VI sem

Unit - 4

Information Theory and coding:- Information mesurement, Average information and information rate, coding for discrete memoryless source, continious channel capacity, Maximum entropy, Huffman and Fano coding, Discrete channel capacity, Trade-off between S/N and Bandwidth, Block code, Hamming code, cyclic code, convolution code, Tree digram, state digram, Trellis digram, Viterbi encoder and decoder, Turbo code.

Information Theory → The mathematical study of the coding of information in the form of sequence of symbols, impulses etc.

⇒ Information means importance.

⇒ if the Probability of occurance of an events is less then the information associated with that event will be more and vice-versa.

Ex:- if a dog bites a men, it's no news, but if a man bites a dog its a news.

explain from

$$I\{xi\} \propto \frac{1}{P\{xi\}}$$

$$I\{xi\} = \log_b\left(\frac{1}{P\{xi\}}\right)$$

or

$$\boxed{I\{xi\} = -\log_b\{P\{xi\}\}}$$

* Unit of $I\{xi\}$ depends upon the base (b) chosen

b	units
2	bits binary
e	nat (natural unit)
10	decit or ~~Hartaley~~ Hartley

⇒ it is standard to use $b=2$. the conversion of these Units to other unit can be obteined by the following relations:

$$\log_2 a = \frac{\log_{10} a}{\log_{10} 2}$$

Ex-1 Calculate the amount of information if it is given that $P(xi) = 1/4$.

Soln:- we know that the amount of information is given as

$$I(xi) = \log_2\left(\frac{1}{P(xi)}\right) = \frac{\log_{10} \frac{1}{P(xi)}}{\log_{10} 2}$$

Substituting given value of $P(xi)$ in above eqn

$$I(xi) = \frac{\log_{10} 4}{\log_{10} 2} = 2 \text{ bits}$$

$$\frac{\log_{10} 4}{\log_{10} 2} = \frac{2 \cdot \log_{10} 2}{\log_{10} 2} = 2$$

Scanned by CamScanner

Q.2 A source is generating 3 possible systems with probabilities of $\frac{1}{4}, \frac{1}{2}, \frac{1}{4}$. Find the information associated with each of the symbols.

Soln: Given $P(x_1) = \frac{1}{4}$, $P(x_2) = \frac{1}{2}$, $P(x_3) = \frac{1}{4}$

$$I(x_1) = \log_2\left(\frac{1}{P(x_1)}\right) = \log_2 4 = 2 \text{ bits}$$

$$I(x_2) = \log_2\left(\frac{1}{P(x_2)}\right) = \log_2 2 = 1 \text{ bit}$$

$$I(x_3) = \log_2\left(\frac{1}{P(x_3)}\right) = \log_2 4 = 2 \text{ bits}$$

Note: the probability of occurance of x_2 is high so that the information associated with x_2 will be less.

Q.3 Show that $I(x_i x_j) = I(x_i) + I(x_j)$ if x_i and x_j are independent.

Soln: If x_i and x_j are independent then

$$P(x_i x_j) = P(x_i) \cdot P(x_j)$$

Now

$$I(x_i x_j) = \log \frac{1}{P(x_i x_j)}$$

$$= \log \frac{1}{P(x_i) \cdot P(x_j)}$$

$$= \log \frac{1}{P(x_i)} + \log\left(\frac{1}{P(x_j)}\right)$$

$$= I(x_i) + I(x_j)$$

ex-4 Calculate the amount of information if binary digits (binits) occurs with equal likelihood in a binary PCM systems.

Soln: In binary PCM systems there are only two binary levels i.e. 1 or 0. Since they occurs with equal likelihood, their probability of occurance would be

$$P(x_1)\{\text{'0' level}\} = P(x_2)\{\text{'1' levels}\} = \frac{1}{2}$$

Therefore the amount of information content will be given as

$$I(x_1) = \log_2\left(\frac{1}{P(x_1)}\right) = \log_2 2 = 1$$

$$I(x_2) = \log_2\left(\frac{1}{P(x_2)}\right) = \log_2 2 = 1$$

$$\boxed{I(x_1) = I(x_2) = 1}$$

Hence the correct identification of binary digit (binit) in binary PCM carries 1 bit of information.

Q.5 In a binary source, 0s occur three times as often as 1s. Find the information contained in 0s and 1s. Express the information in bits, nats, and Hartleys

Soln: If Prob. of symbol 0 occuring in data is $P(0)$, and 1 for $P(1)$

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

$3(P(1)) + P(1) = 1$

$P(0) = 3/4$, $P(1) = 1/4$

Information contained in 0s and 1s in bits

$I(x_0) = \log_2\left(\frac{1}{P(0)}\right) = -\log_2 P(0) = -\log_2 0.75 = 0.418$

$I(x_1) = -\log 0.25 = 2.0$ bits

Information in Hartleys

$P(0) = -\log_{10} 0.75 = 0.125$ Hartleys

A key measure in information theory is "entropy" → H → Entropy quantifies the amount of uncertainty involved in the value of random variable or the outcomes of a random process.

Average information (Entropy) :-

In a practical commn system, we usually transmit long sequence of symbols from an information source. Thus we are more interested in the average information that a source produces than the information content of a single symbol.

Unit of H is bits/symbol, nats/symbols, Hartleys/symbols

Mathematically, it is given as

* it is measure of the avg information content per source symbol.

$$\boxed{H = \sum_i I(x_i) \cdot P(x_i)}$$

$$\boxed{H = \sum_i P(x_i) \log_2 \left(\frac{1}{P(x_i)}\right)}$$

or

Entropy is the avg information in the symbols produced by a msg source.

$$\boxed{H = -\sum_i P(x_i) \cdot \log_2 P(x_i)}$$

* Information Rate (R) :- The avg rate at which the information must be transferred is called the source of information rate

⇒ Units of R is bits/sec.

⇒ Now,

$$R = \frac{\text{bits}}{\text{Symbols}} \times \frac{\text{Symbol}}{\text{Sec}}$$ — Symbol (r) Rate

$$\boxed{R = H \times r}$$

⇒ Information Rate = Symbol Rate * Entropy

* binits stands for binary digits as message or code elements that is different from information bits.

