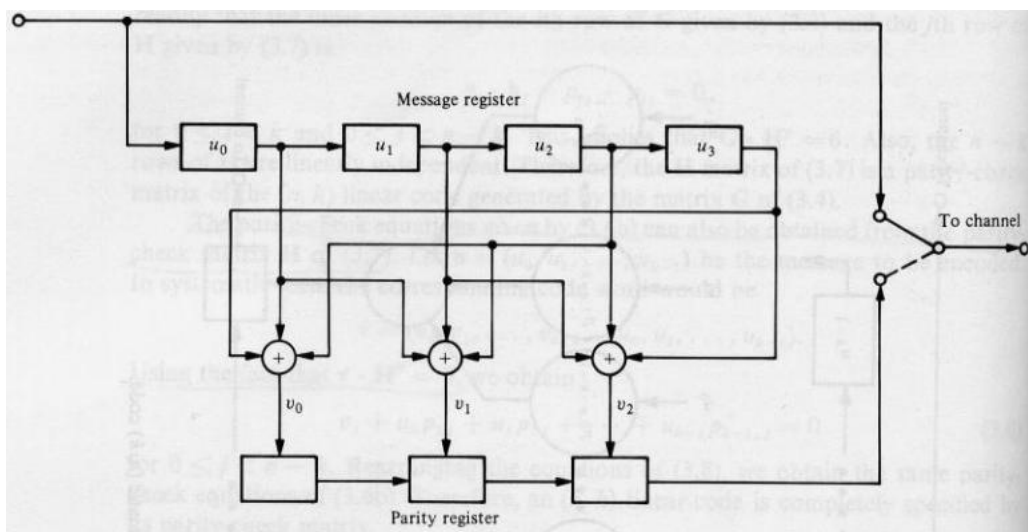
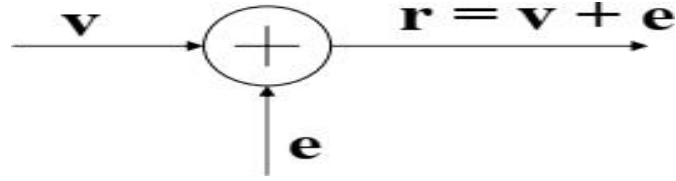


Figure: Encoding Circuit for $(7,4)$ code

Syndrome and Error Detection

Let $v = (v_0, v_1, \dots, v_{n-1})$ be a code word that was transmitted over a noisy channel. Let $r = (r_0, r_1, \dots, r_{n-1})$ be the received vector at the output of the channel



Where

$e = r + v = (e_0, e_1, \dots, e_{n-1})$ is an n -tuple and the n -tuple 'e' is called the error vector (or error pattern). The condition is

$$e_i = 1 \text{ for } r_i \neq v_i$$

$$e_i = 0 \text{ for } r_i = v_i$$

Upon receiving r , the decoder must first determine whether r contains transmission errors. If the presence of errors is detected, the decoder will take actions to locate the errors, correct errors (FEC) and request for a retransmission of v .

When r is received, the decoder computes the following $(n - k)$ -tuple.

$$s = r \cdot HT$$

$$s = (s_0, s_1, \dots, s_{n-k-1})$$

where s is called the syndrome of r .

The syndrome is not a function of the transmitted codeword but a function of error pattern. So we can construct only a matrix of all possible error patterns with corresponding syndrome.

When $s = 0$, if and only if r is a code word and hence receiver accepts r as the transmitted code word. When $s \neq 0$, if and only if r is not a code word and hence the presence of errors has been detected. When the error pattern e is identical to a nonzero code word (i.e., r contain errors but $s = r \cdot HT = 0$), error patterns of this kind are called undetectable error patterns. Since there are $2^k - 1$ non-zero code words, there are $2^k - 1$ undetectable error patterns. The syndrome digits are as follows:

$$s_0 = r_0 + r_{n-k} p_{00} + r_{n-k+1} p_{10} + \dots + r_{n-1} p_{k-1,0}$$

$$s_1 = r_1 + r_{n-k} p_{01} + r_{n-k+1} p_{11} + \dots + r_{n-1} p_{k-1,1}$$

.

$$s_{n-k-1} = r_{n-k-1} + r_{n-k} p_{0,n-k-1} + r_{n-k+1} p_{1,n-k-1} + \dots + r_{n-1} p_{k-1,n-k-1}$$

The syndrome s is the vector sum of the received parity digits $(r_0, r_1, \dots, r_{n-k-1})$ and the parity-check digits recomputed from the received information digits $(r_{n-k}, r_{n-k+1}, \dots, r_{n-1})$.

The below figure shows the syndrome circuit for a linear systematic (n, k) code.

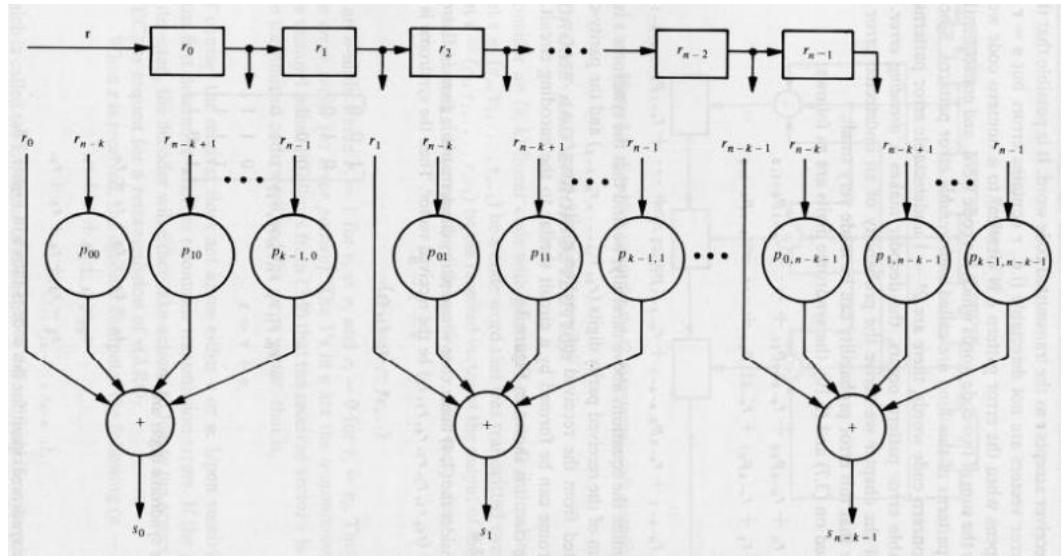


Figure: Syndrome Circuit

Error detection and error correction capabilities of linear block codes:

If the minimum distance of a block code C is $d_{\min}$, any two distinct code vector of C differ in at least $d_{\min}$ places. A block code with minimum distance $d_{\min}$ is capable of detecting all the error pattern of $d_{\min} - 1$ or fewer errors.

