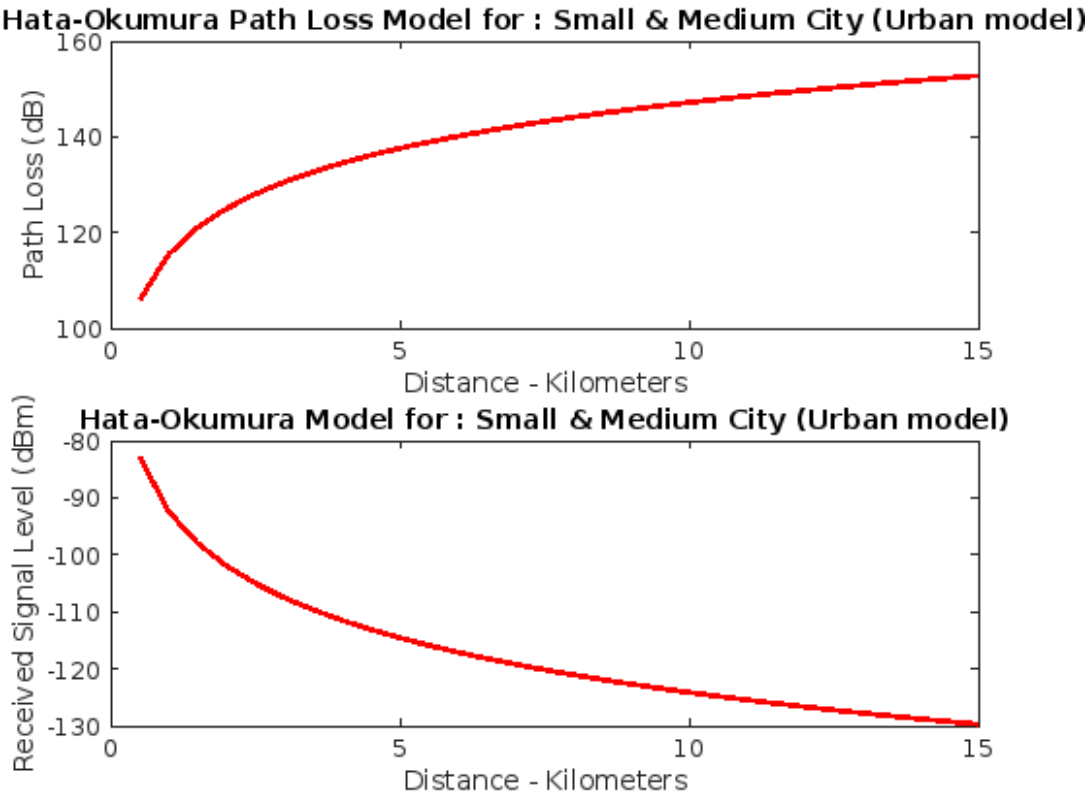
```
%Name: Harshit Rathod
%Roll No.: 304D059
%Exp1-Compute and compare the median loss by employing Hata model
 for various distance for carrier frequencies of 2.1 GHz and 6 GHz. Assume
 transmit and receive antenna heights of 40 m and 2 m in a large city. Plot
 the graph of path loss vs distance.

clc;
clear all;

Hbts= 50 ;
Tbts = 350 ;
Htav= 300;
Hm=3 ;
f=900 ;
d=0.5:0.5:15;
Pt = 0.020;
Gt= 10;

models = {'Big City (Urban model)';'Small & Medium City (Urban model)';'Sub-
urban environment';'Open Rural environment'};
display('Hata-Okumura Model');
display(['1 ' models{1,1}]);
display(['2 ' models{2,1}]);
display(['3 ' models{3,1}]);
display(['4 ' models{4,1}]);

environment = str2double(inputdlg('Select your choice of environment: '));
if environment < 1 || environment > 4
    error('Invalid Selection');
end

modelName = models{environment};
display(['Chosen Model: ' modelName]);

switch environment
    case 1
        C=0;
        if f<=200
            aHm=8.29*(log10(1.54*Hm))^2-1.1;
        else
            aHm=3.2*(log10(11.75*Hm))^2-4.97;
        end
    case 2
        C=0;
        aHm = (1.1*log10(f)-0.7)*Hm-(1.56*log10(f)-0.8);
    case 3
        aHm = (1.1*log10(f)-0.7)*Hm-(1.56*log10(f)-0.8);
        C=-2*(log10(f/28))^2-5.4;
    case 4
        aHm = (1.1*log10(f)-0.7)*Hm-(1.56*log10(f)-0.8);
        C=-4.78*(log10(f))^2+18.33*log10(f)-40.98;
```

```
    otherwise
        error('Invalid model selection');
end

Hb=Hbts+Tbts-Htav;
A = 69.55 + 26.16*log10(f) - 13.82*log10(Hb)-aHm;
B = 44.9 - 6.55*log10(Hb);
PL=A+B*log10(d)+C;
subplot(2,1,1)
plot(d,PL,'r','LineWidth',2);
title(['Hata-Okumura Path Loss Model for : ' modelName]);
xlabel('Distance - Kilometers');
ylabel('Path Loss (dB)');
Pr = 10*log10(Pt*1000)+Gt-PL;
subplot(2,1,2)
plot(d,Pr,'r','LineWidth',2);
title(['Hata-Okumura Model for : ' modelName]);
xlabel('Distance - Kilometers');
ylabel('Received Signal Level (dBm)');
```

```
Hata-Okumura Model
1 Big City (Urban model)
2 Small & Medium City (Urban model)
3 Sub-urban environment
4 Open Rural environment
Chosen Model: Small & Medium City (Urban model)
```

