Govt. of N.C.T of Delhi

G.B PANT ENGINEERING COLLEGE

Okhla Industrial Estate Phase-III, New Delhi – 110020

Department of Electronics and Communication Engineering

B. Tech (CSE) – 5th Semester

ETEC-357 Digital Communication Lab Practical File

Submitted by:
RAVI PAWAR
03620902718
CSE 5th SEM

Submitted to:
Mr. Padam Saini

Index

S.no.	Title	Page No.	Remarks

Practical 1

Expt. No. 1

Date

Page No.

AIM

To study Sampling Theorem

APPARATUS REQUIRED

System Running with MATLAB software

THEORY

Sampling:

By sampling a continuous time signal at isolated equally spaced points in time, we obtain sequence of numbers

$n \in \{\ldots, -2, -1, 0, 1, 2, 3, \ldots\}$

T_s is the Sampling Period

Many Signal originate as continuous-time signal, eg-> Conventional music or voice

Frequency Domain:

- Replicates Spectrum of Continuous-Time signal at offsets that are integer multiples of sampling frequency.
- Fourier Series of impulse train where $\omega_c = 2\pi f_s$
- As sampling rate increases, sampled waveform looks more and more like the original.
- Many applications (eg. communication systems) care more about frequency content in the waveform and not its shape
- Zero crossing: frequency content of sinusoid when it meets the axis.

Teacher's Signature :

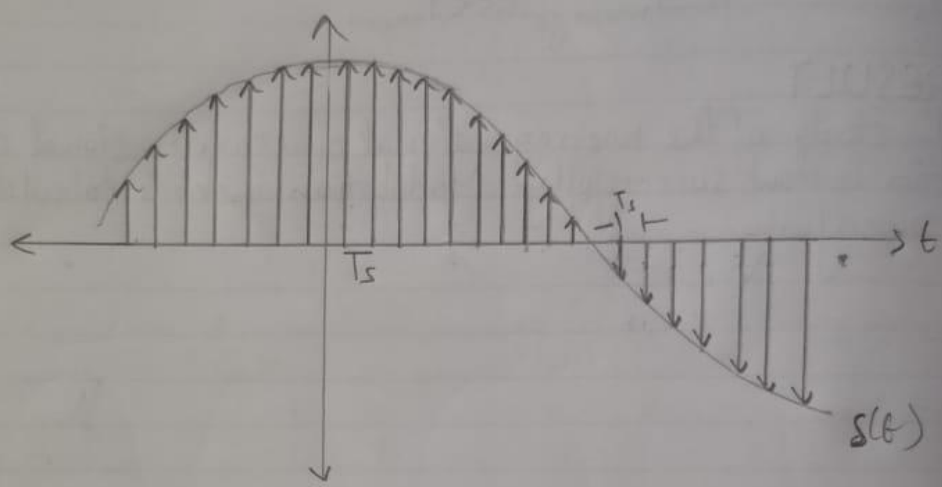


Fig: Sampled Analog Waveform

Expt. No. ______ Date ______ Page No. ______

Shanan Sampling Theorem

A continuous time signal $x(t)$ with frequency no higher than f_{max} can be reconstructed from its samples $x[n] = x(nT_s)$ by taking samples at a rate f_s which is greater than $2f_{max}$.

$$\text{Nquist rate} = 2f_{max}$$
$$\text{Nquist frequency} = f_s/2$$

Consider a sinusoid $\sin(2\pi f_{max} t)$
use a sampling period of $T_s = 1/f_s = 1/2f_{max}$

Assumptions

1. Continuous Time signal has no frequency content above f_{max}
2. Sampling time is exactly the same b/w any two samples
3. Sequence of no. obtained by sampling is represented in exact precision.
4. Conversion of sequence to continuous time is ideal.

Aliasing

→ Analog sinusoid.
$x(t) = A\cos(2\pi f_0 t + \phi)$
→ Sampled at $T_s = 1/f_s$
$x[n] = x(T_s n) = A\cos(2\pi f_0 T_s n + \phi)$
→ Keeping the sampling period same, sample
$y(t) = A\cos(2\pi(f_0 + l f_s)t + \phi)$
where l is an integer
$$y[n] = y(T_s n)$$
$$= A\cos(2\pi(f_0 + l f_s)T_s n + \phi)$$
$$= A\cos(2\pi f_0 T_s n + 2\pi l f_s T_s n + \phi)$$

Teacher's Signature : ______

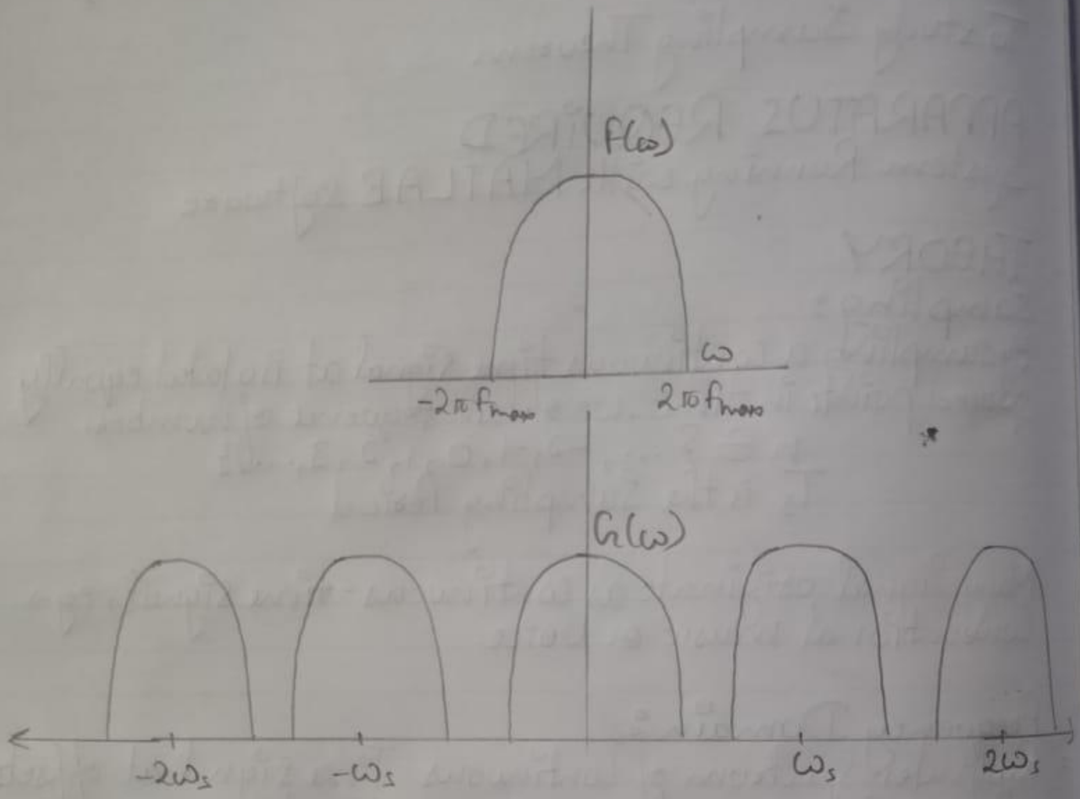


Fig:- Frequency Domains.

Expt. No. ____________ Date ____________

Page No. ________

$= A\cos(2\pi f_0 T_s n + 2\pi l n + \phi)$

$= A\cos(2\pi f_0 T_s n + \phi)$

$= x[n]$

Here, $F_s T_s = 1$

Since l is an integer

$\cos(n \pm 2\pi l) = \cos(n)$

$\Rightarrow$ $y[n]$ indistinguishable from $x[n]$.

