EXPERIMENT NO. 4

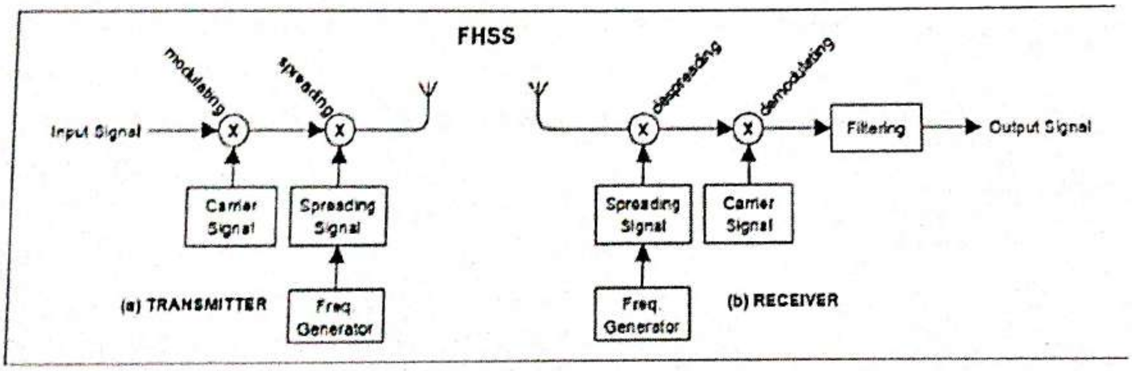
AIM: Study of FH-SS transmitter and receiver using suitable hardware setup/kit.

OBJECTIVE: Experimental Study of Generation &detection of FH-SS coherent BPSK & its spectrum

APPARATUS: FH-SS kit, DSO, CRO, Connecting wires.

THEORY: -

In Frequency Hoping Spectrum, available channel bandwidth is broken into a large number of non-overlapping frequency slots. Data is modulated onto time-varying, Pseudo random carrier frequencies. Transmitter "hops" between different narrowband Channels with centre frequency fi, and bandwidth B (instantaneous bandwidth). Spectrum BW (bandwidth) over which hopping occurs is called the total hopping. Data sent by hopping transmitter carrier to seemingly random channels which are known only to desired receiver. On each channel, data bursts are sent using conventional narrowband modulation hopping period/hop duration. There are generally two types of hopping schemes employed in FHSS.



Fast Hopping: In this scheme the hopping rate is kept equal to or greater than the Data rate. The hopping rate should not be many time faster than the data rate. Generally, it can be 2-3 times faster than the data rate.

Slow Hopping: In this scheme the hopping rate is kept lower than the data rate. Also, the data rates which can be supported by FHSS system are quite lower than the data rates supported by DSSS systems.

FH-SS Transmitter:

FHSS is a method of transmitting radio signals by rapidly switching a carrier among many frequency channels, using a pseudorandom or pseudonoise sequence known to both transmitter and receiver. The advantage of this system is that the signal sees a different channel and a different set of interfering signals during each hop.

III) Observe o/p of FHSS Demodulator. We will not get proper detected o/p as we ha

PN seq. to FHSS Demodulator.

IV) Now Connect PN sequence to PN i/p of FHSS demodulator

V) Observe Detected o/p.

Questions:

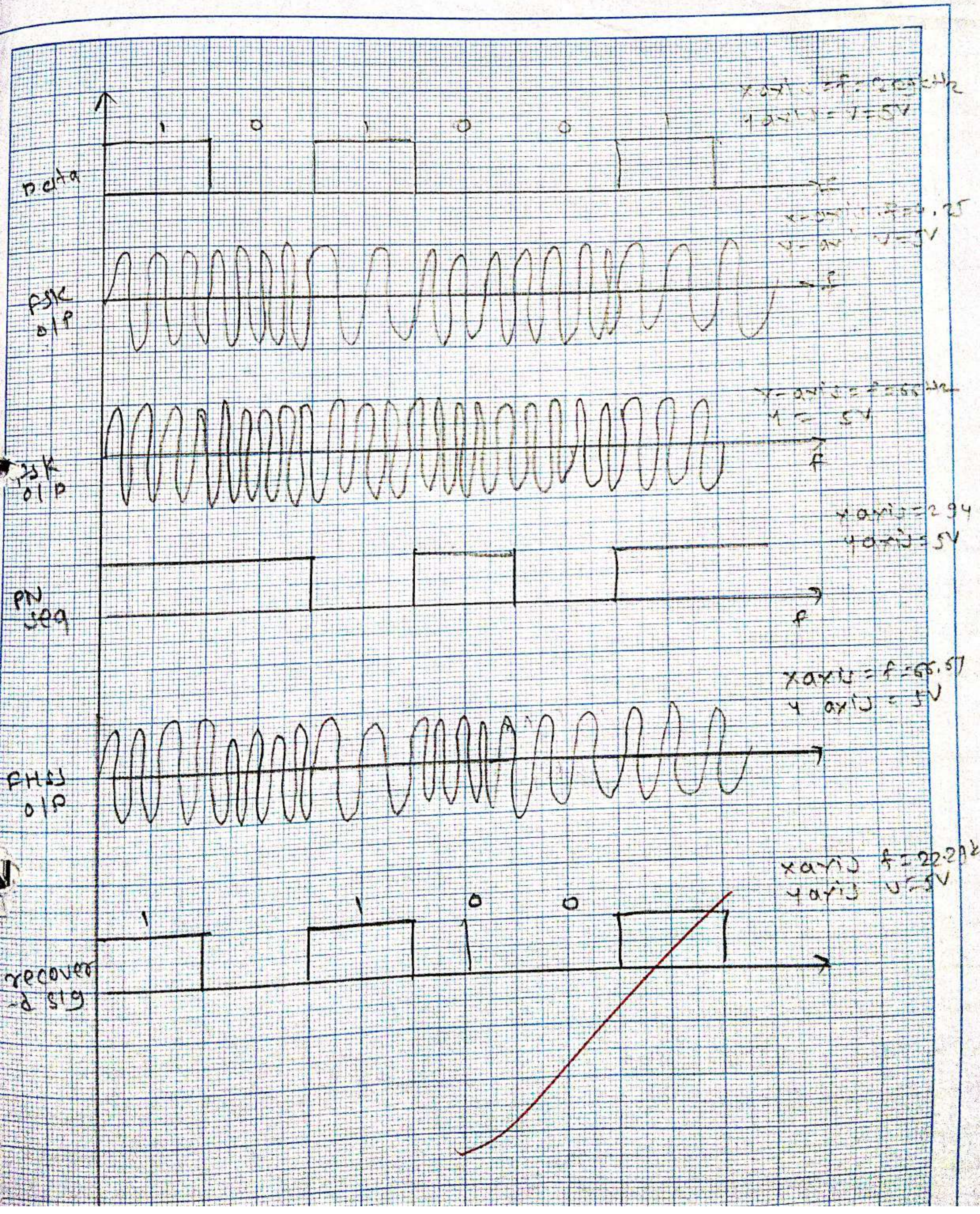
1. Explain FH-SS?
2. Difference between FH-SS and DS-SS?
3. Explain Hop rate, Symbol Rate for FH-SS?
4. How to calculate Processing gain, jamming margin for FH-SS?
5.

