# Name-Pravin Patil

# Roll no.-3138

**Experiment no.1 Program:**

%Distance vs loss (dB) for small city for Hata Model' clc; clear all; close

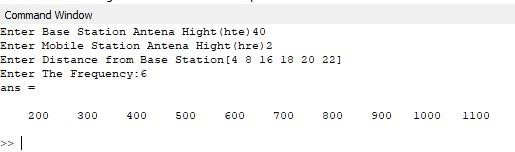
all;

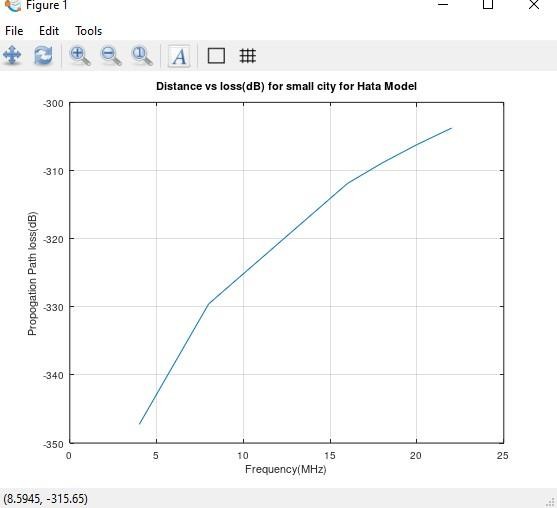
Hte=input('Enter Base Station Antena Hight (hte)'); Hre=input('Enter Mobile Station Antena Hight(hre)'); d= input('Enter Distance from Base Station'); f=input('Enter The Frequency:'); for i=1:length(d) CH=0.8+((1.1\*log(d))-0.7)\*Hre-1.56\*log(d);

LU=69.55+26.16\*log(d)-132.82\*log(Hte)-CH+(44.9-6.55\*log(Hte))\*log(f); end figure(1) plot(d,LU) title('Distance vs loss (dB) for small city for hata Model'); xlabel ('Frequency(MHz)'); ylabel('Propogation Path loss(dB)'); grid on;

[200 300 400 500 600 700 800 900 1000 1100]

# Output:





**Experiment no .2 simulate ber performance over a wireline AWGN**

**Code:-**

clear all; close all; clc; graphics\_toolkit("fltk"); fc=1000; % Frequency for "0" bits t=linspace(0,1/1000,50); e0=cos(2\*pi\*fc\*t); % BPSK output for "1" e1=-cos(2\*pi\*fc\*t); % BPSK output for "0" b=mod(randperm(16),2); bnot=1-b; n=['The binary data is ',num2str(b)]; bpsk1=[ ];bpsk2=[ ];bin=[ ]; for i=1:length(b) bpsk1=[bpsk1,b(i)\*e0]; bpsk2=[bpsk2,bnot(i)\*e1]; bin=[bin,b(i)\*ones(1,50)]; end; bpskout=bpsk1+bpsk2; tm=[0:length(bpsk1)-1]; plot(tm,bin,'r--'); axis([0 length(bin) 0 1.5]); hold on;

plot(tm,bpskout,'b'); axis([0 length(tm) -1.5 1.5]); hold off; xlabel('Time index'); ylabel('Amplitude'); legend('Random binary','BPSK output'); title('Simulation of Binary Phase Shift keying'); gtext(n);

% Script of Eb/N0 Vs BER for BPSK modulation scheme

clear; clc;

N=1000; %Number of input bits EbN0dB = -5:1:5; data=randn(1,N)>=0; % Generating a uniformly distributed random 1s and 0s bpskModulated = 2\*data-1; % Mapping 0->-1 and 1->1

M=2; % Number of Constellation points M=2^k for BPSK k=1

Rm=log2(M); % Rm=log2(M) for BPSK M=2

Rc=1; %Number of Constellation points M=2^k for BPSK k=1

BER = zeros(1,length(EbN0dB)); %Place holder for BER values for each Eb/N0 index=1; for k=EbN0dB,

%Adding noise with variance according to the required Eb/N0 EbN0 = 10.^(k/10); %Converting Eb/N0 dB value to linear scale noiseSigma = sqrt(1./(2\*Rm\*Rc\*EbN0)); %Standard deviation for AWGN Noise noise = noiseSigma\*randn(1,length(bpskModulated)); received = bpskModulated + noise;