CONCLUSION

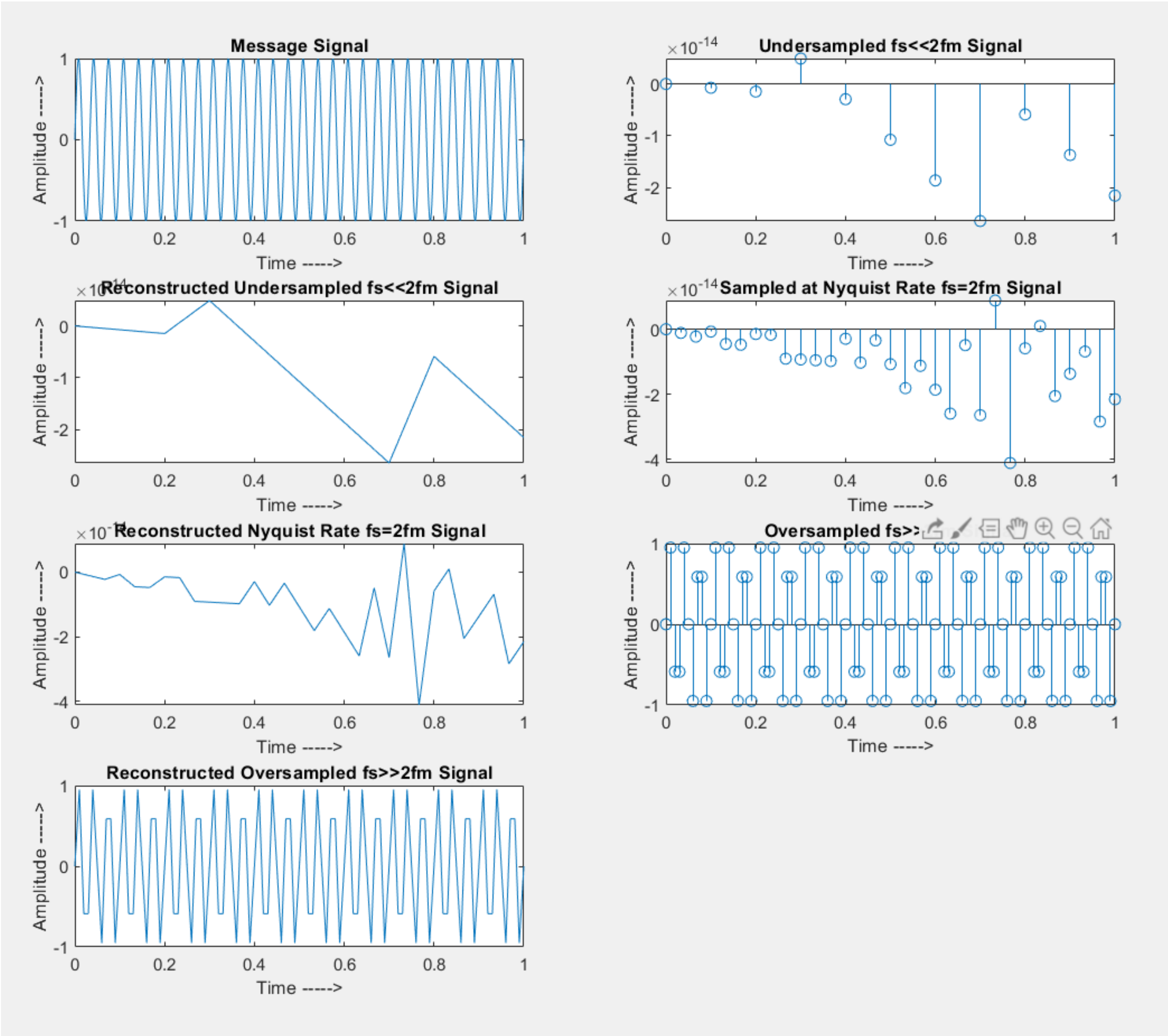
Hence, we successfully studied about Sampling Theorem.

Teacher's Signature : ____________

Program (Sampling Theorem)

MATLAB PROGRAM TO IMPLEMENT SAMPLING THEOREM

```
t = 0:0.001:1;
fm = input ('Enter the modulating signal frequency = ');
x = sin(2*pi*fm*t);
subplot (4,2,1);
plot(t,x);
xlabel('Time ----->');
ylabel('Amplitude ----->');
title('Message Signal');
fs1 = input('Enter Sampling Frequency < Modulating Signal Frequency = ');
fs2 = input('Enter Sampling Frequency = Modulating Signal Frequency = ');
fs3 = input('Enter Sampling Frequency > Modulating Signal Frequency = ');

%Sampling at fs<<2fm
n = 0:1/fs1:1;
x1 = sin(2*pi*fm*n);
subplot(4,2,2);
stem(n,x1);
xlabel('Time ----->');
ylabel('Amplitude ----->');
title('Undersampled fs<<2fm Signal');
subplot(4,2,3);
plot(n,x1);
xlabel('Time ----->');
ylabel('Amplitude ----->');
title('Reconstructed Undersampled fs<<2fm Signal');

%Sampling at fs=2fm
n = 0:1/fs2:1;
x2 = sin(2*pi*fm*n);
subplot(4,2,4);
stem(n,x2);
xlabel('Time ----->');
ylabel('Amplitude ----->');
title('Sampled at Nyquist Rate fs=2fm Signal');
subplot(4,2,5);
plot(n,x2);
xlabel('Time ----->');
ylabel('Amplitude ----->');
title('Reconstructed Nyquist Rate fs=2fm Signal');

%Sampling at fs>>2fm
n = 0:1/fs3:1;
x3 = sin(2*pi*fm*n);
subplot(4,2,6);
stem(n,x3);
xlabel('Time ----->');
ylabel('Amplitude ----->');
title('Oversampled fs>>2fm Signal');
subplot(4,2,7);
plot(n,x3);
xlabel('Time ----->');
ylabel('Amplitude ----->');
title('Reconstructed Oversampled fs>>2fm Signal');
```



Message Signal
Undersampled fs<<2fm Signal
Reconstructed Undersampled fs<<2fm Signal
Sampled at Nyquist Rate fs=2fm Signal
Reconstructed Nyquist Rate fs=2fm Signal
Oversampled fs>>
Reconstructed Oversampled fs>>2fm Signal
Amplitude ----->
Time ----->
×10^-14

Practical 2

Expt. No. 2 Date Page No.

AIM

To study Pulse Code Modulation (PCM)

THEORY

Pulse Code Modulation (PCM) is a method used to digitally represent sampled analog signals. It is the standard form for digital audio in computers and various Blu-ray, Compact Disc and DVD formats, as well as other uses such as digital telephone systems. A PCM stream is a digital representation of an analog signal, in which the magnitude of the analog signal is sampled regularly at uniform intervals, with each sample being quantized to the nearest value within a range of digital steps.

Basis of Pulse Code Modulation.

The three steps for developing an equivalent PCM digital signal from an analog signal are-

1. Sampling
2. Quantization
3. Coding

1. Sampling

The foundation of PCM is based on Nyquist Sampling Theorem. If a unlimited signal is sampled at regular intervals of time and at a rate equal to or higher than twice the highest significant signal frequency, then the sample contains all the information of the original signal. The original signal may then be reconstructed by use of a low pass filter.

Teacher's Signature :

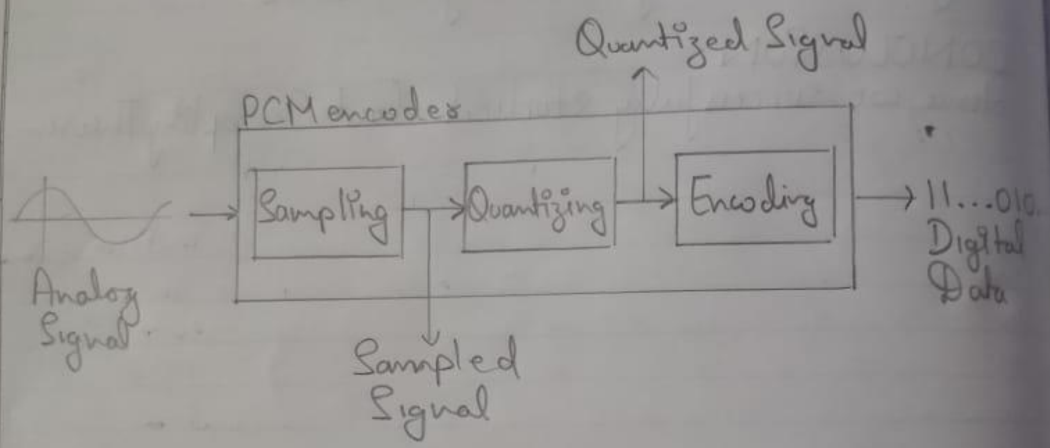


Fig:- Block Diagram Pulse Code Modulation

Date ______

Expt. No. ______ Page No. ______

② Quantization.

The digitization of analog signals involves the rounding off of the values which are approximately equal to the analog values. The method of sampling chooses a few points on the analog signal and then these points are joined to round off the value of a near stabilized value. Such process is called Quantization.

③ Coding

Older PCM system uses 7 bit code, and modern systems use an 8 bit code with its improved quantizing distortion performance. The companding and ~~expanding~~ coding is done together simultaneously. The compression and later expansion functions are logarithmic. A pseudologarithmic curve made up of linear segments imparts finer granularity to low level signals and less granularity to the higher-level signals.

Advantages of Pulse Code Modulation

1. Immune to channel induced noise and distortion.
2. Repeaters can be employed along the transmitting channel
3. Encoders allow secured data transmission.
4. It ensures uniform transmission quality

Disadvantages of Pulse Code Modulation.

