Name: Parth

Roll Number: 2020UCA1812

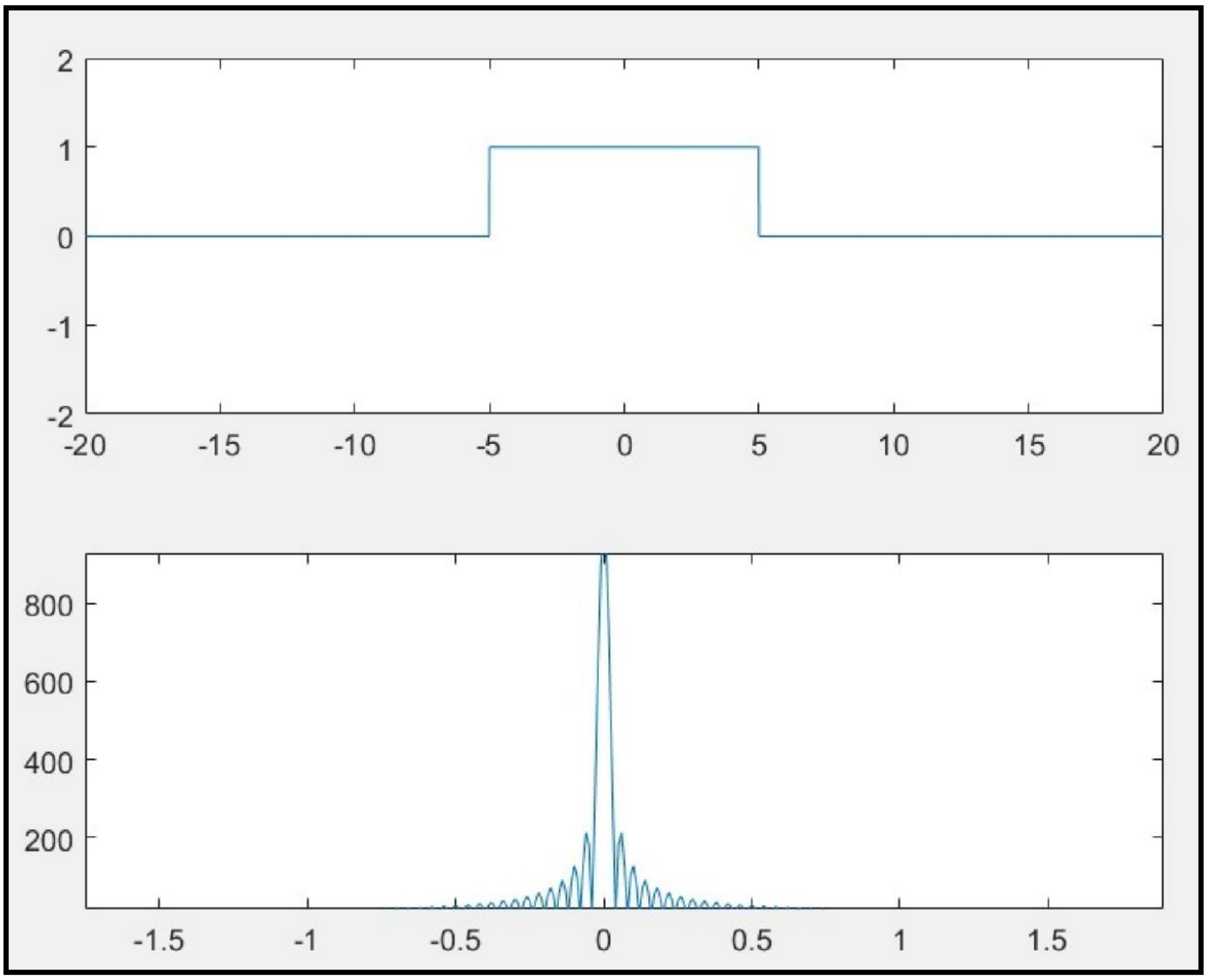
Semester - IV

Data Communication CAECC12

Netaji Subhas University of Technology

1. To plot the spectrum of a pulse of width 10.

```
  clc clear all
close all   w =
10; t = [-
20:0.1:20];
  x = rectpuls(t,
w); figure(1);
subplot(211);
plot(t, x); axis([-
20 20 0 2]);
  X = fft(x);
subplot(212); plot(t,
fftshift(abs(X)));
```

