**BAPATLA ENGINEERING COLLEGE: BAPATLA (AUTONOMOUS)**

**WIRELESS AND MOBILECOMMUNICATION LAB MANUAL**

**[CODE: 18ECL72]**

Logo

Description automatically generated

**Prepared by:**

Mrs. D. Swetha , M.Tech

Assistant Professor

Mr. Sk.Idrish, M.Tech, (Ph.D)

Assistant Professor

Mr. Y. VENKATA KRISHNA,

LAB ASSISTANT

**IV B.Tech.–I Semester**

**WIRELESS AND MOBILE COMMUNICATIONS LAB**

**(18ECL72)**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Lectures | 0 | Tutorial | 0 | | Practical | 3 | Credits | 1 | |
| Continuous Internal Assessment | | | : | 50 | Semester End Examination | | | : | 50 |

**COURSE OUTCOMES (CO):**

|  |  |
| --- | --- |
| **Course Outcome (CO)** | **Course Outcome description** |
| **CO 1** | Understand and analyze fading channel models |
| **CO 2** | Understand DSSS modulation technique |
| **CO3** | Design and analyze OFDM and MIMO systems |

**CO, PO AND PSO MAPPING:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **18ECL72** | **WMC Lab** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO 12** | **PSO 1** | **PSO 2** | **PSO 3** |
| **CO 1** | Understand and analyze fading channel models | 3 | 2 |  | 2 | 2 |  |  |  |  |  |  |  | 1 | 1 | 3 |
| **CO 2** | Understand DSSS modulation technique | 2 | 3 |  | 2 | 2 |  |  |  |  |  |  |  | 2 | 2 | 2 |
| **CO 3** | Design and analyze OFDM and MIMO systems | 3 | 2 |  |  | 2 |  |  |  |  |  |  |  | 2 | 2 | 3 |

**LIST OF EXPERIMENTS**

**The following experiments can be performed using Matlab/Simulink/Scilab/Virtual Labs.**

1. Simulation of Friss Transmission equation.
2. Simulation of Rayleigh fading Channel model.
3. Calculate the probability that the received signal level crosses a certain sensitivity level.
4. Study the outage probability, LCR & ADF in SISO for Selection Combining and MRC.
5. Study the effect of handover threshold and margin on SINR and call drop probability and handover probability.
6. Study the effect of delay spread on frequency selectivity.
7. Plot BER-SNR and Bit Rate-SNR graphs for No Fading channel
8. Plot BER-SNR and Bit Rate-SNR graphs for Flat Fading channel
9. Simulation of Okumura Outdoor Propagation Model.
10. Simulation of log normal shadowing radio propagation model.
11. Simulation of Walsh Hadamard Code.
12. Study distribution of downlink C/I due to different parameters.
13. Implement Direct Sequence Spread Spectrum modulation technique.
14. Design OFDM based Transmitter and Receiver for different channel environments.
15. Design OFDM system with 2x2, 2x4, 4x4 MIMO systems.

|  |  |  |
| --- | --- | --- |
| **Signature of the Lab In-charge**  **(Mr.B.Achyut, Asst.Prof.)** |  | **Signature of the HOD**  **(Dr. N. Venkateswara Rao, Prof.)** |

**1. Simulation of Friis Free space equation**

**Aim:-** To write a MATLAB code for simulation of Friis free space equation

**Software:-** MATLAB

**Code:-**

clc;

clear all;

close all;

PtdBm=52;

GtdBi=25;

GrdBi=15;

frequency=5\*10^9;

lambda=3\*10^8/frequency;

d =41935000\*(1:1:200);

L=1;

Pt=10^((PtdBm-30)/10);

Gt=10^(GtdBi/10);

Gr=10^(GrdBi/10);

Pr= Pt\*(Gt\*Gr\*lambda^2)./((4\*pi.\*d).^2\*L);

PrdBm=10\*log10(Pr)+30;