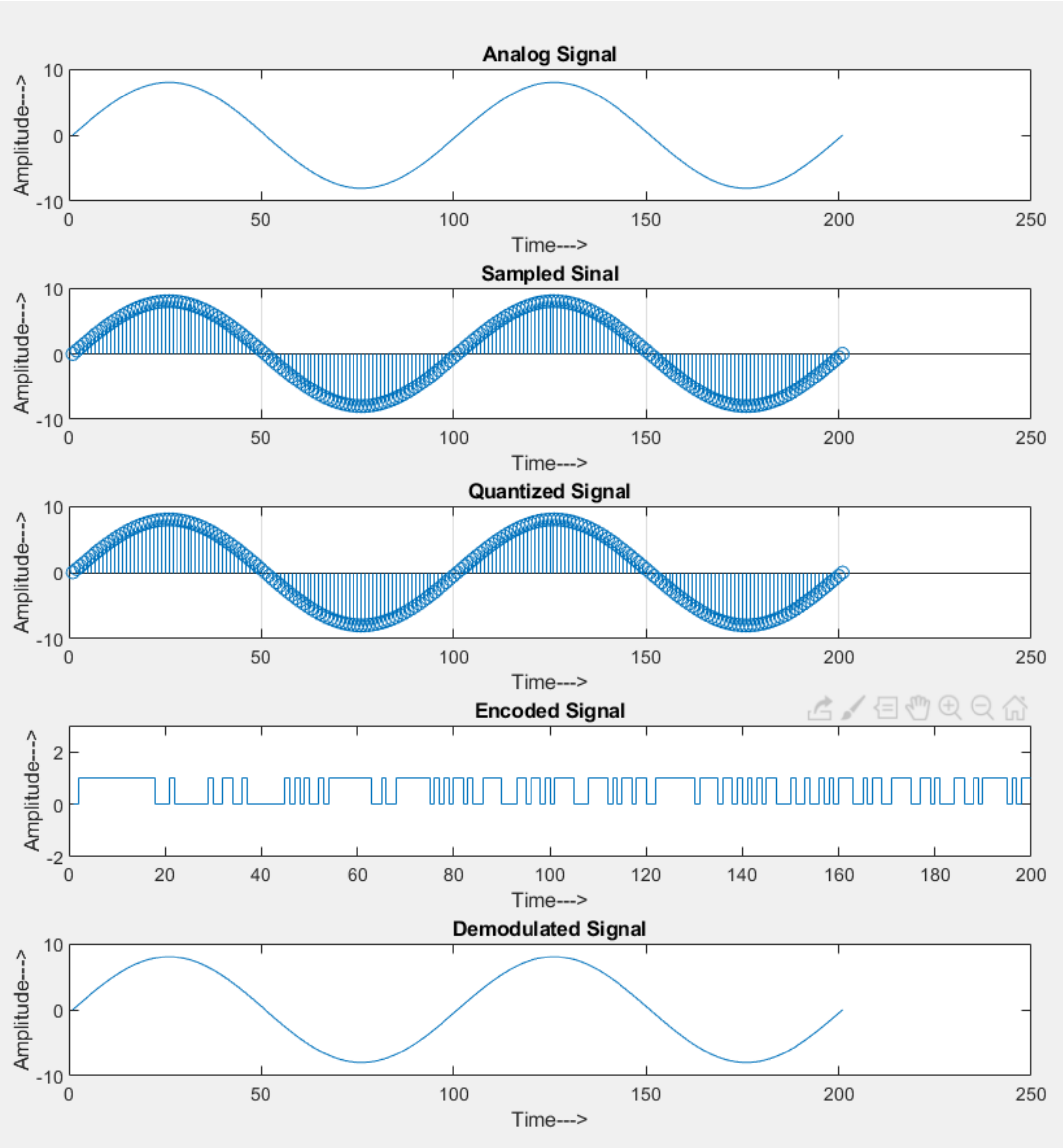
1. Pulse Code Modulation Increases the transmission Bandwidth
2. A PCM system is somewhat more complex than other system.

Teacher's Signature : ______

MATLAB PROGRAM TO IMPLEMENT PULSE CODE MODULATION AND DEMODULATION

```
n=input('Enter n value for n-bit PCM system : ');
n1=input('Enter number of samples in a period : ');
L=2^n;

% Sampling Operation
x=0:2*pi/n1:4*pi;
s=8*sin(x);
subplot(5,1,1);
plot(s);
title('Analog Signal');
ylabel('Amplitude--->');
xlabel('Time--->');
subplot(5,1,2);
stem(s);grid on; title('Sampled Sinal'); ylabel('Amplitude--->'); xlabel('Time--->');

% Quantization Process
vmax=8;
vmin=-vmax;
del=(vmax-vmin)/L;
part=vmin:del:vmax;
code=vmin-(del/2):del:vmax+(del/2);
[ind,q]=quantiz(s,part,code);
l1=length(ind);
l2=length(q);
for i=1:l1
if(ind(i)~=0)
ind(i)=ind(i)-1;
end
i=i+1;
end
for i=1:l2
if(q(i)==vmin-(del/2))
q(i)=vmin+(del/2);
end
end
subplot(5,1,3);
stem(q);grid on;
title('Quantized Signal');
ylabel('Amplitude--->');
xlabel('Time--->');

% Encoding Process
code=de2bi(ind,'left-msb');
k=1;
for i=1:l1
for j=1:n
coded(k)=code(i,j);
j=j+1;
k=k+1;
end
i=i+1;
end
subplot(5,1,4); grid on;
```

```
stairs(coded);
axis([0 2*n1 -2 3]); title('Encoded Signal');
ylabel('Amplitude--->');
xlabel('Time--->');

% Demodulation Of PCM signal
qunt=reshape(coded,n,length(coded)/n);
index=bi2de(qunt','left-msb');
q=del*index+vmin+(del/2);
subplot(5,1,5); grid on;
plot(q);
title('Demodulated Signal');
ylabel('Amplitude--->');
xlabel('Time--->');
```



Practical 3

Date

Expt. No. 3

Page No.

AIM

To study Delta Modulation (DM).

APPARATUS

System running MATLAB software

THEORY

Delta Modulation is an Analog to Digital and Digital to Analog Signal Conversion Technique used for Transmission of voice information where quality is not of primary importance. DM is the simplest form of Differential Pulse-Code Modulation (DPCM) where the difference b/w successive samples are encoded into n-bit data streams. In delta modulation, the transmitted data are reduced to 1-bit stream. It's main features are—

- The analog signal is approximated with a series of segment
- Each segment of the approximated signal is compared to the preceding bits and the successive bits are determined by this comparison
- Only the change of information is sent, that is, only an increase or decrease of the signal amplitude from the previous sample is sent whereas a no-change condition causes the modulated signal to remain at the same 0 or 1 state of the previous sample.

To achieve high SNR ratio, delta modulation must use over sampling techniques, that is, the analog signal is

Teacher's Signature :

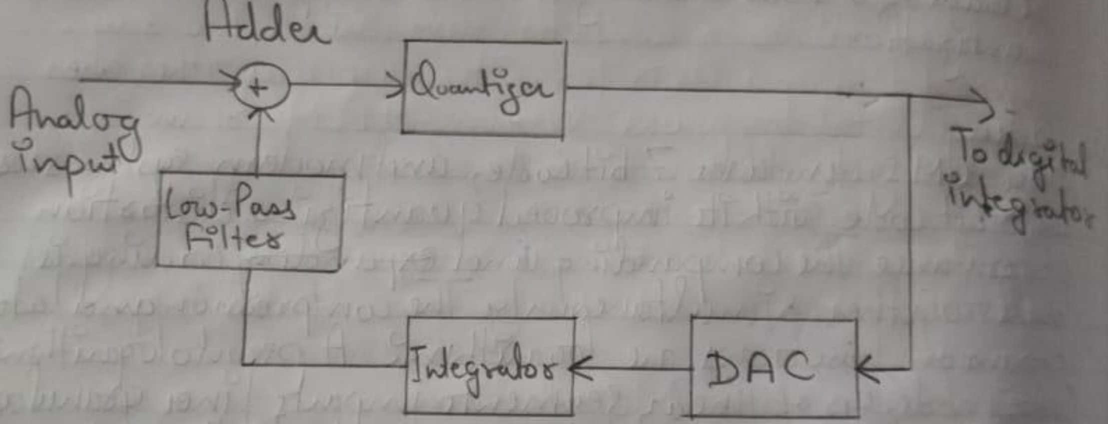
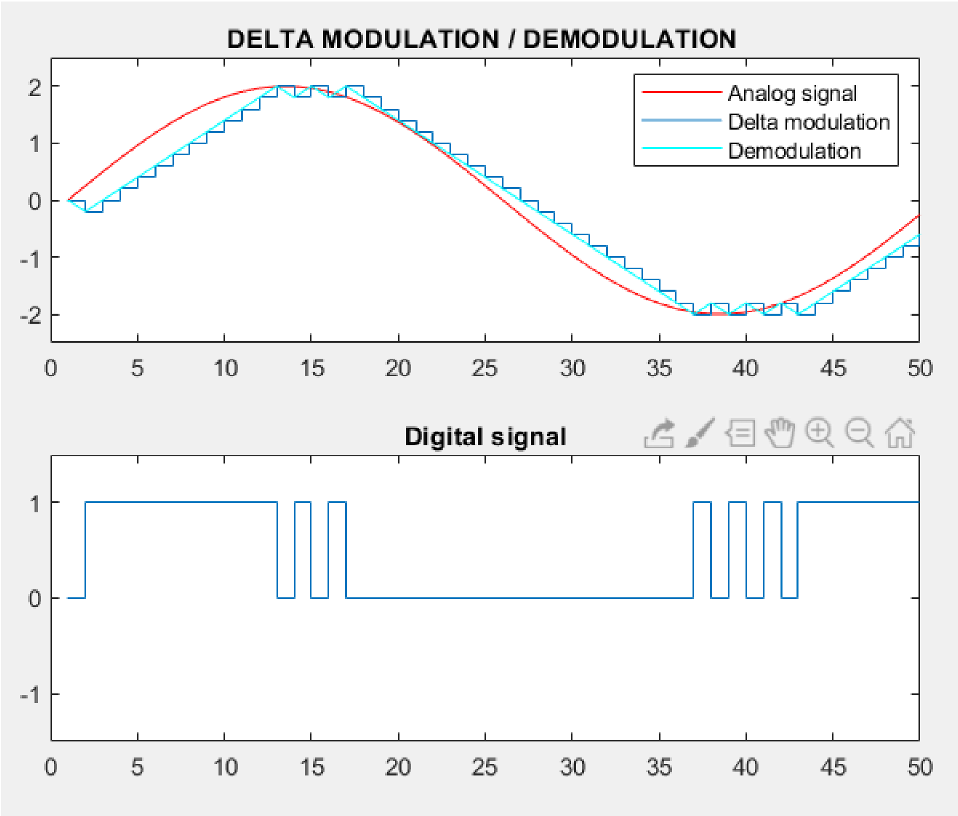