Conclusion:

In this experiment we performed h/w setup and observed the FH-SS waveforms. we converted given binary data 101001 into FH-SS waveform.

Timely submission (10)	Journal Presentation(10)	Performance(10)	Understanding(10)	Oral(10)	Total (50)
					45
Sub Teacher Sign:					

3/10



Data
1 0 1 0 0 1
FSK o/p
PN seq
x axis = 2.94
y axis = 5V
x axis = f = 65.57
y axis = 5V
FHSS o/p
recover -d sig
1 1 0 0
x axis
y axis V = 5V

1) Explain FHSS
FHSS is a wireless commm technique where the carrier freqn of transmitted s/g changes rapidly to a predefined hopping pattern known to both transmitter and receiver.
- The hopping occurs many times per sec
Appln - Bluetooth, cordless phones etc.

2) Compaire FHSS and DSSS

FHSS	DSSS
1) Rapidly hops the carrier s/g among many narrow freqn channels according to a psuedo-random seq.	1) multiplies the data s/g with high rate pseudo-random code, spreading it across a wide continuous band.
2) Hopping occurs in steps	2) spreading occurs instantly over the entire B/W
3) Need less instantaneous B/W for each loop	3) Require large continuous B/W at all times.
4) simpler RF front end	4) complex

3)

3) Explain hop rate, symbol rate for FH-SS

- Hop rate

Hop rate is the number of $freq^n$ changes (hops) the FHSS transmitter makes per second.

units Hops/sec

- eg If the system changes its carrier $freq^n$ 100 times every second the hop rate is 100 hops/s.

- Symbol rate

- symbol rate also called baud rate is the number of data symbols transmitted per second on the carrier $freq^n$.

- units symbols per second (baud)

eg If each hop transmits 4 symbols and there are 100 hops per second. the symbol rate $4 \times 100 = 400$ symbols/sec

4) How to calculate processing gain, jamming margin for FHSS

1) Processing Gain (Gp)

$$Gp = \frac{B_{ss}}{B_m}$$

B_{ss} spread spectrum B.W

B_m msg sig BW

2) Jamming margin

$$Jm = Gp - \left(\frac{E_b}{N_0}\right)_{req.}$$

EXPERIMENT NO. 7

AIM: To Simulate and study of Performance of M-ary PSK.

OBJECTIVE: Write a MATLAB code for M ary PSK.

SOFTWARE: MATLAB Software version________

THEORY:

The word binary represents two bits. M represents a digit that corresponds to the number of conditions, levels, or combinations possible for a given number of binary variables.

This is the type of digital modulation technique used for data transmission in which instead of one bit, two or more bits are transmitted at a time. As a single signal is used for multiple bit transmission, the channel bandwidth is reduced.

M-ary Equation

If a digital signal is given under four conditions, such as voltage levels, frequencies, phases, and amplitude, then M = 4.

The number of bits necessary to produce a given number of conditions is expressed mathematically as

N=log2M

Where

N is the number of bits necessary

M is the number of conditions, levels, or combinations possible with N bits.

The above equation can be re-arranged as

2N=M

For example, with two bits, 22 = 4 conditions are possible.

Types of M-ary Techniques

In general, Multi-level M−ary modulation techniques are used in digital communications as the digital inputs with more than two modulation levels are allowed on the transmitter's input. Hence, these techniques are bandwidth efficient.

There are many M-ary modulation techniques. Some of these techniques, modulate one parameter of the carrier signal, such as amplitude, phase, and frequency.

M-ary PSK

This is called as M-ary Phase Shift Keying M−ary PSK

The phase of the carrier signal, takes on M different levels.

Representation of M-ary PSK:

Algorithm

1) Initialize envuronment
clear all provious data close existing using clc & clear all;

2) Input M-ary PSK value;