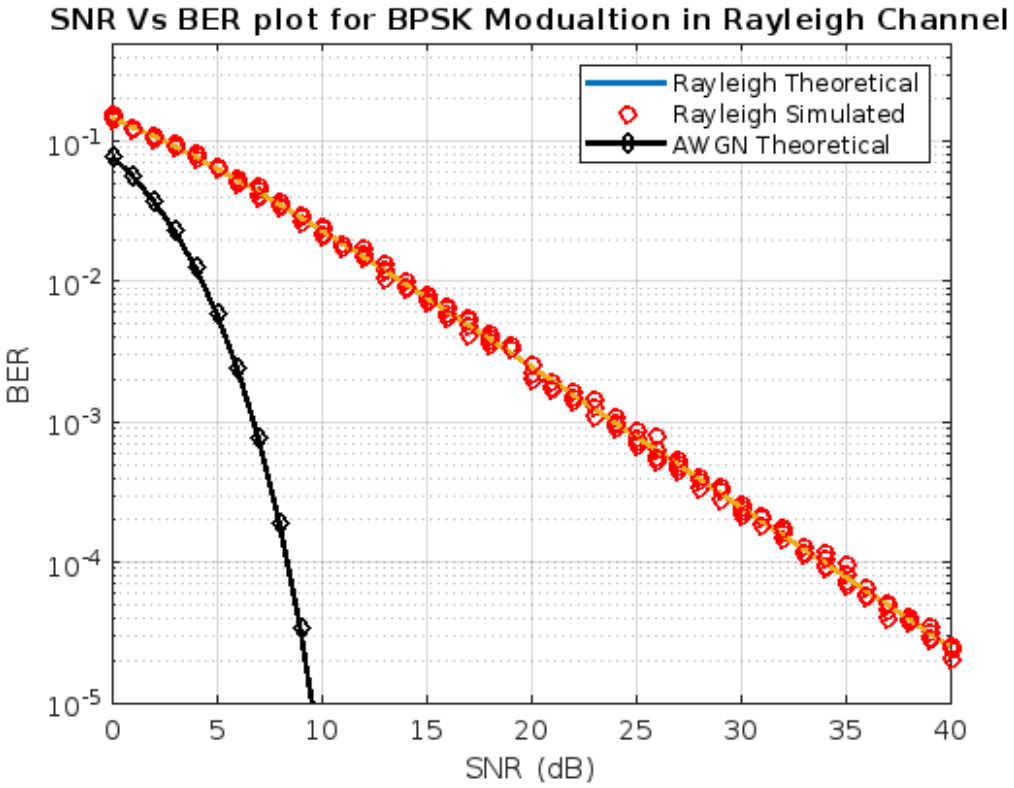


Published with MATLAB® R2023a

```
%Name: Harshit Rathod

%Roll No.: 304D059

clear all;

format long;
bit_count = 10000;
SNR = 0: 1: 40;

for aa = 1: 1: length(SNR)

T_Errors = 0;
    T_bits = 0;

    while T_Errors < 100

        uncoded_bits  = round(rand(1,bit_count));
tx = -2*(uncoded_bits-0.5);
        N0 = 1/10^(SNR(aa)/10);
        h = 1/sqrt(2)*[randn(1,length(tx)) + 1j*randn(1,length(tx))];
rx = h.*tx + sqrt(N0/2)*(randn(1,length(tx))+1j*randn(1,length(tx)));
rx = rx./h;
        rx2 = rx < 0;
        diff = uncoded_bits - rx2;
        T_Errors = T_Errors + sum(abs(diff));
T_bits = T_bits + length(uncoded_bits);

    end

    BER(aa) = T_Errors / T_bits;
    disp(sprintf('bit error probability = %f',BER(aa)));

end


SNRLin = 10.^(SNR/10);
theoryBer = 0.5.*(1-sqrt(SNRLin./(SNRLin+1)));

figure(1);
semilogy(SNR,theoryBer,'-','LineWidth',2);
hold on;

figure(1);
semilogy(SNR,BER,'or','LineWidth',2);
hold on;
xlabel('SNR (dB)');
ylabel('BER');
title('SNR Vs BER plot for BPSK Modualtion in Rayleigh Channel');

figure(1);
theoryBerAWGN = 0.5*erfc(sqrt(10.^(SNR/10)));
semilogy(SNR,theoryBerAWGN,'blad-','LineWidth',2);
```

