Wireless Communication (IPCC)

Course Code: EC71 **Credits: 1:0:0**
Prerequisites: Communication Systems **Contact Hours: 14**
Course Coordinator(s): T D Senthilkumar & Flory Francis

List of Experiments

1. Bit Error Rate (BER) for different digital modulation schemes under the AWGN and fading channel
2. Study of large-scale path loss models
3. Bit error rate analysis of digital communication receivers with Maximal Ratio Combining (MRC) receive diversity in frequency-flat and slowly varying fading channel
4. Bit error rate analysis of digital communication receivers with Equal Gain Combining (EGC) receive diversity in frequency-flat and slowly varying fading channel
5. BER performance of 2x1 MISO under the Rayleigh fading channel
6. Analyze the performance of the multicarrier modulation scheme
7. OFDM transmitter and receiver in AWGN channel

Experiment 1

Analyze Bit Error Rate (BER) performance for BPSK signals over AWGN and Rayleigh channel. Compare the results with theoretical results.

Aim

Write a code to compute Bit Error Rate (BER) performance for BPSK signals over AWGN channel. Compare the results with theoretical results**.**

Algorithm

Step1: Generate random binary data

Step2: Construct the constellation symbols for the generated binary sequence

Step3: Generate complex normal random variable for additive white Gaussian noise (AWGN)

Step4: Add AWGN noise with the transmitted symbol

Step5: Construct maximum likelihood (ML) receiver to decode the symbol from the received signal

Step6: Compare the decoded sequence with the original sequence to estimate the number of errors

Step7: Average bit error rate (ABER) can be calculated by computing the ratio between total number of errors and total number of bits

Repeat the procedure for different modulation schemes and plot the performance curve ABER Vs SNR

```
clc;
close all;
clear all;
% Number of information bits
m= 10^5;
%Range of SNR values
snr_dB = [0:1:20];
for j=1:1:length(snr_dB)
n_err = 0;
n_bits = 0;
while n_err < 100
% Generate sequence of binary bits
inf_bits=round(rand(1,m));
% BPSK modulator
x=2*(inf_bits-0.5);
% Noise variance
N0=1/10^(snr_dB(j)/10);
% Send over Gaussian Link to the receiver
y=x + sqrt(N0/2)*(randn(1,length(x))+i*randn(1,length(x)));
% Decision making at the Receiver
est_bits=y > 0;
% Calculate Bit Errors
diff=inf_bits-est_bits;
n_err=n_err+sum(abs(diff));
n_bits=n_bits+length(inf_bits);
end
% Calculate Bit Error Rate
BER(j)=n_err/n_bits;
end
% AWGN Theoretical BER
theoryBerAWGN=0.5*erfc(sqrt(10.^(snr_dB/10)));
```

```
semilogy(snr_dB,BER,'or','LineWidth',2);
hold on;
semilogy(snr_dB,theoryBerAWGN,'blad-','LineWidth',2);
legend(‘AWGN Simulated', 'AWGN Theoretical');
axis([0 20 10^-5 0.5]);
xlabel('SNR (dB)');
ylabel('BER');
grid on;
```

Experiment 2

Analyze Bit Error Rate (BER) performance for BPSK signals over Rayleigh fading channel. Compare the results with theoretical results.

Aim

Write a code to compute Bit Error Rate (BER) performance for BPSK signals over AWGN and Rayleigh channel. Compare the results with theoretical results**.**