Q-1 A source is generating 4 possible symbols with the probability of $\frac{1}{8}$, $\frac{1}{8}$, $\frac{1}{4}$, $\frac{1}{2}$. Find the entropy and information rate if the source is generating 1 symbol/msec.

Given :-

$P(x_1) = \frac{1}{8}$, $P(x_2) = \frac{1}{8}$, $P(x_3) = \frac{1}{4}$, $P(x_4) = \frac{1}{2}$

Symbol rate = 1 symbol/m sec.

$$\boxed{r = 1000 \text{ symbol/sec}}$$

$$H = \sum_{i=1}^{4} P(x_i) \log_2\left(\frac{1}{P(x_i)}\right)$$

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

$$\boxed{H = 1.75 \text{ bits/symbol}}$$

$$R = r \times H$$

$$= 1000 \times 1.75$$

$$\boxed{R = 1.75 \text{ kbps}}$$

Scanned by CamScanner

* Entropy is measure of uncertainty.

Analysis:-

(1) Tx —— $x_1\ x_2\ x_3\ x_4 \ldots$

Case-1:

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

$H = \sum_{i=1}^{2} P(x_i) \log_2 \frac{1}{P(x_i)}$

$= \frac{1}{2} \log_2 2 + \frac{1}{2} \log_2 2$

$= 1$ bits/symbol

$= H_{max}$

when the prob. are equal.

case-2

$P(x_i) = 1,\ P(x_2) = 0$

$H = -\sum_{i=1}^{2} P(x_i) \log_2 P(x_i)$

$= 0 + 0$

$= 0$ bits/symbol

$= H_{min}$

when the probability of 1 is 1 and other is zero.

(2) Tx —— $x_1\ x_2\ x_3$

case(1):- $P(x_1) = P(x_2) = P(x_3) = 1/3$

$\boxed{H_{max} = \log_2 3 \text{ bit/symbol}}$

case(2)

$P(x_1) = 1,\ P(x_2) = P(x_3) = 0$

$H_{min} = 0$ bit/symbol

Conclusion:

Tx —— $x_1\ x_2\ x_3 \ldots x_m$

$P(x_1) = P(x_2) = P(x_3) = \ldots P(x_m) = \frac{1}{M}$

$\boxed{H_{max} = \log_2 M}$ bits/symbol ; $\boxed{H_{min} = 0}$ bits/symbol

Channel capacity:-

⇒ it specified the no of bits allowed by the channel in 1 sec.

* Channel Capacity, C = bits/sec

Tx —— R bits/sec —— C → bits/sec

Hence,

$\boxed{C \geq R \leftarrow \text{No information loss}}$

Shanon Hartley Law:-

⇒ It given the relation b/w channel capacity (C) and Bandwidth (BW)

⇒ Mathematically,

Normal S/N (not in dB)

$\boxed{C = B \log_2 \left(1 + \frac{S}{N}\right)}$ → bits/sec

C = channel capacity (bits/sec)

B = channel B.W (Hz)

S = Channel power expected at channel O/P (avg signal power)

N = Noise power

$\left(\frac{S}{N}\right)_{dB} = 10 \log_{10}\left(\frac{S}{N}\right)$

$(S/N)_{dB}$	(S/N)
10 dB	10
20 dB	100
15 dB	$10^{1.5} =$

$10\,dB = 10 \log_{10}(S/N)$

$1 = \log_{10}(S/N)$

$(S/N) = 10^1 = 10$

Q-1 For a channel of BW = 4 KHz
$(S/N) = 15\,dB$
Find the channel capacity

Soln: As $S/N = 15\,dB$

$= 10^{1.5} = 31.6$

So,

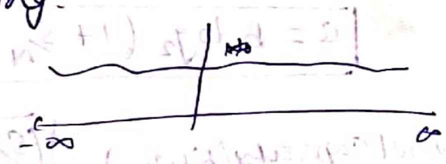
$C = B \log_2 (1 + S/N)$

$= 4 \log_2 (1 + 31.2)$

$\boxed{C = 20.1\ Kbps}$

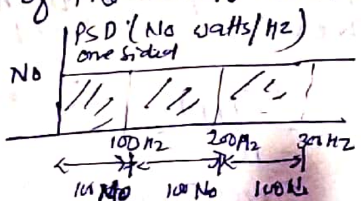
capacity of AWGN (Additive White Gaussian Noise) Channel :-

⇒ White noise has the frequency spectrum as following.



it covers the all frequency components.

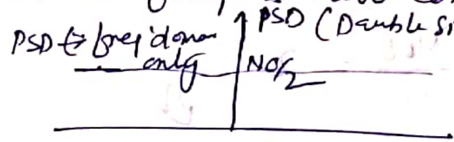
⇒ The PSD of the white Noise is given as



* freq. b/w 0 to 100 Hz will be affected by 100 No watts of power.

* freq. b/w 100 to 200 Hz will be affected by 100 No Watt of power.

If the -ve freq. is also considered then



Regarding white noise, its power is given as

$N (watts) = \frac{watts}{Hz} \times Hz$

$\boxed{N = N_0 B}$

* Default power spectral density is one sided.

Note:- Each of the frequency component transmitted through a channel is affected by same amount of white noise power.

The channel BW is given as

$\uparrow C = B \uparrow \log_2 (1 + \frac{S}{N})$
(linear)

For a AWGN channel,

$\uparrow C = B \uparrow \log_2 \{1 + \frac{S}{N_0 B \uparrow}\}$
(nonlinear)

Conclusion :
For AWGN channel as $B \to \infty$
channel capacity becomes $\boxed{C_\infty = 1.44\ S/N_0}$

Proof: As we know that

$$C = B \log_2 \left\{ 1 + \frac{S}{N_0 B} \right\}$$

$$C = \frac{N_0}{S} \cdot \frac{S}{N_0} B \log \left\{ 1 + \frac{S}{N_0 B} \right\}$$

Let $\frac{N_0 B}{S} = x$

as $B \to \infty \Rightarrow x \to \infty$

So,

$$C_\infty = \frac{S}{N_0} \lim_{x \to \infty} x \cdot \log_2 \left(1 + \frac{1}{x} \right)$$

$$= \cancel{\frac{S}{N_0} \log_e 2}$$

$$= \frac{S}{N_0} \log_e 2$$

$$\boxed{C_\infty = 1.44\, S/N_0}$$

OR

Bandwidth S/N Trade off :-

for a noiseless channel ($S/N = \infty$) has an infinite capacity, on the other hand, the channel capacity does not become infinite as the BW ~~become~~ approaches infinite because with an increase of BW, the noise power also increases.