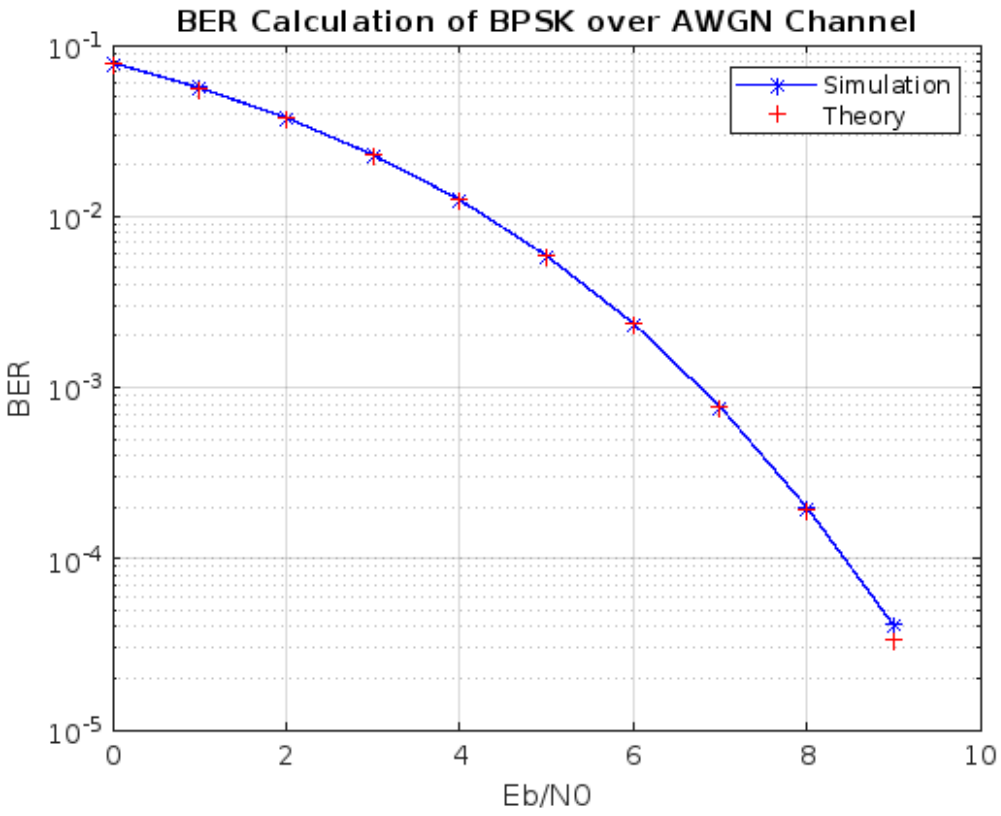
```
legend('Rayleigh Theoretical','Rayleigh Simulated', 'AWGN Theoretical');
axis([0 40 10^-5 0.5]);
grid on;
```

```
bit error probability = 0.157400
bit error probability = 0.119800
bit error probability = 0.106300
bit error probability = 0.090200
bit error probability = 0.080300
bit error probability = 0.063700
bit error probability = 0.049800
bit error probability = 0.047600
bit error probability = 0.035600
bit error probability = 0.028600
bit error probability = 0.021500
bit error probability = 0.017600
bit error probability = 0.014700
bit error probability = 0.013600
bit error probability = 0.009200
bit error probability = 0.007250
bit error probability = 0.005550
bit error probability = 0.004900
bit error probability = 0.003967
bit error probability = 0.003350
bit error probability = 0.002550
bit error probability = 0.001683
bit error probability = 0.001486
bit error probability = 0.001437
bit error probability = 0.000973
bit error probability = 0.000867
bit error probability = 0.000556
bit error probability = 0.000537
bit error probability = 0.000388
bit error probability = 0.000345
bit error probability = 0.000212
bit error probability = 0.000182
bit error probability = 0.000172
bit error probability = 0.000119
bit error probability = 0.000106
bit error probability = 0.000098
bit error probability = 0.000058
bit error probability = 0.000052
bit error probability = 0.000038
bit error probability = 0.000029
bit error probability = 0.000025
```