%Threshold Detector

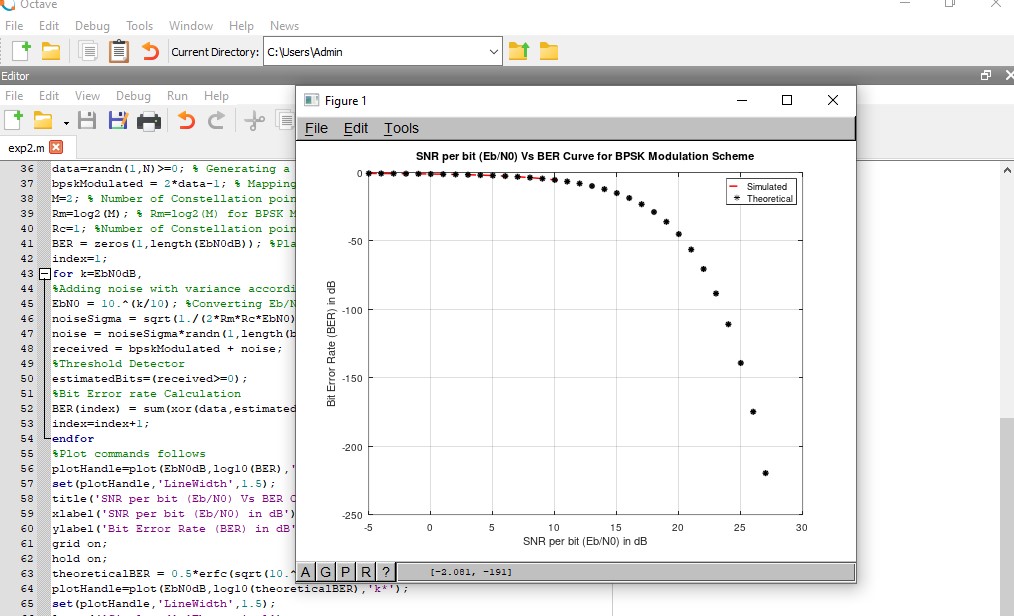
estimatedBits=(received>=0);

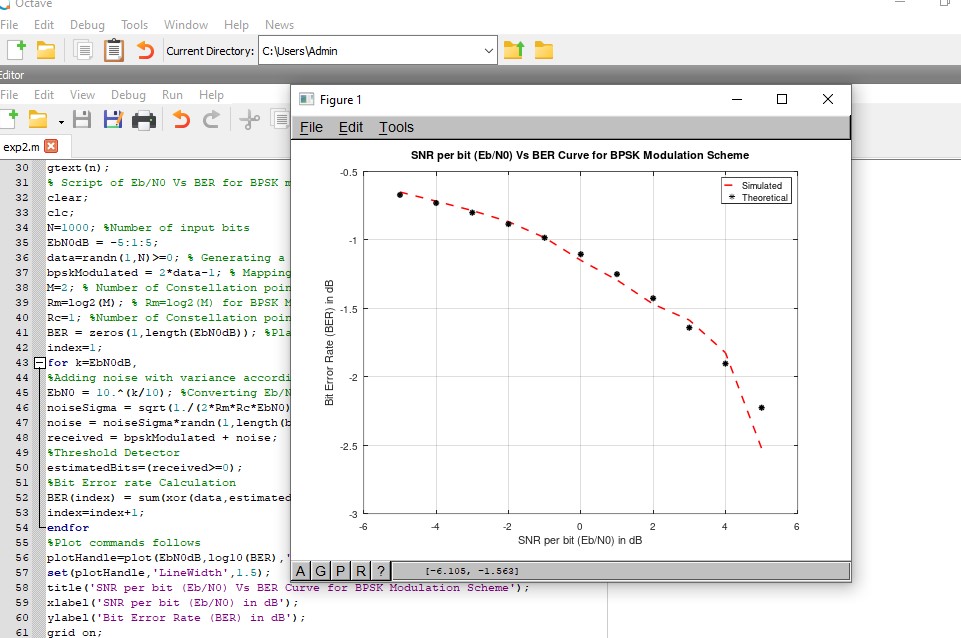
%Bit Error rate Calculation

BER(index) = sum(xor(data,estimatedBits))/length(data); index=index+1; endfor

%Plot commands follows plotHandle=plot(EbN0dB,log10(BER),'r--'); set(plotHandle,'LineWidth',1.5); title('SNR per bit (Eb/N0) Vs BER Curve for BPSK Modulation Scheme'); xlabel('SNR per bit (Eb/N0) in dB'); ylabel('Bit Error Rate (BER) in dB'); grid on; hold on; theoreticalBER = 0.5\*erfc(sqrt(10.^(EbN0dB/10))); plotHandle=plot(EbN0dB,log10(theoreticalBER),'k\*'); set(plotHandle,'LineWidth',1.5); legend('Simulated','Theoretical'); grid on