3) Define symbol set

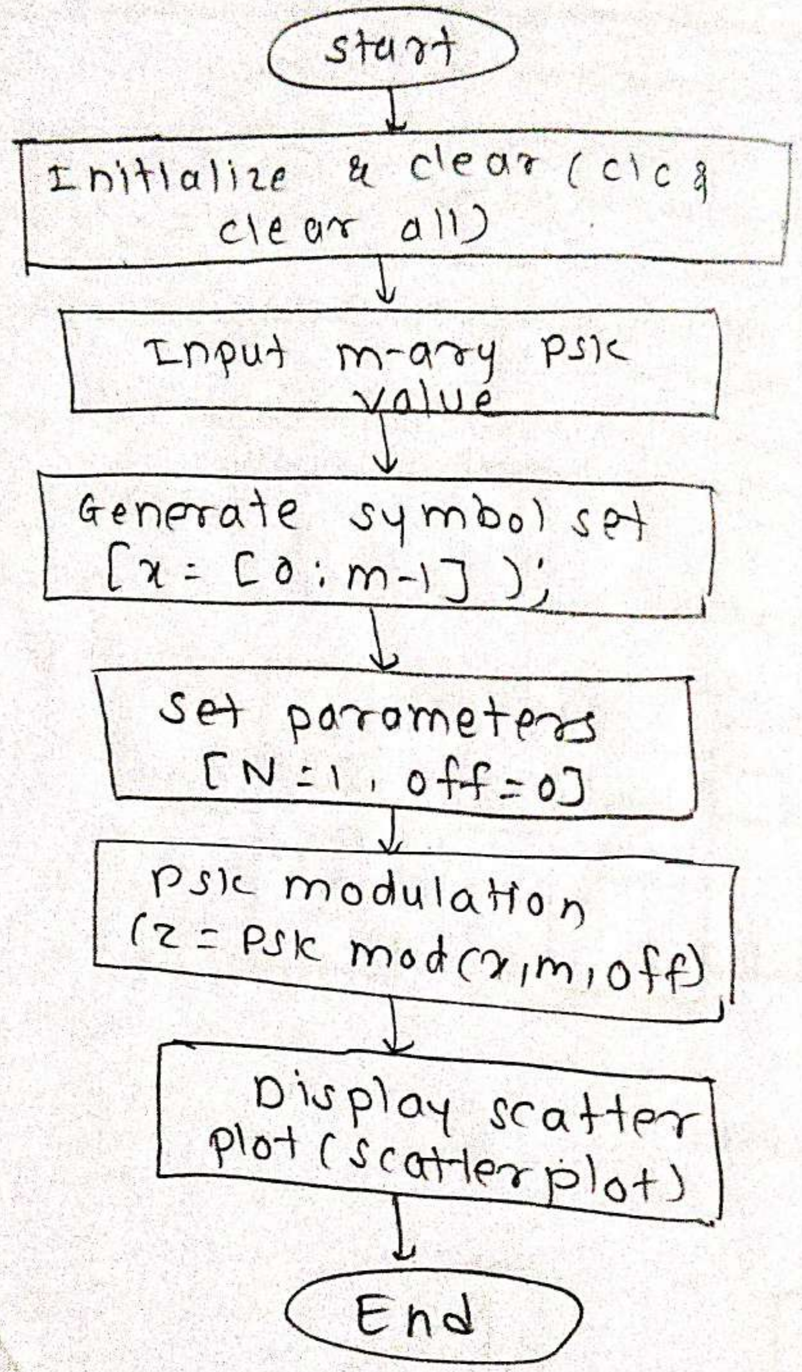
4) set parameters

i) set no. of symbols N to 1

ii) Set the phase offset off to 0

5) perform PSK modulation

6) scatter plot;
Generate & display a scatter ptot of the PSK


start
Initialize & clear (clc & clear all)
Input m-ary psk value
Generate symbol set [x = [0 : m-1]);
set parameters [N = 1 , off = 0]
psk modulation (z = psk mod(x, m, off)
Display scatter plot (scatterplot)
End

Flowchart:

Algorithm:

Program:

Input,

Output,

Result:

Questions:

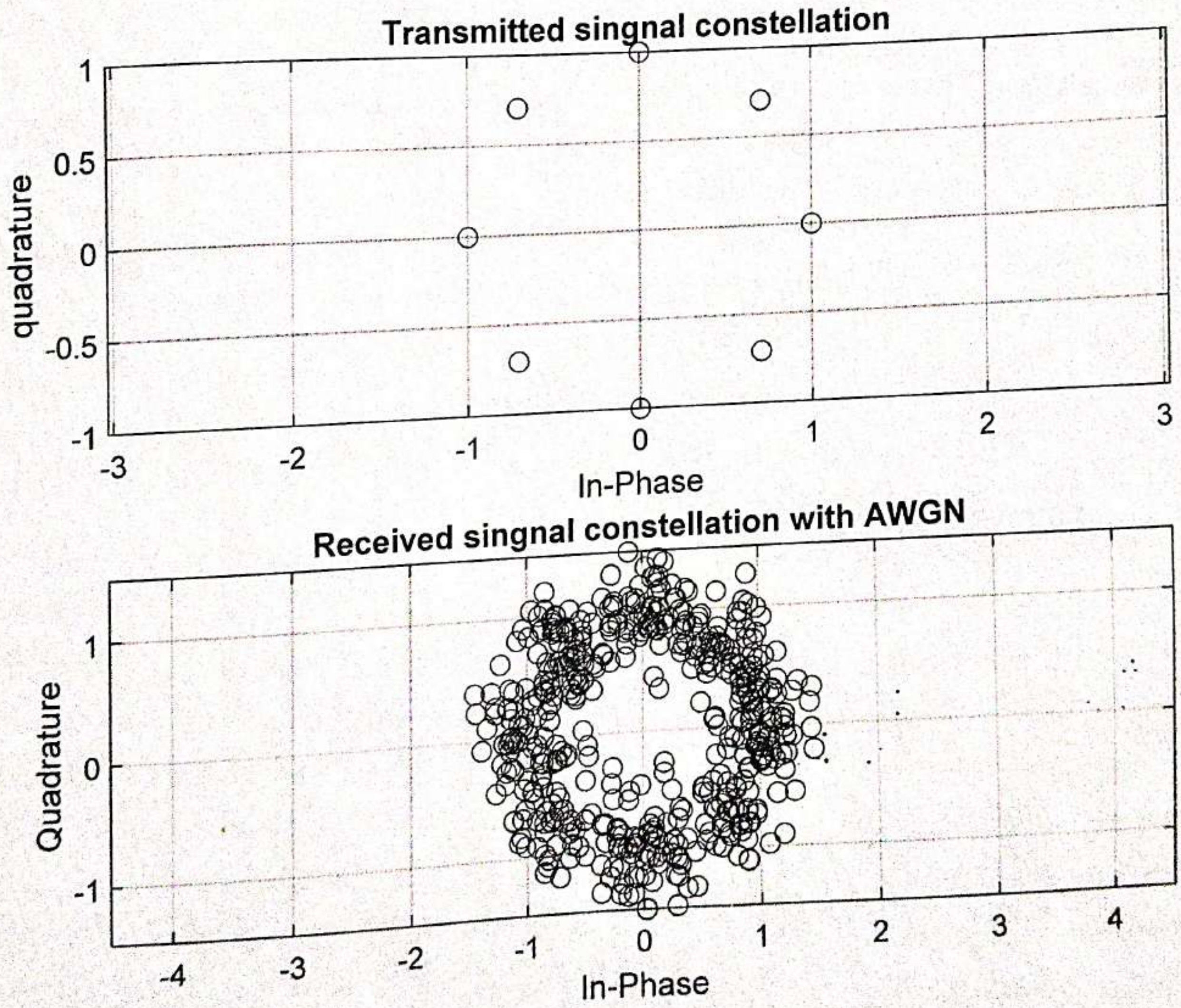
1. What is Mary PSK?
2. Difference between BPSK, QPSK and Mary PSK?
3. Calculate the Bandwidth of Mary PSK?

Conclusion:

In this experiment we successfully implement the code to perform M-ary PSK. We observed the transmitted signal constellation and received signal constellation

Timely submission (10)	Journal Presentation(10)	Performance(10)	Understanding(10)	Oral(10)	Total (50)
					43
Sub Teacher Sign:					

```
axis equal;
subplot(2,1,2);
plot(real(rxSignal),imag(rxSignal),'o');
title('Received singnal constellation with AWGN');
xlabel('In-Phase');
ylabel('Quadrature');
grid on;
axis equal;
```



1) What is Mary PSK

M-ary Phase shift keying is a digital modulation in which each symbol is represented by one of M equally spaced phases of a carrier. Each symbol conveys $\log_2 M$ bits. Typical examples.