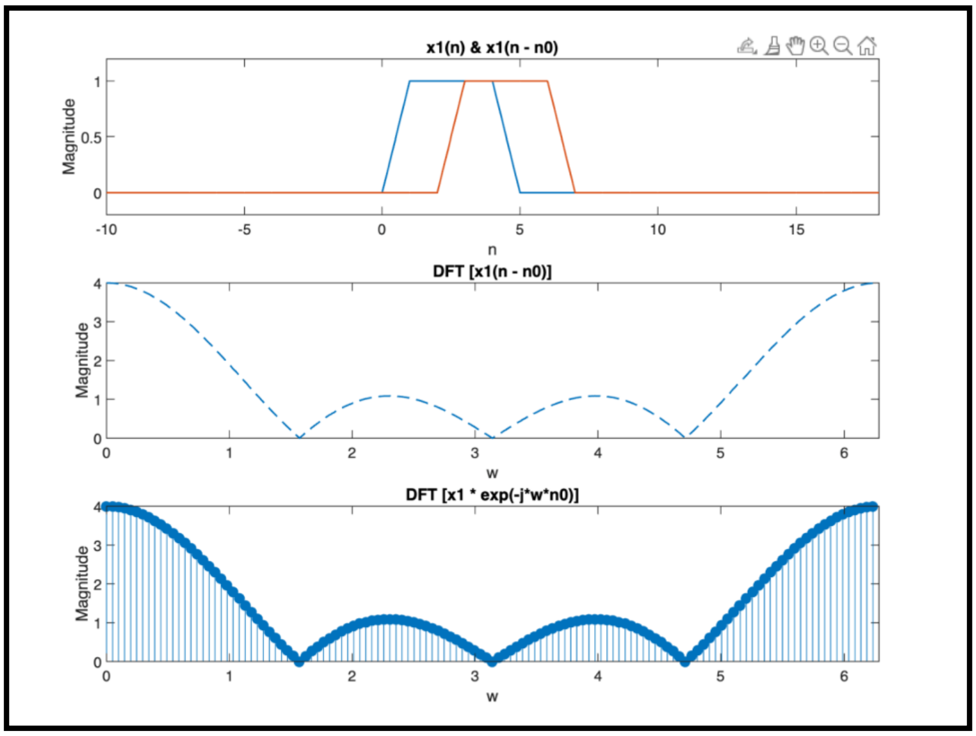


2. To verify following properties of Fourier Transform:

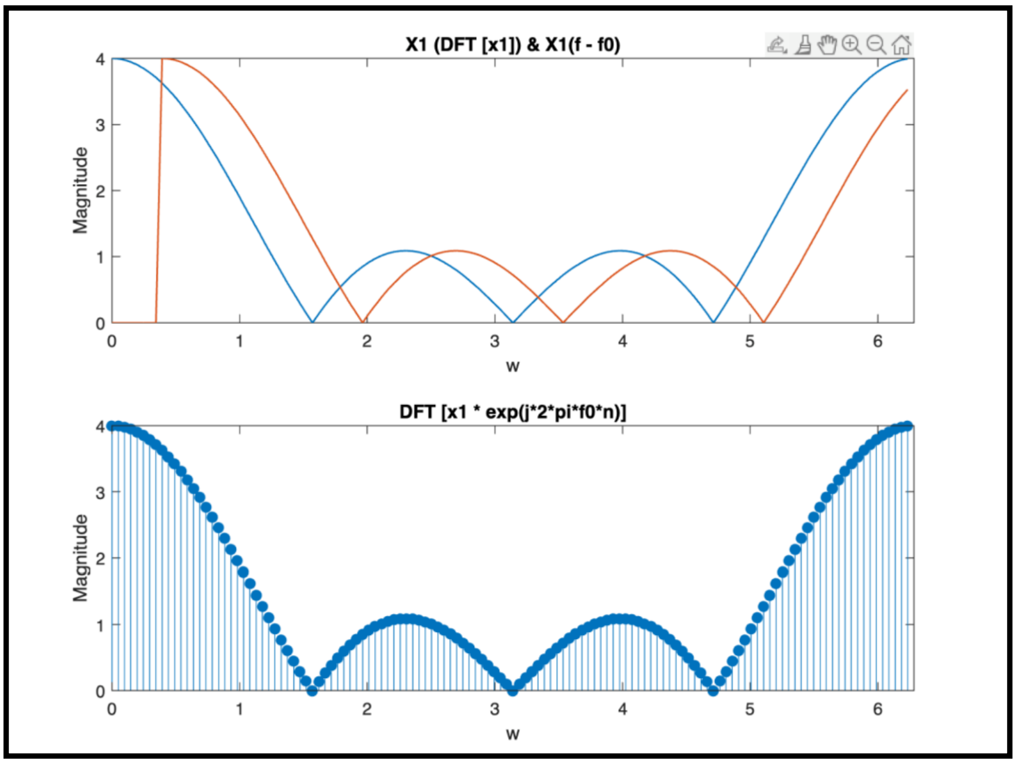
i. Time Shifting, ii. Frequency shifting, iii. Convolutional

```
% 1. Time Shifting n0
= 2;
y = [zeros(1, n0), x1(1 : N - n0)];
Y = fft(y);
Y1 = X1 .* exp(-1i * w * n0);
```

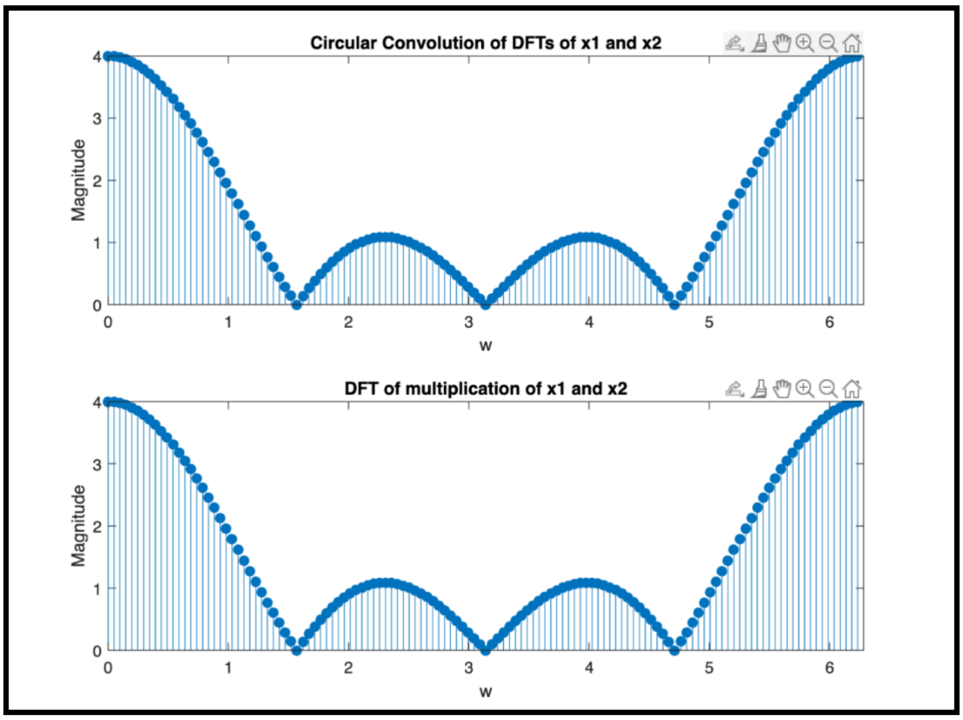
```
figure(3); subplot(311);
plot(-10 : N, [zeros(1, 11), x1], 'LineWidth', 1);
hold on;
plot(-10 : N, [zeros(1, 11), y], 'LineWidth', 1);
xlim([-10 18]); ylim([-0.2
1.2]); xlabel('n');
ylabel('Magnitude');
title('x1(n) & x1(n - n0)');
subplot(312);
plot(w, abs(Y), '--', 'LineWidth', 1);
xlim([0 2 * 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)]');
```



```
%2. Frequency Shifting f0
= 8;
Y = [zeros(1, f0), X1(1 : N - f0)]; y1
= x1 .* exp(1i * 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)]');
```