Thus for a fixed signal power, and in the presence of white Gaussian noise, the channel capacity approaches an upper limit with increasing BW.
This is explained below.
we have,

$$C = B \log \left(1 + \frac{S}{N} \right)$$

$$= B \log \left(1 + \frac{S}{N_0 B} \right)$$

$$= \frac{S}{N_0} \cdot \frac{N_0}{S} \cdot B \log \left(1 + \frac{S}{N_0 B} \right)$$

$$= \frac{S}{N_0} \cdot \frac{N_0 B}{S} \log \left(1 + \frac{S}{N_0 B} \right)$$

Let $\frac{S}{N_0 B} = x$

$$= \frac{S}{N_0} \cdot \frac{1}{x} \log (1 + x)$$

$$= \frac{S}{N_0} \cdot \log (1+x)^{1/x}$$

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

if $x = \frac{S}{N_0 B}$; then as B approaches infinite, x approaches zero

$$\lim_{B \to \infty} \left(1 + \frac{S}{N_0 B} \right)^{\frac{N_0 B}{S}} = e$$

Hence,

$$\lim_{B \to \infty} C = \frac{S}{N_0} \log_2 e$$

$$\boxed{C_\infty = 1.44 \frac{S}{N_0}}$$

Imp

Let us now consider the tradeoff b/w the BW and S/N ratio

$$\frac{S}{N} = 15, \quad B = 5\text{ KHz}$$

$$C = B \log\left(1+\frac{S}{N}\right)$$

$$= 5 \log(1+15)$$

$$= 5 \log_2 16 \Rightarrow 5 \frac{\log_{10} 16}{\log_{10} 2}$$

$$= 5 \times 4$$

$$= 20 \text{ K bits/sec}$$

If the S/N ratio is increases to 31, the BW for the same channel capacity can be found form

$$C = 20 = B \log(1+31)$$

$$B = \frac{20}{\log 32} = \frac{20}{5} = 4\text{ KHz}$$

With a 4 KHz BW, the noise power will be 4/5 times the noise power at 5 KHz.

Thus the signal power will have to increase by factor $\frac{4}{5} \times \frac{31}{15} = 1.65$.

Q-1 For AWGN of having BW 4 KHz, two sided noise PSD is given by 10^{-12} watts/Hz. Find the channel capacity required to get signal power of 0.1 mW at the o/p of the channel.

Given :- $BW = 4\text{ KHz}$

$$N_0/2 = 10^{-12}$$

$$\boxed{N_0 = 2 \times 10^{-12}}$$

$$N_0 B = 2 \times 10^{-12} \times 4 \times 10^3$$

$$= 8 \times 10^{-9} \text{ watts}$$

$$C = B \log_2\left(1 + \frac{S}{N_0 B}\right)$$

$$= 4 \times 10^3 \log_2\left(1 + \frac{0.1 \times 10^{-3}}{8 \times 10^{-9}}\right)$$

$$\boxed{C = 54.44 \text{ kbps}}$$

Q. 2 A voice grade channel for the telephone n/w has BW of 7.4 KHz. Calculate the information capacity of the telephone channel for signal to noise ratio of 30 dB

Soln:- Given $\frac{S}{N}\Big|_{dB} = 30\,dB$

$$\left(\frac{S}{N}\right)_{dB} = 30\,dB = 10 \log\left(\frac{S}{N}\right)$$

$$30 = 10 \log\left(\frac{S}{N}\right)$$

$$\frac{S}{N} = 10^3 = 1000$$

Channel Capacity $C = B \log_2\left(1+\frac{S}{N}\right)$

$$= 3.4 \times 10^3 \log_2(1+1000)$$

$$= 3.4 \times 10^3 \log_2(1001)$$

$$\boxed{C = 33.8 \times 10^3 \text{ b/s}}$$

---

* Coding offers most significant application of the information theory. The main purpose of coding is to improve the efficiency of the commn system in some sense.

Entropy coding :-

⇒ The design of a variable length code such that its average code word length approaches the entropy of the DMS is referred to as entropy coding. In this coding section we presents two example of entropy coding:

(I) Huffman encoding

(II) Shanon - Fano Coding

(i) Huffman encoding :-

We describe an important class of prefix codes known as Huffman codes.

⇒ In general, Huffman encoding results in an optimum code. Thus, it is the code that has the highest efficiency.

⇒ We presents the procedure using a specific example.

Suppose that we wish to code five words x_1, x_2, x_3, x_4 and x_5 with probabilities $P(x_1) = 0.4$, $P(x_2) = 0.19$, $P(x_3) = 0.16$, $P(x_4) = 0.15$, $P(x_5) = 0.1$.

The Huffman code is accomplished in four steps as follows.

Step-1 * Arrange the messages in order of decreasing probabilities.

* if there are equal probabilities choose any of the various ordering.

* The two source symbols of lowest probability are assigned a "0" and a "1"

Words	Probability
x_1	0.4
x_2	0.19
x_3	0.16
x_4	0.15] 0
x_5	0.1] 1

Step 2 ⇒ Combine the bottom two entries to form a new entry with a probability that is the sum of original probability.

⇒ if necessary, reorder the list so that new probabilities are in decreasing order.

word	Probability (stage-I)		Probability (stage-II)
x_1	0.4		0.4
x_2	0.19		0.25
x_3	0.16		0.19
x_4	0.15	0	0.16
x_5	0.1	1	

Step-3 ⇒ Continue combining in pairs until only two entries remain.