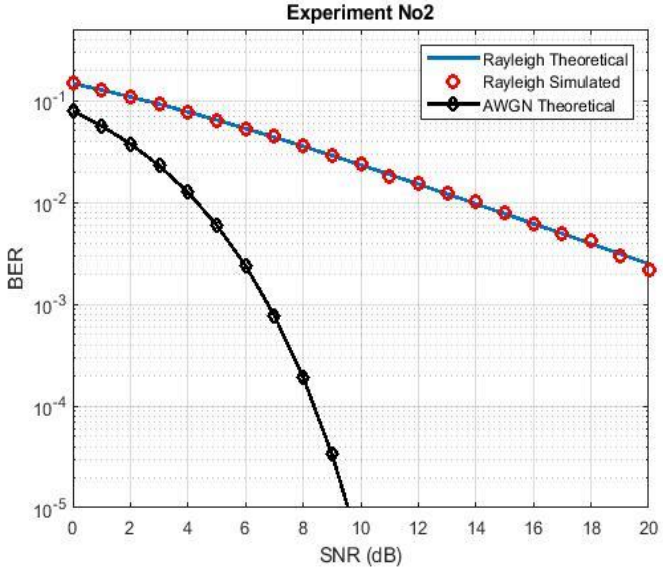
Algorithm

Step1: Generate random binary data
Step2: Construct the constellation symbols for the generated binary sequence
Step3: Generate complex channel fading coefficient
Step4: Generate complex normal random variable for additive white Gaussian noise (AWGN)
Step5: Add AWGN noise with the transmitted symbol
Step6: Construct maximum likelihood (ML) receiver to decode the symbol from the received signal
Step7: Compare the decoded sequence with the original sequence to estimate the number of errors
Step8: Average bit error rate (ABER) can be calculated by computing the ratio between total number of errors and total number of bits

Repeat the procedure for different modulation schemes and plot the performance curve ABER Vs SNR

```
clc;
close all;
clear all;
% Number of information bits
m= 10^5;
%Range of SNR values
snr_dB = [0:1:20];
for j=1:1:length(snr_dB)
n_err = 0;
n_bits = 0;
while n_err < 100
% Generate sequence of binary bits
inf_bits=round(rand(1,m));
% BPSK modulator
x=-2*(inf_bits-0.5);
% Noise variance
N0=1/10^(snr_dB(j)/10);
% Rayleigh channel fading
h=1/sqrt(2)*[randn(1,length(x)) + i*randn(1,length(x))];
% Send over Gaussian Link to the receiver
y=h.*x + sqrt(N0/2)*(randn(1,length(x))+i*randn(1,length(x)));
% decision metric
y=y./h;
% Decision making at the Receiver
est_bits=y < 0;
% Calculate Bit Errors
diff=inf_bits-est_bits;
n_err=n_err+sum(abs(diff));
n_bits=n_bits+length(inf_bits);
end
```

```
% Calculate Bit Error Rate
BER(j)=n_err/n_bits;
end
% Rayleigh Theoretical BER
snr = 10.^(snr_dB/10);
theoryBer=0.5.*(1-sqrt(snr./(snr+1)));
% AWGN Theoretical BER
theoryBerAWGN=0.5*erfc(sqrt(10.^(snr_dB/10)));
semilogy(snr_dB,theoryBer,'-','LineWidth',2);
hold on;
semilogy(snr_dB,BER,'or','LineWidth',2);
hold on;
semilogy(snr_dB,theoryBerAWGN,'blad-','LineWidth',2);
legend('Rayleigh Theoretical','Rayleigh Simulated', 'AWGN Theoretical');
axis([0 20 10^-5 0.5]);
xlabel('SNR (dB)');
ylabel('BER');
grid on;
```



Experiment 3

Study of Log-Distance path loss propagation model

Aim

a) write a matlab program to calculate the path loss for log distance path loss indoor propagation mode.
b) To simulate the log distance path loss model using Matlab.
c) To obtain graphical representation by varying various parameters and by considering various terrains.

Theory:

Log-distance Path Loss Model

Log distance path loss model is a generic model and an extension to Friis Free space model. It is use to predict the propagation loss for a wide range of environments, whereas, the Friis Free space model is restricted to unobstructed clear path between transmitter and the receiver.

In the field region of the transmitter (d≥df), if PL(d_0) is the path loss measured in dB at a distance d_0 from the transmitter, them the path loss (the loss in signal power measure in dB when moving from distance d_0 to d at an arbitrary distance $d>d_0$ is given by