$M = 2 \rightarrow$ BPSK
$M = 4 \rightarrow$ QPSK
$M = 8 \rightarrow$ 8-PSK

2) Difference betn BPSK, QPSK and M-ary PSK

Parameters	BPSK	QPSK	M-ary PSK
1) size	2 points	4 points	M equally spaced phase pts.
2) Bits/sym.	1	2	$\log_2 M$
3) symbol rate R_s	R_b	$R_b/2$	$R_b/\log_2 M$
4) B/W	$B = (1+\alpha) R_b$	$B = (1+\alpha)\frac{R_b}{2}$	$B = (1+\alpha)\frac{R_b}{\log_2 M}$
5) spectral efficiency (η)	$\frac{1}{1+\alpha}$	$\frac{2}{1+\alpha}$	$\frac{\log_2 M}{1+\alpha}$
6) complexity	simple	moderate	High

3) Calculate the bandwidth of M-ary PSK

$$B = (1+\alpha) R_s = (1+\alpha) \frac{R_b}{\log_2 M}$$

R_b = bit rate
R_s = symbol rate
α = roll-off factor

special cases

- BPSK ($M=2$)

$$R_s = R_b$$
$$B = (1+\alpha) R_b$$

- QPSK ($M=4$)

$$R_s = \frac{R_b}{2}$$

$$B = (1+\alpha) \frac{R_b}{2}$$

- In general (M-PSK)

$$B = (1+\alpha) \frac{R_b}{\log_2 M}$$

Bandwidth efficiency

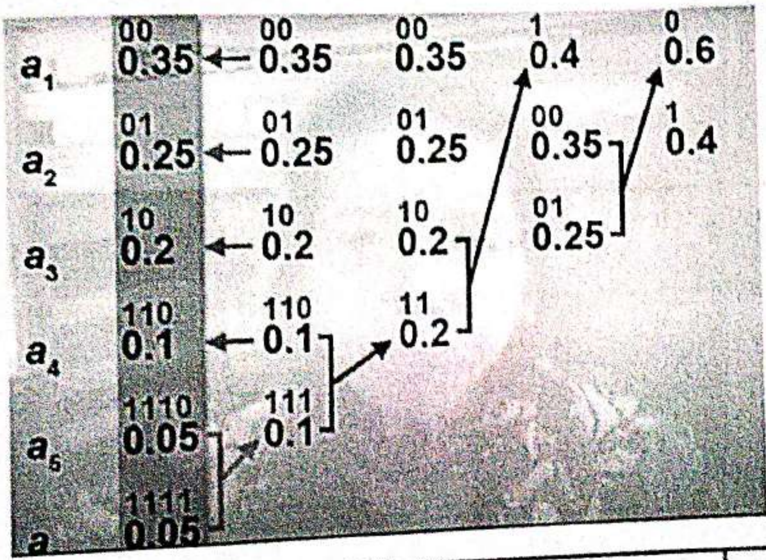
$$\eta = \frac{R_b}{B} = \frac{\log_2 M}{1+\alpha}$$

EXPERIMENT NO. 8

Aim: To understand Source Coding Theorem. Study of Huffman coding.
Objective: Study of Huffman coding Encoder and Decoder.
Software Used: MATLAB 7.6 software

Huffman coding is a lossless data compression algorithm. In this algorithm, a variable-length code is assigned to input different characters. The code length is related to how frequently characters are used. Most frequent characters have the smallest codes and longer codes for least frequent characters.

Huffman coding is an entropy coding algorithm used for lossless data compression, developed by David Huffman in 1952. The idea is to assign variable-length codes to input characters, the most frequent character gets the smallest code and the least frequent character gets the largest code. The Huffman algorithm is a so-called "greedy" approach to solving this problem in the sense that at each step, the algorithm chooses the best available option. It turns out that this is sufficient for finding the best encoding.



Message	Probability	Digits obtained by tracing	Codeword obtained by reading digits of column-3 from LSB side	No. of digits
m_0	$p_0 = 0.4$	1	1	(1)
m_1	$p_1 = 0.2$	10	01	(2)
m_2	$p_2 = 0.2$	000	000	(3)
m_3	$p_3 = 0.1$	0100	0010	(4)
m_4	$p_4 = 0.1$	1100	0011	(4)