⇒ The codeword for each symbol is found by starting at the right with the most significant bit working backword and tracking the sequence of 0s and 1s assigned to that symbol as well as its successors.

Symbol	step-1	step-2	step-3	step-4
x_1	0.4 ← 1	0.4 ← 1	0.4 ← 1	0.6] 0
x_2	0.19 ← 000	0.25 ← 01	0.35 00] 0	0.4] 1
x_3	0.16 ← 001	0.19 00] 0	0.25 01] 1	
x_4	0.15 010] 0	0.16 00] 1		
x_5	0.1 011] 1			

0.19 → 0

0.16 → 1

0.25 → 01

Prob. Final = 0.6 → 0

0.4 → 1

Finally, the code words of the Huffman code for the source are tabulated in Table (below)

Symbol	Probability	Code word
x_1	0.4	1
x_2	0.19	000
x_3	0.16	001
x_4	0.15	010
x_5	0.1	011

The entropy is given by $H(X) = -\sum_{i=1}^{5} P(x_i)\log_2 P(x_i)$

$H(X) =$ bits/message

$$= -0.4\log_2 0.4 - 0.19\log_2 0.19 - 16\log_2 0.16 - 0.15\log_2 0.15 - 0.1\log_2 0.1$$

$$\boxed{H(X) = 2.15}$$

Average length is given by $L = 0.4\times(1) + 0.19\times(3) + 0.16\times(3) + 0.1\times(3) = 2.2$

codeword codeword code

L = letters/message

$$\eta = \frac{H(X)}{L} = \frac{2.15}{2.2} = 0.977$$

$$\boxed{\eta = 97.7\%}$$

⇒ One disadvantage of the Huffman Code is that we can not start assigning code words until the entire combination process is completed. That is, every one of the columns must be developed before the first code can be assigned.

** The coding process is often performed by a special purpose micro computer.

Shannon-Fano Code:

✓ ⇒ Shannon-Fano code is similar to the Huffman code, with a major differences being that the operations are being performed in a forward, rather than backward, direction.

↳ Thus the storage requirements are considerably relaxed and the code is easier to impliment.

⇒ While it often leads to average lengths that that are the same as those of the Huffman code,

the results of Shannon-Fano coding are not always good as those of Haffman Coding.

⇒ We again illustrate the techniques with an example of that of Huffman codes presented earlier;

Step-1 Arrange the message in order of decreasing probabilities. if there are equal probabilities choose any of the various orderings.

Word	Probability
x_1	0.4
x_2	0.19
x_3	0.16
x_4	0.15
x_5	0.1

Step 2 ⇒ Partition the message into the most equiprobable subsets that is start at the top or bottom and devide the group into two sets.

⇒ we find the total Probability of upper set and the total probability of the lower set.

⇒ we choose the deviding line that results in the closest two Probability.

⇒ we now assign a 0 to all members of one of the two sets and a 1 to all members of the others (the choice is arbitrary).

⇒ suppose we choose 0 for the top set and 1 for the bottom. this is illustrated here.

Words	Probability		step-1
x_1	0.4		0
x_2	0.19	→0.59	0
x_3	0.16	→0.41	1
x_4	0.15		1
x_5	0.1		1

ie is devided in two groups, 0.59 and 0.41 almost similar

Step-3 continue this process, each time partitioning the sets with as nearly equal probability as possible until further partitioning is not possible.

Finally Shannon-Fano code is constructed as follows:

Symbols	Probability	step-I	step-2	step-3	Code
x_1	0.4	0	0		00
x_2	0.19	0	1		01
x_3	0.16	1	0		10
x_4	0.15	1	1	0	110
x_5	0.1	1	1	1	111

This entropy is given by

$$H(x) = -\sum_{i=1}^{5} P(x_i) \log_2 P(x_i)$$

$$= -0.4 \log_2 0.4 - 0.19 \log_2 0.19 - 0.16 \log_2 0.16 - 0.15 \log_2 0.15 - 0.1 \log_2 0.1$$

$$\boxed{H(x) = 2.15}$$

Average length is given by no of code words

$$L = \sum_{i=1}^{5} P(x_i) \cdot n_i$$

$$= 0.4(2) + 0.19(2) + 0.16(2) + 0.15(3) + 0.1(3)$$

$$= 2.25$$?.

$$\eta = \frac{H(x)}{L} = \frac{2.15}{2.25} = 0.956 = 95.6\%$$

Error control coding is classified in two types;
1. Block Code.
2. Convolution code.

⇒ **Block Code:-** In the

⇒ Block code is a code having all its words of the same length,

⇒ K bit block is converted into an n-bit block, or each block of k message bits is encoded into a block of n-bits $(n>k)$.

⇒ The resultant block code called an (n,k) block code.

Message block (K-bits) → Channel encoder → Code block

Structure of code words: K bit + n-k bits = n bits

$\boxed{k + r = n}$ Code words

Left most bit of the code word corresponds to the msg bit → message

Right most bit corresponds to parity bit → Check bits

$\boxed{n>k}$

(n, k)
(7, 4)
Code rate $(r) = k/n$
$r = 4/7 = 0.57$
Code rate doesn't exceed to 1

⇒ Check bits (r) are derived from the message bits and are added to them.

⇒ The n-bit block of a channel encoder output is called a code word.

* Parity bits help in error detection and correction, also in locating data i.e. In the data being transmitted

(a) **Parity check code:-**

⇒ The simplest possible block code is when the no of check bits is one. These are known as parity check code.

⇒ when total no of 1s in the code word is Even then this is called even parity check code.

⇒ and when the total no of 1s in the code word is Odd, then this is called odd parity check code.

Following example explains the parity check code

Message	Code for even parity (check bit 'r')	Code for odd parity (check bit 'r')
010011 [no of 1's odd]	1	0
101110 [no of 1's is even]	0	1

(b) **Binary code space:-**

→ In this section certain important concepts such as the weight code, Hamming distance etc.