output:-





**Exp.3. compute Rms delay spread for a give power profile and plot the graph of power vs delay**

**Experiment No.3**

**Title:** Compute the RMS delay spread for a given Power profile and plot the graph of Power vs Delay.

**Name :** Pravin Patil

**Roll No:** 3138

**Program :**

Define new function, File→New Function→meas\_continuous\_PDP function [meanDelay,rmsDelay,symbolRate,coherenceBW] = meas\_continuous\_PDP(fun,lowerLim,upperLim)

%Function to calculate mean Delay, RMS delay spread, maximum symbol

%rate that a signal can be transmitted without ISI and the coherence

%BW for the PDP equation specified as function handle(fun)

% example: fun = @(tau) exp(-tau/0.00001); %given PDP equation

%lowerLim - lower limit for integration

%upperLim - upper limit for integration moment\_1 = @(x) x.\*fun(x); meanDelay = integral(moment\_1,lowerLim,upperLim)/integral(fun,lowerLim,upperLim); moment\_2 = @(y) ((y-meanDelay).ˆ2).\*fun(y); rmsDelay =

sqrt(integral(moment\_2,lowerLim,upperLim)/integral(fun,lowerLim,upperLim)); symbolRate = 1/(10\*rmsDelay); %maximum symbol rate to avoid ISI coherenceBW = 1/(50\*rmsDelay);%for 0.9 correlation

%coherenceBW = 1/(5\*rmsDelay);%for 0.5 correlation

Endfunction

Run following script after saving above function in a file.

fun = @(tau) 2\*exp(-tau/1e-6);

[meanDelay, rmsDelay, symbolRate, coherenceBW] = meas\_continuous\_PDP(fun,0,10e-

6);

tau =

[

0

:0.01

e

-

6:5

e

-

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'

,

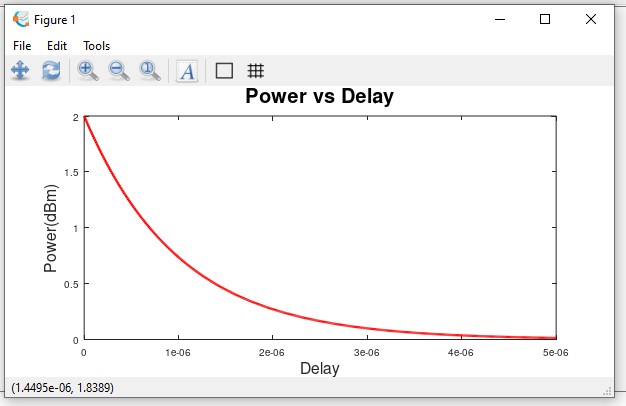
1

6

)

;

**Output :**



**Ex[perimnet .4 perform a link – budget ananlysis for a wireless communication system**

**Experiment no . 5 BER Performance of multi-antena Ralyigh channel for snr**

**Program:**

clc;

clear all;

clc;

clear all;

N=1000000;

L=4;

for SNRdb=[0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60];

SNR=10.^(SNRdb/10);

display(SNR);

p=(SNR/(2+SNR)).^0.5;

display(p);

q=((1-p)/2).^L;

%r=((1+p)/2);

for index=0:1:L-1;

n=L+index-1;

display(n);

k=index;

display(index);

n\_factorial=factorial(n);

k\_factorial=factorial(k);

display(k\_factorial);

n\_minus\_k\_factorial=factorial(n-k);

combi=n\_factorial/(k\_factorial.\* n\_minus\_k\_factorial);

%syms index

f = combi.\*(((1+p)/2).^index);

V = subs(f, k, 0:3)

BER = sum(V)

display(BER);

%syms index;

%BER = symsum((combi.\*(r.^index)),index,0,3);

BERdb=10.\*log10(BER);

BERdb\_perbit=BERdb/N;

display(BERdb\_perbit);

%plot(SNRdb,BER,'r');

end

ezplot(BERdb\_perbit,[0,60]);

hold on;

end

xlabel('SNR in Decibel');

ylabel('BER in Decibel per bit');

grid on;