Fig: Block Diagram Delta Modulation

MATLAB PROGRAM TO IMPLEMENT DELTA MODULATION AND DEMODULATION

```
a=2;
t=0:2*pi/50:2*pi;
x=a*sin(t);
l=length(x);
subplot(2,1,1);
plot(x,'r');
delta=0.2;
hold on
xn=0;
for i=1:l;
if x(i)>xn(i)
d(i)=1;
xn(i+1)=xn(i)+delta;
else
d(i)=0; xn(i+1)=xn(i)-delta;
end
end
stairs(xn)
if d(i)==0
xn(i+1)=xn(i)-delta;
else
xn(i+1)=xn(i)+delta;
end
plot(xn,'c');
ylim([-2.5,2.5]);
xlim([0,50]);
legend('Analog signal','Delta modulation','Demodulation')
title('DELTA MODULATION / DEMODULATION ');
subplot(2,1,2);
stairs(d);
ylim([-1.5,1.5]);
xlim([0,50]);
title('Digital signal');
```


DELTA MODULATION / DEMODULATION
Analog signal
Delta modulation
Demodulation
2
1
0
-1
-2
0
5
10
15
20
25
30
35
40
45
50
Digital signal
1
0
-1
0
5
10
15
20
25
30
35
40
45
50

Practical 4

Expt. No. 4

Date

Page No.

AIM

To generate and demodulate amplitude shift keyed (ASK) signal using MATLAB

APPARATUS

System Running MATLAB

THEORY

Generation of ASK

Amplitude shift keying - ASK - is a modulation process, which imparts to a sinusoid two or more discrete amplitude levels. These are related to the number of levels adopted by the digital message. For a binary message sequence there are two levels, one of which is typically zero. The data rate is a sub multiple of the carrier frequency. Thus the modulated waveform consists of bursts of a sinusoid. One of the disadvantages of ASK, compared with FSK and PSK, for example, is that it has not got a constant envelope. This makes its processing (eg, power amplification) more difficult, since linearity becomes an important factor. However, it does make for ease of demodulation with an envelope detector.

Demodulation.

ASK signal has a well defined envelope. Thus it enable to demodulation by an envelope detector. Some sort of decision-making circuitry is necessary for detecting the message. The signal is recovered by using a correlator

Teacher's Signature :

Expt. No. ______ Date ______

Page No. ______

and decision making circuitry is used to recover the binary sequence.

ASK modulation

1. Generate Carrier Signal
2. Start FOR loop
3. Generate Binary Data, Message Signal (on-off form)
4. Generate ASK Modulated Signal
5. Plot message signal and ASK modulated signal
6. End FOR loop.
7. Plot the binary data and carrier

ASK demodulation.

1. Start FOR loop
2. Perform Correlation of ASK signal with carrier to get decision variable
3. Make decision to get demodulated Binary data. If $x>0$ choose 1 else choose 0
4. Plot the demodulated Binary data.

Teacher's Signature : ______

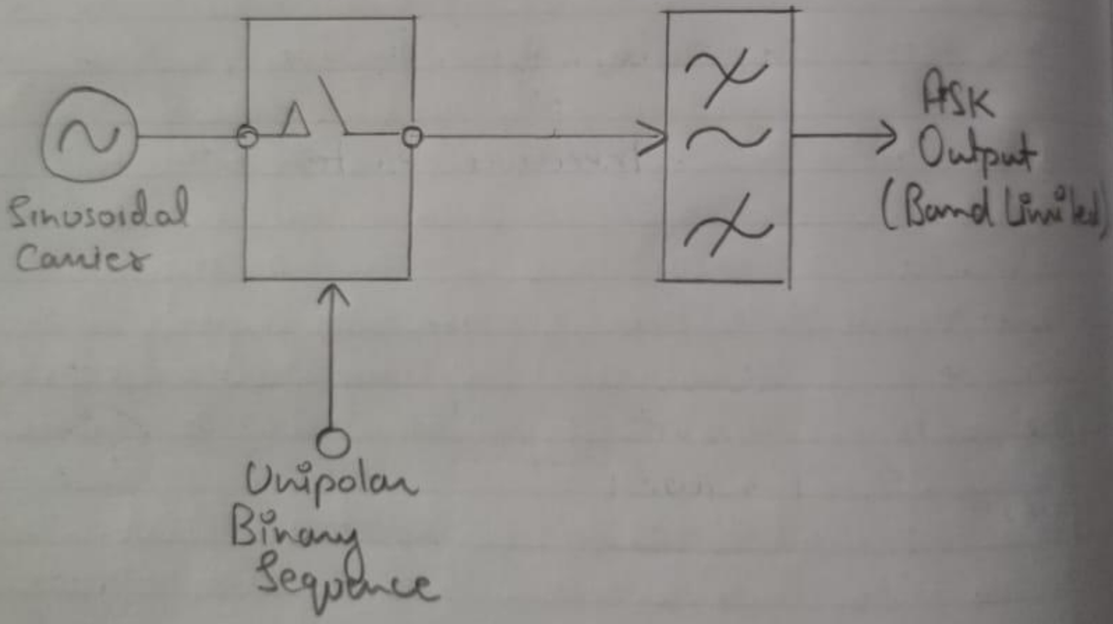
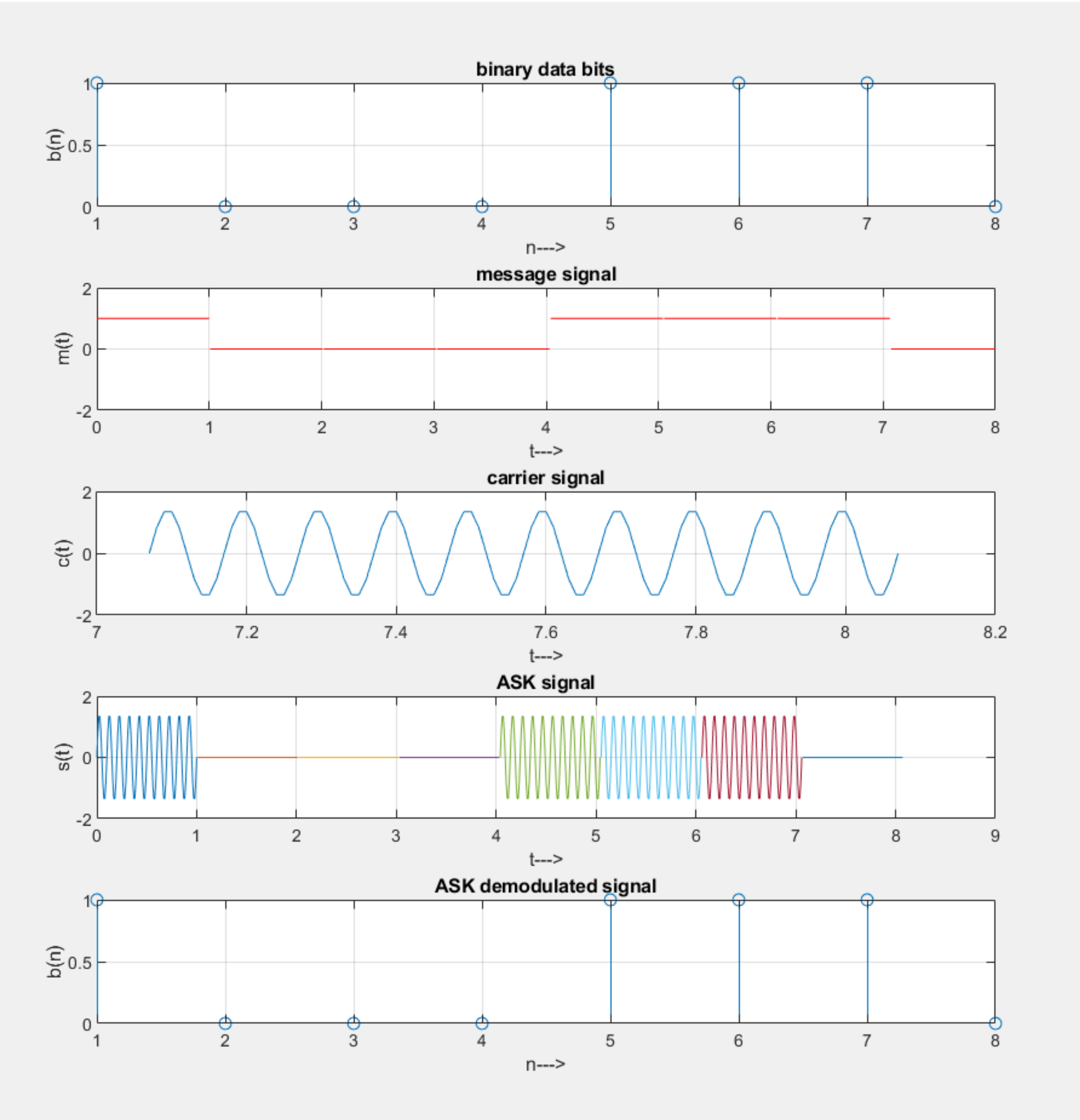