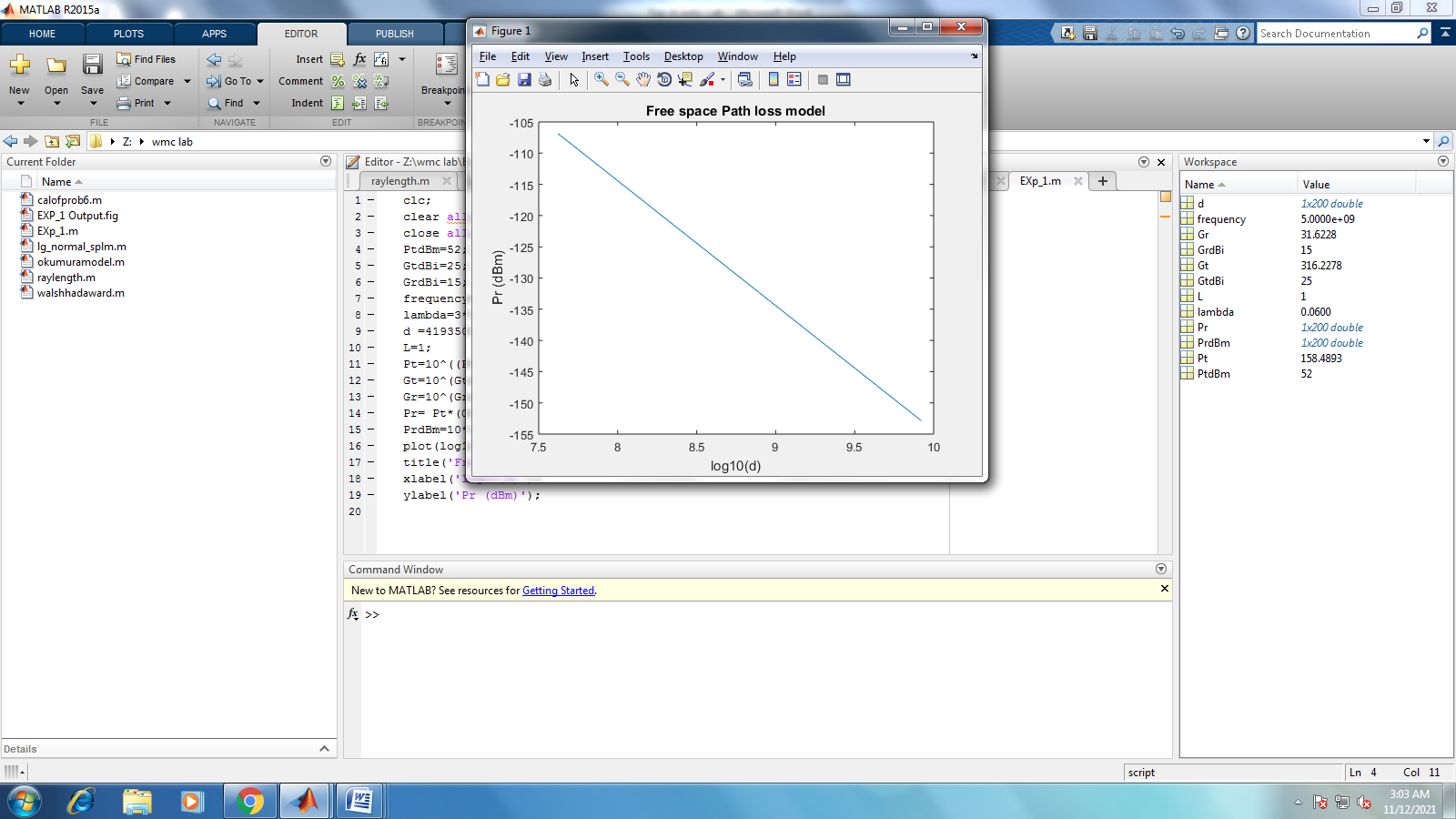
plot(log10(d),PrdBm);

title('Free space Path loss model')

xlabel('log10(d)');

ylabel('Pr (dBm)');

OUTPUT:-



**2. Simulation of RayLeigh fading channel model**

**Aim:-** To write a MATLAB code for simulation of RayLeigh fading channel model

**Software:-** MATLAB

**Code:-**

N=1000000;

variance = 0.2;

x = randn(1, N);

y = randn(1, N);

r = sqrt(variance\*(x.^2 + y.^2));

step = 0.1; range = 0:step:3;

h = hist(r, range);

approxPDF = h/(step\*sum(h));

theoretical = (range/variance).\*exp(-range.^2/(2\*variance));

plot(range, approxPDF,'b\*', range, theoretical,'r');

title('Simulated and Theoretical Rayleigh PDF for variance = 0.5')

legend('Simulated PDF', 'Theoretical PDF')

xlabel('r -?');

ylabel('P®-?');

grid;

theta = atan(y./x);

figure(2)

hist(theta);

[counts,range] = hist(theta,100);

step=range(2)-range(1);

approxPDF = counts/(step\*sum(counts));

hold on

plotHandle=plot(range, approxPDF,'r');

set(plotHandle,'LineWidth',3.5);

axis([-2 2 0 max(approxPDF)+0.2])