Published with MATLAB® R2023a

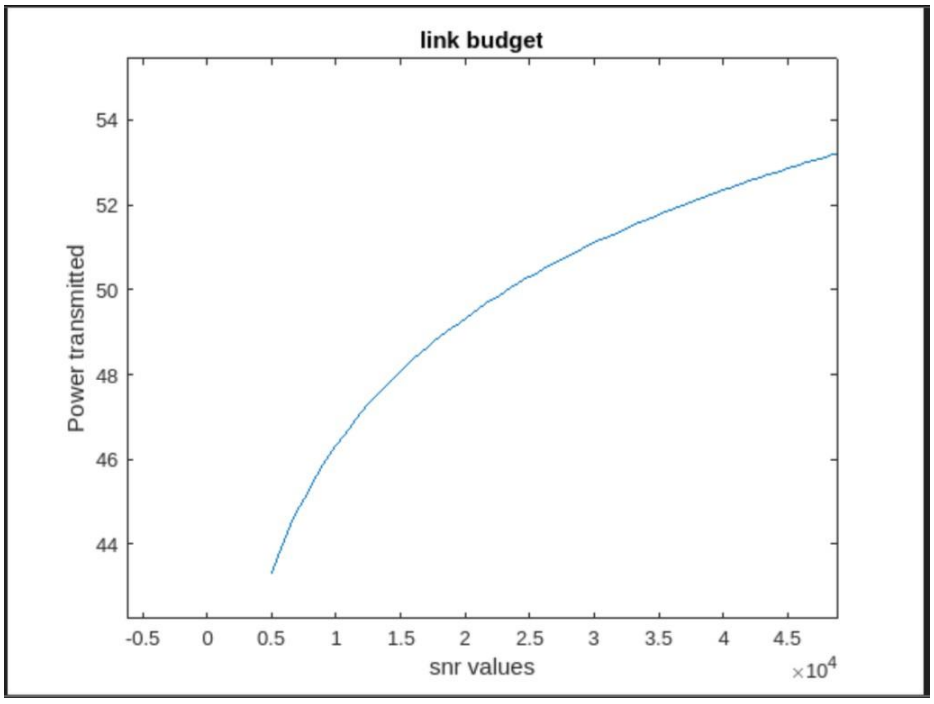
```
%Name: Harshit Rathod
%Roll No.: 304D059
%Exp3 - Simulate BER performance over a wireless AWGN channel with BPSK
transmission for SNR: 0 TO 50 dB.

clc; close
all; clear
; tic;
bit_number = 10^6;
data = randn(1,bit_number)>0.5;
bpsk_data = 2*data-1;
noise = 1/sqrt(2)*(randn(1,bit_number)+1i*randn(1,bit_number));
mean(abs(noise.^2));
SNR = 0:9;
snr_lin= 10.^(SNR/10);
y = zeros(length(SNR),bit_number);
for i = 1:length(SNR)
y(i,:) = real(sqrt(snr_lin(i))*bpsk_data + noise);
end
err = zeros(length(SNR),bit_number);
Err = zeros(10,2);
for i =1:length(SNR) for
    j = 1:bit_number
      if y(i,j)>=0
         y(i,j)=1;
      else
         y(i,j)=0;
      end
end
    err(i,:) = abs(y(i,:)-data);
Err(i,:) = size(find(err(i,:)));
end
ber = zeros(length(SNR),1);
for i=1:length(SNR)
  ber(i) = Err(i,2)/bit_number;
end
theoryBer = 0.5*erfc(sqrt(snr_lin));
semilogy(SNR,ber,'b*-','linewidth',1);
grid on;
hold on;
semilogy(SNR,theoryBer,'r+','linewidth',1);
grid on;
title('BER Calculation of BPSK over AWGN Channel');
xlabel('Eb/N0');
ylabel('BER');
legend('Simulation','Theory');
toc;

Elapsed time is 0.536833 seconds.
```



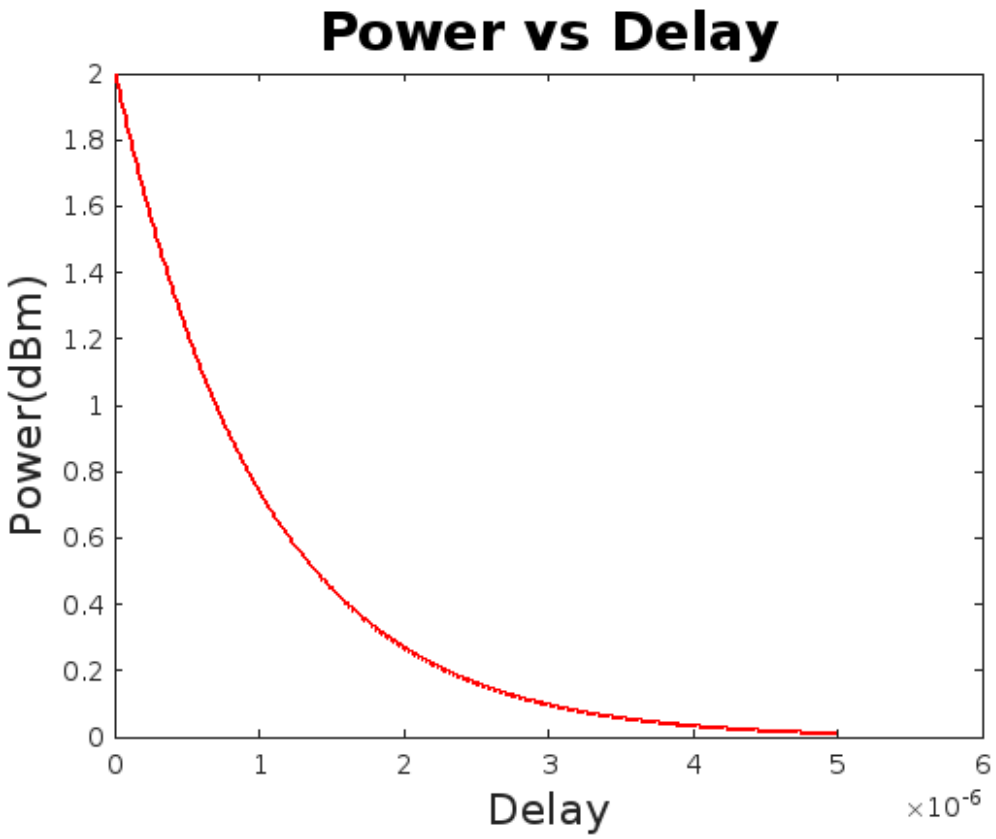
Published with MATLAB® R2023a

```
%Name: Harshit Rathod

%Roll No.: 304D059

Exp4 - Perform a Link-Budget analysis for a wireless communication
system.


clc;
clear all;

L=163.25;
M=10;
k=1.38*10^-23;
t=293;
f=2.1;
Gr=5;
Lc=3;
Gt=12;
BW=30*10^3;
BER=10^-4;
SNR=5000:500:50000;
No=k*t*f;
Sn=No*BW;
I=2*Sn;
snr=10*log10(SNR);
i=10*log10(I);
Pt=snr-Gt+L+M-Gr+Lc+i;
plot(SNR,Pt)
title('link budget');
xlabel('snr values');
ylabel('Power  transmitted')
```