Fig:- ASK generation method

MATLAB PROGRAM TO IMPLEMENT ASK MODULATION AND DEMODULATION

```
%ASK Modulation

%GENERATE CARRIER SIGNAL
Tb=1; fc=10;
t=0:Tb/100:1;
c=sqrt(2/Tb)*sin(2*pi*fc*t);

%generate message signal
N=8;
m=rand(1,N);
t1=0;
t2=Tb;
for i=1:N
t=[t1:.01:t2];
if m(i)>0.5
m(i)=1;
m_s=ones(1,length(t));
else
m(i)=0;
m_s=zeros(1,length(t));
end
message(i,:)=m_s;

%product of carrier and message
ask_sig(i,:)=c.*m_s;
t1=t1+(Tb+.01);
t2=t2+(Tb+.01);

%plot the message and ASK signal
subplot(5,1,2);axis([0 N -2 2]);plot(t,message(i,:),'r');
title('message signal');xlabel('t--->');ylabel('m(t)');grid on
hold on
subplot(5,1,4);plot(t,ask_sig(i,:));
title('ASK signal');xlabel('t--->');ylabel('s(t)');grid on
hold on
end
hold off

%Plot the carrier signal and input binary data
subplot(5,1,3);plot(t,c);
title('carrier signal');xlabel('t--->');ylabel('c(t)');
grid on
subplot(5,1,1);stem(m);
title('binary data bits');xlabel('n--->');ylabel('b(n)');
grid on


% ASK Demodulation
t1=0;t2=Tb;
for i=1:N
t=[t1:Tb/100:t2]

%correlator
x=sum(c.*ask_sig(i,:));

%decision device
if x>0
```

```
demod(i)=1;
else
demod(i)=0;
end
t1=t1+(Tb+.01);
t2=t2+(Tb+.01);
end

%plot demodulated binary data bits
subplot(5,1,5);stem(demod);
title('ASK demodulated signal'); xlabel('n--->');ylabel('b(n)');
grid on;
```



Practical 5

Expt. No. 5

Date

Page No.

AIM

To generate and demodulate phase shift keyed (PSK) signal using MATLAB

APPARATUS

System Running MATLAB

THEORY

Generation of PSK signal

PSK is a digital modulation scheme that conveys data by changing, or modulating, the phase of a reference signal (the carrier wave). PSK uses a finite number of waves phases, each assigned a unique pattern of binary digits. Usually, each phase encodes an equal number of bits. Each pattern of bits forms the symbol that is represented by the particular phase. The demodulator, which is designed specifically for the symbol-set used by the modulator, determines the phase of the received signal and maps it back to the symbol it represents, thus recovering the original data.

In a coherent binary PSK system, the pair of signal $S_1(t)$ and $S_2(t)$ used to represent binary symbols 1 & 0 are defined by

$$S_1(t) = \sqrt{2E_b/T_b}\ \cos 2\pi f_c t$$

$$S_2(t) = \sqrt{2E_b/T_b}\ (2\pi f_c t + \pi)$$

$$= -\sqrt{2E_b/T_b}\ \cos 2\pi f_c t$$

Teacher's Signature :

Expt. No. ______ Date ______

Page No. ______

where $0 \le t < T_b$

E_b = Transmitted Signal Energy per bit

The Carrier Frequency $f_c = n/T_b$ for some fixed integer n

PSK Modulation
1. Generate Carrier Signal
2. Start FOR loop
3. Generate Binary data, message signal in polar form
4. Generate PSK modulated signal.
5. Plot message signal and PSK modulated signal.
6. End FOR loop.
7. Plot the binary data and Carrier.

PSK Demodulation.
1. Start FOR loop,
Perform Correlation of PSK signal with carrier to get decision variable
2. Make decision to get demodulated binary data. If $x > 0$, choose '1' else choose '0'
3. Plot the demodulated Binary data.

Teacher's Signature : ______

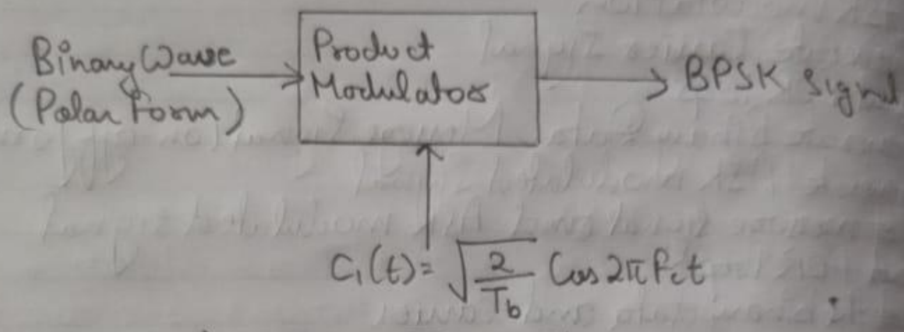


fig: BPSK Transmitter

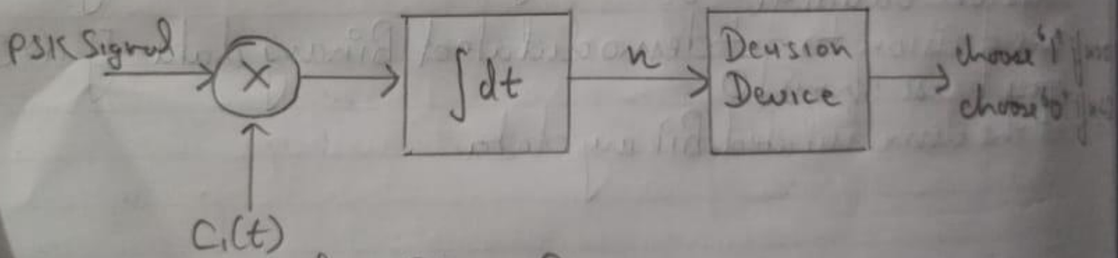
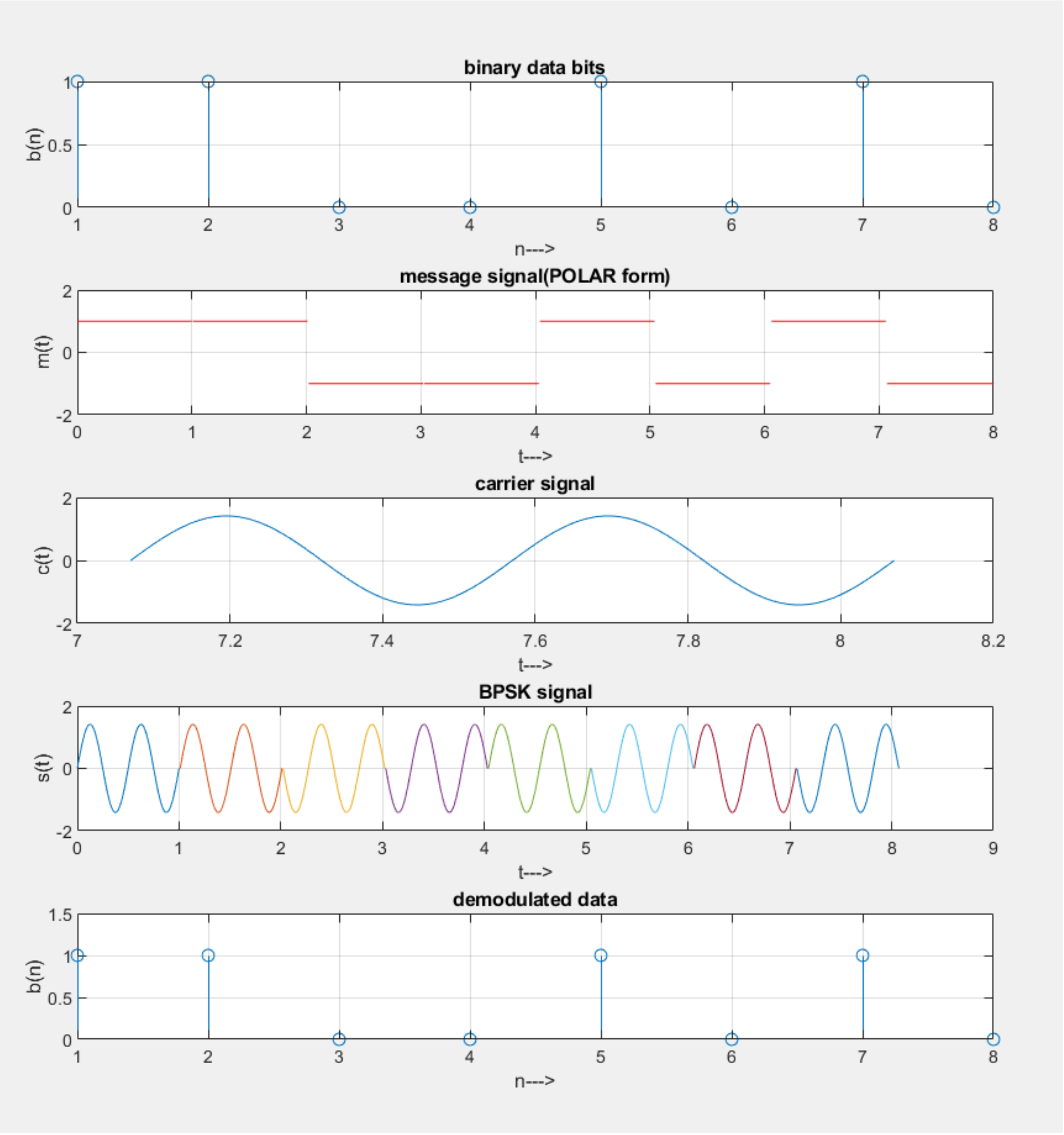


