COECC12-DATA COMMUNICATION

LAB FILE

AMOGH GARG

2020UCO1688

NETAJI SUBHAS UNIVERSITY OF TECHNOLOGY

EXPERIMENT-1

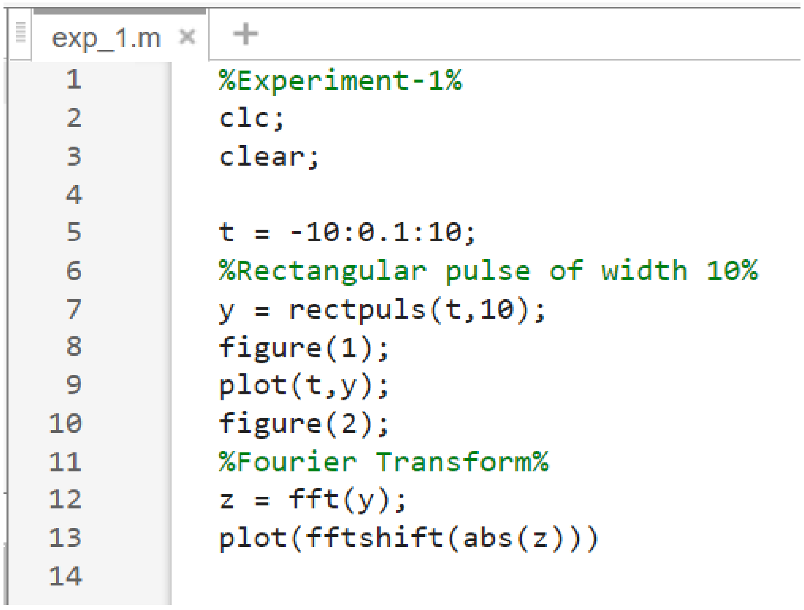
AIM: To plot the spectrum of a pulse of width 10.

THEORY: The Fourier transform is a mathematical function that decomposes a waveform, which is a function of time, into the frequencies that make it up. The result produced by the Fourier transform is a complex valued function of frequency. The absolute value of the Fourier transform represents the frequency value present in the original function and its complex argument represents the phase offset of the basic sinusoidal in that frequency.

$$f(x) = \int_{-\infty}^{\infty} F(k)\, e^{2\pi i k x}\, dk$$

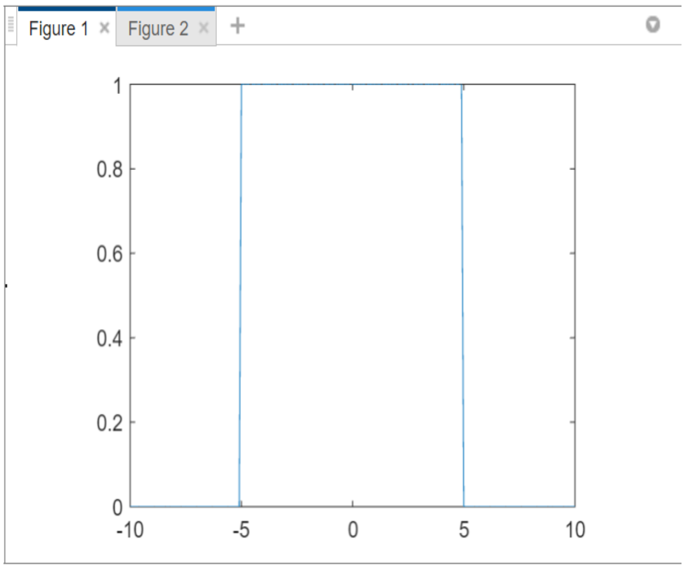
$$F(k) = \int_{-\infty}^{\infty} f(x)\, e^{-2\pi i k x}\, dx$$

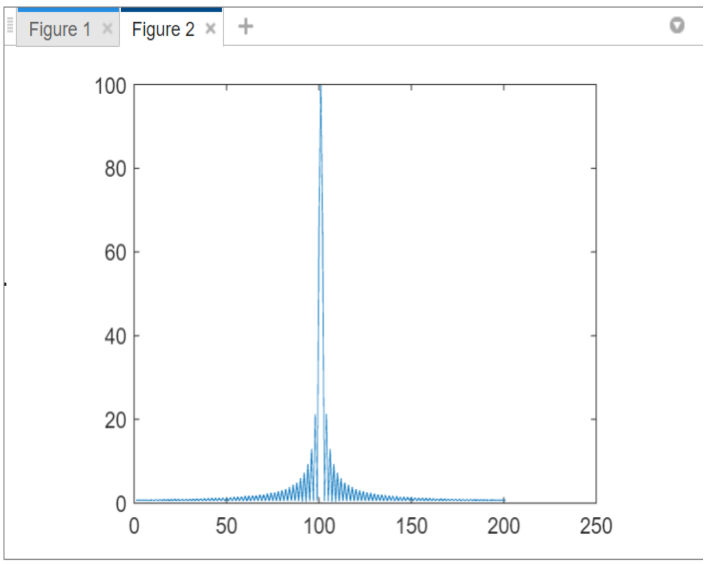
CODE:



```
%Experiment-1%
clc;
clear;

t = -10:0.1:10;
%Rectangular pulse of width 10%
y = rectpuls(t,10);
figure(1);
plot(t,y);
figure(2);
%Fourier Transform%
z = fft(y);
plot(fftshift(abs(z)))
```

OUTPUT:





EXPERIMENT-2

AIM: To verify following properties of Fourier Transform:

i. Time Shifting, ii. Frequency shifting, iii. Convolutional

THEORY:

1)

$$x(t) \leftrightarrow X(\omega)$$

$$x(t-t_0) \leftrightarrow X(\omega)e^{-j\omega t_0}$$

Proof : $F[x(t-t_0)] = \int_{-\infty}^{\infty} x(t-t_0)e^{-j\omega t}dt$

$$= \int_{-\infty}^{\infty} x(u)e^{-j\omega(u+t_0)}du = e^{-j\omega t_0}X(\omega)$$