Output –


link budget
Power transmitted
54
52
50
48
46
44
-0.5
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
snr values
×10^4

```
%Name: Harshit Rathod
%Roll No: 304D059
%Exp 5

fun = @(tau) 2*exp(-tau/1e-6);
[meanDelay, rmsDelay, symbolRate, coherenceBW] =
meas_continuous_PDP(fun,0,10e-6);
tau = 0:0.01e-6:5e-6;
fun1 = 2*exp(-tau/1e-6);
plot(tau, fun1, 'r', 'LineWidth', 2);
title('Power vs Delay', 'Fontsize',20);
xlabel('Delay', 'Fontsize',16); ylabel
('Power(dBm)', 'Fontsize',16);
```

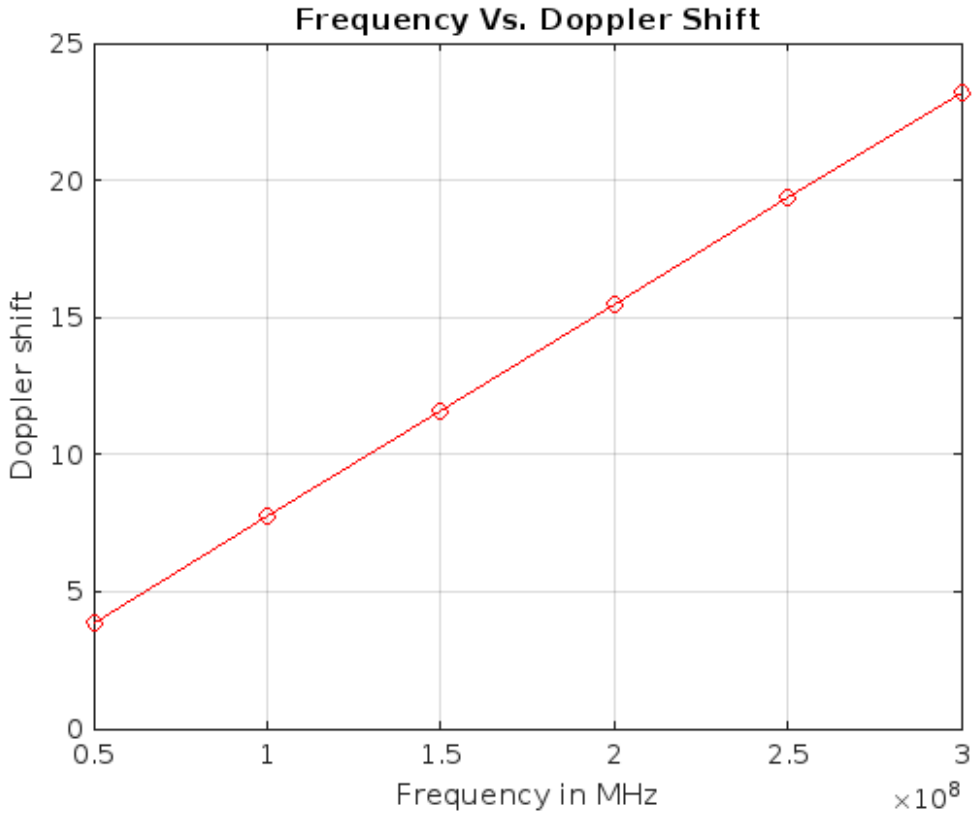


Published with MATLAB® R2023a

```
%Name: Harshit Rathod
%Roll No.: 304D059
% Exp6-Compute doppler shift of the received signal for different carrier
 frequency of mobile generations by considering vehicle is moving at 60  miles
per hour at an angle of 30 degree with the line joining the  base station.

clear; v
=26.82;
f=50000000:50000000:300000000;
C=0.866;
s = 3*10^8;

for aa = 1: 1: length(f)
lamda= s./f;
    a = v./lamda;

deltaf  =  a.*C;

disp(deltaf);
    F=f+deltaf;

end

plot(f,deltaf,'r-o')
grid on;
title('Frequency Vs. Doppler Shift');