However, it cannot detect all the error pattern of $d_{\min}$ errors because there exists at least one pair of code vectors that differ in $d_{\min}$ places and there is an error pattern of $d_{\min}$ errors that will carry one into the other. The random-error-detecting capability of a block code with minimum distance $d_{\min}$ is $d_{\min} - 1$.

An (n, k) linear code is capable of detecting $2^n - 2^k$ error patterns of length n . Among the $2^n - 1$ possible non zero error patterns, there are $2^k - 1$ error patterns that are identical to the $2^k - 1$ non zero code words. If any of these $2^k - 1$ error patterns occurs, it alters the transmitted code word v into another code word w , thus w will be received and its syndrome is zero.

If an error pattern is not identical to a nonzero code word, the received vector r will not be a code word and the syndrome will not be zero.

Hamming Codes:

These codes and their variations have been widely used for error control in digital communication and data storage systems.

For any positive integer $m \geq 3$, there exists a Hamming code with the following parameters:

Code length: $n = 2^m - 1$

Number of information symbols: $k = 2^m - m - 1$

Number of parity-check symbols: $n - k = m$

Error-correcting capability: $t = 1 (d_{\min} = 3)$

The parity-check matrix H of this code consists of all the non zero m -tuple as its columns (2^m-1)

In systematic form, the columns of H are arranged in the following form

$$H = [I_m \ Q]$$

where I_m is an $m \times m$ identity matrix

The sub matrix Q consists of $2^m - m - 1$ columns which are the m -tuples of weight 2 or more. The columns of Q may be arranged in any order without affecting the distance property and weight distribution of the code.

In systematic form, the generator matrix of the code is

$$G = [Q^T \ I_{2^m-m-1}]$$

where Q^T is the transpose of Q and I_{2^m-m-1} is an $(2^m - m - 1) \times (2^m - m - 1)$ identity matrix.

Since the columns of H are nonzero and distinct, no two columns add to zero. Since H consists of all the nonzero m -tuples as its columns, the vector sum of any two columns, say h_i and h_j , must also be a column in H , say h_l . $h_i + h_j + h_l = 0$. The minimum distance of a Hamming code is exactly 3.

Using H' as a parity-check matrix, a shortened Hamming code can be obtained with the following parameters :

Code length: $n = 2^m - 1 - 1$

Number of information symbols: $k = 2^m - m - 1 - 1$

Number of parity-check symbols: $n - k = m$

Minimum distance : $d_{\min} \geq 3$

When a single error occurs during the transmission of a code vector, the resultant syndrome is nonzero and it contains an odd number of 1's ($e \times H'^T$ corresponds to a column in H'). When double errors occurs, the syndrome is nonzero, but it contains even number of 1's.

Decoding can be accomplished in the following manner:

- i) If the syndrome s is zero, we assume that no error occurred
- ii) If s is nonzero and it contains odd number of 1's, assume that a single error occurred. The error pattern of a single error that corresponds to s is added to the received vector for error correction.
- iii) If s is nonzero and it contains even number of 1's, an uncorrectable error pattern has been detected.

Problems:

1.

The parity check bits of a (8,4) block code are generated by

$$c_0 = m_0 + m_1 + m_3$$

$$c_1 = m_0 + m_1 + m_2$$

$$c_2 = m_0 + m_2 + m_3$$

$$c_3 = m_1 + m_2 + m_3$$

where m_1, m_2, m_3 and m_4 are the message digits.

- Find the generator matrix and the parity check matrix for this code.
- Find the minimum weight of this code.
- Find the error-detecting capabilities of this code.
- Show through an example that this code can detect three errors/codeword.

Solution

$$1(a) \quad \mathbf{c} = [c_0 \cdots c_3] = [b_0 \cdots b_3 m_0 \cdots m_3] = [m_0 \cdots m_3] \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \end{bmatrix} \mathbf{I}_4$$

$$\text{Therefore, } \mathbf{G} = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \end{bmatrix} \mathbf{I}_4$$

$$\text{and then } \mathbf{H} = \begin{bmatrix} 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix}$$

(b)

m	C
0000	0000 0000
0001	1011 0001
0010	0111 0010
0011	1100 0011
0100	1101 0100
0101	0110 0101
0110	1010 0110
0111	0001 0111
1000	1110 1000
1001	0101 1001
1010	1001 1010
1011	0010 1011
1100	0011 1100
1101	1000 1101
1110	0100 1110
1111	1111 1111

Therefore, minimum weight = 4

(c) $d_{\min} = \text{minimum weight} = 4$

Therefore, error-detecting capability = $d_{\min} - 1 = 3$

(d) Suppose the transmitted code be 00000000 and the received code be 11100000.

$$\mathbf{s} = \mathbf{rH}^T = [11100000] \begin{bmatrix} 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix}^T = [1110] \neq \mathbf{0}$$

Binary Cyclic codes:

Cyclic codes are the sub class of linear block codes.

Cyclic codes can be in systematic or non systematic form.

Definition:

A linear code is called a cyclic code if every cyclic shift of the code vector produces some other code vector.

Properties of cyclic codes:

- (i) Linearity
- (ii) Cyclic

Linearity: This property states that sum of any two code words is also a valid code word.

$$X_1 + X_2 = X_3$$

Cyclic: Every cyclic shift of valid code vector produces another valid code vector.

Consider an n-bit code vector

$$X = \{x_{n-1}, x_{n-2}, \dots, x_1, x_0\}$$

Here $x_{n-1}, x_{n-2}, \dots, x_1, x_0$ represent individual bits of the code vector 'X'.