fig: BPSK Receiver

MATLAB PROGRAM TO IMPLEMENT PSK MODULATION AND DEMODULATION

```
% PSK modulation

%GENERATE CARRIER SIGNAL
Tb=1;
t=0:Tb/100:Tb;
fc=2;
c=sqrt(2/Tb)*sin(2*pi*fc*t);

%generate message signal
N=8;
m=rand(1,N);
t1=0;t2=Tb;
for i=1:N
t=[t1:.01:t2];
if m(i)>0.5
m(i)=1;
m_s=ones(1,length(t));
else
m(i)=0;
m_s=-1*ones(1,length(t));
end
message(i,:)=m_s;

%product of carrier and message signal
bpsk_sig(i,:)=c.*m_s;

%Plot the message and BPSK modulated signal
subplot(5,1,2);axis([0 N -2 2]);plot(t,message(i,:),'r');
title('message signal(POLAR form)');xlabel('t--->');ylabel('m(t)');
grid on; hold on;
subplot(5,1,4);plot(t,bpsk_sig(i,:));
title('BPSK signal');xlabel('t--->');ylabel('s(t)');
grid on; hold on;
t1=t1+1.01; t2=t2+1.01;
end
hold off

%plot the input binary data and carrier signal
subplot(5,1,1);stem(m);
title('binary data bits');xlabel('n--->');ylabel('b(n)');
grid on;
subplot(5,1,3);plot(t,c);
title('carrier signal');xlabel('t--->');ylabel('c(t)');
grid on;
10

% PSK Demodulation
t1=0;t2=Tb;
for i=1:N
t=[t1:.01:t2];

%correlator
x=sum(c.*bpsk_sig(i,:));

%decision device
if x>0
    demod(i)=1;
```

```
else
demod(i)=0;
end
t1=t1+1.01;
t2=t2+1.01;
end

%plot the demodulated data bits
subplot(5,1,5);
stem(demod);
ylim([0,1.5])
title('demodulated data');xlabel('n--->');ylabel('b(n)');
grid on
```



Practical 6

Date

Expt. No. 6 Page No.

AIM

To generate and demodulate frequency Shift Keyed (FSK) signal using MATLAB

APPARATUS

System running MATLAB software

THEORY

Generation of FSK

Frequency Shift Keying (FSK) is a frequency modulation scheme in which digital information is transmitted through discrete frequency changes of a carrier wave. The simplest FSK is binary FSK (BFSK). BFSK uses a pair of discrete frequencies to transmit Binary (0s and 1s) information. With this scheme, the "1" is called the mark frequency and "0" is called the space frequency.

In binary FSK system, symbols 1 & 0 are distinguished from each other by transmitting one of the two sinusoidal waves that differ in frequency by a fixed amount.

$$S_i(t) = \sqrt{\frac{2E_b}{T_b}} \cos 2\pi f_i t$$

$$0 \le t \le T_b$$

E_b = Transmitted Energy / bit

Transmitted frequency $= f_i = (n_c + i)/T_b$

n = constant (integer)

T_b = bit interval.

Teacher's Signature :

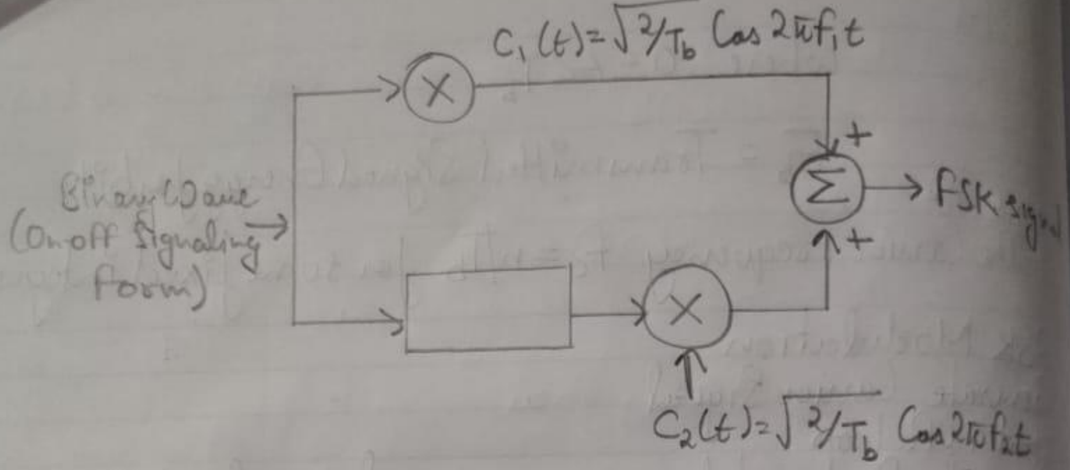


fig → BFSK Transmitter

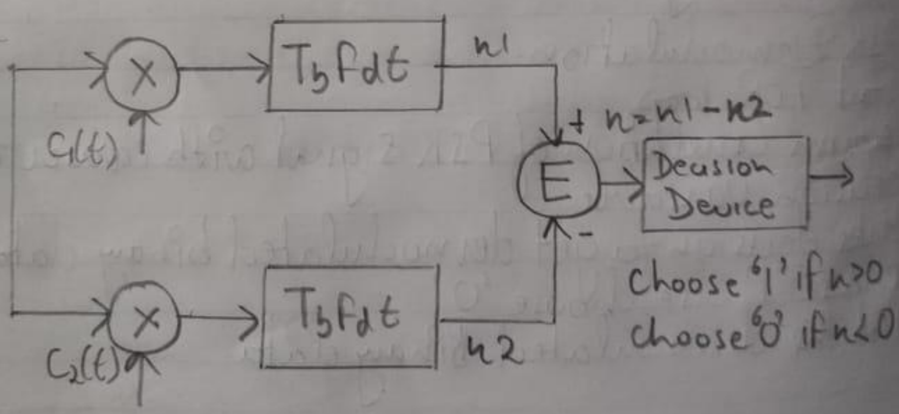


fig : BFSK Receiver

Date ______

Expt. No. ______ Page No. ______

FSK modulation.

1. Generate two Carrier Signals
2. Start FOR loop
3. Generate Binary data, message signal and inverted message signal
4. Multiply carrier1 with message signal and carrier 2 with inverted message signal
5. Perform addition to get FSK modulated signal.
6. Plot message signal and FSK modulated signal
7. End FOR loop.
8. Plot the Binary data and Carriers.