hold off

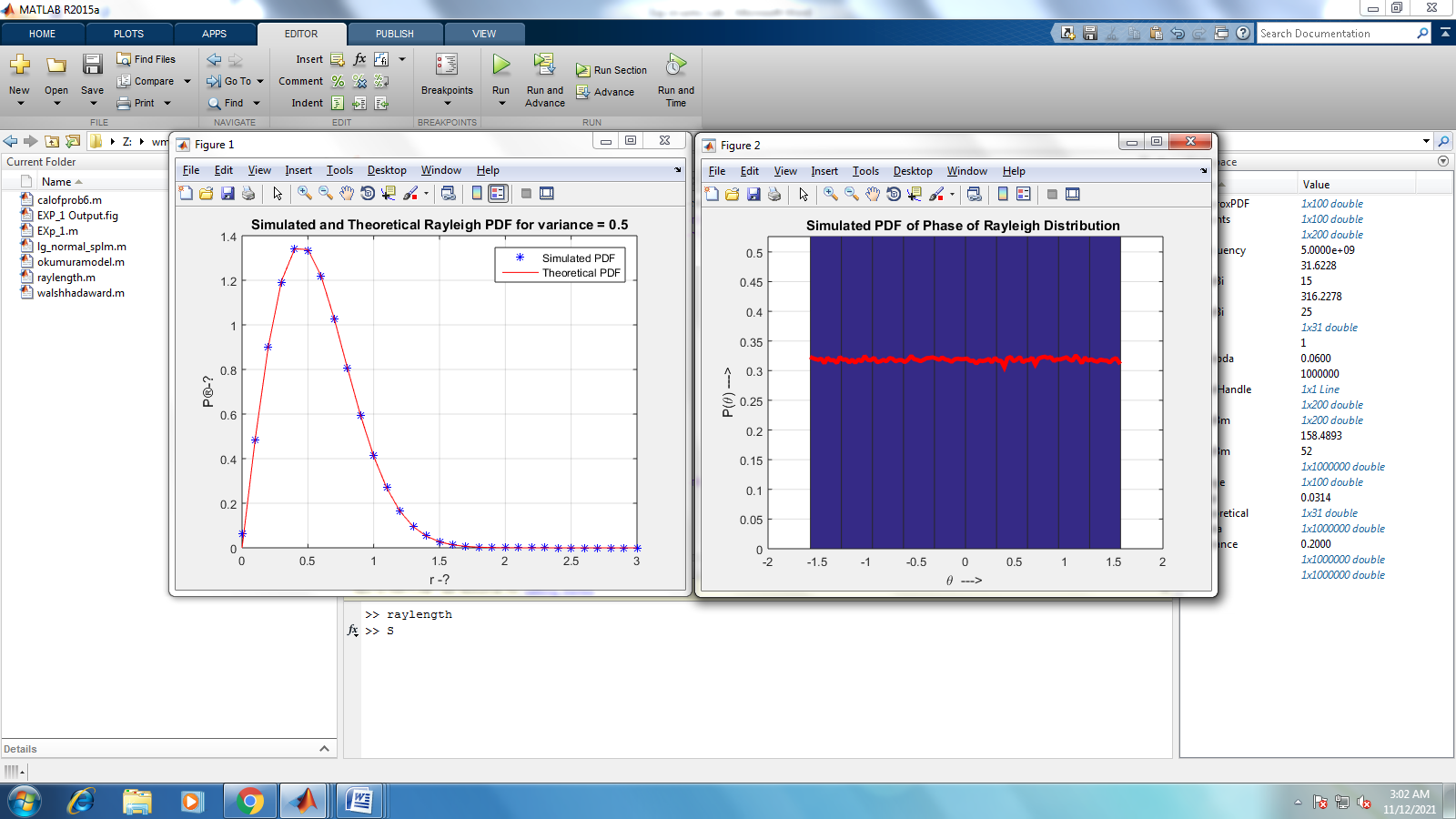
title('Simulated PDF of Phase of Rayleigh Distribution ');

xlabel('\theta --->');

ylabel('P(\theta) --->');

grid;

**OUTPUT:-**



**3. Simulation of Log normal/ Log distance shadowing path loss model**

**Aim:-** To write a MATLAB code for simulation of Friis free space equation

**Software:-** MATLAB

**Code:-**

clear all;

close all;

PtdBm=52;

GtdBi=25;

GrdBi=15;

frequency=1\*10^9;

d =41935000\*(1:1:500) ;

L=1;

Pt=10^((PtdBm-30)/10);

Gt=10^(GtdBi/10);

Gr=10^(GrdBi/10);

lambda=3\*10^8/frequency;

Pr= Pt\*(Gt\*Gr\*lambda^2)./((4\*pi.\*d).^2\*L);

X = sigma\*randn(size(Pr));

propLoss = Pr./Pt;

PLdBm = 10\*log10(propLoss)+10\*log10(X);

PrdBm=10\*log10(Pr)+30;

plot(log10(d),10\*log10(propLoss),'G','LineWidth',2);

title('Log Normal Shadowing model')

xlabel('log10(d)');