Hamming weight:-

⇒ $\boxed{\text{The weight of the code = no of non-zero components}}$

Code words	~~weight~~ weight
010110	3
101000	2
000000	0

Ex
$C = \{01234, 43210\}$
$w[01234] = 4$ coz one zero is present
$w[43210] = 4$
$d = 4$

(c) **Hamming distance:-**

⇒ no of components in which two code words differs.

Ex
0 1 0 0
1 1 1 1
diff, x (same), diff, diff ⇒ $\boxed{d=3}$

Ex
$u = 1010$
$v = 0111$
$w = 1001$

Perform XOR operation, Hamming distance

$C = (000, 101, 011, 110)$

0000, 0101, 1010, 1111

Cyclic code:-

- * C is a linear code.
- ** A code is said to be cyclic code if the cyclic shift of each code word is also a code word. i.e. whenever $a_0, \dots a_{n-1} \in C$, then also $a_{n-1}, a_0 \dots a_{n-2} \in C$.

Theorem ⟹ If $g(x)$ is a polynomial of degree $(n-k)$ that is a factor of $(1+x^n)$, then $g(x)$ generates an (n,k) cyclic code (C) in which the code polynomials $C(x)$ for the data word,

$d = d_0, d_1, d_2 \dots d_{k-1}$ is generated by

$$\boxed{C(x) = d(x) \cdot g(x)}$$

where, $d(x) = d_0 + d_1 x + d_2 x^2 + \dots d_{k-1} x^{k-1}$ is the data polynomial corresponding to the data word.

polynomial $g(x)$ is called generator polynomial (matrix).

Ex.

$$\begin{array}{ccccccc} x^6 & \dots & & & & x^1 & x^0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 & 0 \end{array}$$

$C(x) = x^6 + x^4 + x^1 + 1$

polynomial representing coding

Generator Matrix:-

⟹ In an (n,k) linear block code C, we defined a code vector c and data vector d as follow.

$c = [c_1, c_2, c_3 \dots c_n]$

$d = [d_1, d_2, d_3 \dots d_n]$

⟹ If a data bit appear in specified location of C, then the code C is called systematic otherwise non-systematic.

$c_i = d_i, \; i = 1, 2 \dots k$

$c_1 = d_1$

$c_2 = d_2$

$c_3 = d_3$

$\vdots$

$c_k = d_n$

$c_{k+1} = P_{11} d_1 + P_{21} d_2 + \dots P_{k1} d_k$

$c_{k+2} = P_{12} d_1 + P_{22} d_2 + \dots P_{k2} d_k$

$c_n = P_{1,n-k} d_1 + P_{2,n-k} d_2 + \dots P_{k,n-k} d_k$

$$\boxed{C = dG}$$

$m = n-k$, $\boxed{P \text{ matrix} = m \times k}$

$$C = [d_1 \; d_2 \; d_3 \dots d_n] \begin{bmatrix} 1 & 0 & 0 & 0 & \dots & 0 & P_{11} & P_{21} & \dots & P_{m1} \\ 0 & 1 & 0 & 0 & \dots & 0 & P_{21} & P_{22} & \dots & P_{m2} \\ 0 & 0 & 1 & 0 & 0 & \dots & & & & \\ 0 & 0 & 0 & 1 & \dots & 0 & P_{31} & P_{32} & \dots & P_{m3} \\ \vdots & & & & & & & & & \\ 0 & 0 & 0 & \dots & & 1 & P_{1k} & P_{2k} & \dots & P_{mk} \end{bmatrix}$$

We know that, $C = dG$, then

$$\boxed{G = [I_k \; P^T]}$$

P^T = Transpose of P

I_k = Unity matrix

XOR

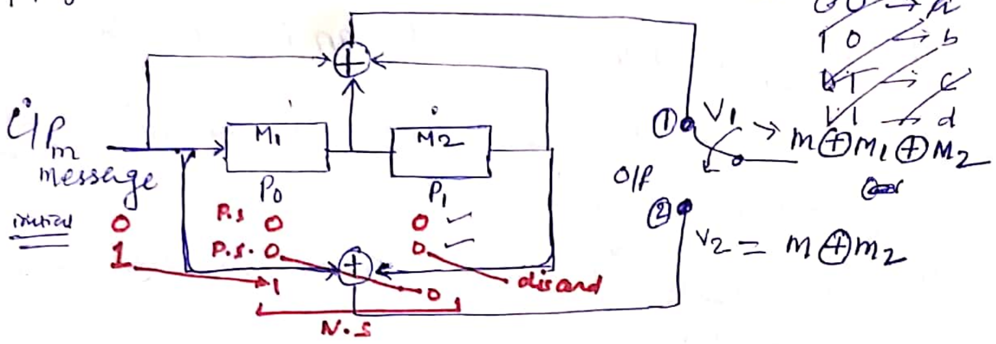
0	0	0
0	1	1
1	0	1
1	1	0

① for 1: $1 \oplus 0 \oplus 0 = 1$, $1 \oplus 0 = 1$

Convolution code

encoder convolution

1st
00 → a
10 → b
01 → c
11 → d



M_1, M_2 → Memory [Actually shows Present state (P.S.)
or P_0, P_1 — here two memory, so that four combination 00, 01, 10, 11

✓ Case ① input is either 0 or 1 then we assume P.S. start with 00 initially.

Input (I/P) (m)	Present state (P.S) P_0 P_1		Next state (N.S)	O/P V_1 V_2
0	0 0	a	0 0 → a	0 0
1	0 0		1 0 → b	1 1
0	1 0	b	0 1 → c	1 0
1	1 0		1 1 → d	0 1
0	0 1	c	0 0 → a	1 1
1	0 1		1 0 → b	0 0
0	1 1	d	0 1 → c	0 1
1	1 1		1 1 → d	1 0

last one is discard; 1 0 (n.s.)