FSK demodulation

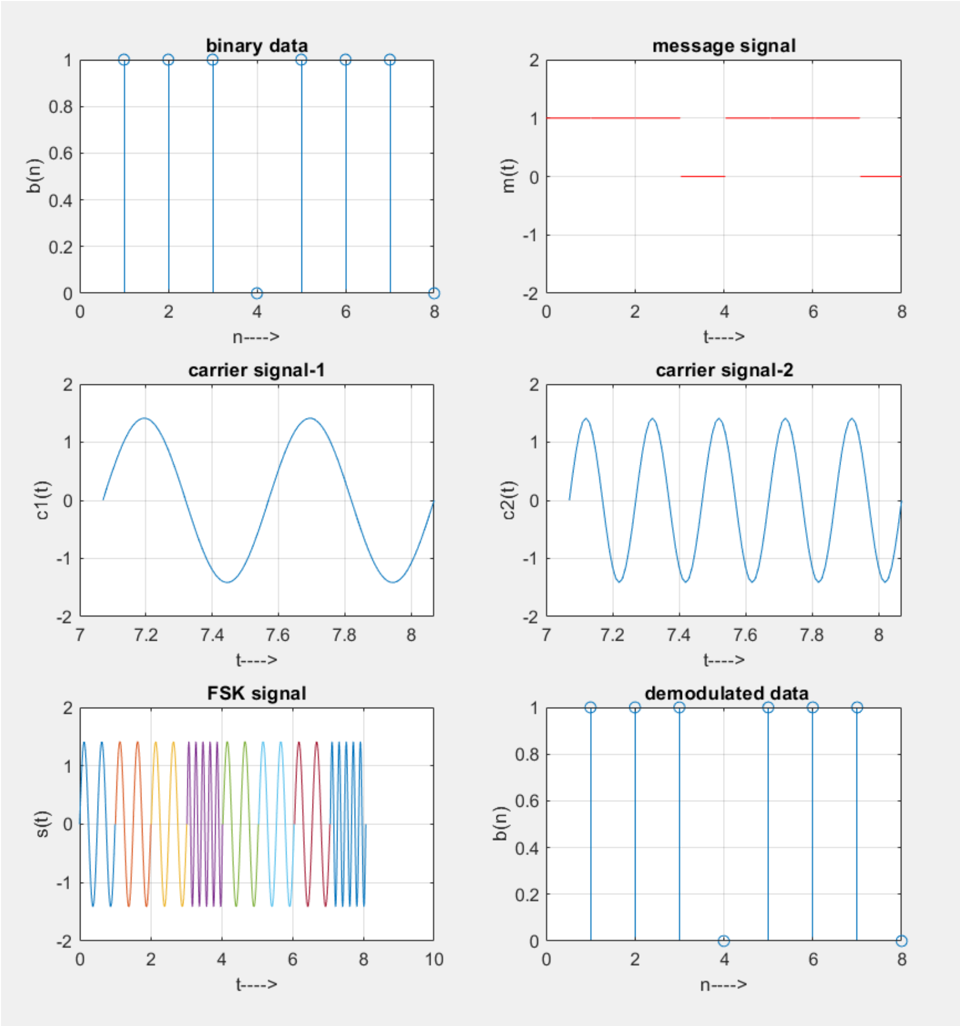
1. Start FOR loop
2. Perform Correlation of FSK modulated signal with carrier 1 and carrier 2 to get two decisions variables x1 and x2
3. Make decision x = x1 - x2 to get demodulated binary data. If x>0 choose '1' else choose '0'
4. Plot the demodulated Binary data.

Teacher's Signature : ______

MATLAB PROGRAM TO IMPLEMENT FSK MODULATION AND DEMODULATION

```
% FSK Modulation

%GENERATE CARRIER SIGNAL
Tb=1; fc1=2;fc2=5;
t=0:(Tb/100):Tb;
c1=sqrt(2/Tb)*sin(2*pi*fc1*t);
c2=sqrt(2/Tb)*sin(2*pi*fc2*t);

%generate message signal
N=8;
m=rand(1,N);
t1=0;t2=Tb;
for i=1:N
t=[t1:(Tb/100):t2];
if m(i)>0.5
m(i)=1;
m_s=ones(1,length(t));
invm_s=zeros(1,length(t));
else
m(i)=0;
m_s=zeros(1,length(t));
invm_s=ones(1,length(t));
end
message(i,:)=m_s;

%Multiplier
fsk_sig1(i,:)=c1.*m_s;
fsk_sig2(i,:)=c2.*invm_s;
fsk=fsk_sig1+fsk_sig2;

%plotting the message signal and the modulated signal
subplot(3,2,2);axis([0 N -2 2]);
plot(t,message(i,:),'r');
title('message signal');
xlabel('t---->');
ylabel('m(t)');
grid on;
hold on;
subplot(3,2,5);
plot(t,fsk(i,:));
title('FSK signal');
xlabel('t---->');
ylabel('s(t)');
grid on;hold on;
t1=t1+(Tb+.01);
t2=t2+(Tb+.01);
end
hold off

%Plotting binary data bits and carrier signal
subplot(3,2,1);stem(m);
title('binary data');
xlabel('n---->');
ylabel('b(n)');
grid on;
subplot(3,2,3);
plot(t,c1);
title('carrier signal-1');
```

```
xlabel('t---->');
ylabel('c1(t)');
grid on;
subplot(3,2,4);
plot(t,c2);
title('carrier signal-2');
xlabel('t---->');
ylabel('c2(t)');
grid on;


% FSK Demodulation
t1=0;
t2=Tb;
for i=1:N
t=[t1:(Tb/100):t2];

%correlator
x1=sum(c1.*fsk_sig1(i,:));
x2=sum(c2.*fsk_sig2(i,:));
x=x1-x2;

%decision device
if x>0
demod(i)=1;
else
demod(i)=0;
end
t1=t1+(Tb+.01);
t2=t2+(Tb+.01);
end

%Plotting the demodulated data bits
subplot(3,2,6);
stem(demod);
title(' demodulated data');
xlabel('n---->');
ylabel('b(n)');
 grid on;
```


binary data
b(n)
n---->
message signal
m(t)
t---->
carrier signal-1
c1(t)
t---->
carrier signal-2
c2(t)
t---->
FSK signal
s(t)
t---->
demodulated data
b(n)
n---->

Practical

Date

Expt. No. 7

Page No.

AIM

To generate and demodulate quadrature phase shifted (QPSK) signal using MATLAB

THEORY

Generation of Quadrature PSK signal

QPSK is also known as quaternary PSK, quadriphase PSK, 4-PSK, or 4-QAM. It is a phase modulation technique that transmits two bits in four modulation states.

Phase of the carrier takes on one of the four equally spaced value such as $\frac{\pi}{4}$, $\frac{3\pi}{4}$, $\frac{5\pi}{4}$ and $\frac{7\pi}{4}$

$$S_i(t) = \sqrt{\frac{2E}{T}} \cos\left(2\pi f_c t + (2i-1)\pi/4\right), \quad 0 \le t \le T$$

$$= 0$$

where $i = 1, 2, 3, 4,$

$E =$ Tx signal energy per symbol

$T =$ symbol duration.

Each of the possible value of phase corresponds to a pair of bits called dibits. Thus, the gray encoded sets of digits 10, 00, 01, 11

$$S_i(t) = \sqrt{\frac{2E}{T}} \cos\left[(2i-1)\pi/4\right] \cos(2\pi f_c t) -$$

$$\sqrt{\frac{2E}{T}} \sin\left[(2i-1)\frac{\pi}{4}\right] \sin(2\pi f_c t)$$

Teacher's Signature :

Expt. No. ______ Date ______

Page No. ______

Original binary output.

QPSK modulation

1. Generate quadrature carriers
2. Start FOR loop
3. Generate binary data, message signal (bipolar form)
4. Multiply carrier1 with odd bits of message signal and carrier 2 with even bits of message signal
5. Perform addition of odd and even modulated signals to get the QPSK modulated signal.
6. Plot QPSK modulated Signal.
7. End FOR loop.
8. Plot the binary data and carriers.

QPSK demodulation.

1. Start FOR loop.
2. Perform correlation of QPSK modulated signal with quadrature carriers to get two decision variable x1 and x2.
3. Make decision on x1 & x2 and multiplexed to get demodulated Binary data.
   - If x1>0 and x2>0, choose '11'
   - If x1>0 and x2<0, choose '10'
   - If x1<0 and x2>0, choose '01'
   - If x1<0 and x2<0, choose '00'
4. End FOR loop.
5. Plot demodulated data.