If the above code vector is cyclically shifted to left side i.e., One cyclic shift of X gives,

$$X' = \{x_{n-2}, \dots, x_1, x_0, x_{n-1}\}$$

Every bit is shifted to left by one position.

Algebraic Structures of Cyclic Codes:

The code words can be represented by a polynomial. For example consider the n-bit code word $X = \{x_{n-1}, x_{n-2}, \dots, x_1, x_0\}$.

UNIT IV

Binary Cyclic Codes

Binary Cyclic codes:

Cyclic codes are the sub class of linear block codes.

Cyclic codes can be in systematic or non systematic form.

Definition:

A linear code is called a cyclic code if every cyclic shift of the code vector produces some other code vector.

Properties of cyclic codes:

(i) Linearity

(ii) Cyclic

Linearity: This property states that sum of any two code words is also a valid code word.

$$X_1 + X_2 = X_3$$

Cyclic: Every cyclic shift of valid code vector produces another valid code vector.

Consider an n-bit code vector

$$X = \{x_{n-1}, x_{n-2}, \dots, x_1, x_0\}$$

Here $x_{n-1}, x_{n-2}, \dots, x_1, x_0$ represent individual bits of the code vector 'X'.

If the above code vector is cyclically shifted to left side i.e., One cyclic shift of X gives,

$$X' = \{x_{n-2}, \dots, x_1, x_0, x_{n-1}\}$$

Every bit is shifted to left by one position.

Algebraic Structures of Cyclic Codes:

The code words can be represented by a polynomial. For example consider the n-bit code word $X = \{x_{n-1}, x_{n-2}, \dots, x_1, x_0\}$.

This code word can be represented by a polynomial of degree less than or equal to (n-1) i.e.,

$$X(p) = x_{n-1}p^{n-1} + x_{n-2}p^{n-2} + \dots + x_1p + x_0$$

Here X(p) is the polynomial of degree (n-1)

p- Arbitrary variable of the polynomial

The power of p represents the positions of the codeword bits i.e.,

p^{n-1} – MSB

p^0 -- LSB

p -- Second bit from LSB side

Polynomial representation due to the following reasons

- (i) These are algebraic codes, algebraic operations such as addition, multiplication, division, subtraction etc becomes very simple.
- (ii) Positions of the bits are represented with help of powers of p in a polynomial.

Generation of code words in Non-systematic form:

Let $M = \{m_{k-1}, m_{k-2}, \dots, m_1, m_0\}$ be 'k' bits of message vector. Then it can be represented by the polynomial as,

$$M(p) = m_{k-1}p^{k-1} + m_{k-2}p^{k-2} + \dots + m_1p + m_0$$

Let X(p) be the code word polynomial

$$X(p) = M(p)G(p)$$

G(p) is the generating polynomial of degree 'q'

For (n,k) cyclic codes, $q = n - k$ represent the number of parity bits.

The generating polynomial is given as

$$G(p) = p^q + g_{q-1}p^{q-1} + \dots + g_1p + 1$$

Where $g_{q-1}, g_{q-2}, \dots, g_1$ are the parity bits.

If $M_1, M_2, M_3, \dots$ etc are the other message vectors, then the corresponding code vectors can be calculated as

$$X_1(p) = M_1(p) G(p)$$

$$X_2(p) = M_2(p) G(p)$$

$$X_3(p) = M_3(p) G(p)$$

Generation of Code vectors in systematic form:

$$X = (k \text{ message bits} : (n-k) \text{ check bits}) = (m_{k-1}, m_{k-2}, \dots, m_1, m_0 : c_{q-1}, c_{q-2}, \dots, c_1, c_0)$$

$$C(p) = c_{q-1}p^{q-1} + c_{q-2}p^{q-2} + \dots + c_1p + c_0$$

The check bit polynomial is obtained by

$$C(p) = \text{rem} \left[\frac{p^q M(p)}{G(p)} \right]$$

Generator and Parity Check Matrices of cyclic codes:

Non systematic form of generator matrix:

Since cyclic codes are sub class of linear block codes, generator and parity check matrices can also be defined for cyclic codes.

The generator matrix has the size of $k \times n$.

Let generator polynomial given by equation

$$G(p) = p^q + g_{q-1}p^{q-1} + \dots + g_1p + 1$$

Multiply both sides of this polynomial by p^i i.e.,

$$p^i G(p) = p^{i+q} + g_{q-1}p^{i+q-1} + \dots + g_1p^{i+1} + p^i \text{ and } i=(k-1), (k-2), \dots, 2, 1, 0$$

Systematic form of generator matrix:

Systematic form of generator matrix is given by

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

The t^{th} row of this matrix will be represented in the polynomial form as follows

$$t^{\text{th}} \text{ row of } G = p^{n-t} + R_t(p)$$

Where $t = 1, 2, 3, \dots, k$

Lets divide p^{n-t} by a generator matrix $G(p)$. Then we express the result of this division in terms of quotient and remainder i.e.,

$$\frac{p^{n-t}}{G(p)} = \text{Quotient} + \frac{\text{Remainder}}{G(p)}$$

Here remainder will be a polynomial of degree less than q , since the degree of $G(p)$ is ' q '.