xlabel('Frequency in MHz');ylabel('Doppler shift');
```

```
Columns 1 through 3

   3.871020000000000   7.742039999999999  11.613060000000001

Columns 4 through 6

  15.484079999999999  19.355100000000000  23.226120000000002

Columns 1 through 3

   3.871020000000000   7.742039999999999  11.613060000000001

Columns 4 through 6

  15.484079999999999  19.355100000000000  23.226120000000002

Columns 1 through 3

   3.871020000000000   7.742039999999999  11.613060000000001

Columns 4 through 6

  15.484079999999999  19.355100000000000  23.226120000000002

Columns 1 through 3
```

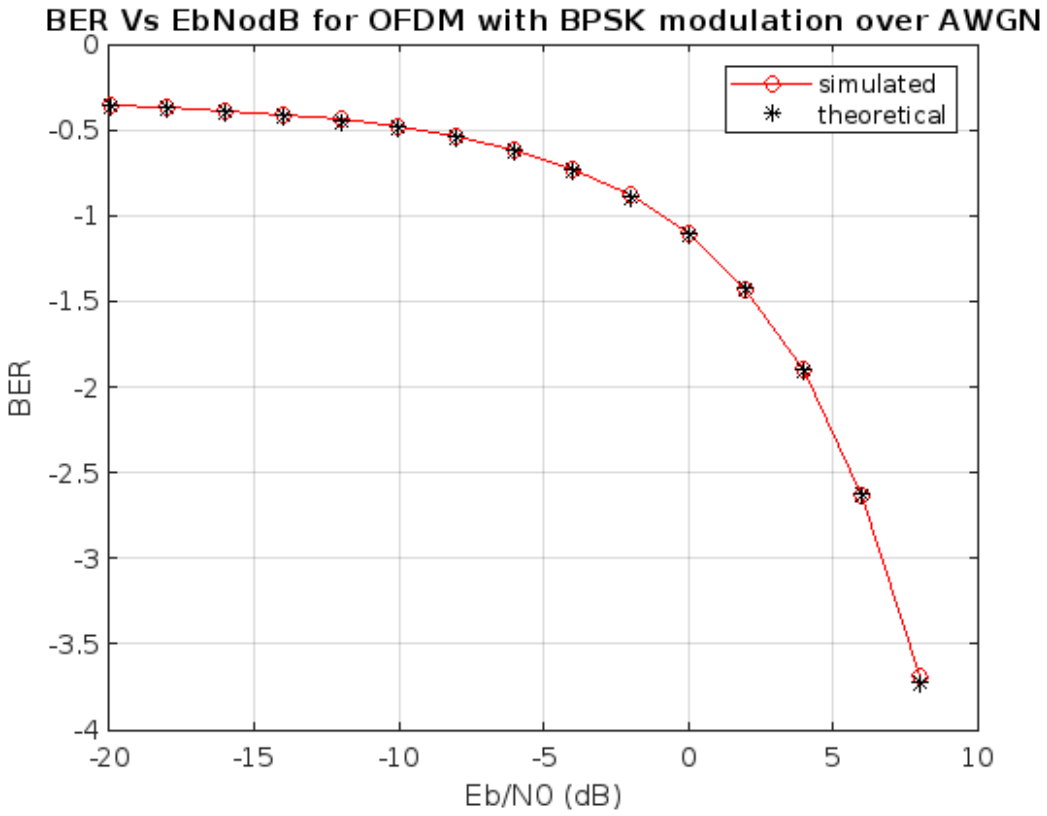
```
    3.871020000000000   7.742039999999999  11.613060000000001

  Columns 4 through 6

   15.484079999999999  19.355100000000000  23.226120000000002

  Columns 1 through 3

    3.871020000000000   7.742039999999999  11.613060000000001

  Columns 4 through 6

   15.484079999999999  19.355100000000000  23.226120000000002

  Columns 1 through 3

    3.871020000000000   7.742039999999999  11.613060000000001

  Columns 4 through 6

   15.484079999999999  19.355100000000000  23.226120000000002
```