Algorithm

1) start

2) create a priority queue containing all the symbols with their probabilities

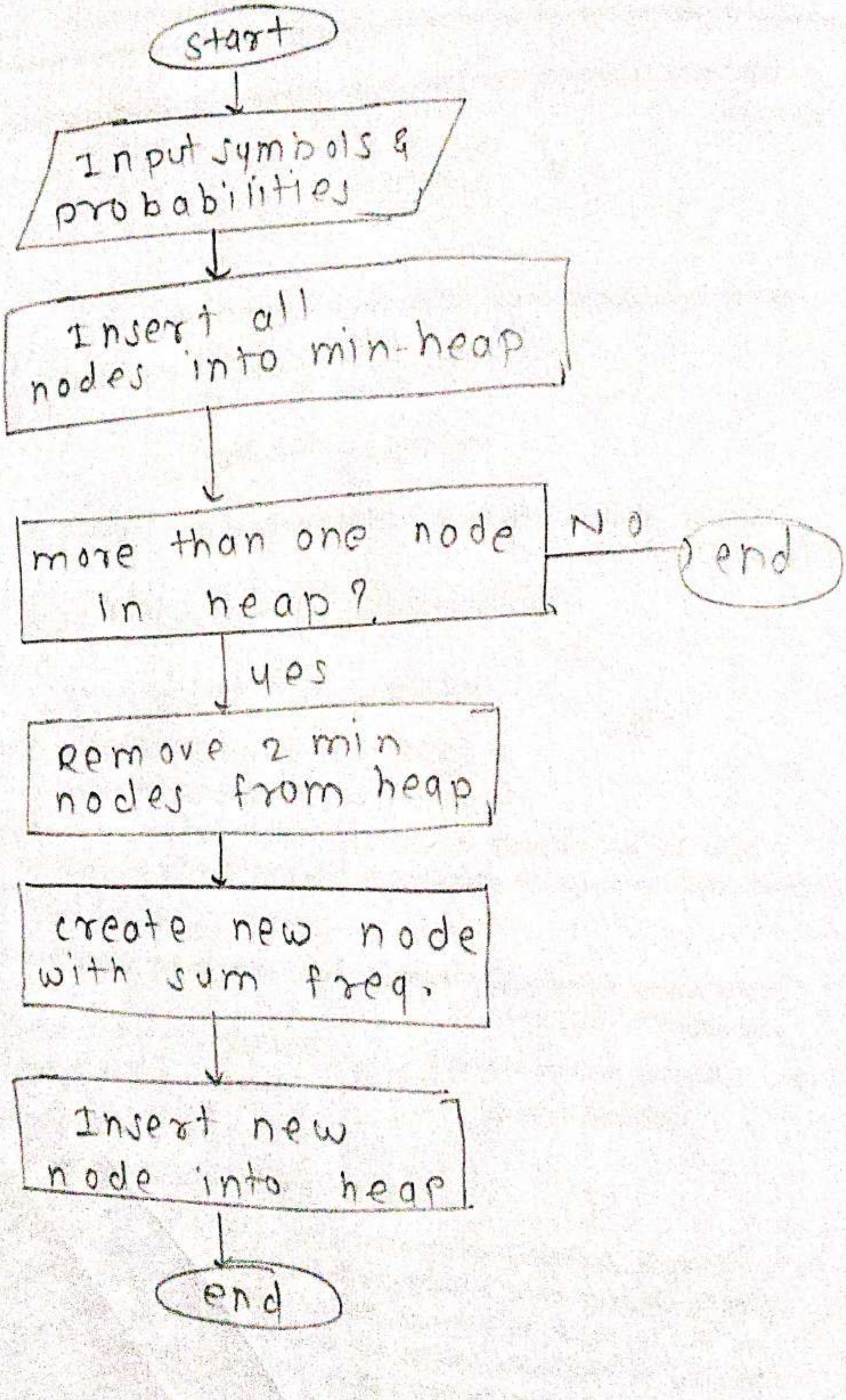
3) while there is more than one node in the queue.
   - Remove the nodes with smallest probabilities.
   - create a new internal node with these two nodes as children.
   - The probability of the new node = sum of the two smallest probabilities
   - Insert this new node back into the priority queue.

4) when only one node is left, it becomes the root of the huffman tree.

5) Assign codes to each symbol.
   - traverse the tree from root to each leaf.
   - Assign 0 for the left branch and 1 for the right branch.

6) output the codewords for all symbols.

7) Stop.


start
Input symbols &
probabilities
Insert all
nodes into min-heap
more than one node
in heap?
No
end
yes
Remove 2 min
nodes from heap
create new node
with sum freq.
Insert new
node into heap
end

Flowchart:

Algorithm:

Program:

Input,

Output,

Result:

Questions:

Solve the following two numerical:

1. Apply Huffman Coding for the symbols [A E H N G S] generated by a DMS with probabilities [0.19,0.15,0.2,0.16,0.4,0.08] Also calculate coding efficiency.
2. Encode the following symbols using Huffman source coding technique and calculate coding efficiency [1/4,1/8,1/16, 1/16, 1/16,1/4,1/16,1/8]

Conclusion:

In this experiment we perform huffmam coding. We write the code for huffman coding and huffman decoding and observed the output. we performed the s/w experiment on matlab.

Timely Submission(10)	Journal Presentation(10)	Performance (10)	Understanding(10)	Oral(10)	Total (50)
					44
Sub Teacher Sign					

1) Apply huffman coding for the symbols [A E H N G S] generated by DMS with probabilities [0.19, 0.15, 0.2, 0.16, 0.4, 0.08] Also calculate efficiency.

[m]	(P)	(P)				
A	~~0.19~~	0.4	0.4	0.4	0.43	0.75
E	~~0.16~~	0.2	0.23	0.35	0.4	0.43
H	~~0.15~~	0.19	0.20	0.23	0.35	
N	~~0.~~	0.16	0.19	0.20		
G		0.15	0.16			
S		0.08				

m	P	codeword	n
A	0.4	00	2
E	0.2	11	2
H	0.19	010	3
N	0.16	011	3
G	0.15	100	3
S	0.08	101	3

$$L = \sum_{k=1}^{6} (P_k \cdot n_k)$$

$$L = (0.4 \times 2) + (0.2 \times 2) + (0.19 \times 3) + (0.16 \times 3) + (0.15 \times 3) + (0.08 \times 3)$$

$$L = 2.94$$

$$H = \sum_{k=1}^{6} P_k \log_2 (1/P_k) = 2.573$$

$$\text{efficiency} = \frac{H}{L} \times 100$$

efficiency = 87.51 %

NMIET

2) Encode the following symbols using Huffman source coding technique and calculate coding efficiency [1/4, 1/8, 1/16, 1/16, 1/16, 1/4, 1/16, 1/8]

[m]	[P]					
x_1	1/4	1/4	1/4	1/4	1/4	1/4
x_2	1/4	1/4	1/4	1/4	1/4	1/4 0
x_3	1/8	1/8	1/8	1/8	1/8 0	1/80 1
x_4	1/8	1/8	1/8	1/8 0	1/72 1	
x_5	1/16	1/16	1/16 0	1/64 1		
x_6	1/16	1/16 0	1/48 1			
x_7	1/16 0	1/32 1				
x_8	1/16 1					

[m]	[P]	codeword	n
x_1	1/4	0	1
x_2	1/4	10	2
x_3	1/8	110	3
x_4	1/8	1110	4
x_5	1/16	11110	5
x_6	1/16	111110	6
x_7	1/16	1111110	7
x_8	1/16	1111111	7

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

$$L = 2.758$$

$$H = \sum_{k=1}^{n} p_k \log_2(1/p_k) \qquad H = 2.75$$

$$\eta = \frac{H}{L} \times 100 \qquad \eta = 98.92\,\%$$

```
%Huffman Coding & Decoding Program
clc;
clear all;
close all;
code_length=0;
x= input('Enter number of sysmbols');
for m = 1:x
   symbols(m) = input('Enter the symbol number:');
   p(m) = input('Enter the probability:');
end
Hx=0;
for m =1:x
   [dict,avglen] = huffmandict(symbols,p);
   hcode =huffmanenco(m, dict);
   display(hcode);
   disg= huffmandeco(hcode, dict);
   display(disg);
   code_length= length(hcode);
   display(code_length);
   Hx= Hx+(p(m)*(-log(p(m)))/(log(2)));

end
display(Hx);
efficiency= (Hx/avglen)*100;
```