The degree of quotient will depend upon value of t

Lets represent Remainder = $R_t(p)$

Quotient = $Q_t(p)$

$$\frac{p^{n-t}}{G(p)} = Q_t(p) + \frac{R_t(p)}{G(p)}$$

$$p^{n-t} = Q_t(p)G(p) + R_t(p)$$

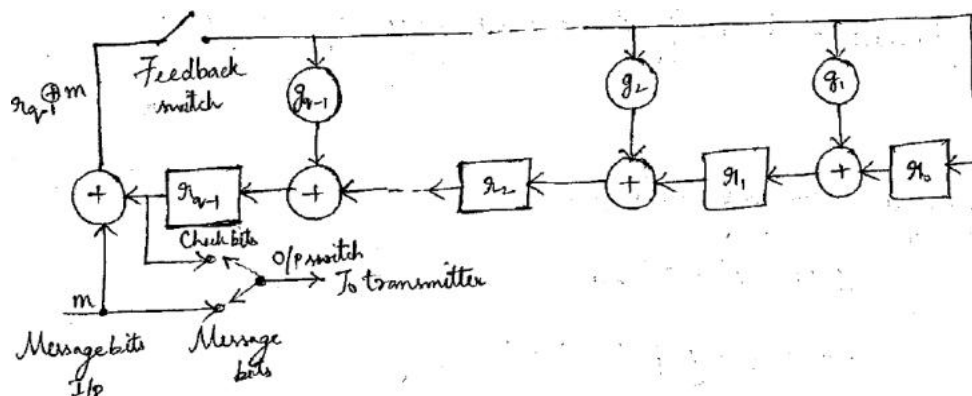
And $t = 1, 2, \dots, k$

$$p^{n-t} + R_t(p) = Q_t(p)G(p)$$

Represents t^{th} row of systematic generator matrix

Parity check matrix $H = [P^T : I_q]_{q \times n}$

Encoding using an (n-k) Bit Shift Register:



The feedback switch is first closed. The output switch is connected to message input. All the shift registers are initialized to zero state. The ' k ' message bits are shifted to the transmitter as well as shifted to the registers.

After the shift of 'k' message bits the registers contain 'q' check bits. The feedback switch is now opened and output switch is connected to check bits position. With the every shift, the check bits are then shifted to the transmitter.

The block diagram performs the division operation and generates the remainder. Remainder is stored in the shift register after all message bits are shifted out.

Syndrome Decoding, Error Detection and Error Correction:

In cyclic codes also during transmission some errors may occur. Syndrome decoding can be used to correct those errors.

Lets represent the received code vector by Y.

If 'E' represents the error vector then the correct code vector can be obtained as

$$\mathbf{X}=\mathbf{Y}+\mathbf{E} \text{ or } \mathbf{Y}=\mathbf{X}+\mathbf{E}$$

In the polynomial form we can write above equation as

$$Y(p) = X(p)+E(p)$$

$$X(p) = M(p)G(p)$$

$$Y(p)= M(p)G(p) + E(p)$$

$$\frac{Y(p)}{G(p)} = Quotient + \frac{Remainder}{G(p)}$$

If $Y(p)=X(p)$

$$\frac{X(p)}{G(p)} = Quotient + \frac{Remainder}{G(p)}$$

$$\frac{Y(p)}{G(p)} = Q(p) + \frac{R(p)}{G(p)}$$

$$Y(p)=Q(p)G(p) + R(p)$$

Clearly R(p) will be the polynomial of degree less than or equal to q-1

$$Y(p) = Q(p) G(p) + R(p)$$

$$M(p)G(p)+E(p)=Q(p)G(p)+R(p)$$

$$E(p)=M(p)G(p)+Q(p)G(p)+ R(p)$$

$$E(p)=[M(p)+Q(p)]G(p)+R(p)$$

This equation shows that for a fixed message vector and generator polynomial, an error pattern or error vector 'E' depends on remainder R.

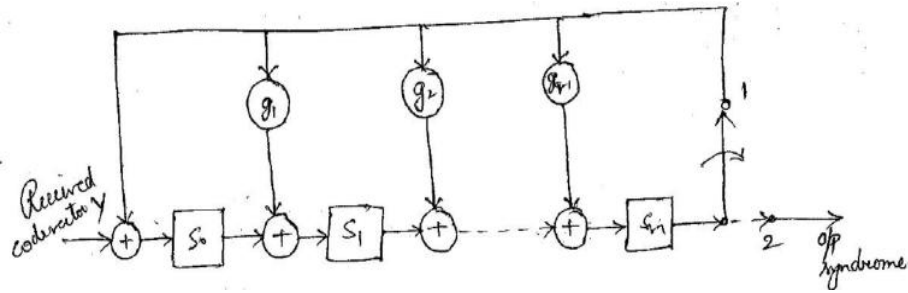
For every remainder 'R' there will be specific error vector. Therefore we can call the remainder vector 'R' as syndrome vector 'S', or $R(p)=S(p)$. Therefore

$$\frac{Y(p)}{G(p)} = Q(p) + \frac{S(p)}{G(p)}$$

Thus Syndrome vector is obtained by dividing received vector Y (p) by G (p) i.e.,

$$S(p) = rem\left[\frac{Y(p)}{G(p)}\right]$$

Block Diagram of Syndrome Calculator:



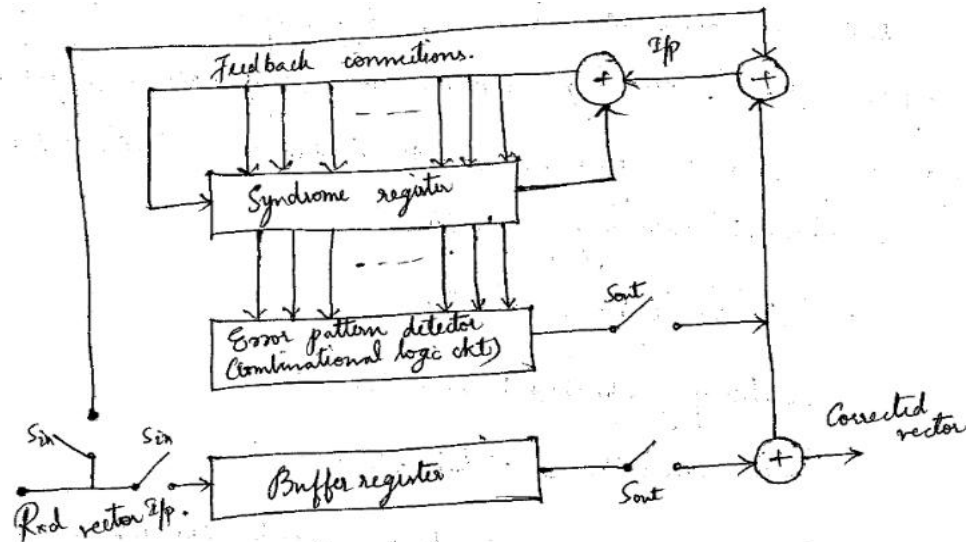
There are 'q' stage shift register to generate 'q' bit syndrome vector. Initially all the shift register contents are zero & the switch is closed in position 1.