Teacher's Signature ______

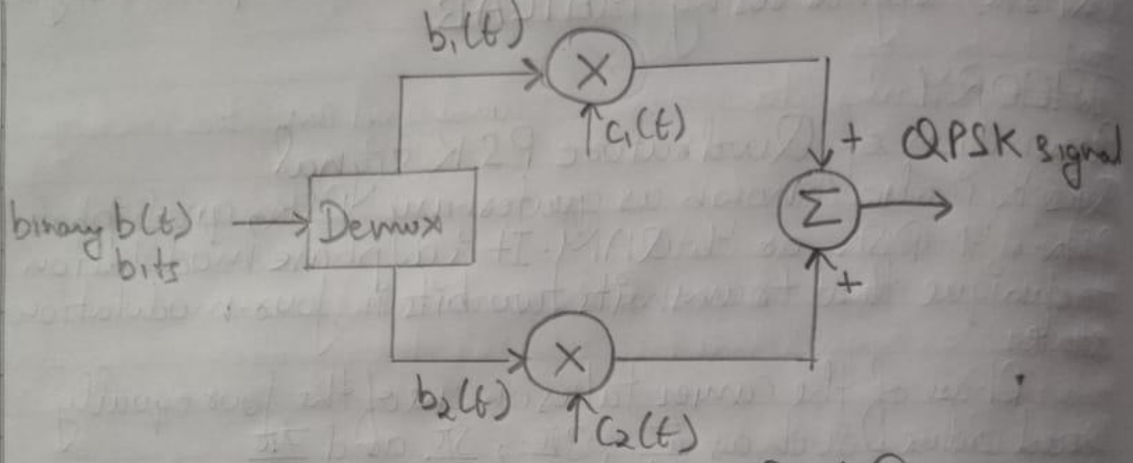
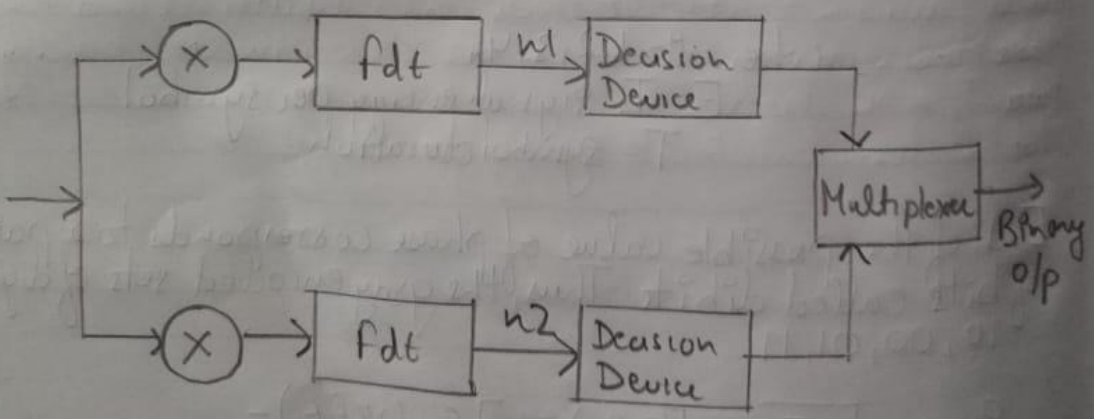


fig: QPSK Transmitter Block Diagram



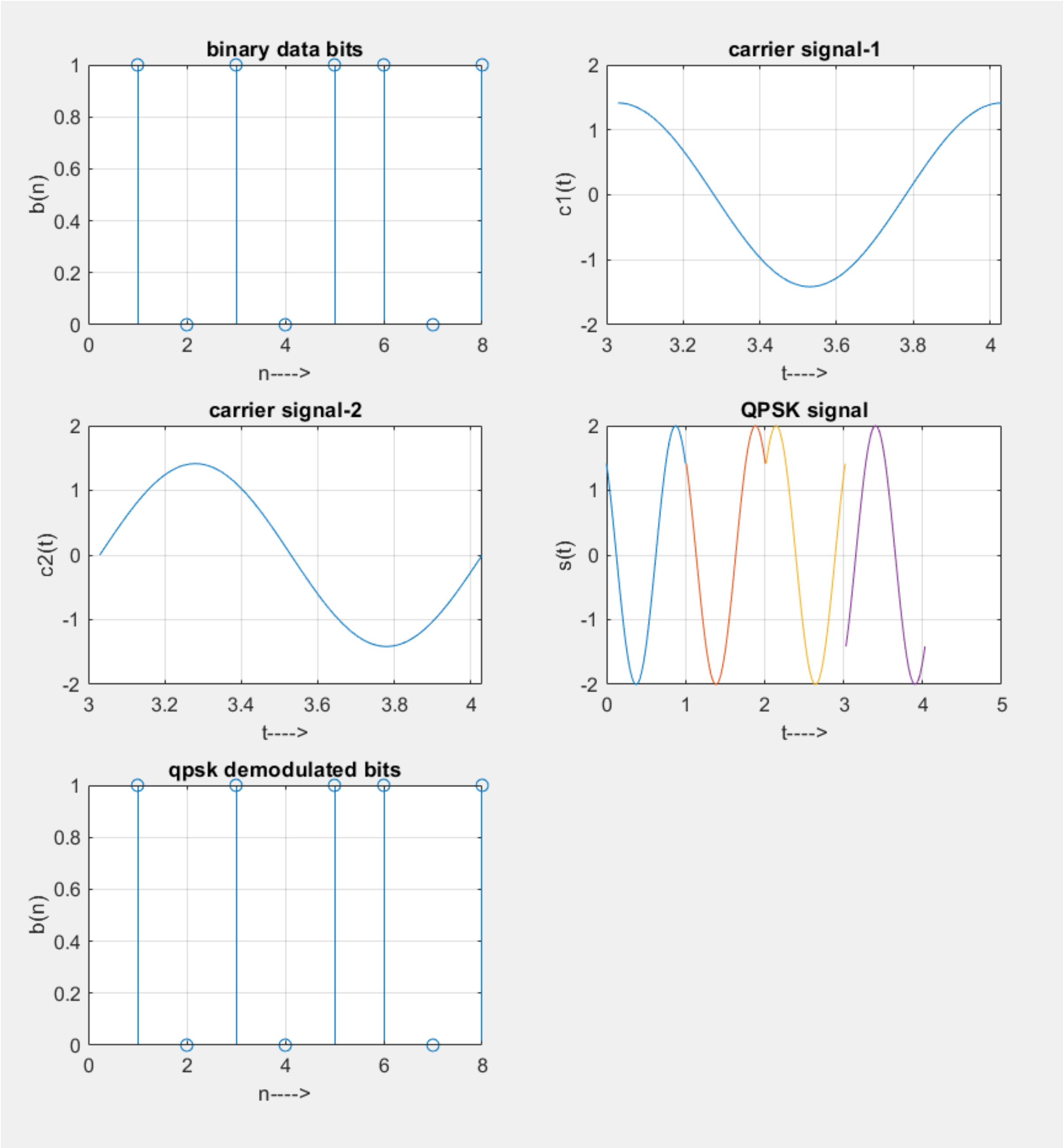
Input Debits	Phase of QPSK Signal	Co-ordinates of msg Signals	
		S1	S2
10	$\pi/4$	$\sqrt{E/2}$	$-\sqrt{E/2}$
00	$3\pi/4$	$-\sqrt{E/2}$	$-\sqrt{E/2}$
01	$5\pi/4$	$-\sqrt{E/2}$	$+\sqrt{E/2}$
11	$7\pi/4$	$+\sqrt{E/2}$	$+\sqrt{E/2}$

Four message Points of QPSK

MATLAB PROGRAM TO IMPLEMENT QPSK MODULATION AND DEMODULATION

```
% QPSK Modulation

%GENERATE QUADRATURE CARRIER SIGNAL
Tb=1;t=0:(Tb/100):Tb;fc=1;
c1=sqrt(2/Tb)*cos(2*pi*fc*t);
c2=sqrt(2/Tb)*sin(2*pi*fc*t);

%generate message signal
N=8;m=rand(1,N);
t1=0;t2=Tb;
for i=1:2:(N-1)
t=[t1:(Tb/100):t2];
if m(i)>0.5
m(i)=1;
m_s=ones(1,length(t));
else
m(i)=0;
m_s=-1*ones(1,length(t));
end

%odd bits modulated signal
odd_sig(i,:)=c1.*m_s;
if m(i+1)>0.5
%21
m(i+1)=1;
m_s=ones(1,length(t));
else
m(i+1)=0;
m_s=-1*ones(1,length(t));
end

%even bits modulated signal
even_sig(i,:)=c2.*m_s;

%qpsk signal
qpsk=odd_sig+even_sig;

%Plot the QPSK modulated signal
subplot(3,2,4);
plot(t,qpsk(i,:));
title('QPSK signal');
xlabel('t---->');
ylabel('s(t)');
grid on;
hold on;
t1=t1+(Tb+.01);
t2=t2+(Tb+.01);
end
hold off

%Plot the binary data bits and carrier signal
subplot(3,2,1);
stem(m);
title('binary data bits');
xlabel('n---->');
ylabel('b(n)');
grid on;
```

```
subplot(3,2,2);
plot(t,c1);
title('carrier signal-1');
xlabel('t---->');
ylabel('c1(t)');
grid on;
subplot(3,2,3);
plot(t,c2);
title('carrier signal-2');
xlabel('t---->');
ylabel('c2(t)');
grid on;

% QPSK Demodulation
t1=0;t2=Tb;
for i=1:N-1
t=[t1:(Tb/100):t2]

%correlator
x1=sum(c1.*qpsk(i,:));
x2=sum(c2.*qpsk(i,:));

%decision device
if (x1>0&&x2>0)
demod(i)=1;
demod(i+1)=1;
elseif (x1>0&&x2<0)
demod(i)=1;
demod(i+1)=0;
elseif (x1<0&&x2<0)
demod(i)=0;
demod(i+1)=0;
elseif (x1<0&&x2>0)
demod(i)=0;
demod(i+1)=1;
end
t1=t1+(Tb+.01);
t2=t2+(Tb+.01);
end
subplot(3,2,5);
stem(demod);
title('qpsk demodulated bits');
xlabel('n---->');
ylabel('b(n)');
grid on;
```


binary data bits
b(n)
n---->
carrier signal-1
c1(t)
t---->
carrier signal-2
c2(t)
t---->
QPSK signal
s(t)
t---->
qpsk demodulated bits
b(n)
n---->