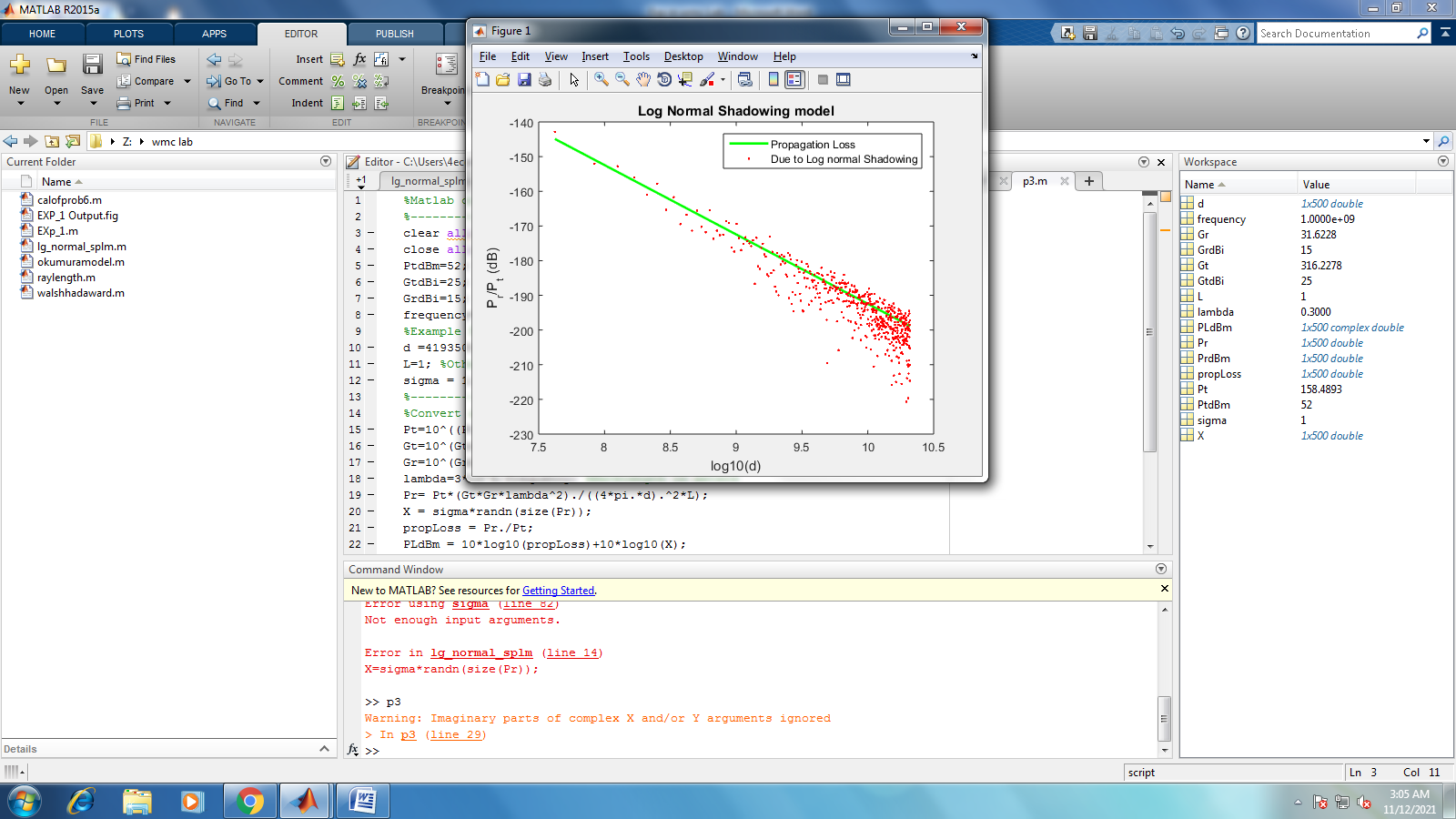
ylabel('P\_r/P\_t (dB)');

hold on;

plot(log10(d),PLdBm,'r.');

legend('Propagation Loss','Due to Log normal Shadowing');

OUTPUT:-



**4. Simulation of Okumura Outdoor Propagation Model**

**Aim:-** To write a MATLAB code for simulation of Friis free space equation

**Software:-** MATLAB

**Code:-**

clc;

clear all;

close all;

Hte=30:1:100;

Hre=input('Enter the receiver antenna height 3m<hre<10m : ');

d =input('Enter distance from base station 1Km<d<100Km : ');

f=input('Enter the frequency 150Mhz<f<1920Mhz : ');

c=3\*10^8;

lamda=(c)/(f\*10^6);

Lf = 10\*log((lamda^2)/((4\*pi)^2)\*d^2);

Amu = 35;

Garea = 9;

Ghte = 20\*log(Hte/200);

if(Hre>3)

Ghre = 20\*log(Hre/3);

else

Ghre = 10\*log(Hre/3);

end

L50 = Lf+Amu-Ghte-Ghre-Garea;

display('Propagation pathloss is : ');

disp(L50);

plot(Hte,L50,'LineWidth',1.5);

title('Okumura Model Analysis');

xlabel('Transmitter antenna Height (Km)');

ylabel('Propagation Path loss(dB) at 50 Km');

grid on;

**OUTPUT:-**

Enter the receiver antenna height 3m<hre<10m : 10

Enter distance from base station 1Km<d<100Km : 100

Enter the frequency 150Mhz<f<1920Mhz : 900

Propagation pathloss is :

Columns 1 through 11

59.3736 58.7178 58.0828 57.4674 56.8704 56.2906 55.7272 55.1792 54.6458 54.1263 53.6200

Columns 12 through 22

53.1261 52.6442 52.1736 51.7138 51.2643 50.8247 50.3946 49.9735 49.5612 49.1571 48.7611

Columns 23 through 33

48.3727 47.9917 47.6179 47.2509 46.8905 46.5365 46.1887 45.8468 45.5107 45.1801 44.8549

Columns 34 through 44

44.5349 44.2199 43.9098 43.6045 43.3037 43.0074 42.7154 42.4277 42.1440 41.8642 41.5884