The received vector Y is shifted bit by bit into the shift register. The contents of flip flops keep changing according to input bits of Y and values of g_1, g_2 etc.

After all the bits of Y are shifted, the 'q' flip flops of shift register contain the q bit syndrome vector. The switch is then closed to position 2 & clocks are applied to shift register. The output is a syndrome vector $S = (S_{q-1}, S_{q-2} \dots S_1, S_0)$

Decoder of Cyclic Codes:

Once the syndrome is calculated, then an error pattern is detected for that particular syndrome. When the error vector is added to the received code vector Y, then it gives corrected code vector at the output.



The switch named S_{out} is opened and S_{in} is closed. The bits of the received vector Y are shifted into the buffer register as well as they are shifted in to the syndrome calculator. When all the n bits of the received vector Y are shifted into the buffer register and Syndrome calculator the syndrome register holds a syndrome vector.

Syndrome vector is given to the error pattern detector. A particular syndrome detects a specific error pattern.

S_{in} is opened and S_{out} is closed. Shifts are then applied to the flip flop of buffer registers, error register, and syndrome register.

The error pattern is then added bit by bit to the received vector. The output is the corrected error free vector.

Unit-5

Convolution codes

Definition of Convolutional Coding

A convolutional coding is done by combining the fixed number of input bits. The input bits are stored in the fixed length shift register and they are combined with the help of mod-2 adders. This operation is equivalent to binary convolution and hence it is called *convolutional coding*. This concept is illustrated with the help of simple example given below.

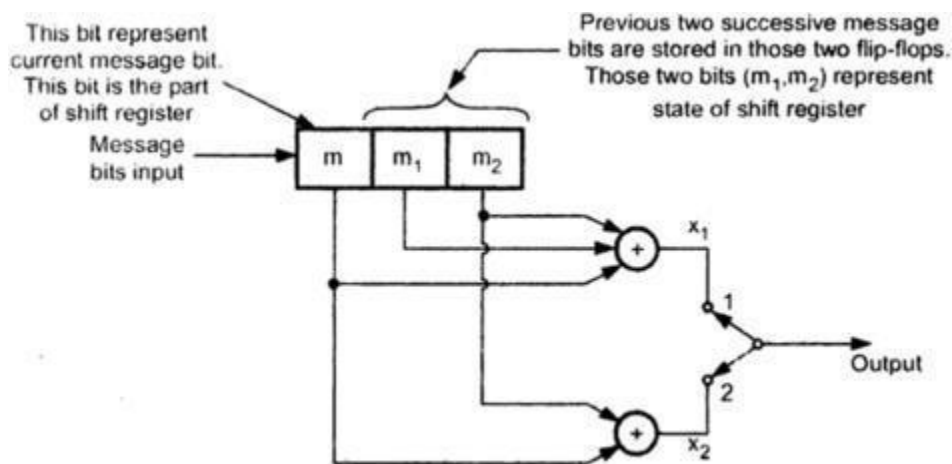


Fig. 4.4.1 Convolutional encoder with $k = 3$, $k = 1$ and $n = 2$

Operation :

Whenever the message bit is shifted to position 'm', the new values of x_1 and x_2 are generated depending upon m , m_1 and m_2 . m_1 and m_2 store the previous two message bits. The current bit is present in m . Thus we can write,

$$x_1 = m \oplus m_1 \oplus m_2 \quad \dots (4.4.1)$$

and
$$x_2 = m \oplus m_2 \quad \dots (4.4.2)$$

The output switch first samples x_1 and then x_2 . The shift register then shifts contents of m_1 to m_2 and contents of m to m_1 . Next input bit is then taken and stored in m . Again x_1 and x_2 are generated according to this new combination of m , m_1 and m_2 (equation 4.4.1 and equation 4.4.2). The output switch then samples x_1 then x_2 . Thus the output bit stream for successive input bits will be,

$$X = x_1x_2x_1x_2x_1x_2 \dots \text{ and so on} \quad \dots (4.4.3)$$

Here note that for every input message bit two encoded output bits x_1 and x_2 are transmitted. In other words, for a single message bit, the encoded code word is two bits i.e. for this convolutional encoder,

Number of message bits, $k = 1$

Number of encoded output bits for one message bit, $n = 2$

4.4.1.1 Code Rate of Convolutional Encoder

The code rate of this encoder is,

$$r = \frac{k}{n} = \frac{1}{2} \quad \dots (4.4.4)$$

In the encoder of Fig. 4.4.1, observe that whenever a particular message bit enters a shift register, it remains in the shift register for three shifts i.e.,

First shift $\rightarrow$ Message bit is entered in position ' m '.

Second shift $\rightarrow$ Message bit is shifted in position m_1 .

Third shift $\rightarrow$ Message bit is shifted in position m_2 .

And at the fourth shift the message bit is discarded or simply lost by overwriting. We know that x_1 and x_2 are combinations of m , m_1 , m_2 . Since a single message bit remains in m during first shift, in m_1 during second shift and in m_2 during third shift; it influences output x_1 and x_2 for 'three' successive shifts.

4.4.1.2 Constraint Length (K)

The constraint length of a convolution code is defined as the number of shifts over which a single message bit can influence the encoder output. It is expressed in terms of message bits.

For the encoder of Fig. 4.4.1 constraint length $K = 3$ bits. This is because in this encoder, a single message bit influences encoder output for three successive shifts. At the fourth shift, the message bit is lost and it has no effect on the output.

4.4.1.3 Dimension of the Code

The dimension of the code is given by n and k . We know that ' k ' is the number of message bits taken at a time by the encoder. And ' n ' is the encoded output bits for one message bits. Hence the dimension of the code is (n, k) . And such encoder is called (n, k) convolutional encoder. For example, the encoder of Fig. 4.4.1 has the dimension of $(2, 1)$.

4.4.2 Time Domain Approach to Analysis of Convolutional Encoder

Let the sequence $\{g_0^{(1)}, g_1^{(1)}, g_2^{(1)}, \dots, g_m^{(1)}\}$ denote the impulse response of the adder which generates x_1 in Fig. 4.4.1. Similarly, Let the sequence $\{g_0^{(2)}, g_1^{(2)}, g_2^{(2)}, \dots, g_m^{(2)}\}$ denote the impulse response of the adder which generates x_2 in Fig. 4.4.1. These impulse responses are also called *generator sequences* of the code.