Input:

```
Enter number of sysmbols4
Enter the symbol number:1
Enter the probability:0.4
```

EXPERIMENT NO. 9

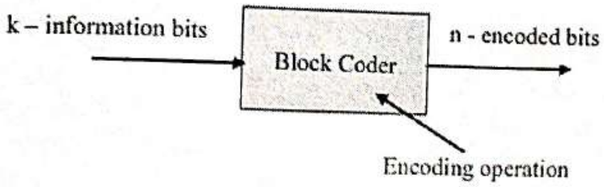
Aim: To understand linear block coding and decoding techniques.
Objective: Study of Linear Block Code Encoder and Decoder.
Software Used: MATLAB 7.6 software

Theory:

Errors are introduced in the data through the channel. The channel noise interferes the signal. The number of errors introduced due to channel noise are minimized by the encoder by adding redundant bits. This increases the overall data rate. Hence channel has to accommodate this increased data rate, and the system becomes slightly complex due to coding techniques. The codes are classified as block or convolution codes.
For a linear code, if two code words are added by modulo 2 arithmetic then, it produces a third code word in the case.

LINEAR BLOCK CODES:

(n, k) Block codes



n-digit codeword made up of k-information digits and (n-k) redundant parity check digits. The rate or efficiency for this code is k/n.

$$\text{Code efficiency } r = \frac{k}{n} = \frac{\text{Number of information bits}}{\text{Total number of bits in codeword}}$$

Note: unlike source coding, in which data is compressed, here redundancy is deliberately added, to achieve error detection.

SYSTEMATIC BLOCK CODES

A systematic block code consists of vectors whose 1st k elements (or last k-elements) are identical to the message bits, the remaining (n-k) elements being check bits. A code vector then takes the form:

$$X = (m_0, m_1, m_2, \ldots\ldots m_{k-1}, c_0, c_1, c_2, \ldots\ldots c_{n-k})$$

Or

$$X = (c_0, c_1, c_2, \ldots\ldots c_{n-k}, m_0, m_1, m_2, \ldots\ldots m_{k-1})$$

Conclusion:

ORAL QUESTIONS

1) What is Hamming weight of a code word?
2) What is Hamming distance, Code rate, Word Length,
3) Define Minimum Hamming Distance, Block length, Constraint Length
4) What is block code, Hamming Code.
5) What are the properties of a linear block code?
6) What is a systematic code, Block code?
7) What is singleton bound, Burst Error?
8) What is parity check matrix?
9) How to generate generator matrix.
10) Types of Error control methods
11) What do you mean by (n, k) code?

Timely Submission(10)	Journal Presenattaion(10)	Performance (10)	Understanding(10)	Oral(10)	Total (50)
Sub Teacher Sign					

1) What is hamming weight of a code word?

→ The hamming weight of a code word is the number of non-zero symbols in the code word.

- Take a binary code word.
- Count how many bits are 1.
- That count = Hamming weight

Example

- code word = 1011001
- Number of 1's = 4
- ~~Aap.~~ Hamming weight = 4.

- Hamming weight is used to measure the error detection and correction capability of a code.
- The minimum Hamming weight among all non-zero code words determines the minimum ~~Happ~~ Hamming distance of the code.

2) What is Hamming distance, code rate, word length?

→ - The hamming distance between two code words is the number of bit positions in which they differ.

- In other words, it is the count of substitutions required to change one code word into another.
- code word 1 = 101100

NMIET

code word 2 = 111110
Difference positions = 2 (2nd and 5th bi
so, Hamming distance = 2
- minimum Hamming distance of a code determines how many errors can be detected or corrected

2) code Rade (R)
- code rate is the ratio of number of information bits (k) to the total number of bits in a code word (n)

$$R = \frac{k}{n}$$

3) word length
The word length is the total number of bits in a code word.
- It represents how many bits are transmitted together as one unit.
Eg- In a Hamming (7,4) code:
word length = 7 bits (4 info + 3 parity

3) Define minimum Hamming distance, Block length, constraint length
→ 1) minimum hamming distance
- The minimum Hamming distance of a c is the smallest hamming distance betw any two distinct valid code words.
- It determines the error detection and correction capability.

- It determines the error detection of a code is
- It determines the error detection and correction capability of the code.

2) Block length (n)
- In block codes, the block length is the total number of bits in a code word
- Denoted as n

3) Constraint length (k)
- In convolutional codes, the constraint length is the number of i/p bits that affect the o/p.
- It represents the memory of the encoder

$$k = m + 1$$

4) What is block code, Hamming code.
1) Block code
- A block code is a type of error-control code where the input message is divided into fixed size blocks of k bits, and each block is encoded into a larger n-bit codeword.
- It is generally (decoded) denoted as (n, k) code.
k = no. of info. bits.
n = no. of bits in codeword.

$$R = \frac{k}{n}$$

2) Hamming code
- The hamming code is a special type of linear block code invented by richard hamming.
- It is used for single-bit error correctio[n] and double-bit error detection.
- It inserts parity bits at positions that are powers of 2 in the code word.

6) What is a systematic code.
- A code is called systematic if the origin[al] msg bits appear unchanged in the codeword, and the parity bits are simply appended.
- This makes decoding easier because the data bits can be read directly.

7) what is singleton bound, Burst Error?

→ The singleton bound is a fundamental limit in coding theory that relates the parameters of the block code.

- For a block code with
  - n = code word length
  - k = number of message bits.
  - d-min = minimum Hamming distance.

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

8) What is parity check matrix

- The parity check matrix is a matrix used in coding theory to detect and sometimes correct errors in a linear block code.

- For a linear block code with
  - n = code word length
  - k = number of information bits
  - (n, k) code.

$$H \cdot C^T = 0$$

- Any valid code word, when multiplied by H^T gives zero.
- If the result is non-zero, it indicates errors in the received code word.
- Typically written as,

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

P^T = transpose of matrix

I_{n-k} = identity matrix

EXPERIMENT NO. 10

AIM: To Simulate Study of Convolutional codes

OBJECTIVE: Simulation study of Convolution codes

SOFTWARE: Code block Software, Turbo C++.

THEORY:

This experiment describes the encoder and decoder structures for convolutional codes. The encoder will be represented in many different but equivalent ways. Also, the main decoding strategy for convolutional codes, based on the Viterbi Algorithm, will be described. A firm understanding of convolutional codes is an important prerequisite to the understanding of turbo codes. 2.1 Encoder Structure

A convolutional code introduces redundant bits into the data stream through the use of linear shift registers as shown in Figure 2.1.