2)

$$F[e^{j\omega_0 t}x(t)] = \int_{-\infty}^{\infty} e^{j\omega_0 t}\, x(t)e^{-j\omega t}dt$$

$$= \int_{-\infty}^{\infty} x(t)e^{-j(\omega-\omega_0)t}dt = X(\omega-\omega_0)$$

$$x(t)\,e^{j\omega_0 t} \leftrightarrow X(\omega-\omega_0)$$

3)

$$Y(\omega) = \int_{-\infty,d\tau}^{\infty} x(\tau) \int_{-\infty,dt}^{\infty} h(t-\tau)e^{-j\omega t}dt\,d\tau \quad \text{let } u = t-\tau$$

$$= \int_{-\infty,d\tau}^{\infty} x(\tau) \int_{-\infty,du}^{\infty} h(u)e^{-j\omega(u+\tau)}du\,d\tau$$

$$= \int_{-\infty,d\tau}^{\infty} x(\tau)e^{-j\omega\tau}d\tau \int_{-\infty,du}^{\infty} h(u)e^{-j\omega u}du$$

$$= X(\omega)H(\omega)$$

CODE:

TIME SHIFTING:

```
n0 = 2;
y = [zeros(1, n0), x1(1 : N - n0)];
Y = fft(y);
Y1 = X1 .* exp(-j * w * n0);
figure(3);
subplot(311);
plot(-10 : N, [zeros(1, 11), x1],
'LineWidth', 1);
hold on;
plot(-10 : N, [zeros(1, 11), y], 'LineWidth',
1);
xlim([-10 18]);
ylim([-0.2 1.2]);
xlabel('n');
ylabel('Magnitude');
title('x1(n) & x1(n - n0)');
subplot(312);
plot(w, abs(Y), '--', 'LineWidth', 1);
xlim([0 2 * pi]);
xlabel('w');
ylabel('Magnitude');
title('DFT [x1(n - n0)]');
subplot(313);
stem(w, abs(Y1), 'filled');
xlim([0 2 * pi]);
xlabel('w');
ylabel('Magnitude');
title('DFT [x1 * exp(-j*w*n0)]');
```

FREQUENCY SHIFTING:

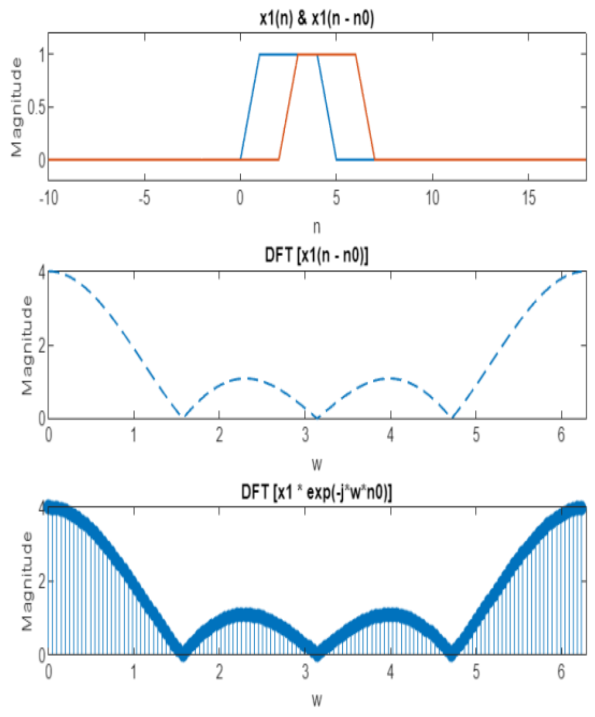
```
f0 = 8;
Y = [zeros(1, f0), X1(1 : N - f0)];
y1 = x1 .* exp(j * 2 * pi * f0 * n);
Y1 = fft(y1);
figure(4);
subplot(211);
plot(w, abs(X1), 'LineWidth', 1);
hold on;
plot(w, abs(Y), 'LineWidth', 1);
xlim([0 2 * pi]);
xlabel('w');
ylabel('Magnitude');
title('X1 (DFT [x1]) & X1(f - f0)');
subplot(212);
stem(w, abs(Y1), 'filled');
xlim([0 2 * pi]);
xlabel('w');
ylabel('Magnitude');
title('DFT [x1 * exp(j*2*pi*f0*n)]');
```

CONVOLUTION:

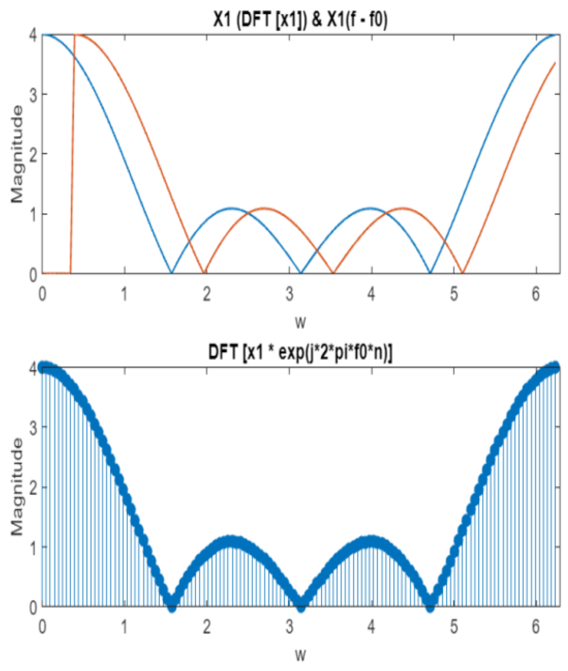
```
N = 128;
x1 = ones(1, 4);
x2 = ones(1, 6);
n1 = length(x1);
n2 = length(x2);
x1 = [x1, zeros(1, N - n1)];
x2 = [x2, zeros(1, N - n2)];
n = 0 : N - 1;
k = 0 : N - 1;
w = 2 * pi * k / N;
X1 = fft(x1);
X2 = fft(x2);
figure(1);
subplot(211);
stem(w, abs(X1), 'filled');
xlim([0 2 * pi]);
xlabel('w');
ylabel('Magnitude');
title('128 - point DFT of x1');
subplot(212);
stem(w, abs(X2), 'filled');
xlim([0 2 * pi]);
xlabel('w');
ylabel('Magnitude');
title('128 - point DFT of x2');
X3 = 1 / N * cconv(X1, X2, N);
x4 = x1 .* x2;
X4 = fft(x4);
figure(2);
subplot(211);
```