Columns 45 through 55

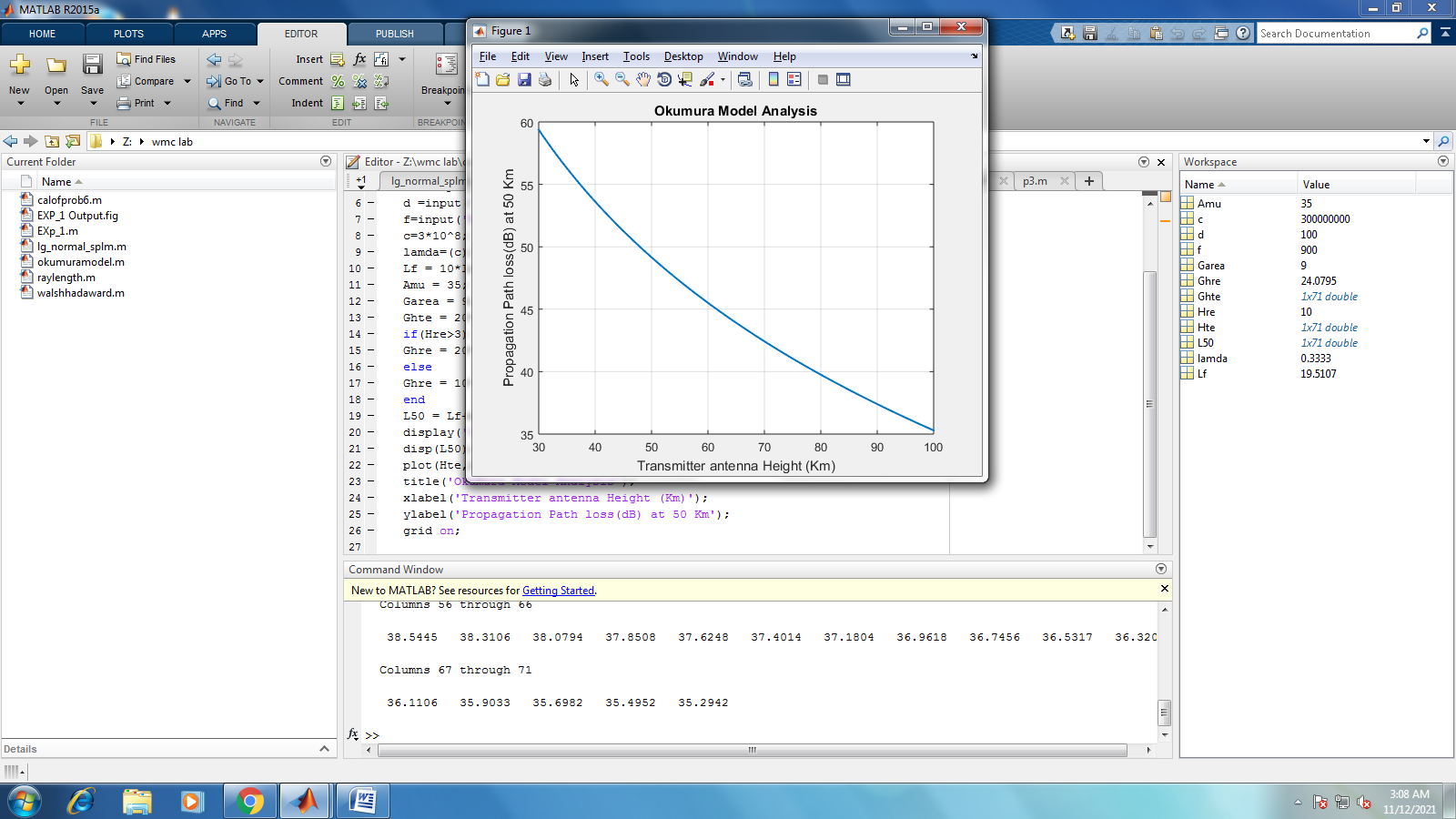
41.3163 41.0478 40.7829 40.5215 40.2634 40.0086 39.7570 39.5086 39.2632 39.0208 38.7812

Columns 56 through 66

38.5445 38.3106 38.0794 37.8508 37.6248 37.4014 37.1804 36.9618 36.7456 36.5317 36.3200

Columns 67 through 71

36.1106 35.9033 35.6982 35.4952 35.2942

****

**5. Simulation OF WALSH HADAMARD CODE**

**Aim:-** To write a MATLAB code for simulation of Friis free space equation

**Software:-** MATLAB

**Code:-**

clc;

clear all;

N=2;

H=[0 0;0 1];

codeSize=input('enter codeSize value');

if bitand(codeSize,codeSize-1)==0

while(N~=codeSize)

N=N\*2;

H=repmat(H,[2,2]);

[m,n]=size(H);

for i=m/2+1:m

for j=n/2+1:n

H(i,j)=~H(i,j);

end

end

end

else

disp('invalid codesize: code must be the power of 2');

end

disp(H);