**Output:**

**Experiment no .6**

**Title:**

**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.**

**Program:**

clc;

clear all;

speed\_miles=60; % Given Speed in miles per hour

speed\_km=60\*1.6; % conversion to km per hour

speed\_meter\_per\_sec=speed\_km\*(5/18); % conversion to m/s

c=3\*(10^8); %speed of light in m/s

fd=(speed\_meter\_per\_sec/c)\*cos(pi/6)\*(1850\*10^6); % Doppler Shift frequency

display(fd);

TE E&TC

**Experiment no .8 Title: Simulate the OFDM system and evaluate frame error rate against SNR**

# Matlab Code:

%Simulation of OFDM system in an AWGN environment clear; clc;

%--------Simulation parameters----------------

nSym=10^4; %Number of OFDM Symbols to transmit EbN0dB = -20:2:8; % bit to noise ratio

%

%--------OFDM Parameters - Given in IEEE Spec--

N=64; %FFT size or total number of subcarriers (used + unused) 64 Nsd = 48; %Number of data subcarriers 48

Nsp = 4 ; %Number of pilot subcarriers 4 ofdmBW = 20 \* 10^6 ; % OFDM bandwidth

%

% Derived Parameters deltaF = ofdmBW/N; %=20 MHz/64 = 0.3125 MHz Tfft = 1/deltaF; % IFFT/FFT period = 3.2us

Tgi = Tfft/4;%Guard interval duration - duration of cyclic prefix

Tsignal = Tgi+Tfft; %duration of BPSK-OFDM symbol

Ncp = N\*Tgi/Tfft; %Number of symbols allocated to cyclic prefix Nst = Nsd + Nsp; %Number of total used subcarriers

nBitsPerSym=Nst; %For BPSK the number of Bits per Symbol is same as num of subcarriers

%

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));

%Monte Carlo Simulation for i=1:length(EsN0dB), for j=1:nSym

% Transmitter

s=2\*round(rand(1,Nst))-1; %Generating Random Data with BPSK modulation

%IFFT block

%Assigning subcarriers from 1 to 26 (mapped to 1-26 of IFFT input)

%and -26 to -1 (mapped to 38 to 63 of IFFT input); Nulls from 27 to 37

%and at 0 position

X\_Freq=[zeros(1,1) s(1:Nst/2) zeros(1,11) s(Nst/2+1:end)];

% Pretending the data to be in frequency domain and converting to time domain x\_Time=N/sqrt(Nst)\*ifft(X\_Freq);

%Adding Cyclic Prefix ofdm\_signal=[x\_Time(N-Ncp+1:N) x\_Time];

% Channel Modeling 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;

% Receiver

%Removing cyclic prefix r\_Parallel=r(Ncp+1:(N+Ncp));

%FFT Block r\_Time=sqrt(Nst)/N\*(fft(r\_Parallel));

%Extracting the data carriers from the FFT output R\_Freq=r\_Time([(2:Nst/2+1) (Nst/2+13:Nst+12)]);

%BPSK demodulation / Constellation Demapper.Force +ve value --> 1, -ve value --> -1 R\_Freq(R\_Freq>0) = +1;

R\_Freq(R\_Freq<0) = -1; s\_cap=R\_Freq;

numErrors = sum(abs(s\_cap-s)/2); %Count number of errors

%Accumulate bit errors for all symbols transmitted errors(i)=errors(i)+numErrors;

end

theoreticalBER(i)=(1/2)\*erfc(sqrt(10.^(EbN0dB(i)/10))); %Same as BER for BPSK over AWGN

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');

**Experiment no .9**

**Title: 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 of users that can be supported in a city of 603 km2 area if cell radios are changed in the steps of 500 m, 700m, 900 m, 1000 m 1200 m and 1500 m**.

**Program:**

clc;

clear all;

Pb=0.02;

Ao=0.04;

r=0.5:0.2:1.5;

AreaCity=603;

A=38.4;

ang=pi/6;

NoUsersPerCell=A/Ao;

display(NoUsersPerCell);

len1=tan(ang)\*r;

len=2\*len1;

display(len);

AreaHexagon=3.\*r.\*len;

display(AreaHexagon);

NoCells=AreaCity./AreaHexagon;

display(NoCells);

NoUser=NoUsersPerCell.\*NoCells;

display(NoUser);

plot(r,NoUser,'b');

xlabel('Radius of Hexagon');

ylabel('NoUser');

title('Graph for Estimating total number of Users In City ');