OUTPUT:

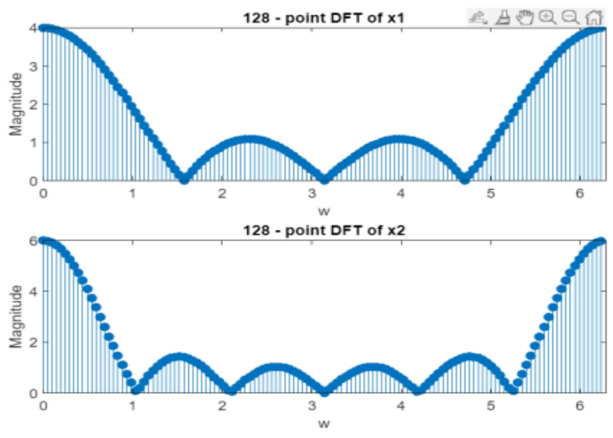
TIME-SHIFTING:

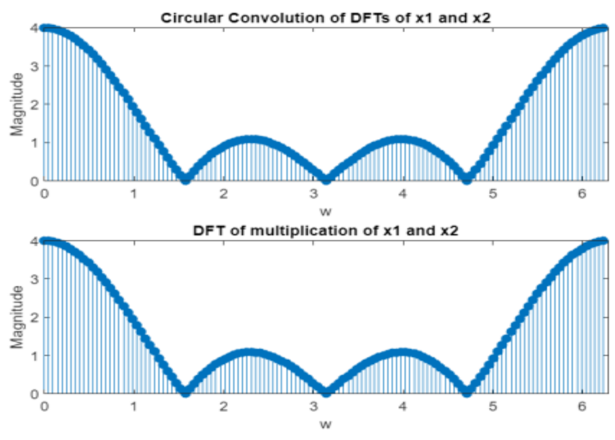


FREQUENCY-SHIFTING:



CONVOLUTION:





EXPERIMENT-3

AIM: Study of uniform, exponential and Gaussian distributed random variables. Draw their pdf and CDF.

THEORY:

The pdf of the uniform distribution is

$$f(x|a,b) = \begin{cases} \left(\frac{1}{b-a}\right) & ; \quad a \le x \le b \\ 0 & ; \quad otherwise \end{cases} .$$

The pdf is constant between a and b.

The cumulative distribution function (cdf) of the uniform distribution is

$$F(x|a,b) = \begin{cases} 0 & ; \quad x < a \\ \frac{x-a}{b-a} & ; \quad a \le x < b \\ 1 & ; \quad x \ge b \end{cases} .$$

The pdf of the exponential distribution is

$$y = f(x|\mu) = \frac{1}{\mu} e^{\frac{-x}{\mu}} .$$

The cumulative distribution function (cdf) of the exponential distribution is

$$p = F(x|u) = \int_0^x \frac{1}{\mu} e^{\frac{-t}{\mu}} dt = 1 - e^{\frac{-x}{\mu}} .$$

The normal probability density function (pdf) is

$$y = f(x|\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x-\mu)^2}{2\sigma^2}} , \quad \text{for } x \in \mathbb{R}.$$

The *likelihood function* is the pdf viewed as a function of the parameters. The maximum likelihood estimates (MLEs) are the parameter estimates that maximize the likelihood function for fixed values of x.

The normal cumulative distribution function (cdf) is

$$p = F(x|\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{x} e^{\frac{-(t-\mu)^2}{2\sigma^2}} dt, \quad \text{for } x \in \mathbb{R}.$$

p is the probability that a single observation from a normal distribution with parameters μ and σ falls in the interval $(-\infty, x]$.

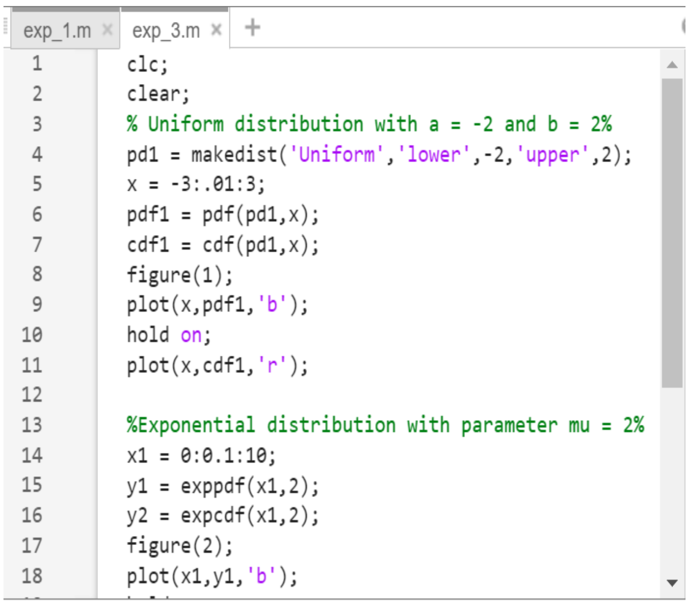
The standard normal cumulative distribution function $\Phi(x)$ is functionally related to the error function `erf`.

$$\Phi(x) = \frac{1}{2}\left(1 - \operatorname{erf}\left(-\frac{x}{\sqrt{2}}\right)\right)$$