**OUTPUT:-**

enter codeSize value8

0 0 0 0 0 0 0 0

0 1 0 1 0 1 0 1

0 0 1 1 0 0 1 1

0 1 1 0 0 1 1 0

0 0 0 0 1 1 1 1

0 1 0 1 1 0 1 0

0 0 1 1 1 1 0 0

0 1 1 0 1 0 0 1

**6. Calculation of probability that the received signal level crosses a certain sensitivity level.**

**Aim:-** To write a MATLAB code for simulation of Friis free space equation

**Software:-** MATLAB

**Code:-**

Pt=10\*10^-3;

f=1\*10^9;

c=3\*10^8;

lam=c/f;

d0=1;

K=(lam/(4\*pi\*d0))^2;

d=1.9;

gam=2;

PrFriss=Pt\*K\*(d0./d).^gam;

gam=4.5;

Pr=Pt\*K\*(d0/d)^gam;

vari=4

Pmin=10\*log10(Pr/4)

x=(Pmin-10\*log10(Pt)-10\*log10(K)+10\*gam\*log10(d/d0))/vari

Prob\_req=qfunc(x)

**OUTPUT:-**

>> calofprob6

vari =

4

Pmin =

-71.0063

x =

-1.5051

Prob\_req =

0.9339

**7. Study the effect of delay spread on frequency selectivity**

**Aim:-** To write a MATLAB code for simulation of Friis free space equation

**Software:-** MATLAB

**Code:-**

clear all

close all

clc

an=[1 0.3 -0.8 0.5 -0.4 0.2];

tn=[0 1 2 3 4 5;0 0.1 0.2 0.3 0.4 0.5]\*(10^-6);

dt=0.01\*10^-6; t=0:dt:15\*10^(-6); t=t(1:end-1);

st=zeros(1,length(t));

st(t<=5\*10^-6)=1;

for c=1:2

for i=1:length(an)

del=round(tn(c,i)/dt);

Ray(i,:)=an(i)\*[zeros(1,del) st(1:length(st)-del)];

end;

yt(c,:)=sum(Ray);

end

subplot(221)

plot(t\*10^6,st)

xlabel('time (\mu sec)');

ylabel('Tx signal');

axis([0 20 -1 2])

grid on

subplot(222)

plot(t\*10^6,yt(1,:));

xlabel('time (\mu sec)');

ylabel('Rx signal');

title('Large delay spread')

axis([0 20 -1 2])

grid on

subplot(224)

plot(t\*10^6,yt(2,:));

xlabel('time (\mu sec)');

ylabel('Rx signal');

title('Small delay spread')

axis([0 20 -1 2])

grid on

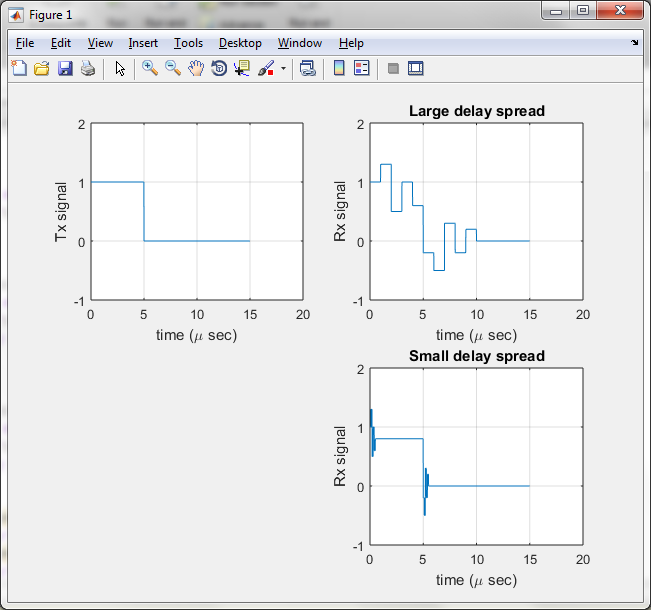
RMS\_delay\_spread\_1=sqrt(var(tn(1,:)))/(10^-6)

display('micro sec');

RMS\_delay\_spread\_2=sqrt(var(tn(2,:)))/(10^-6)

display('micro sec');

**OUTPUT:-**

****

**8. Plot BER-SNR and Bit Rate-SNR graphs for fading channel of SISO with no fading (AWGN)**

**Aim:-** To write a MATLAB code for simulation of Friis free space equation

**Software:-** MATLAB

**Code:-**

clear all

close all

clc

SNR=0:1:20;

SNRn=10.^(SNR/10);

N=800; % Number of bits

M=60; % Number of Experiments

fBER=zeros(1,length(SNR));

for mo=1:M

xbase=sign(randn(1,N)); %BPSK

Sigpow=1;

for ss=1:length(SNR)

% Assuming channel is known at the receiver

noipow=Sigpow/SNRn(ss);

% AWGN channel

ybase=xbase+sqrt(noipow)\*(randn(1,N));

ydet=(sign((ybase)));

err\_pos=find(ydet~=xbase);