Let the incoming message sequence be $\{m_0, m_1, m_2, \dots\}$. The encoder generates the two output sequences x_1 and x_2 . These are obtained by convolving the generator sequences with the message sequence. Hence the name convolutional code is given. The sequence x_1 is given as,

$$x_1 = x_i^{(1)} = \sum_{l=0}^M g_l^{(1)} m_{i-l} \quad i = 0, 1, 2, \dots \quad \dots (4.4.6)$$

Here $m_{i-l} = 0$ for all $l > i$. Similarly the sequence x_2 is given as,

$$x_2 = x_i^{(2)} = \sum_{l=0}^M g_l^{(2)} m_{i-l} \quad i = 0, 1, 2, \dots \quad \dots (4.4.7)$$

Note : All additions in above equations are as per mod-2 addition rules.

As shown in the Fig. 4.4.1, the two sequences x_1 and x_2 are multiplexed by the switch. Hence the output sequence is given as,

$$\{x_i\} = \{x_0^{(1)} x_0^{(2)} x_1^{(1)} x_1^{(2)} x_2^{(1)} x_2^{(2)} x_3^{(1)} x_3^{(2)} \dots\} \quad \dots (4.4.8)$$

$$v_1 = x_i^{(1)} = \{x_0^{(1)} x_1^{(1)} x_2^{(1)} x_3^{(1)} \dots\}$$

$$v_2 = x_i^{(2)} = \{x_0^{(2)} x_1^{(2)} x_2^{(2)} x_3^{(2)} \dots\}$$

Observe that bits from above two sequences are multiplexed in equation (4.4.8). The sequence $\{x_i\}$ is the output of the convolutional encoder.

Transform Domain Approach to Analysis of Convolutional Encoder

In the previous section we observed that the convolution of generating sequence and message sequence takes place. These calculations can be simplified by applying the transformations to the sequences. Let the impulse responses be represented by polynomials. i.e.,

$$g^{(1)}(p) = g_0^{(1)} + g_1^{(1)}p + g_2^{(1)}p^2 + \dots + g_M^{(1)}p^M \quad \dots (4.4.13)$$

$$g^{(2)}(p) = g_0^{(2)} + g_1^{(2)}p + g_2^{(2)}p^2 + \dots + g_M^{(2)}p^M \quad \dots (4.4.14)$$

Thus the polynomials can be written for other generating sequences. The variable 'p' is unit delay operator in above equations. It represents the time delay of the bits in impulse response.

Similarly we can write the polynomial for message polynomial i.e.,

$$m(p) = m_0 + m_1p + m_2p^2 + \dots + m_{L-1}p^{L-1} \quad \dots (4.4.15)$$

Here L is the length of the message sequence. The convolution sums are converted to polynomial multiplications in the transform domain. i.e.,

$\begin{aligned} x^{(1)}(p) &= g^{(1)}(p) \cdot m(p) \\ x^{(2)}(p) &= g^{(2)}(p) \cdot m(p) \end{aligned}$
--

... (4.4.16)

The above equations are the output polynomials of sequences $x_i^{(1)}$ and $x_i^{(2)}$.

Code Tree, Trellis and State Diagram for a Convolution Encoder

Now let's study the operation of the convolutional encoder with the help of code tree, trellis and state diagram. Consider again the convolutional encoder of Fig. 4.4.1. It is reproduced below for convenience.

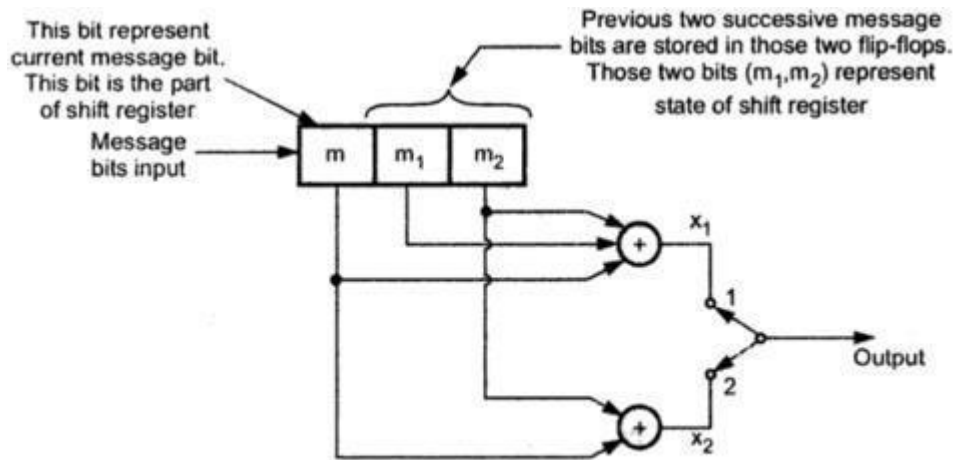


Fig. 4.4.4 Convolutional encoder with $k = 1$ and $n = 2$

States of the Encoder

In Fig. 4.4.4 the previous two successive message bits m_1 and m_2 represents state. The input message bit m affects the 'state' of the encoder as well as outputs x_1 and x_2 during that state. Whenever new message bit is shifted to ' m ', the contents of m_1 and m_2 define new state. And outputs x_1 and x_2 are also changed according to new state m_1, m_2 and message bit m . Let's define these states as shown in Table 4.4.1.

Let the initial values of bits stored in m_1 and m_2 be zero. That is $m_1 m_2 = 00$ initially and the encoder is in state 'a'.

m_2	m_1	State of encoder
0	0	a
0	1	b
1	0	c
1	1	d

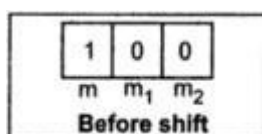
Table 4.4.1 States of the encoder of Fig. 4.4.4

Development of the Code Tree

Let us consider the development of code tree for the message sequence $m = 110$. Assume that $m_1 m_2 = 00$ initially.