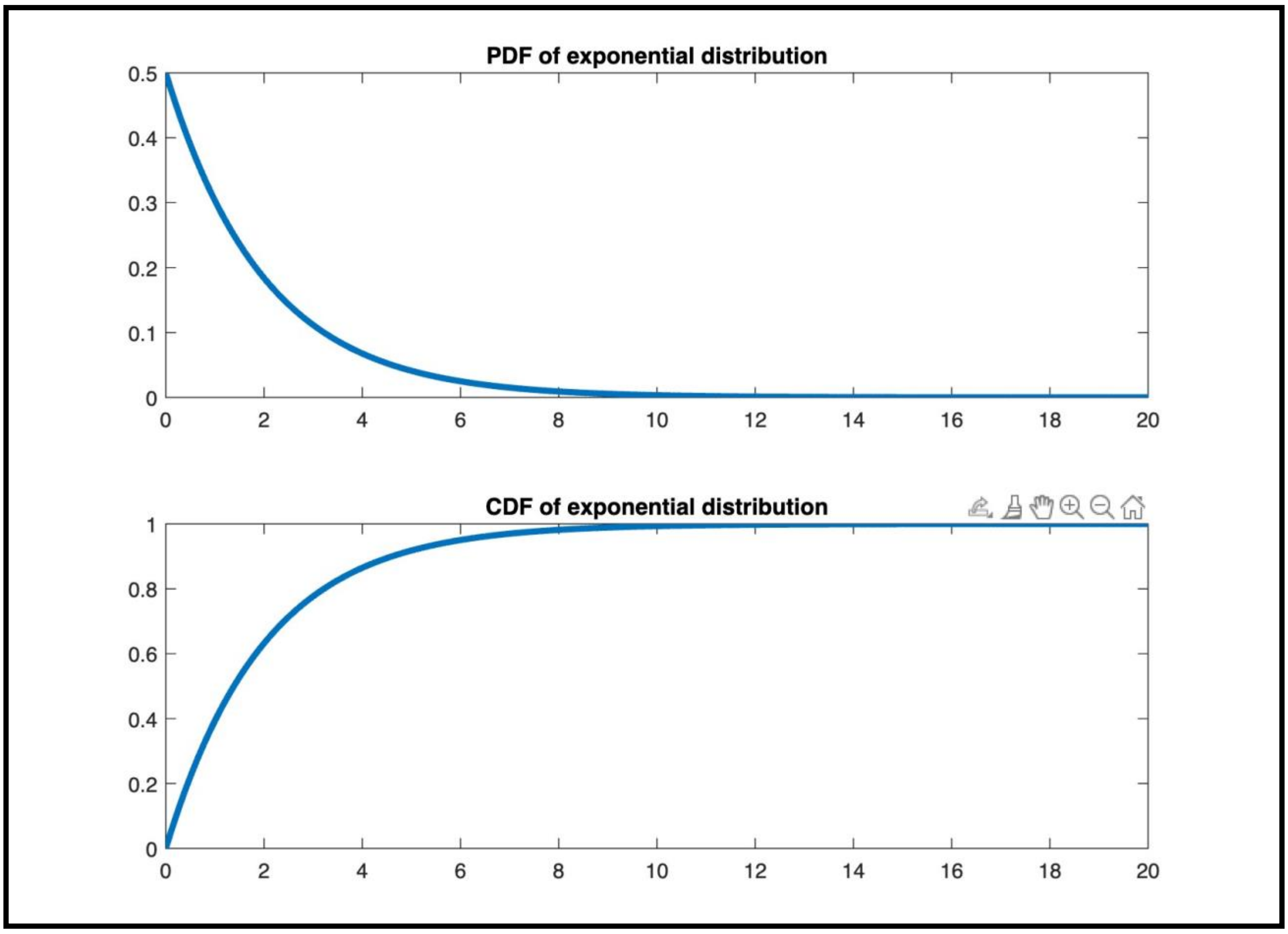


```
% 3. Convolution Property N
= 128; x1 = ones(1, 4); x2 =
ones(1, 6); n1 = length(x1);
n2 = length(x2); x1 = [x1,
zeros(1, N - n1)]; x2 = [x2,
zeros(1, N - n2)]; n = 0 : N
- 1; k = 0 : N - 1;
w = 2 * pi * k / N;
X1 = fft(x1); X2 = fft(x2);
figure(1); subplot(211);
stem(w, abs(X1), 'filled');
xlim([0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

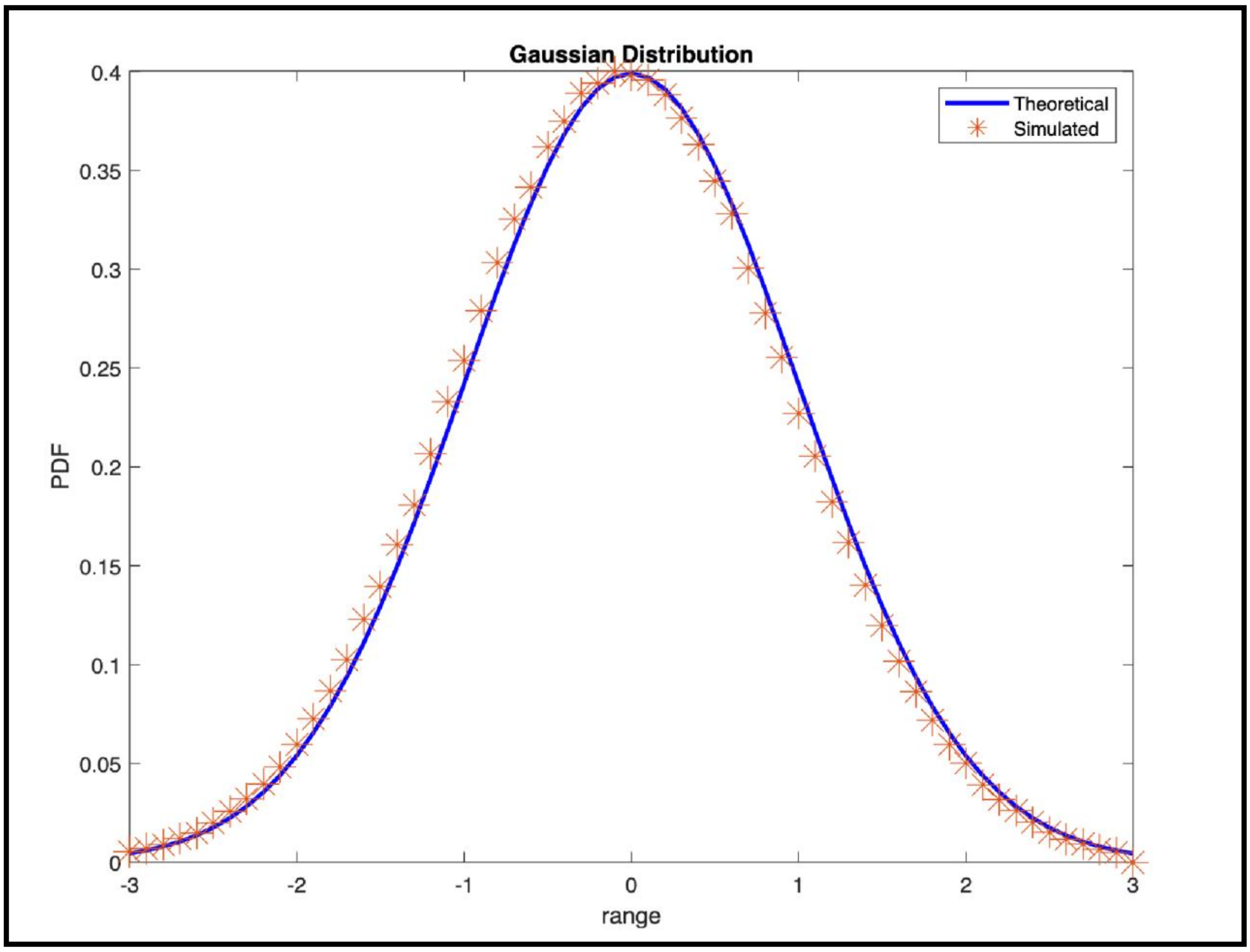


3. Study of uniform, exponential and Gaussian distributed random variables. Draw their PDF and CDF.