BERa(ss)=length(err\_pos)/N;

end

fBER=fBER+BERa;

end

% Theoretical Approximate (Analytical) formulae

BERanaAWGN=0.5\*exp(-SNRn/2);

%Plotting

semilogy(SNR,fBER/M,'k','LineWidth',3)

hold on

semilogy(SNR,BERanaAWGN,'ko','LineWidth',3);

hold on

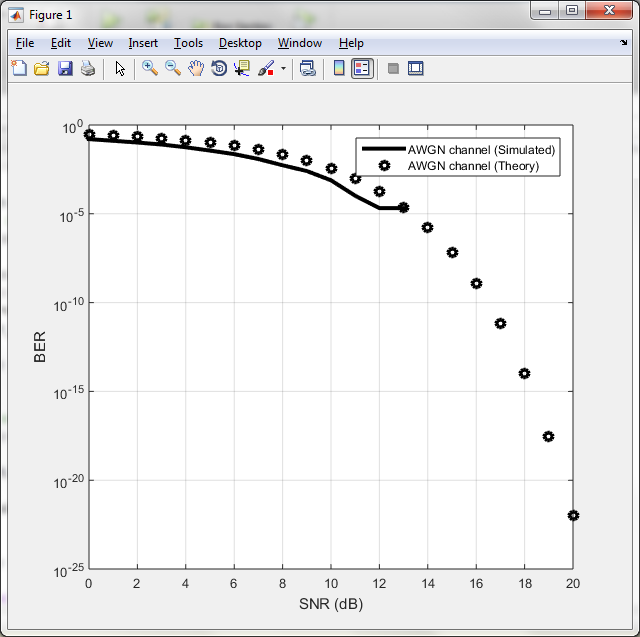
grid on

xlabel('SNR (dB)');

ylabel('BER');

legend('AWGN channel (Simulated)','AWGN channel (Theory)');

**OUTPUT:-**

****

**9. Plot BER-SNR and Bit Rate-SNR graphs for fading channel of SISO with Rayleigh Flat Fading Channel**

**Aim:-** To write a MATLAB code for simulation of Friis free space equation

**Software:-** MATLAB

**Code:-**

clear all

close all

clc

SNR=0:4:40;

SNRn=10.^(SNR/10);

N=800; %Number of bits

M=100; % Number of Experiments

fBER=zeros(1,length(SNR));

for mo=1:M

xbase=sign(randn(1,N)); %BPSK

cc=0.707\*(randn(1,N)+sqrt(-1)\*randn(1,N));

Sigpow=1;

for ss=1:length(SNR) %Assuming channel is known at the receiver

noipow=Sigpow/SNRn(ss); % AWGN channel

ybase=xbase.\*cc+sqrt(noipow)\*(randn(1,N));

yfa=conj(cc).\*ybase; ydet=(sign(real(yfa)));

err\_pos=find(ydet~=xbase);

BERa(ss)=length(err\_pos)/N;

end

fBER=fBER+BERa;

end

% Theoretical Approximate (Analytical) formulae

BERana=1./(2\*SNRn); %Plotting

semilogy(SNR,fBER/M,'k','LineWidth',3)

hold on

semilogy(SNR,BERana,'ko','LineWidth',3);

hold on

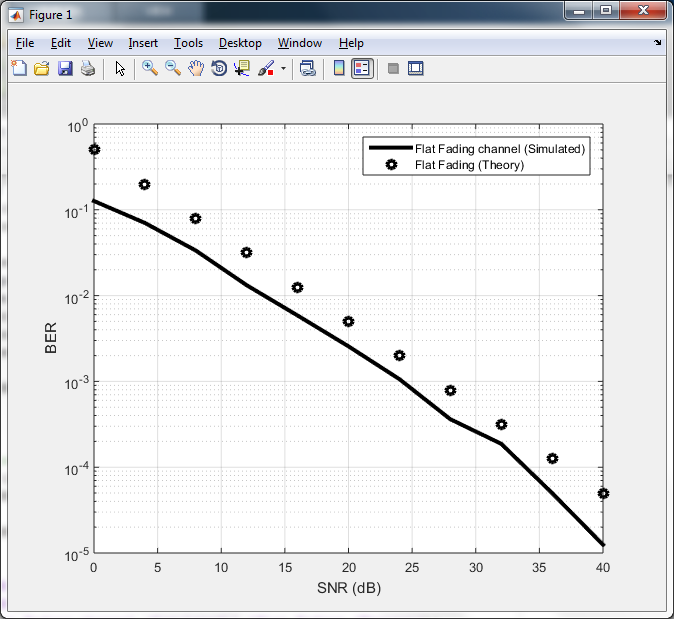
grid on

xlabel('SNR (dB)');

ylabel('BER');

legend('Flat Fading channel (Simulated)','Flat Fading (Theory)');

**OUTPUT:-**

****

**10. Implementation of Direct-Sequence Spread Spectrum modulation technique**

**Aim:-** To write a MATLAB code for simulation of Friis free space equation

**Software:-** MATLAB

**Code:-**

**M\_sequence**;

D=input('Enter the input data sequence: ');

D(D==0)= -1;