Published with MATLAB® R2023a

```
%Name: Harshit Rathod
%Roll No.: 304D059
%Exp7-Program to implement OFDM and evaluate frame error rate against SNR

clear; clc;

nSym=10^4;
EbN0dB = -20:2:8;

N=64;
Nsd = 48;
Nsp = 4 ;
ofdmBW = 20 * 10^6 ;
deltaF = ofdmBW/N;
Tfft = 1/deltaF; Tgi
= Tfft/4; Tsignal =
Tgi+Tfft; Ncp = N*
Tgi/Tfft; Nst = Nsd
+ Nsp;
nBitsPerSym=Nst;
EsN0dB = EbN0dB + 10*log10(Nst/N) + 10*log10(N/(Ncp+N)); % converting to
symbol to noise ratio
errors= zeros(1,length(EsN0dB));
theoreticalBER = zeros(1,length(EsN0dB));

for i=1:length(EsN0dB),
for j=1:nSym

s=2*round(rand(1,Nst))-1;
X_Freq=[zeros(1,1) s(1:Nst/2) zeros(1,11) s(Nst/2+1:end)];
x_Time=N/sqrt(Nst)*ifft(X_Freq);
ofdm_signal=[x_Time(N-Ncp+1:N) x_Time];
noise=1/
sqrt(2)*(randn(1,length(ofdm_signal))+1i*randn(1,length(ofdm_signal)));
r= sqrt((N+Ncp)/N)*ofdm_signal + 10^(-EsN0dB(i)/20)*noise;
r_Parallel=r(Ncp+1:(N+Ncp));
r_Time=sqrt(Nst)/N*(fft(r_Parallel));
R_Freq=r_Time([(2:Nst/2+1) (Nst/2+13:Nst+12)]);
R_Freq(R_Freq>0) = +1;
R_Freq(R_Freq<0) = -1;
s_cap=R_Freq;
numErrors = sum(abs(s_cap-s)/2);
errors(i)=errors(i)+numErrors;
end
theoreticalBER(i)=(1/2)*erfc(sqrt(10.^(EbN0dB(i)/10)));
end
simulatedBER = errors/(nSym*Nst); plot(
EbN0dB,log10(simulatedBER),'r-o'); hold
on;
plot(EbN0dB,log10(theoreticalBER),'k*');
grid on;
title('BER Vs EbNodB for OFDM with BPSK modulation over AWGN');
```

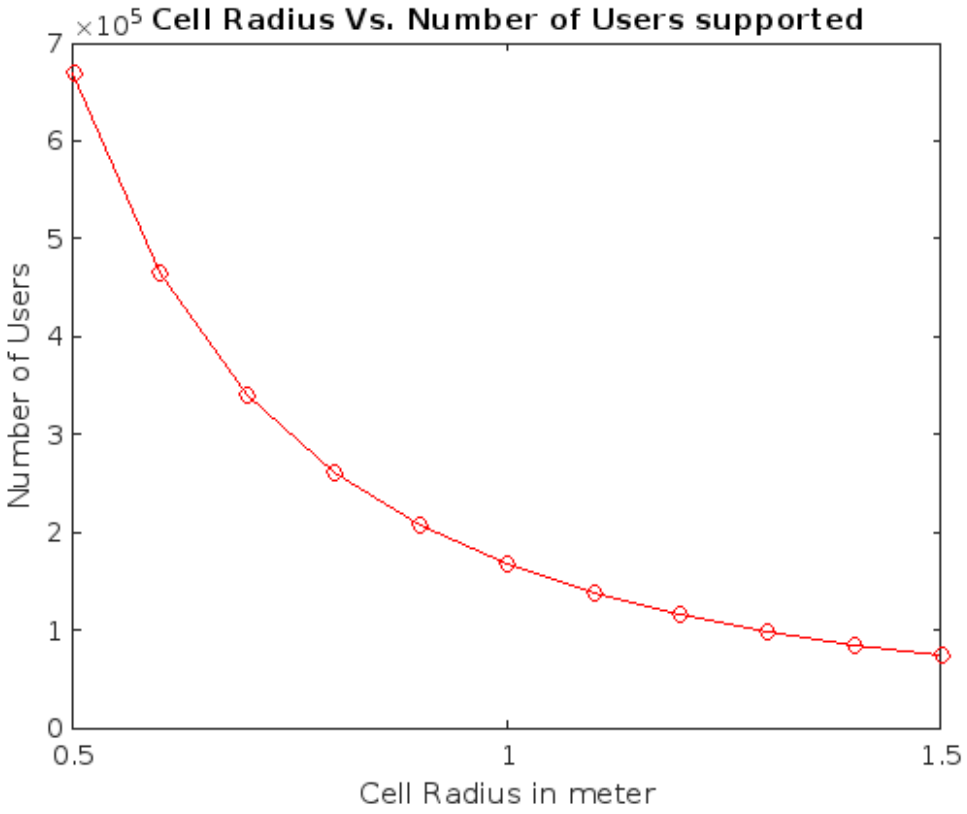
```
xlabel('Eb/N0 (dB)');ylabel('BER');legend('simulated','theoretical');
```