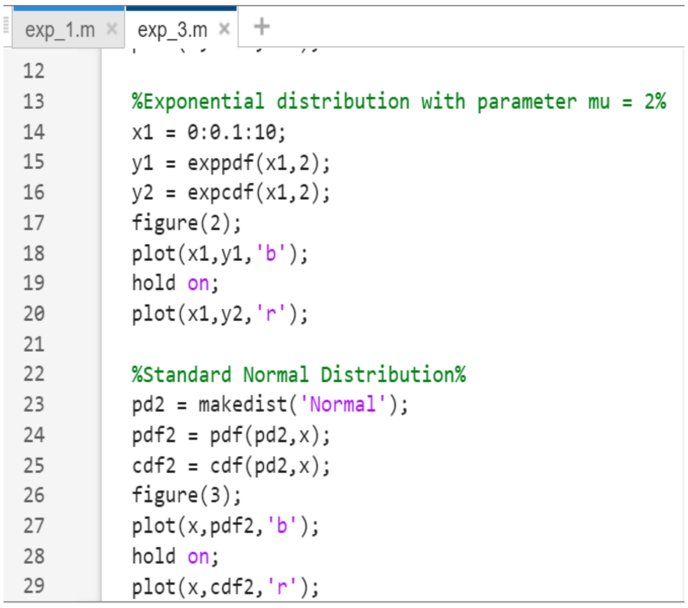
where

$$\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt = 2\Phi\left(\sqrt{2}\,x\right) - 1.$$

CODE:

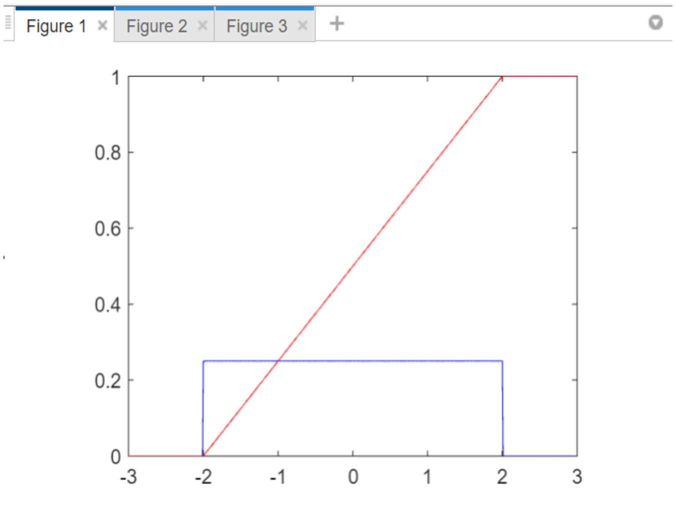


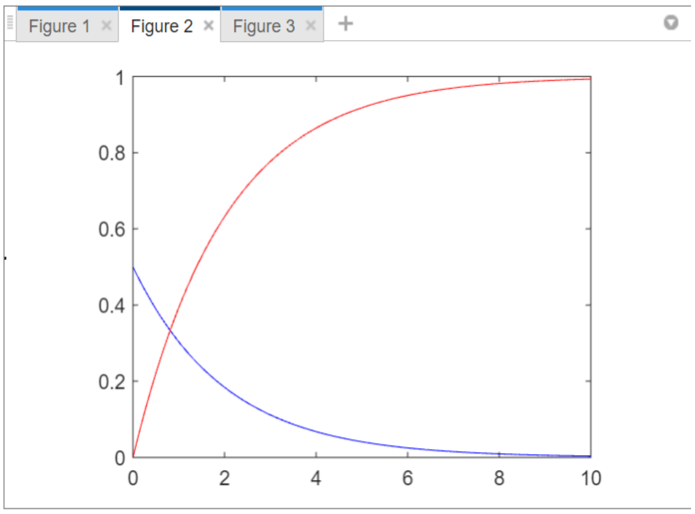
```
clc;
clear;
% Uniform distribution with a = -2 and b = 2%
pd1 = makedist('Uniform','lower',-2,'upper',2);
x = -3:.01:3;
pdf1 = pdf(pd1,x);
cdf1 = cdf(pd1,x);
figure(1);
plot(x,pdf1,'b');
hold on;
plot(x,cdf1,'r');

%Exponential distribution with parameter mu = 2%
x1 = 0:0.1:10;
y1 = exppdf(x1,2);
y2 = expcdf(x1,2);
figure(2);
plot(x1,y1,'b');
```

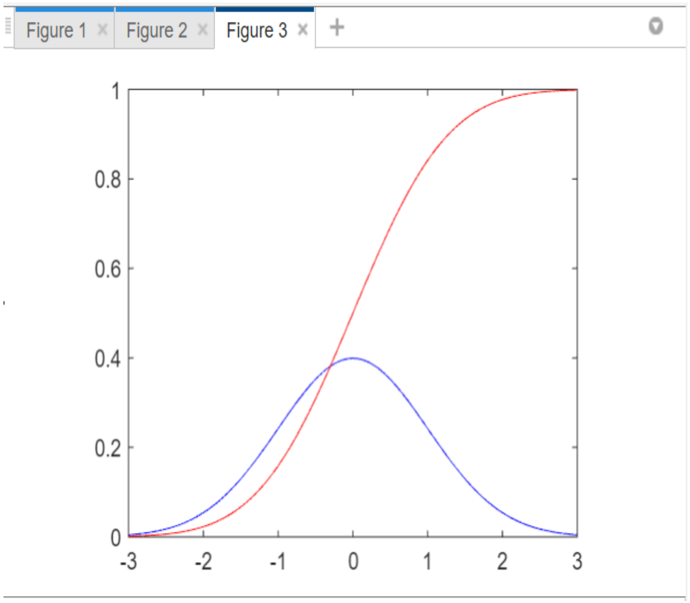


```

%Exponential distribution with parameter mu = 2%
x1 = 0:0.1:10;
y1 = exppdf(x1,2);
y2 = expcdf(x1,2);
figure(2);
plot(x1,y1,'b');
hold on;
plot(x1,y2,'r');

%Standard Normal Distribution%
pd2 = makedist('Normal');
pdf2 = pdf(pd2,x);
cdf2 = cdf(pd2,x);
figure(3);
plot(x,pdf2,'b');
hold on;
plot(x,cdf2,'r');
```