```
x = 0 : 0.1 : 100; lambda = 0.5; fx =
lambda * exp(-lambda * x); subplot(211);
plot(x, fx, 'LineWidth', 3); xlim([0
20]); title("PDF of exponential
distribution"); Fx = 1 - exp(-lambda *
x); subplot(212); plot(x, Fx,
'LineWidth', 3); xlim([0 20]);
title("CDF of exponential distribution");
```

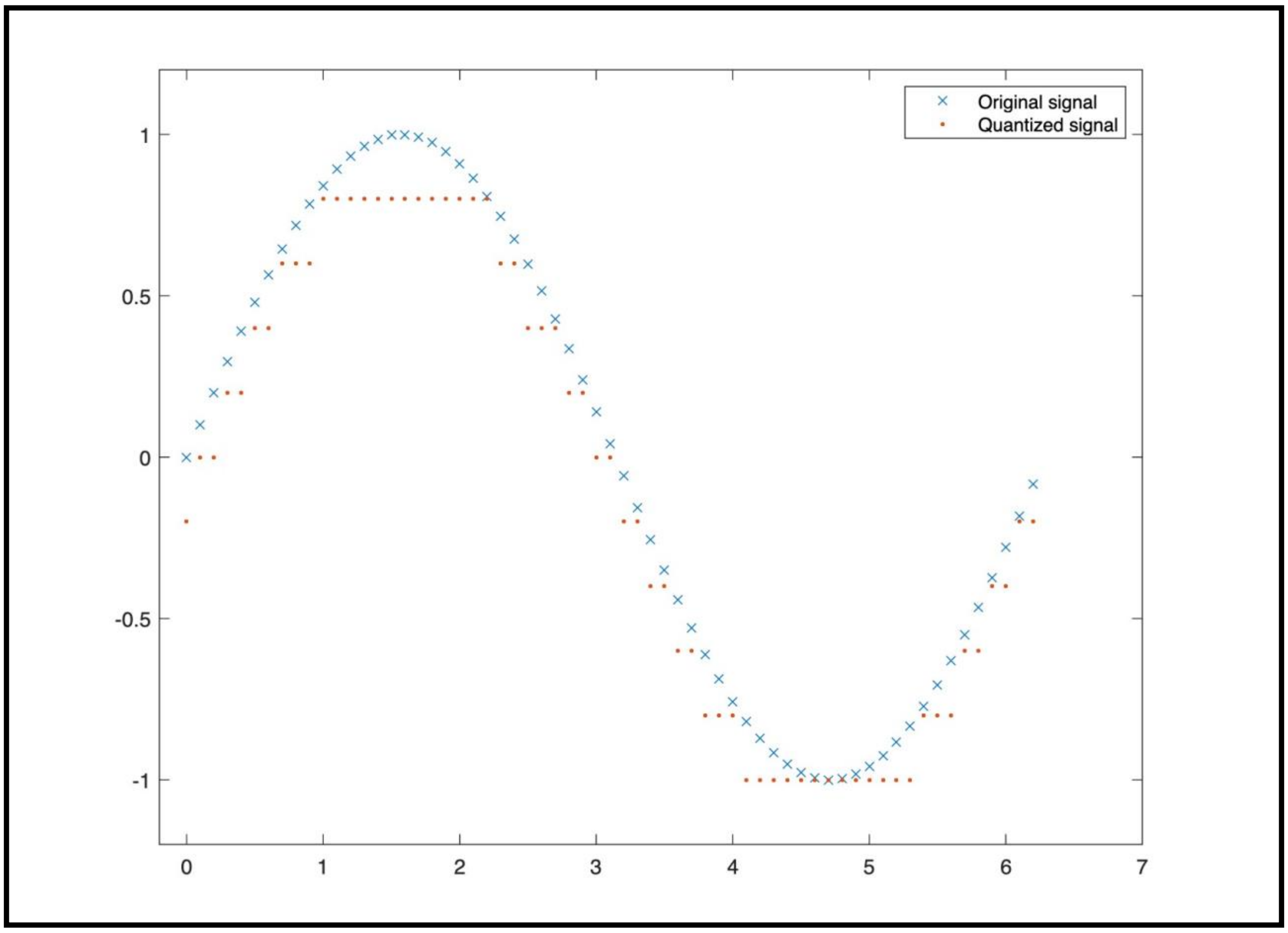