Published with MATLAB® R2023a

```
%Name: Harshit Rathod
%Roll No.: 304P059

%Exp8-Simulate a cellular system with 48 channels per cell and blocking
 probability of 2%. Assume traffic per user is 0.04 E. What is the number pf
 users that can be supported in a city of 603 km 2  area if cell radios are
 changed in the steps of 500 m, 700m, 900 m, 1000 m 1200 m and 1500 m

clear;
n = 48;
Pb = 0.02;
t = 0.04;
area = 603;
prompt = {'Enter total traffic 38.4 for n=48 and pb=0.02:'};
dlgtitle = 'Input';
dims = [1 50];
definput = {'0'};
answer = inputdlg(prompt,dlgtitle,dims,definput);
A = str2double(answer{1});
N = A/t;
disp(N);
cellr = 0.500:0.100:1.500;
for aa = 1:length(cellr)
    cellarea = 6*cellr(aa)^2/sqrt(3);
    cells = area/cellarea;
    users(aa) = N * cells;
end
plot(cellr,users,'r-o')
title('Cell Radius Vs. Number of Users supported');
xlabel('Cell Radius in meter');
ylabel('Number of Users');
```

```
   960
```



Published with MATLAB® R2023a

Name: Harshit Rathod

Roll No.: 304D059

Expt 9 - Simulate mobile environment to evaluate performance parameters using any open source Network Simulator tool.

```
# Simulation parameters setup
set val(chan)   Channel/WirelessChannel    ;# channel type
set val(prop)   Propagation/TwoRayGround   ;# radio-propagation
model
set val(netif)  Phy/WirelessPhy            ;# network interface
type
set val(mac)    Mac/802_11                 ;# MAC type
set val(ifq)    Queue/DropTail/PriQueue    ;# interface queue
type
set val(ll)     LL                         ;# link layer type
set val(ant)    Antenna/OmniAntenna        ;# antenna model
set val(ifqlen) 50                         ;# max packet in ifq
set val(nn)     7                          ;# number of
mobilenodes
set val(rp)     AODV                       ;# routing protocol
set val(x)      1151                       ;# X dimension of
topography
set val(y)      900                        ;# Y dimension of
topography
set val(stop)   10.0                       ;# time of simulation
end

# Initialization
#Create a ns simulator
set ns [new Simulator]

#Setup topography object
set topo [new Topography]
$topo load_flatgrid $val(x) $val(y)
create-god $val(nn)

#Open the NS trace file
set tracefile [open out.tr w]
$ns trace-all $tracefile
```

```
#Open the NAM trace file
set namfile [open out.nam w]
$ns namtrace-all $namfile
$ns namtrace-all-wireless $namfile $val(x) $val(y)
set chan [new $val(chan)];#Create wireless channel

# Mobile node parameter setup
$ns node-config -adhocRouting  $val(rp) \
                -llType        $val(ll) \
                -macType       $val(mac) \
                -ifqType       $val(ifq) \
                -ifqLen        $val(ifqlen) \
                -antType       $val(ant) \
                -propType      $val(prop) \
                -phyType       $val(netif) \
                -channel       $chan \
                -topoInstance  $topo \
                -agentTrace    ON \
                -routerTrace   ON \
                -macTrace      ON \
                -movementTrace ON

# Nodes Definition
#Create 7 nodes
set n0 [$ns node]
$n0 set X_ 338
$n0 set Y_ 305
$n0 set Z_ 0.0
$ns initial_node_pos $n0 20

set n1 [$ns node]
$n1 set X_ 527
$n1 set Y_ 300
$n1 set Z_ 0.0
$ns initial_node_pos $n1 20

set n2 [$ns node]
$n2 set X_ 672
$n2 set Y_ 305
$n2 set Z_ 0.0
$ns initial_node_pos $n2 20

set n3 [$ns node]
$n3 set X_ 867
$n3 set Y_ 304
```

```
$n3 set Z_ 0.0
$ns initial_node_pos $n3 20

set n4 [$ns node]
$n4 set X_ 1051
$n4 set Y_ 302
$n4 set Z_ 0.0
$ns initial_node_pos $n4 20

set n5 [$ns node]
$n5 set X_ 292
$n5 set Y_ 438
$n5 set Z_ 0.0
$ns initial_node_pos $n5 20

set n6 [$ns node]
$n6 set X_ 349
$n6 set Y_ 58
$n6 set Z_ 0.0
$ns initial_node_pos $n6 20

# Generate movement
$ns at 1 " $n6 setdest 890 58 75 "

# Agents Definition
#Setup a TCP connection
set tcp0 [new Agent/TCP]
$ns attach-agent $n5 $tcp0
set sink1 [new Agent/TCPSink]
$ns attach-agent $n6 $sink1
$ns connect $tcp0 $sink1
$tcp0 set packetSize_ 1500

# Applications Definition
#Setup a FTP Application over TCP connection
set ftp0 [new Application/FTP]
$ftp0 attach-agent $tcp0
$ns at 1.0 "$ftp0 start"
$ns at 10.0 "$ftp0 stop"

# Termination
#Define a 'finish' procedure
proc finish {} {
    global ns tracefile namfile
    $ns flush-trace
    close $tracefile
```

```
 close $namfile
    exec nam out.nam &
    exit 0
}

for {set i 0} {$i < $val(nn) } { incr i } {
    $ns at $val(stop) "\$n$i reset"
}

$ns at $val(stop) "$ns nam-end-wireless $val(stop)"
$ns at $val(stop) "finish"
$ns at $val(stop) "puts \"done\" ; $ns halt"
$ns run
```

----------Output----------