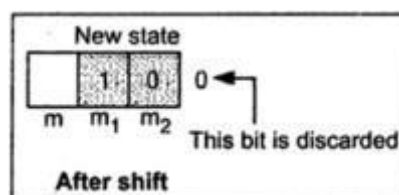
1) When $m=1$ i.e. first bit

The first message input is $m = 1$. With this input x_1 and x_2 will be calculated as



$$x_1 = 1 \oplus 0 \oplus 0 = 1$$

$$x_2 = 1 \oplus 0 = 1$$



The values of $x_1x_2 = 11$ are transmitted to the output and register contents are shifted to right by one bit position as shown.

Thus the new state of encoder is $m_2m_1 = 01$ or 'b' and output transmitted are $x_1x_2 = 11$. This shows that if encoder is in state 'a' and if input is $m = 1$ then the next state is 'b' and outputs are $x_1x_2 = 11$. The first row of Table 4.4.2 illustrates this operation.

The last column of this table shows the code tree diagram. The code tree diagram starts at node or state 'a'. The diagram is reproduced as shown in Fig. 4.4.5.

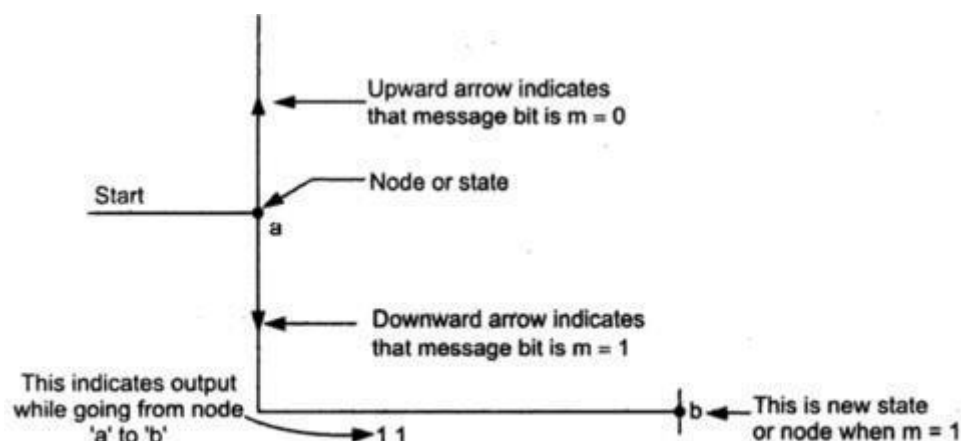
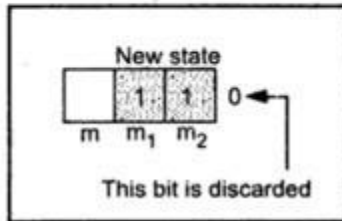
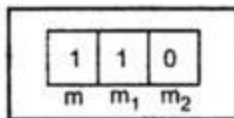


Fig. 4.4.5 Code tree from node 'a' to 'b'

Observe that if $m = 1$ we go downward from node 'a'. Otherwise if $m = 0$, we go upward from node 'a'. It can be verified that if $m = 0$ then next node (state) is 'a' only. Since $m = 1$ here we go downwards toward node b and output is 11 in this node (or state).

2) When $m = 1$ i.e. second bit

Now let the second message bit be 1. The contents of shift register with this input will be as shown below.



$$x_1 = 1 \oplus 1 \oplus 0 = 0$$

$$x_2 = 1 \oplus 0 = 1$$

These values of $x_1x_2 = 01$ are then transmitted to output and register contents are shifted to right by one bit. The next state formed is as shown.

Thus the new state of the encoder is $m_2m_1 = 11$ or 'd' and the outputs transmitted are $x_1x_2 = 01$. Thus the encoder goes from state 'b' to state 'd'.

if input is '1' and transmitted output $x_1x_2 = 01$. This operation is illustrated by Table 4.4.2 in second row. The last column of the table shows the code tree for those first and second input bits.

Example 4.4.8 : Determine the state diagram for the convolutional encoder shown in Fig. 4.4.32. Draw the trellis diagram through the first set of steady state transitions. On the second trellis diagram, show the termination of trellis to all zero state.

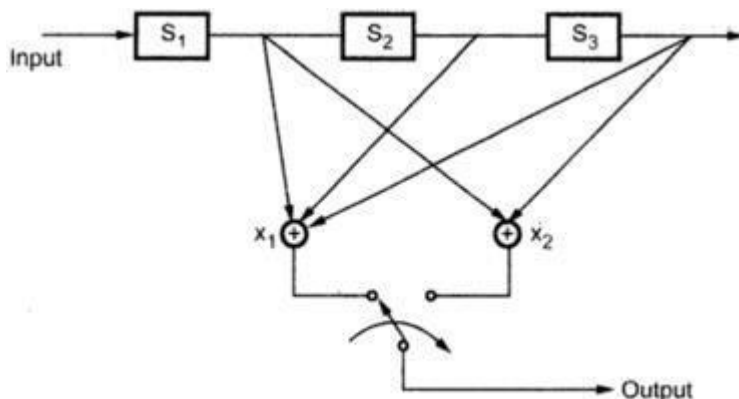


Fig. 4.4.32 Convolutional encoder of example 4.4.8

Sol. : (i) To determine dimension of the code :

For every message bit ($k=1$), two output bits ($n=2$) are generated. Hence this is rate $\frac{1}{2}$ code. Since there are three stages in the shift register, every message bit will affect output for three successive shifts. Hence constraint length, $K=3$. Thus,

$$k = 1, \quad n = 2 \quad \text{and} \quad K = 3$$

ii) To obtain the state diagram :

First, let us define the states of the encoder.