OUTPUT:





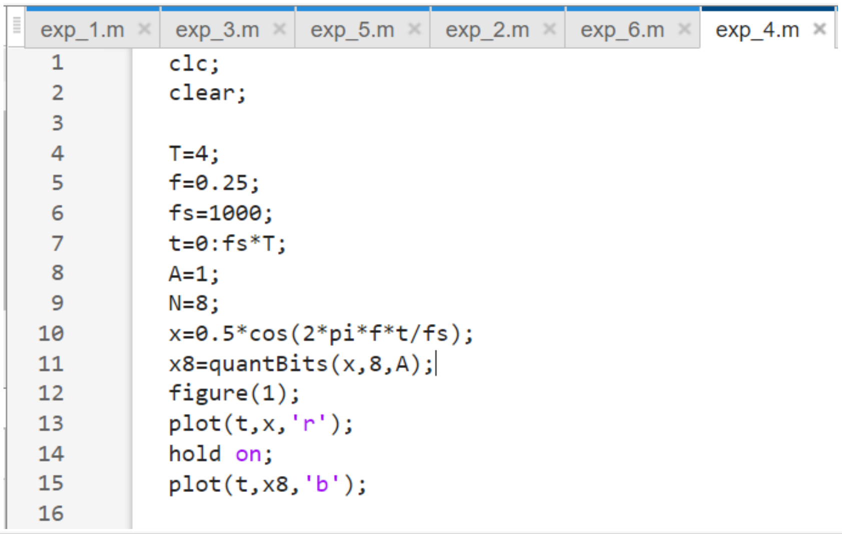


EXPERIMENT-4

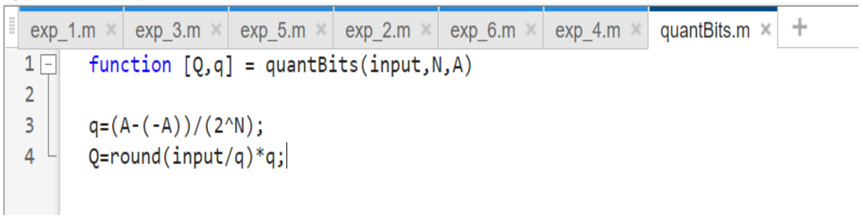
AIM: Study of linear and non-linear quantization. Compute the quantization error of a Gaussian signal.

THEORY: 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 to a near stabilized value. Such a process is called as **Quantization**. The analog-to-digital converters perform this type of function to create a series of digital values out of the given analog signal. The following figure represents an analog signal. This signal to get converted into digital, has to undergo sampling and quantizing. Quantizing of an analog signal is done by discretizing the signal with a number of quantization levels. **Quantization** is representing the sampled values of the amplitude by a finite set of levels, which means converting a continuous-amplitude sample into a discrete-time signal.
The difference between an input value and its quantized value is called a **Quantization Error**.

CODE:

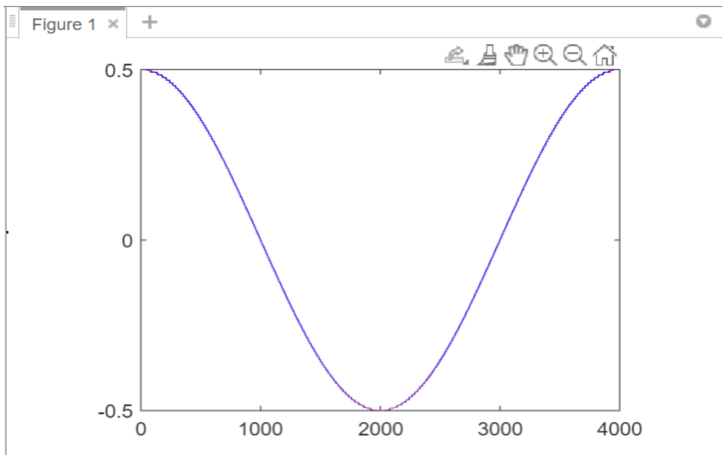

exp_1.m × exp_3.m × exp_5.m × exp_2.m × exp_6.m × exp_4.m ×

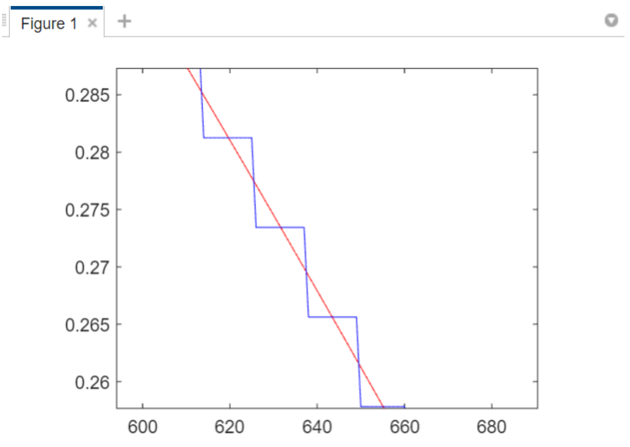
```
clc;
clear;

T=4;
f=0.25;
fs=1000;
t=0:fs*T;
A=1;
N=8;
x=0.5*cos(2*pi*f*t/fs);
x8=quantBits(x,8,A);
figure(1);
plot(t,x,'r');
hold on;
plot(t,x8,'b');

```


exp_1.m × exp_3.m × exp_5.m × exp_2.m × exp_6.m × exp_4.m × quantBits.m ×

```
function [Q,q] = quantBits(input,N,A)

q=(A-(-A))/(2^N);
Q=round(input/q)*q;
```

OUTPUT:

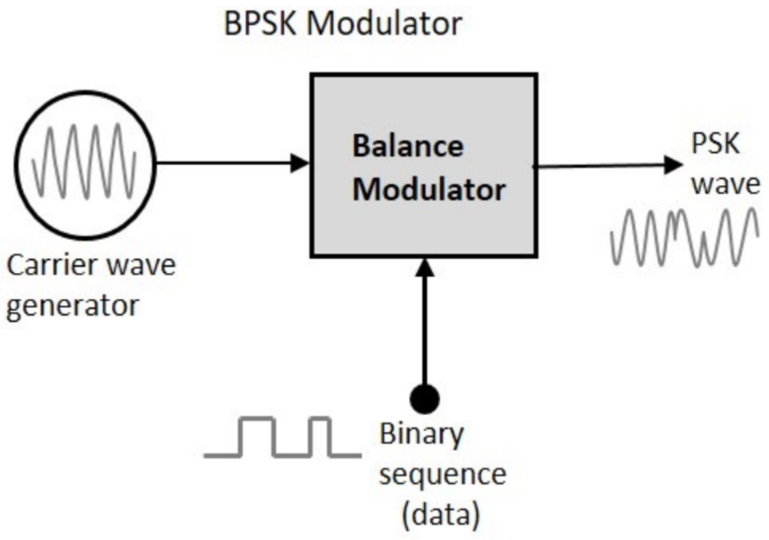


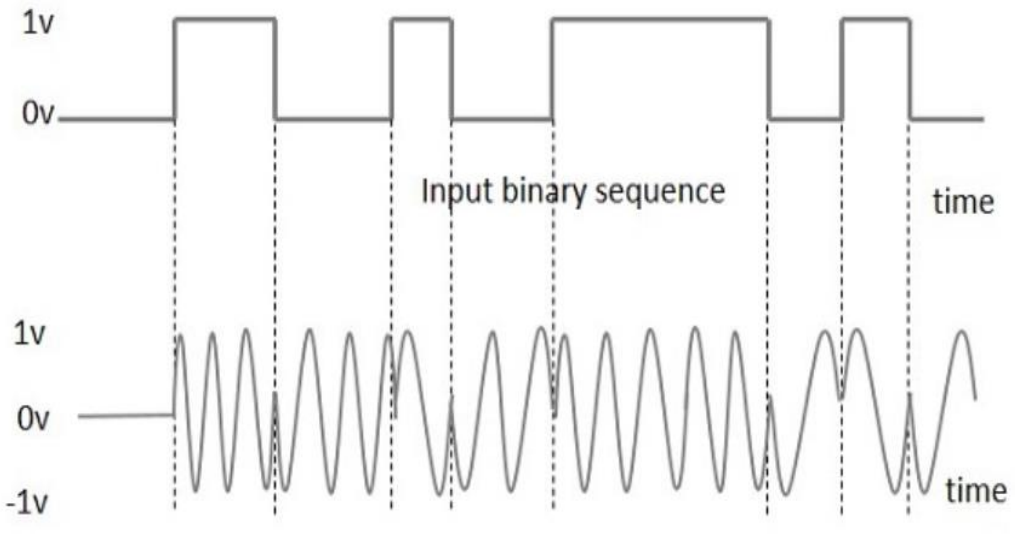


EXPERIMENT-5

AIM: Study of passband digital communication technique BPSK. Calculate the BER of BPSK modulated signal.

THEORY: Phase Shift Keying is the digital modulation technique in which the phase of the carrier signal is changed by varying the sine and cosine inputs at a particular time. PSK technique is widely used for wireless LANs, bio-metric, contactless operations, along with RFID and Bluetooth communications. The block diagram of Binary Phase Shift Keying consists of the balance modulator which has the carrier sine wave as one input and the binary sequence as the other input.