M\_seq(M\_seq==0)=-1;

td=D;

ld=length(D);

td(1:N)=D(1)\*M\_seq;

for i=2:ld

td(1,length(td)+1:length(td)+N)=D(i)\*M\_seq;

end;

Tb=1;

Tpn=Tb/N;

fc=3\*N;

fs=150\*fc;

ts=1/fs;

k=0;

for i=1:N

if M\_seq(i)==1

M\_seq1(1,k+1:k+fs/N)=1;

k=length(M\_seq1);

else

M\_seq1(1,k+1:k+fs/N)=-1;

k=length(M\_seq1);

end;

end;

k=0;

for i=1:ld

if D(i)==1

D1(1,k+1:k+fs)=1;

k=length(D1);

else

D1(1,k+1:k+fs)=-1;

k=length(D1);

end;

end;

k=0;

for i=1:ld\*N

if td(i)==1

td1(1,k+1:k+fs/N)=1;

k=length(td1);

else

td1(1,k+1:k+fs/N)=-1;

k=length(td1);

end;

end;

t=(0:1:fs\*ld-1)\*ts;

tm=(0:1:fs-1)\*ts;

c=cos(2\*pi\*fc\*t);

BPSK=c.\*td1;

figure();

subplot(4,1,1);

plot(t,D1,'b','linewidth',1);

grid on; %Shows grid lines in the graph

legend('NRZ coded Data signal '); %Shows graph names given

subplot(4,1,2);

plot(tm,M\_seq1,'b','linewidth',1);

grid on; %Shows grid lines in the graph

legend('NRZ coded M\_sequence '); %Shows graph names given

subplot(4,1,3);

plot(t,td1,'b','linewidth',1);

grid on; %Shows grid lines in the graph

legend('NRZ coded spread spectrum modulated signal'); %Shows graph names given

subplot(4,1,4);

plot(t,BPSK,'b','linewidth',1);

grid on; %Shows grid lines in the graph

legend('DSSS\_BPSK Modulated Signal'); %Shows graph names given

**M\_SEQUENCE:**

clc; %Clears the command window

clear all; %clears all the varibles used in previous program

close all; %closes all pop up windows like graphs opened in previous window

r=input('enter the length of shiftregister r= '); %reads the shift register length and stores in variable r

N=2^r-1; %calculate PN\_sequence length

tp=input('enter the valid tap array :');

in=input('enter the sequence of input data array of length r : ');

if r~=length(in)

fprintf('\n \n error : length of input data array must be equal to the length of shift registers: \n \n');

else

out=in;

for k=1:N

t\_out=out(tp(1));

for i=1:length(tp)-1

t\_out=xor(t\_out,out(tp(i+1)));

end;

tr1=r;

M\_seq(k)=out(r);

for j=1:r-1

out(tr1)=d\_ff(out(tr1-1));

tr1=tr1-1;

end;

out(1)=t\_out;

if k~=N

if out==in

fprintf('\n \n error : given tap input array is not valid : \n \n');

break;

end;

end;

if k==N

if out==in

fprintf('lenth of the m\_sequence N = %d \n',N);

fprintf('M\_sequence is : \n');

disp(M\_seq);

corre=correlation(M\_seq,N);

t=-2\*N:1:2\*N;

grid on;

plot(t,corre);

grid on;axis([-2\*N-1 2\*N+1 -1 1.5]);

title('autocorrelation of M-sequence : ');

xlabel('timeperiod');

ylabel('amplitude');

else

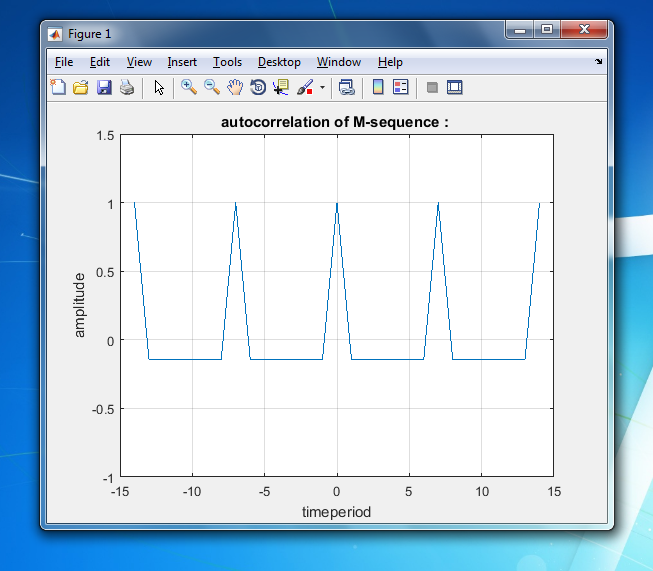
fprintf(' \n \n error : given tap input array is not valid :\n \n');

end;

end;

end;

end;



OUTPUT:

enter the length of shiftregister r= 3

enter the valid tap array :[3 1]

enter the sequence of input data array of length r : [1 0 0]

lenth of the m\_sequence N = 7

M\_sequence is :

0 0 1 1 1 0 1

Enter the input data sequence: [1001010]