```
N = 1000000;
x = randn(1, N); mu = mean(x); sigma2 = var(x); step = 0.1; range = -
3:step:3; h = histc(x, range); f =
((1./(sqrt(2.*pi.*sigma2))).*exp((-(range-mu).^2)./(2.*sigma2)));
figure;
plot(range, f, 'b', 'linewidth', 2); hold on;
simulatedPDF1 = h./(step.*sum(h)); plot(range,
simulatedPDF1, '*', 'markersize', 15); axis ([-3 3
0 max(simulatedPDF1)]); title('Gaussian
Distribution'); xlabel('range'); ylabel('PDF');
legend('Theoretical', 'Simulated', 'Theoretical')
```



4. Study of linear quantisation.

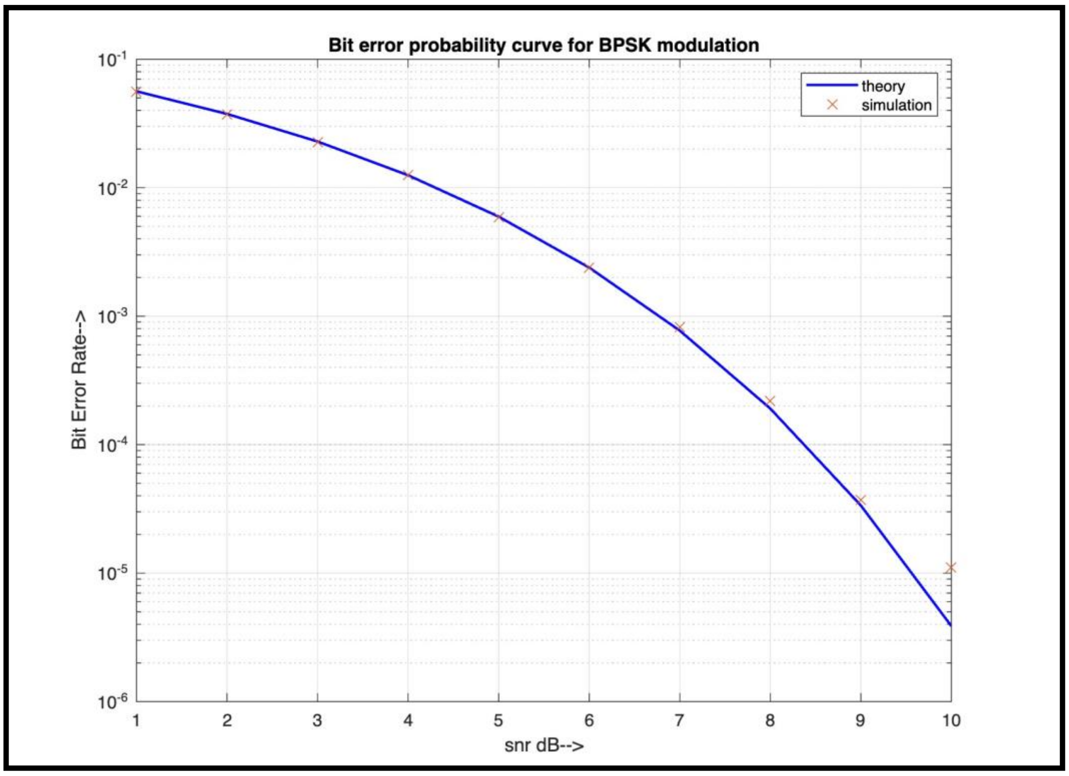
```
% Study of linear quantization. t = 0:.1:2*pi; % Times at
which to sample the sine function sig = sin(t); % Original
signal, a sine wave partition = -1:.2:1; % Length 11, to
represent 12 intervals codebook = -1.2:.2:1; % Length 12, one
entry for each interval [index,quants] =
quantiz(sig,partition,codebook); % Quantize.
plot(t,sig,'x',t,quants,'.') legend('Original
signal','Quantized signal'); axis([-.2 7 -1.2 1.2])
```



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

```
N = 10^6; % number of bits or symbols
% Transmitter
a = rand(1,N)>0.5; % generating 0 and 1   s =
2*a-1; % BPSK modulation 0 -> -1; 1 -> 1
snr_dB = 1:1:10; % multiple Eb/N0 values
snr_ratio = 10.^snr_dB/10;
n = 1/sqrt(2).*(randn(1,N)+1i*randn(1,N));  % mean=0; variance=1;

for i = 1:length(snr_dB)     y =
10^(snr_dB(i)/20).*s + n;

   a_dec = real(y)>0; % receiver -  decision decoding
nErr(i) = size(find(a- a_dec),2); % counting the errors
  end
simBer = nErr/N; % simulated ber
theoryBer = 0.5*erfc(sqrt(10.^(snr_dB/10))); % theoretical ber
figure()
semilogy(snr_dB,theoryBer,'b-','Linewidth',1.5); hold
on
semilogy(snr_dB,simBer,'x','MarkerSize',8); grid
on
legend('theory', 'simulation');
xlabel('snr dB-->'); ylabel('Bit
Error Rate-->');
```

```
title('Bit error probability curve for BPSK modulation');
```



6. Given is a linear block code with the generator matrix G

G =

1 1 0 0 1 0 1

0 1 1 1 1 0 0 1

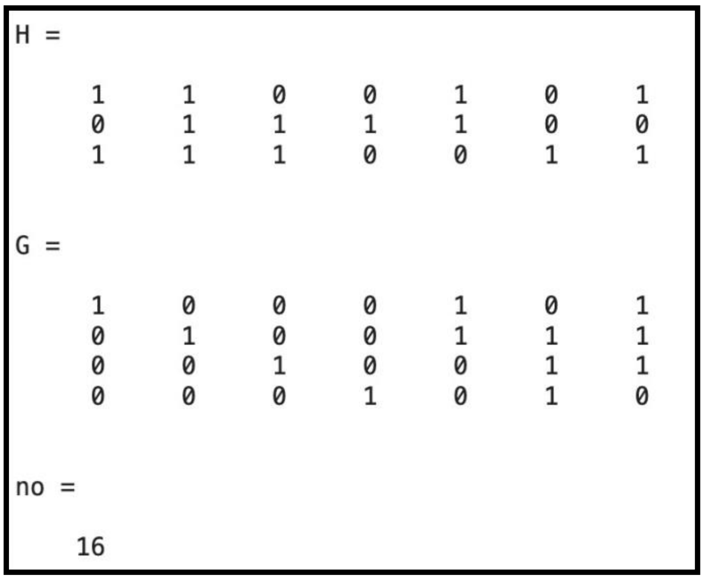
1 1 0 0 1 1

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

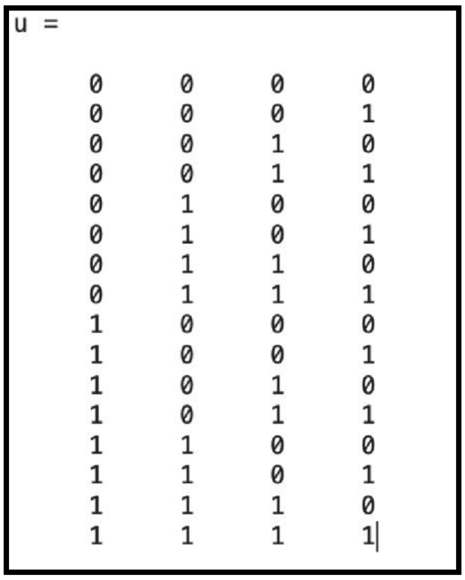
b. Determine the generator matrix G′ of the appropriate systematic (separable) code C’.