$$s_3 s_2 = 00, \quad \text{state 'a'}$$

$$s_3 s_2 = 01, \quad \text{state 'b'}$$

$$s_3 s_2 = 10, \quad \text{state 'c'}$$

$$s_3 s_2 = 11, \quad \text{state 'd'}$$

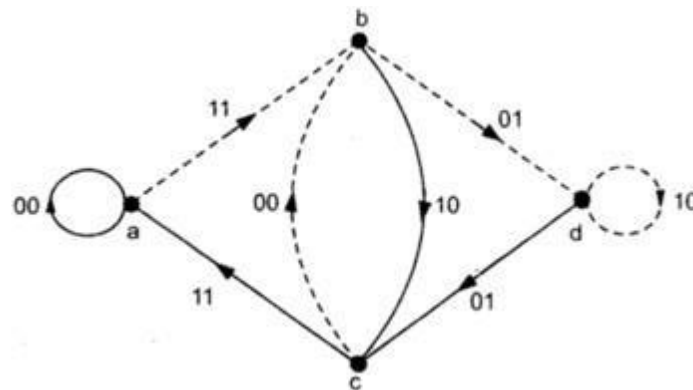
A table is prepared that lists state transitions, message input and outputs. The table is as follows :

Sr. No.	Current state $s_3 s_2$	Input s_1	Outputs $x_1 = s_1 \oplus s_2 \oplus s_3$ $x_2 = s_1 \oplus s_3$		Next state $s_2 s_1$
1	a = 0 0	0	0	0	0 0, i.e. a
		1	1	1	0 1, i.e. b
2	b = 0 1	0	1	0	1 0, i.e. c
		1	0	1	1 1, i.e. d

3	c = 1 0	0	1	1	0 0, i.e. a
		1	0	0	0 1, i.e. b
4	d = 1 1	0	0	1	1 0, i.e. c
		1	1	0	1 1, i.e. d

Table 4.4.8 : State transition table

Based on above table, the state diagram can be prepared easily. It is shown below in Fig. 4.4.33.



iii) To obtain trellis diagram for steady state :

From Table 4.4.9, the code trellis diagram can be prepared. It is steady state diagram. It is shown below.

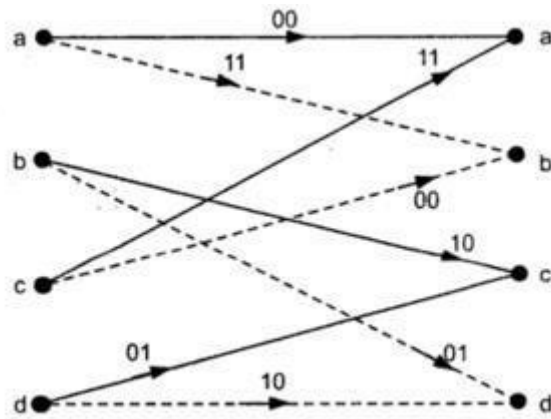


Fig. 4.4.34 Code trellis diagram for steady state

Decoding methods of Convolution code:

1.Veterbi decoding

2.Sequential decoding

3.Feedback decoding

Veterbi algorithm for decoding of convolution codes(maximam likelihood decoding):

Let represent the received signal by y .

Convolutional encoding operates continuously on input data

Hence there are no code vectors and blocks such as.

Metric: it is the discrepancy between the received signal y and the decoding signal at particular node. this metric can be added over few nodes a particular path

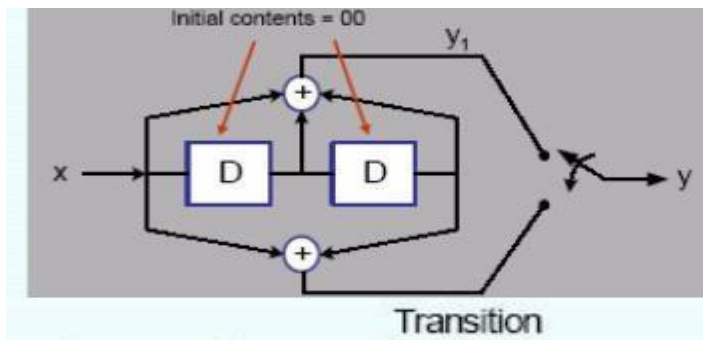
Surviving path: this is the path of the decoded signal with minimum metric

In veterbi decoding a metric is assigned to each surviving path

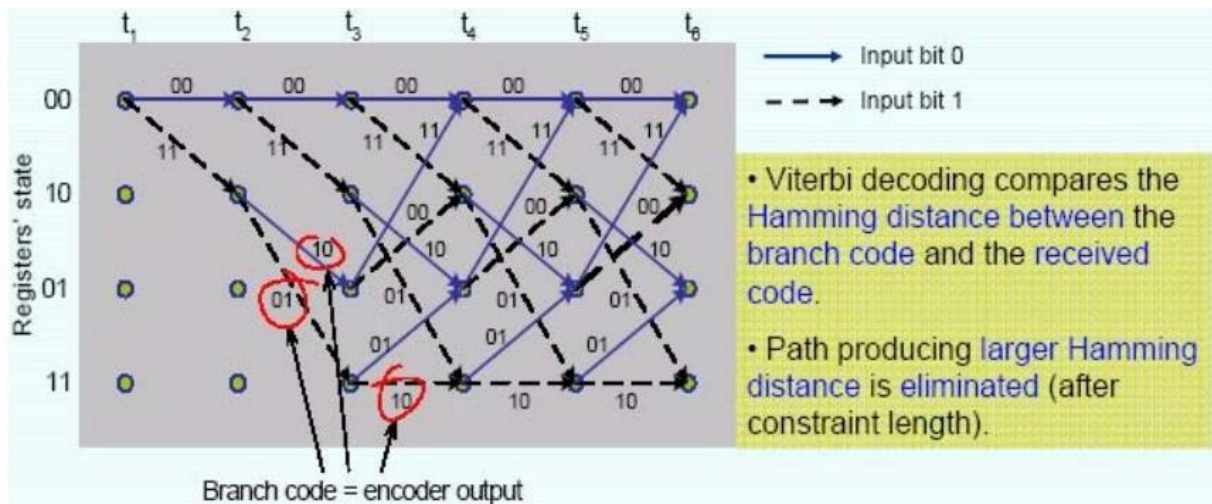
Metric of the particular is obtained by adding individual metric on the nodes along that path.

Y is decoded as the surviving path with smallest metric.

Example:



Exe:



Input data : $m = 1\ 1\ 0\ 1\ 1$

Codeword : $X = 11\ 01\ 01\ 00\ 01$

Received code : $Z = 11\ 01\ 01\ 10\ 01$