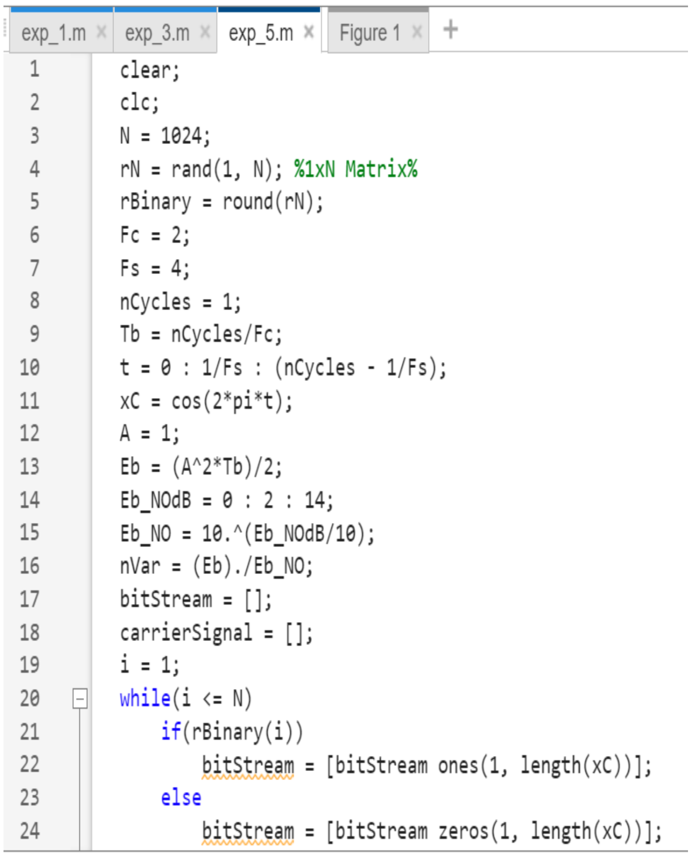




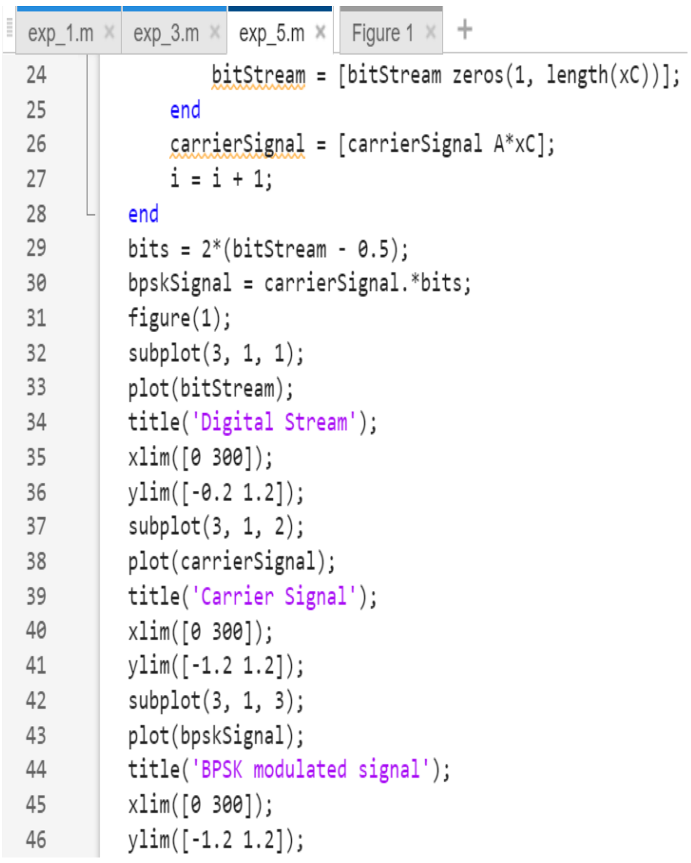
BPSK Modulated output wave

Bit error rate (BER) of a communication system is defined as the ratio of a number of error bits and a total number of bits transmitted during a specific period. It is the likelihood that a single error bit will occur within received bits, independent of the rate of transmission.

CODE:

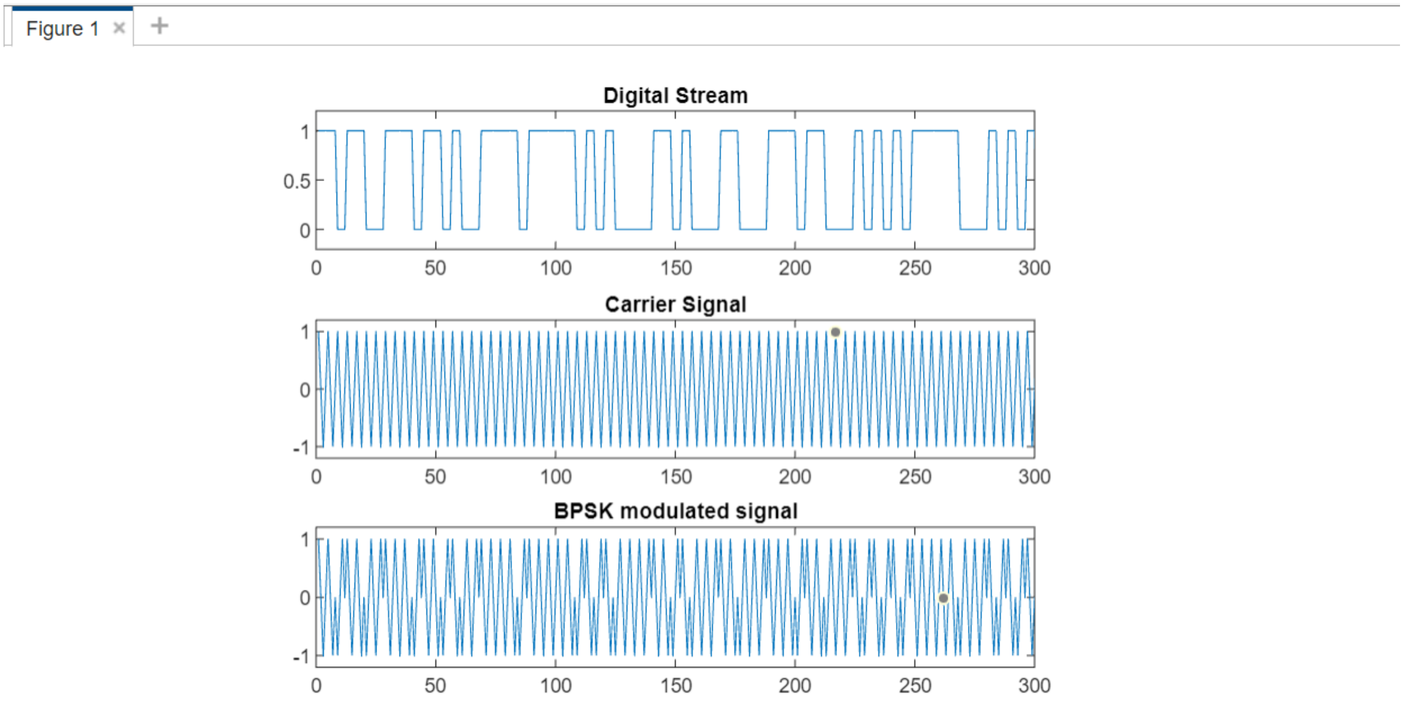


```
clear;
clc;
N = 1024;
rN = rand(1, N); %1xN Matrix%
rBinary = round(rN);
Fc = 2;
Fs = 4;
nCycles = 1;
Tb = nCycles/Fc;
t = 0 : 1/Fs : (nCycles - 1/Fs);
xC = cos(2*pi*t);
A = 1;
Eb = (A^2*Tb)/2;
Eb_N0dB = 0 : 2 : 14;
Eb_N0 = 10.^(Eb_N0dB/10);
nVar = (Eb)./Eb_N0;
bitStream = [];
carrierSignal = [];
i = 1;
while(i <= N)
    if(rBinary(i))
        bitStream = [bitStream ones(1, length(xC))];
    else
        bitStream = [bitStream zeros(1, length(xC))];
```



```
        bitStream = [bitStream zeros(1, length(xC))];
    end
    carrierSignal = [carrierSignal A*xC];
    i = i + 1;
end
bits = 2*(bitStream - 0.5);
bpskSignal = carrierSignal.*bits;
figure(1);
subplot(3, 1, 1);
plot(bitStream);
title('Digital Stream');
xlim([0 300]);
ylim([-0.2 1.2]);
subplot(3, 1, 2);
plot(carrierSignal);
title('Carrier Signal');
xlim([0 300]);
ylim([-1.2 1.2]);
subplot(3, 1, 3);
plot(bpskSignal);
title('BPSK modulated signal');
xlim([0 300]);
ylim([-1.2 1.2]);
```