→ P.T.O

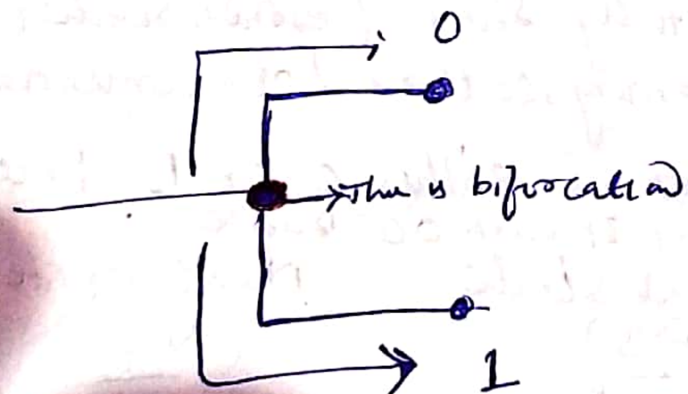
In case of N.S. :- every bit is shifted
✓ input is shifted in the first memory elements
✓ and first memory element is shifted to the 2nd memory element. then finally get N.S.

Tree digram:

(I) Every branch represents an input bit

(II) The upper branch of the bifurcation (Division of a branch) denotes a bit '0'

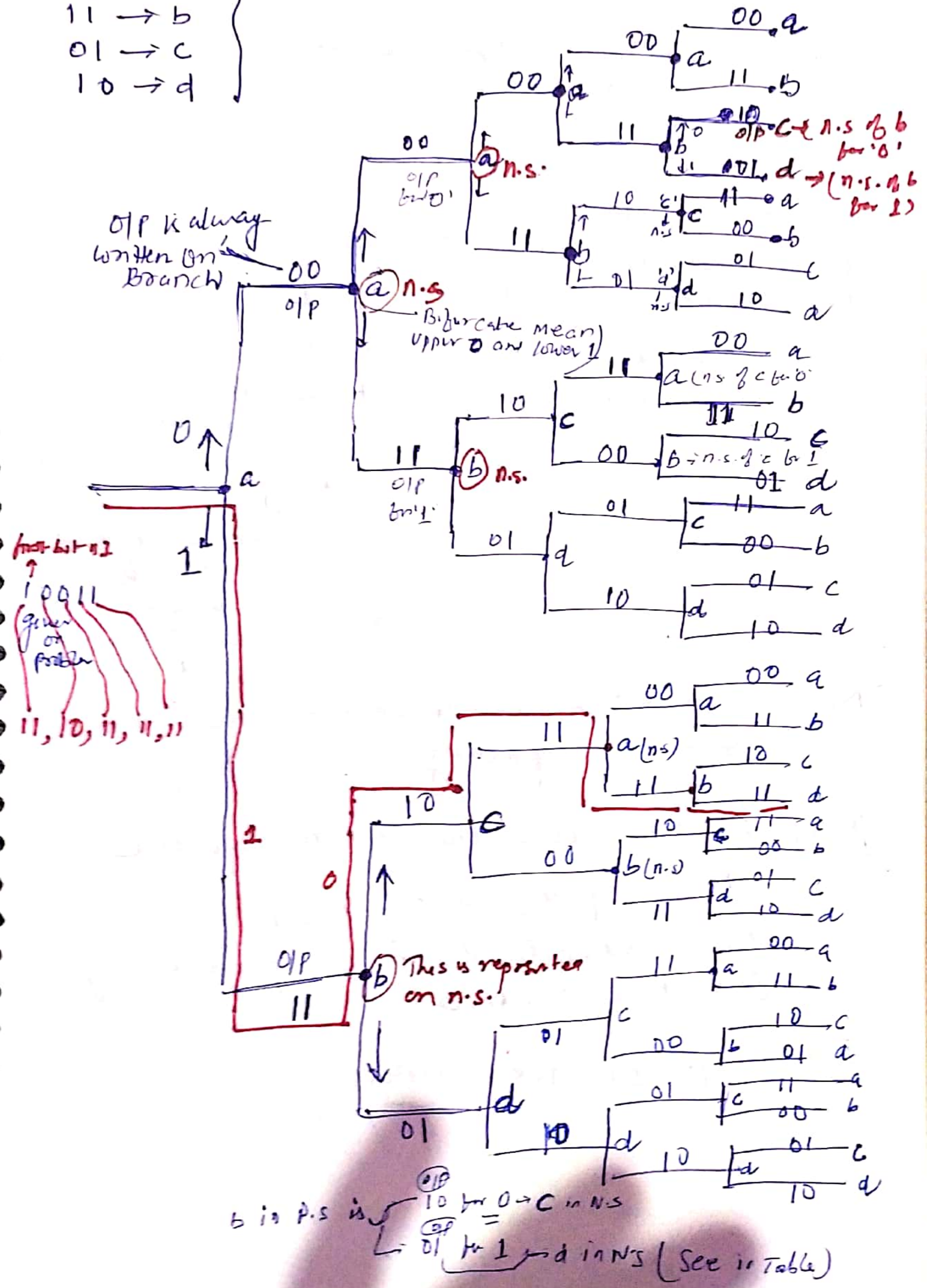
(III) The lower branch of the bifurcation denotes bit '1'