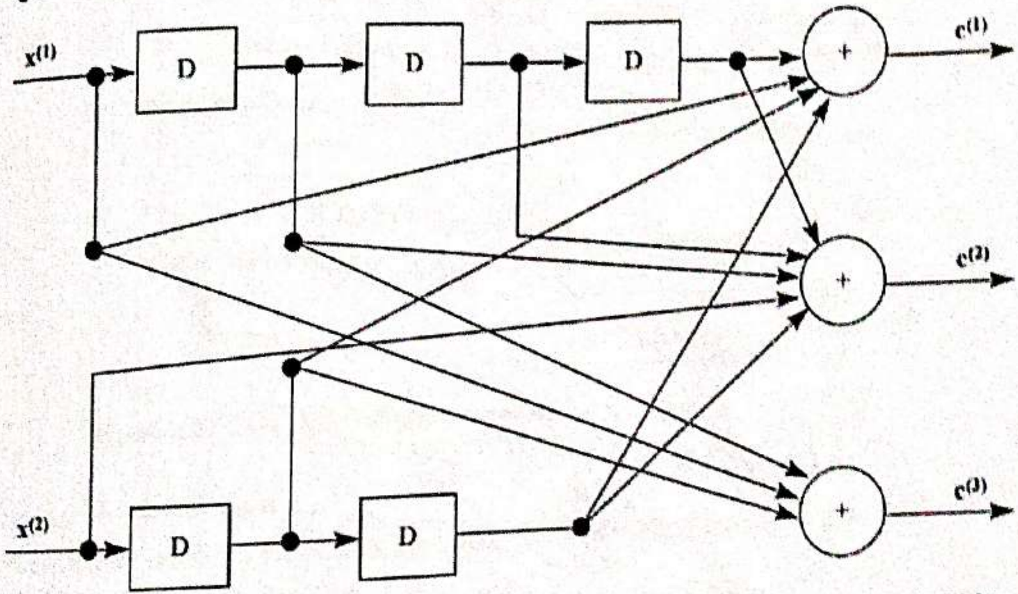


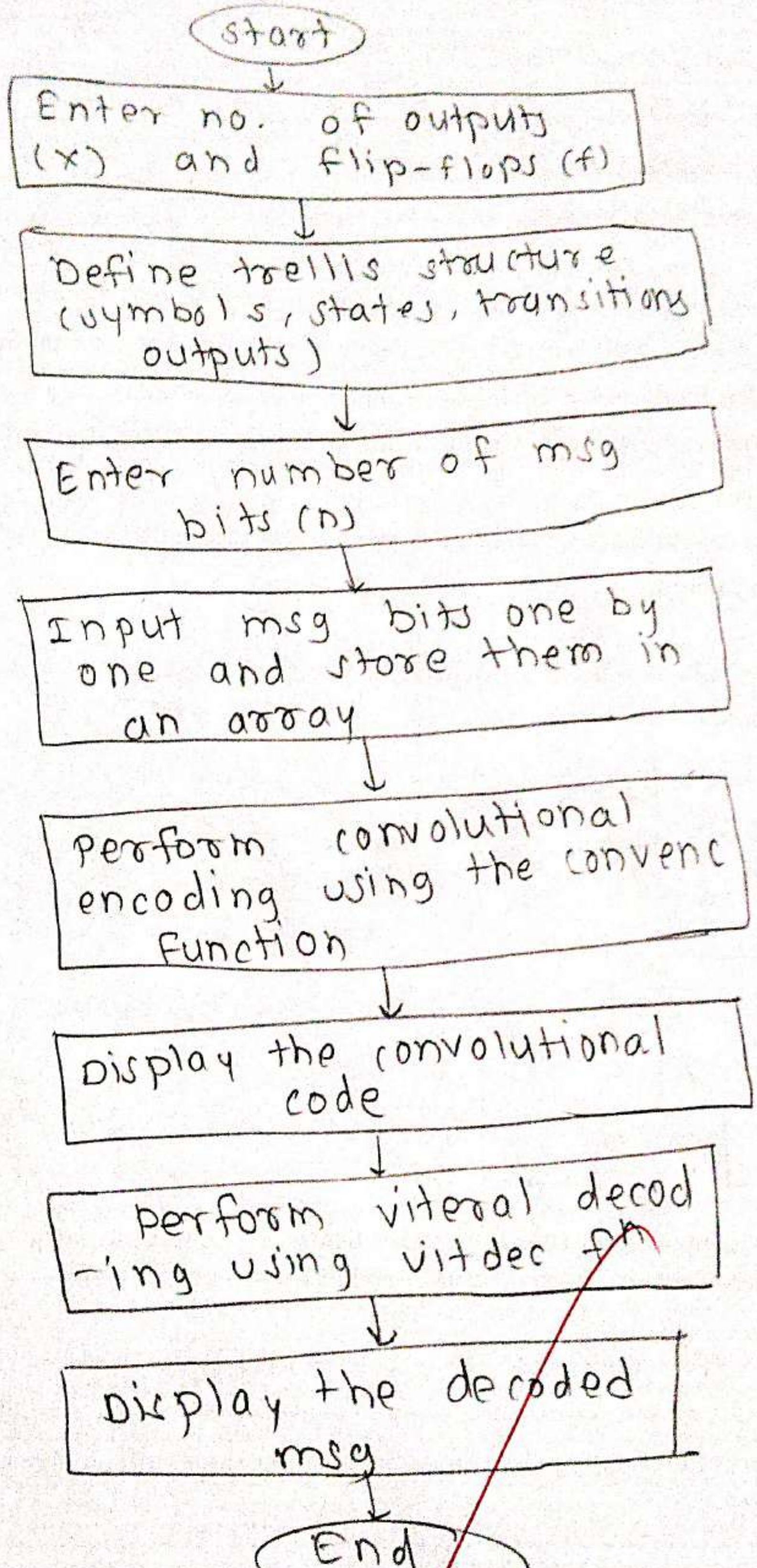
Figure 2.1: Example convolutional encoder where $x^{(i)}$ is an input information bit stream and $c^{(i)}$ is an output encoded bit stream [Wic95].

The information bits are input into shift registers and the output encoded bits are obtained by modulo-2 addition of the input information bits and the contents of the shift registers. The connections to the modulo-2 adders were developed heuristically with no algebraic or combinatorial foundation.