OUTPUT:



EXPERIMENT-6

AIM: Given is a linear block code with the generator matrix G =

1 1 0 0 1 0 1

0 1 1 1 1 0 0

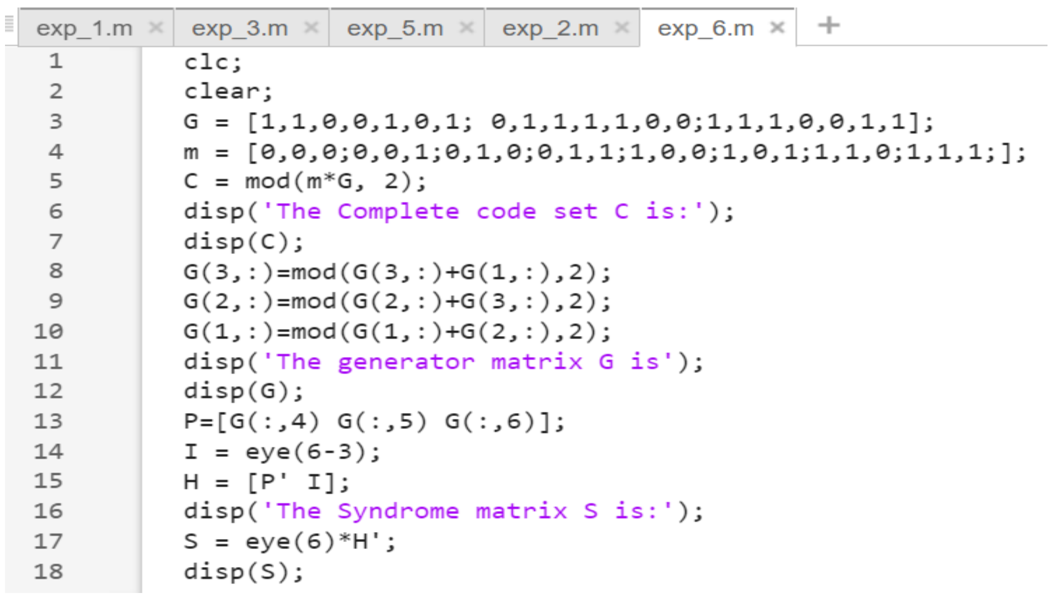
1 1 1 0 0 1 1

a. Calculate the number of valid code words N and the code rate RC. Specify the complete Code set C.

b. Determine the generator matrix G’ of the appropriate systematic (separable) code C’.

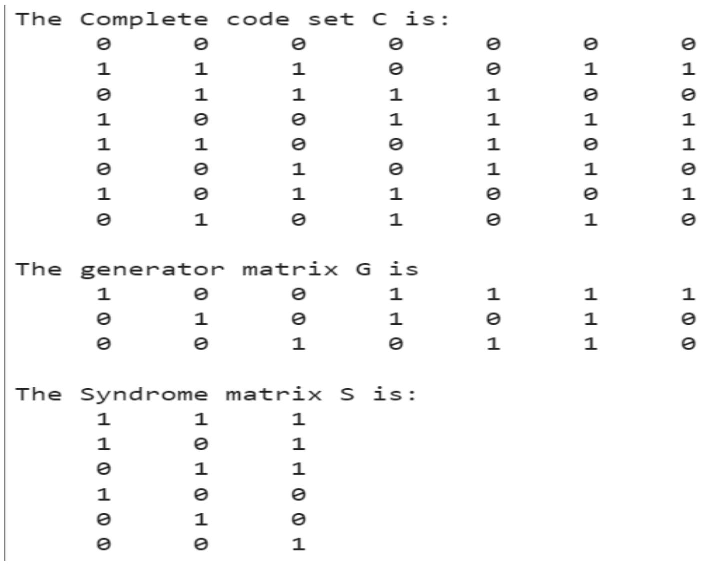
THEORY: Linear block code is a type of error-correcting code in which the actual information bits are linearly combined with the parity check bits so as to generate a linear codeword that is transmitted through the channel. Another major type of error-correcting code is convolution code. In the linear block code technique, the complete message is divided into blocks and these blocks are combined with redundant bits so as to deal with error detection and correction.

CODE:



```
clc;
clear;
G = [1,1,0,0,1,0,1; 0,1,1,1,1,0,0;1,1,1,0,0,1,1];
m = [0,0,0;0,0,1;0,1,0;0,1,1;1,0,0;1,0,1;1,1,0;1,1,1;];
C = mod(m*G, 2);
disp('The Complete code set C is:');
disp(C);
G(3,:)=mod(G(3,:)+G(1,:),2);
G(2,:)=mod(G(2,:)+G(3,:),2);
G(1,:)=mod(G(1,:)+G(2,:),2);
disp('The generator matrix G is');
disp(G);
P=[G(:,4) G(:,5) G(:,6)];
I = eye(6-3);
H = [P' I];
disp('The Syndrome matrix S is:');
S = eye(6)*H';
disp(S);
```

OUTPUT:



```
The Complete code set C is:
     0     0     0     0     0     0     0
     1     1     1     0     0     1     1
     0     1     1     1     1     0     0
     1     0     0     1     1     1     1
     1     1     0     0     1     0     1
     0     0     1     0     1     1     0
     1     0     1     1     0     0     1
     0     1     0     1     0     1     0

The generator matrix G is
     1     0     0     1     1     1     1
     0     1     0     1     0     1     0
     0     0     1     0     1     1     0

The Syndrome matrix S is:
     1     1     1
     1     0     1
     0     1     1
     1     0     0
     0     1     0
     0     0     1
```