```
close all; clear all; clc;
%% Code to solve various problems on linear block code
% Given H Matrix
H = [1 1 0 0 1 0 1; 0 1 1 1 1 0 0; 1 1 1 0 0 1 1]
k = 4; n = 7;
% Generating G Matrix
% Taking the H Matrix Transpose
P = H';
% Making a copy of H Transpose Matrix
L = P;
% Taking the last 4 rows of L and storing
L((5:7), : ) = [];
% Creating a Identity matrix of size K x K
I = eye(k);
% Making a 4 x 7 Matrix
G = [I L]
% Generate U data vector, denoting all information sequences
no = 2 ^ k
% Iterate through an Unit-Spaced Vector
for i = 1 : 2^k
```

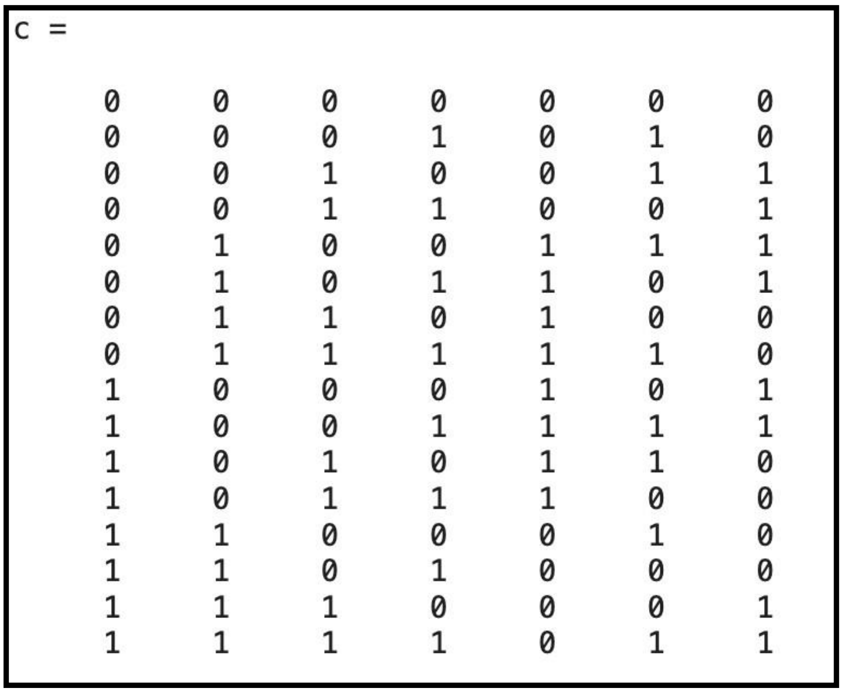
```
% Iterate through Vector with Specified Increment
% or in simple words here we are decrementing 4 till we get 1
for j = k : -1 : 1 if rem(i - 1, 2 ^ (-j + k + 1)) >=
      2 ^ (-j + k)
      u(i, j) =
      1; else
      u(i, j) =
      0; end

      % To avoid displaying each iteration/loop value
      echo off;
end
end

echo on;
u
% Generate CodeWords c
= rem(u * G, 2) % Find
the min distance
w_min = min(sum((c(2 : 2^k, :))')) %
Given Received codeword
r = [0 0 0 1 0 0 0];
r
p = [G(:, n - k + 2 : n)];
%Find Syndrome ht

= transpose(H) s =

rem(r * ht, 2)

for i = 1 : 1 : size(ht)
if(ht(i,1:3)==s) r(i) =
1-r(i); break;
end end disp('The Error is in
bit:') disp(i) disp('The Corrected
Codeword is :')
disp(r)
```



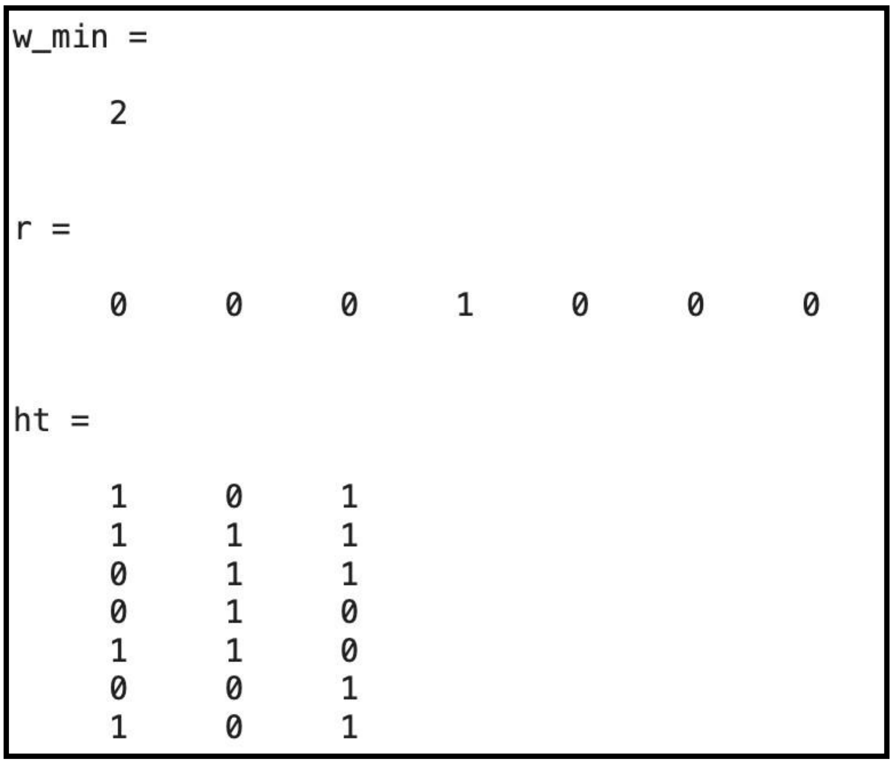
```
H =