(IV) The encoded sequence path has to be traced from left to right

V_1 V_2

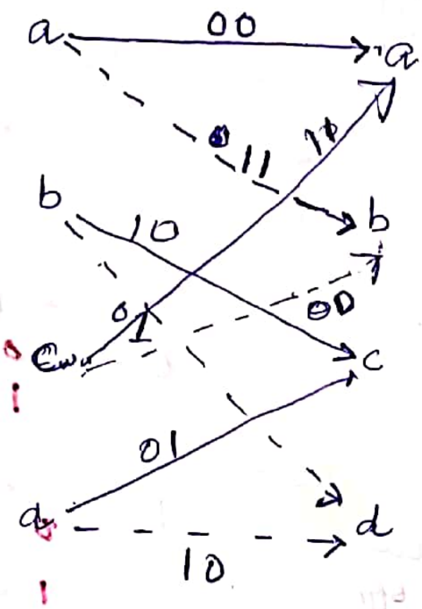
00 → a
11 → b
01 → c
10 → d



Trellis Diagram :-

a→ for 0 → ——→ Solid line
for 1 → - - - - → dash line

this is also o/p



case(1) when 'a' then it moving towards a & b [from N.S]
o/p — {00, 11}

case(II) when b then its moving towards c & d (N.S)
o/p is for 0 → 10
for 1 → 01

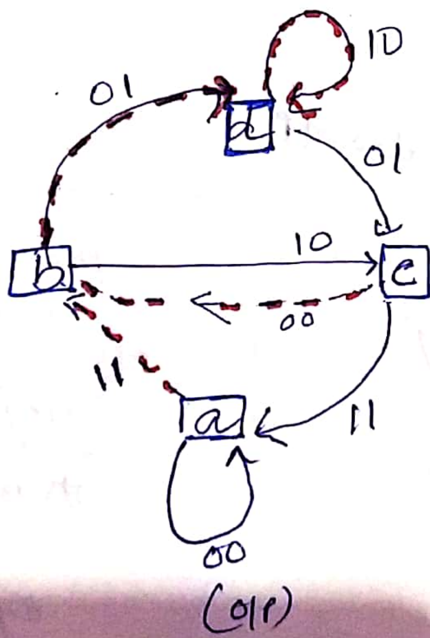
case(3) when c then its moving towards a and b
o/p, 11, 00

case(4) d then c, d
o/p. 01, 10

State Diagram :- There are four state, a, b, c, d

0 → Solid line

1 → dash line



	N.S	O/P
a → 0	a	00
1	b	11

Scanned by CamScanner

Continue $\boxed{\text{cyclic code}}$

See Previous Page explantn

In code word (7, 4)

1110 110

→ not differnciate msg bit and parity check bit in codeword

→ if given codeword we dont know the arrangement of which side is msg and which side is parity (or mixed up) then this type of codewords is known as ~~sy~~ Non-Systematic Codeword.

This type of code is generally found in cyclic code.

Ex. of non-systematic code is Convolution code.

→ we can identify msg bit and parity check bit

→ if arrangement of given codewords is known which first four bit is msg and last 3 bit parity or first 3 bit is msg or last four bit msg then type of code is known as systematic Code

Example of systematic code is "linear Block Code"

Cyclic code word

Non-Systematic Codeword

$X(x) = M(x) \cdot g(x)$

Codeword — polynomial of msg — generator Polynomial

→ both are given generally

→ if not given then we assume msg then solve.

Systematic Codeword

① $X(x) = X^{n-k} M(x)$

② $\frac{X^{n-k} M(x)}{G(x)} \to R$ — Q(x) + R(x) / Q(x) G(x)

③ $X(n) = \frac{X^{n-k} M(x) + R(x)}{G(x)}$

Remaind

Q.2 Construct a Non-Systematic and Systematic (7, 4) cyclic code using a generator polynomial $g(x) = x^3 + x^2 + 1$

Soln:

$c(x) = M(x) \cdot g(x)$

Let $M(x) = 1010$

$= 1 \cdot x^3 + 0 \cdot x^2 + 1x + 0 \cdot x^0$

$= x^3 + x$

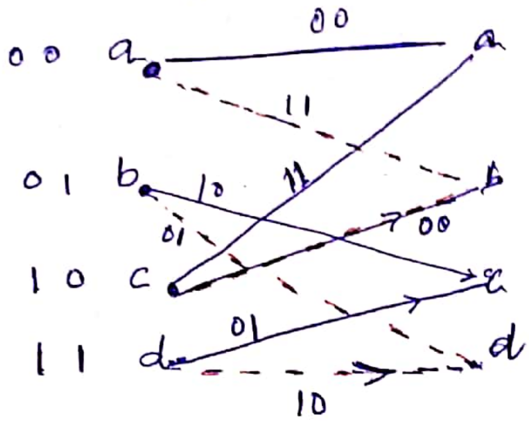
Viterbi Algorithm (VA) for Decoding of Convolution code:

* The Viterbi Algorithm is used to decode ~~the~~ Convolutional codes and any structure or system that can be discribed by a trellis.

✓ Befor starting, we should have idea about Hamming distance suppose,

1	1	0
1	0	0
^	v	^

$\boxed{\text{Hamming distance} = 1}$ so on

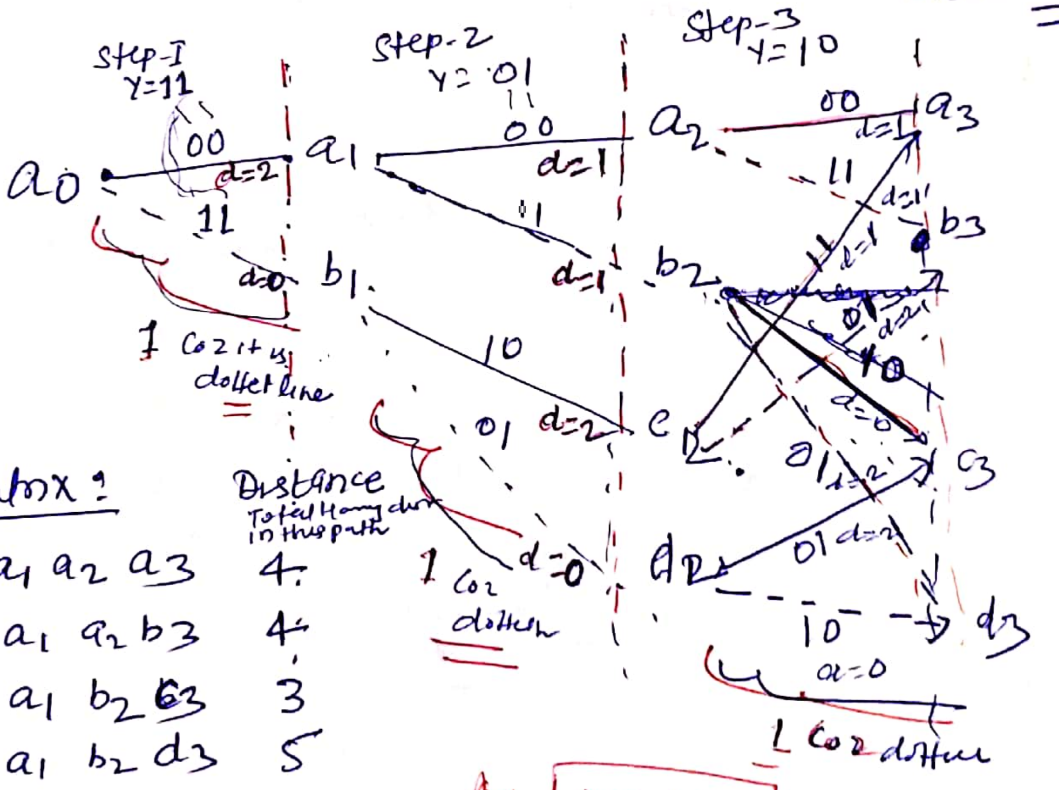


** We count the total no of errors on each possible path and chose the path with the fewest errors. We do this one step at a time

** Viterbi decoding tells us to choose the smallest number at each state and eliminated the other path.

↳ suppose we have received a input, and that input we want to decode.