The code rate r for a convolutional code is defined as

$$r = \frac{k}{n}$$

Decoder



Start
Enter no. of outputs (x) and flip-flops (f)
Define trellis structure (symbols, states, transitions outputs)
Enter number of msg bits (n)
Input msg bits one by one and store them in an array
Perform convolutional encoding using the convenc function
Display the convolutional code
Perform viteral decod-ing using vitdec fn
Display the decoded msg
End

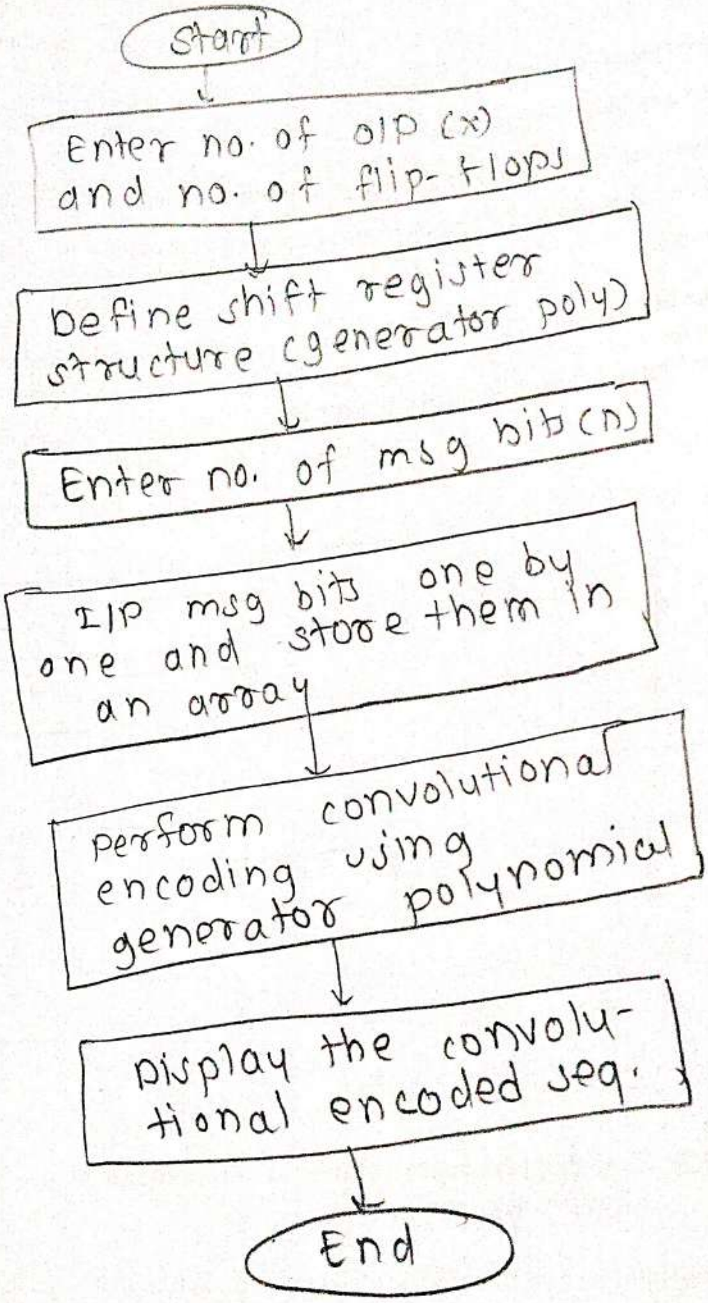
Algorithm for convolution code decoder

1) start
2) Input parameter
3) Define the trellis structure.
4) Input the msg
5) perform convolutional encoding
6) Display the convolution code
7) perform viterbi decoding
8) Display the decoded msg.
9) stop

Algorithm for convolution code encoder

1) start
2) Input parameters
3) Define the shift register structure
4) Input the msg bits.
5) perform convolutional encoding using generator polynomial.
6) Generate the encoded output seq.
7) Display the convolutional encoded msg.
8) Stop

* encoder



3 State D
encoder. T
shows the

Fi

In the state
information
denoted as
convolution
x={1011} (
produces th
the state dia

Flowchart:

Algorithm:

Program:

Input,

Output,

Result:

Questions:

1. Define code rate related to convolutional codes?
2. Define coding gain and constraint Length?
3. Explain the methods of decoding of convolutional coding?

Conclusion:

In this experiment we successfully implemented the code for convolution decoder and encoder. Here we get the expected output. On matlab we performed this experiment.

Timely Submission(10)	Journal Presenattaion(10)	Performance (10)	Understanding(10)	Oral(10)	Total (50)
					43
Sub Teacher Sign					9/10

Name:SakshiDhumal
Roll No:13

```
% Convolution Encoder
x=input('Enter number of outputs:');
f=input('Enter number of flip flops:');

%Define the trellis structure
trellis=struct('numInputSymbols',2,'numOutputSymbols',4,...
'numStates',4,'nextStates',[0 2;2 3;3 1;1 3],...
'outputs',[0 3;3 0;2 1;1 2]);

%input the message
n=input('enter number of message bits:');
msg=zeros(1,n);
fori=1:n
msg(i)=input('enter message bits:');

end

%perform the convolution encoding
code=convenc(msg,trellis);

%display the convolution code
fprintf('the convolution code is:\n');
fori=1:length(code)
fprintf('%d',code(i));
end
fprintf('\n');
```

OUTPUT:

Enter number of outputs:5

Enter number of flip flops:3

enter number of message bits:5

enter message bits:1

enter message bits:0

enter message bits:1

enter message bits:1

enter message bits:0

1) Define code rate related to convolutional codes?

→ Code rate (R) in convolutional codes is defined as the ratio of no. of I/P to the no. of O/P bits generated by the encoder per encoding per encoding cycle.

$$R = \frac{k}{n}$$

where, k = no. of i/p bits.
n = no. of o/p bits produced for the those k i/p bits.

e.g- If a convolutional encoder takes 1 i/p bit & produces 2 o/p bits. then the code rate is,

$$R = \frac{1}{2}$$

2) Define coding gain & constraint length?

→ 1) Coding Gain

Coding gain is improvement in signal-to-noise ratio achieved by using an error control code compared to an uncoded system for the same probability of error.

Coding Gain = SNR (uncoded - SNR (coded)

units → dB

2) Constraint length (k)

The constraint length of a convolution code is no. of bits in the encoder's memory that affect the generation of the o/p bits.

$k = m+1$

Defines memory & complexity of code

Q.3) Explain methods of coding of convolution coding?

1) sequential Decoding

sequential decoding is probabilities search method that explores possible paths in the trellis without evaluating of them