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


G =

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


no =

    16
```



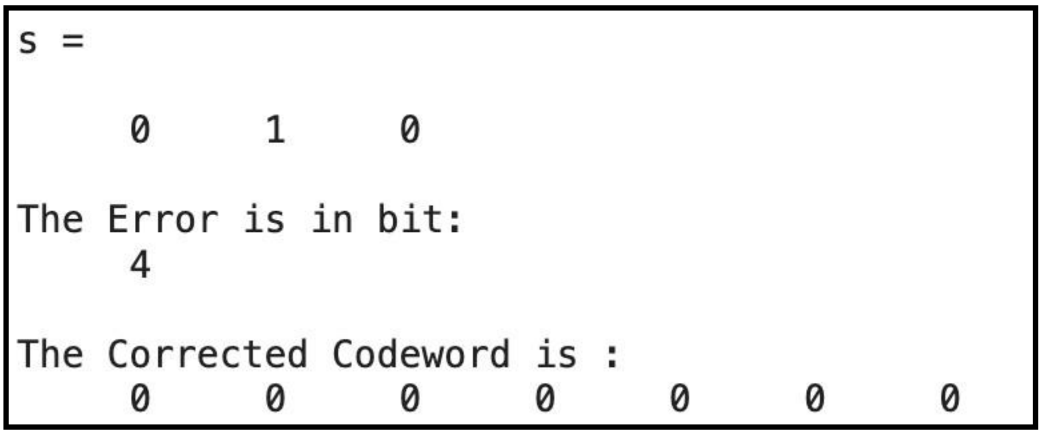
```
u =

     0     0     0     0
     0     0     0     1
     0     0     1     0
     0     0     1     1
     0     1     0     0
     0     1     0     1
     0     1     1     0
     0     1     1     1
     1     0     0     0
     1     0     0     1
     1     0     1     0
     1     0     1     1
     1     1     0     0
     1     1     0     1
     1     1     1     0
     1     1     1     1
```



```
c =

     0     0     0     0     0     0     0
     0     0     0     1     0     1     0
     0     0     1     0     0     1     1
     0     0     1     1     0     0     1
     0     1     0     0     1     1     1
     0     1     0     1     1     0     1
     0     1     1     0     1     0     0
     0     1     1     1     1     1     0
     1     0     0     0     1     0     1
     1     0     0     1     1     1     1
     1     0     1     0     1     1     0
     1     0     1     1     1     0     0
     1     1     0     0     0     1     0
     1     1     0     1     0     0     0
     1     1     1     0     0     0     1
     1     1     1     1     0     1     1
```



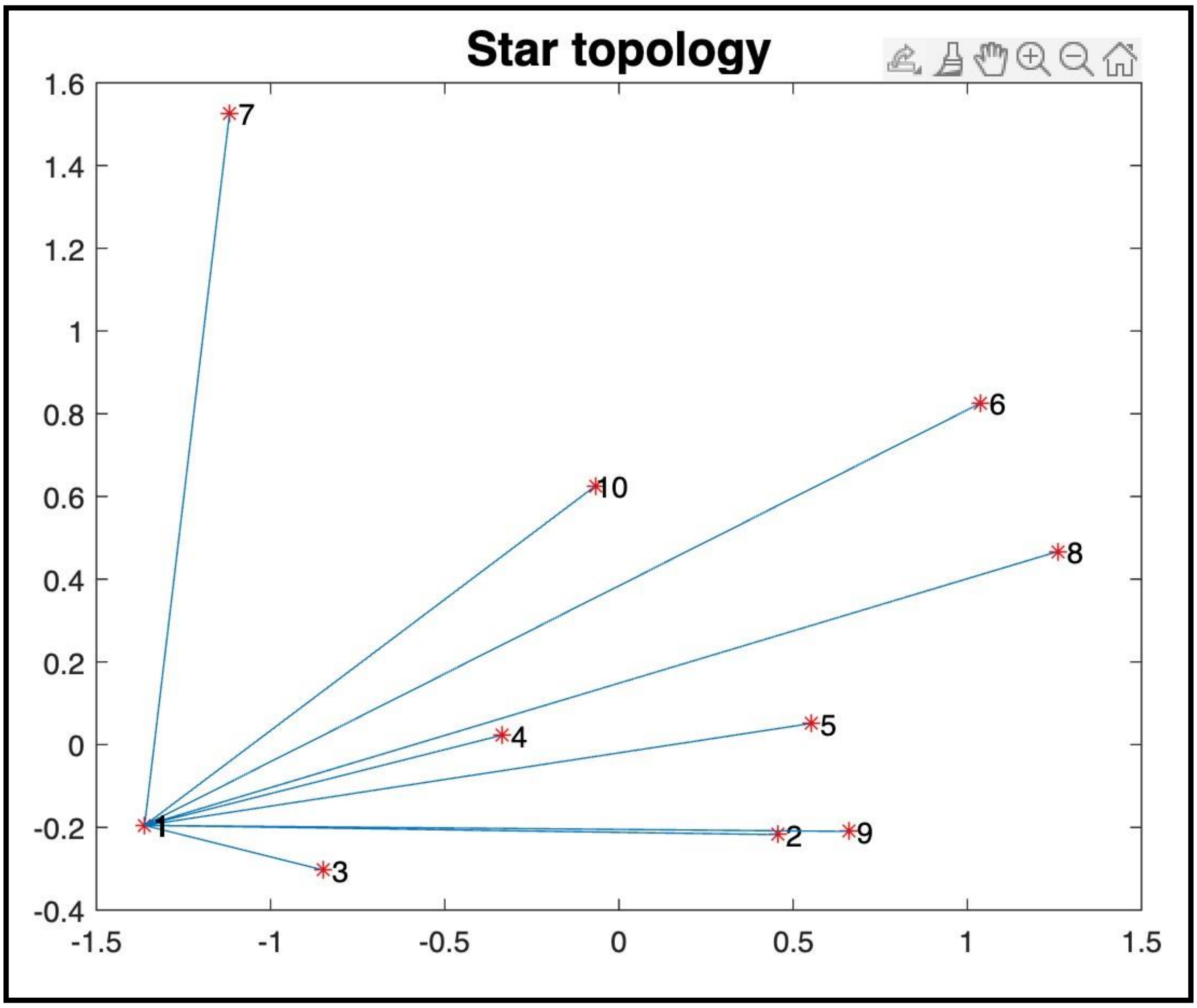
```
w_min =