↳ ~~sa~~ Receive input Y = 11 01 10 → code word then How to decode this



Path Matrix:

	Path	Distance (Total Hamming dis in this path)
1.	a0 a1 a2 a3	4
2	a0 a1 a2 b3	4
3	a0 a1 b2 c3	3
4	a0 a1 b2 d3	5
5	a0 b1 c2 a3	3
6	a0 b1 c2 b3	3
7	a0 b1 d2 c3	2
8	a0 b1 d2 d3	$\boxed{0}$

Ans $\boxed{X = 111}$ encoded

* We choose the path with the smallest sum. The information sequence (i.e. input to the encoder) 111 corresponding to path

→ If we get '0' then that is the best path, then in Receive Code there is no Error

Turbo Codes:-

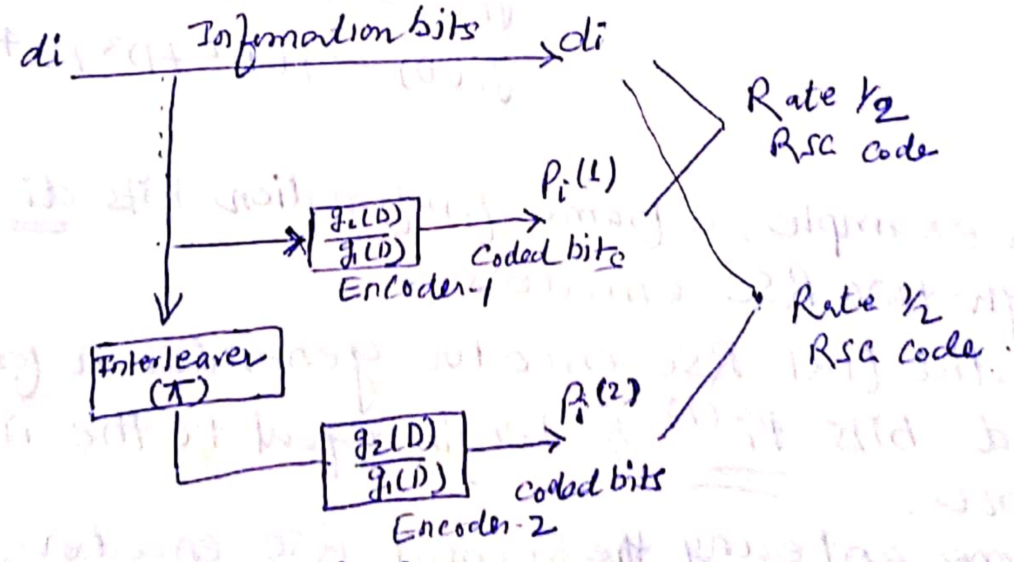
⇒ Turbo Codes are combination of Block and convolutional Codes.

⇒ They require the a block to be formed before encoding can begin, but for encoding shift register are used, just like in convolutional code instead of computing parity bits.

⇒ Turbo Codes are used at least two convolutional components encoders and two maximum a posteriori (MAP) algorithm components decoders.

✓⇒ Turbo Codes perform well in the low SNR environment.

✓⇒ At high SNR, Reed-Solomon code have better Performance than Turbo Codes.



(a)

⇒ Thus turbo consists of two recursive systematic convolutional (RSC) codes.

↳ Unit delay as D, the 1X2 generator matrix of the rate 1/2 RSC code is of the form.

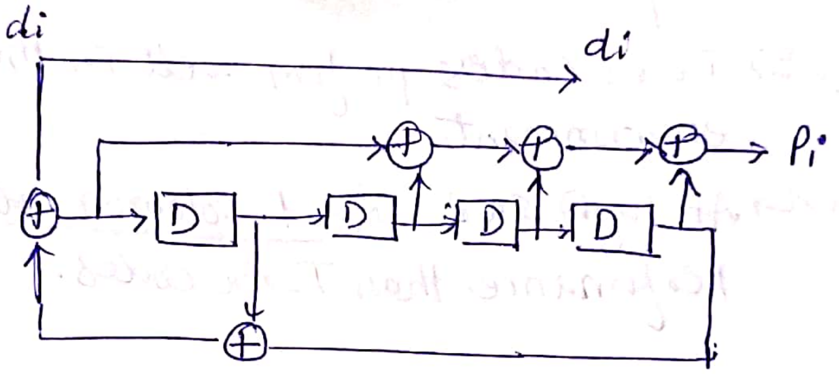
P.T.O

$$G(D) = \left[1 \quad \frac{g_2(D)}{g_1(D)}\right]$$

In perticular, the example turbo codes of Berrou et. al. was specified by

$$G(D) = \left[1 \quad \frac{1+D^2+D^3+D^4}{1+D+D^4}\right]$$

The simple implementation of encoder is shown figure below.



$$g_1(D) = 1+D+D^4$$
$$g_2(D) = 1+D^2+D^3+D^4$$

⇒ In this example, a frame of information bits di is sent through two RSC encoders.
Thus, the first RSC encoder generates a frame of Coded bits $P_i^{(1)}$ of length equal to the information frame.
⇒ Before entering the second RSC encoder, the information bits are interleaved by a random block interleaver π. the second encoder will generate a different coded bit fram $P_i^{(2)}$