     2


r =

     0     0     0     1     0     0     0


ht =

     1     0     1
     1     1     1
     0     1     1
     0     1     0
     1     1     0
     0     0     1
     1     0     1
```



```
s =

     0     1     0

The Error is in bit:
     4

The Corrected Codeword is :
     0     0     0     0     0     0     0
```

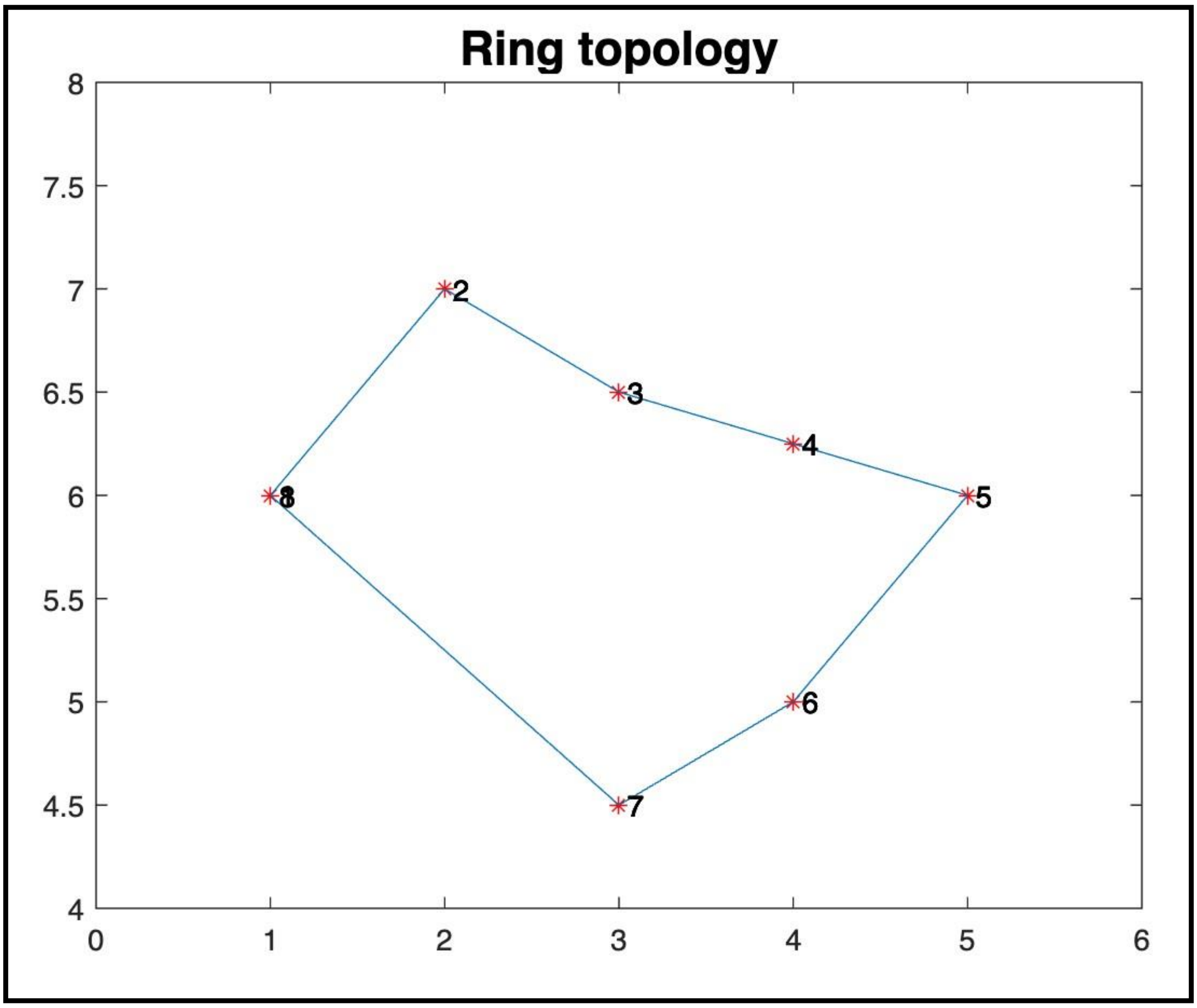
7. Configure and study a network using star topology.

```
clear all;
x=randn(10,1);
y=randn(10,1);
plot(x,y,'r*');
hold on; for
center=1 for
clients=2:10
line([x(center,1),x(clients,1)],[y(center,1),y(clients,1)])
text(x(center,1),y(center,1),sprintf('%2.0f',center))
text(x(clients,1),y(clients,1),sprintf('%2.0f',clients))
title('Star topology','fontsize',16)
end
end
```



8. Configure and study a network using Ring topology.

```
clear all;
x=[1, 2, 3, 4, 5, 4, 3, 1]; y=[6,
7, 6.5, 6.25, 6, 5, 4.5, 6];
plot(x,y,'r*');
hold on; for
clients=2:8
     line([x(clients-1),x(clients)],[y(clients-1),y(clients)])
text(x(clients-1),y(clients-1),sprintf('%2.0f',clients-1))
text(x(clients),y(clients),sprintf('%2.0f',clients))
title('Ring topology','fontsize',16)
xlim([0, 6])
ylim([4, 8])
end
```


Ring topology
8
7.5
7
6.5
6
5.5
5
4.5
4
0
1
2
3
4
5
6
1
8
2
3
4
5
6
7