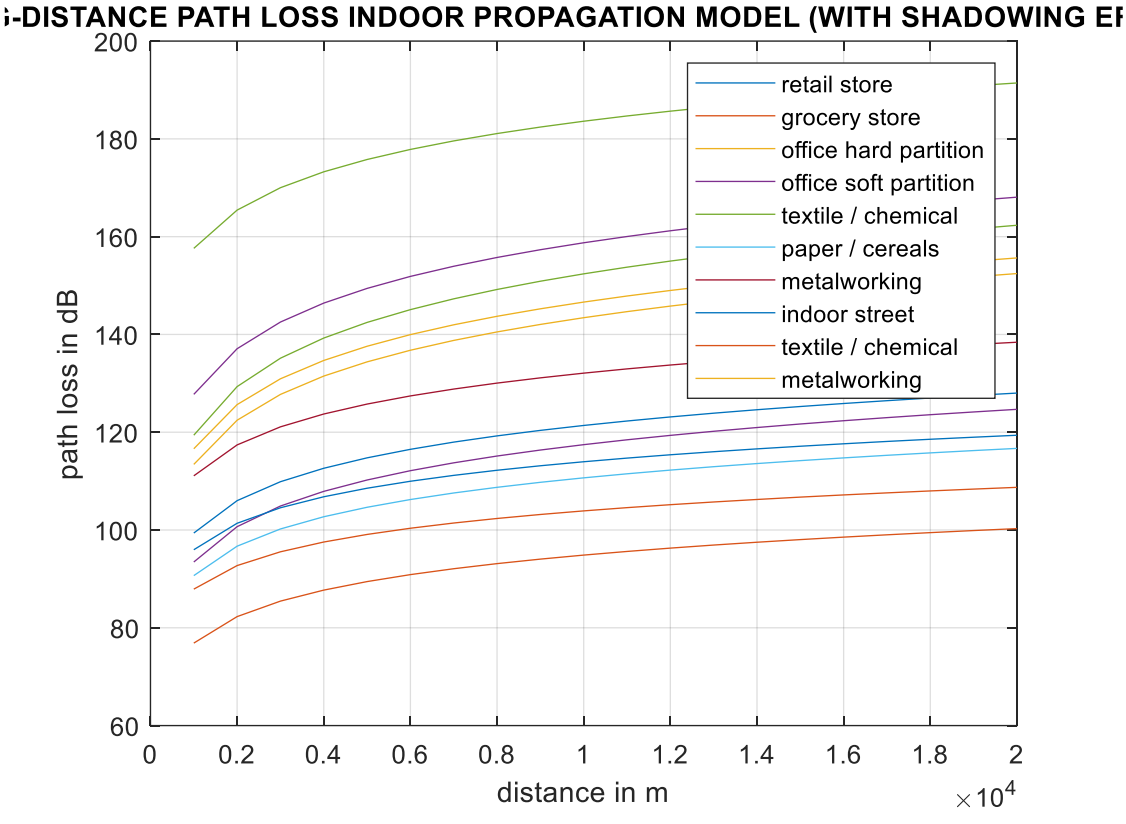
$$PL_{d_0 \to d}(dB) = PL(d_0) + 10n\log_{10}\left(\frac{d}{d_0}\right) + X \qquad d_f \le d_0 \le d$$

PL(d_0) = path loss dB at a distance d_0, PL($d>d_0$) = path loss in dB at arbitrary distance d, n= path loss exponent, x=zero-mean Gaussian distributed random variable (in dB) with standard deviation – σ, this variable is used only when there is a shadowing effect. If there is no shadowing effect, then this variable is zero, taking log of the Normal (Gaussian)-variable results in “Log-Normal” fading.

%LOG-DISTANCE PATH LOSS INDOOR PROPAGATION MODEL with shadow

```
clc
clear all;
close all;
d0=input('enter the reference distance:');
d=1000:1000:20000;
n= [2.2 1.8 3.0 2.4 2.6 2.0 2.1 1.8 1.6 3.0 3.1 3.3];
f= [914 914 1500 900 1900 1300 4000 1300 1300 900 4000 1300];
sigma= [8.7 5.2 7.0 9.6 14.1 3.0 7.0 6.0 5.8 7.0 9.7 6.8];

for i=1:12
   lambda(i)=3e8/(f(i)*10^6);
   PL_d0(i)=-10*log10((lambda(i)^2)/((4*pi*d0)^2));
   X(i)=sigma(i)*randn(size(PL_d0(i)));
   disp(randn(size(PL_d0(i))));
end
for i=1:12
for j=1:20
PL(i,j)=PL_d0(i)+10*n(i)*log10(d(j)/d0)+X(i);
```

```
end
end

%DISTANCE VS PATH LOSS

plot (d,PL);
legend ('retail store', 'grocery store', 'office hard partition', 'office soft partition', 'textile / chemical', 'paper / cereals', 'metalworking', 'indoor street', 'textile / chemical', 'metalworking');
xlabel ('distance in m');
ylabel ('path loss in dB');
title ('LOG-DISTANCE PATH LOSS INDOOR PROPAGATION MODEL (WITH SHADOWING EFFECT');
grid on;

enter the reference distance:30
    3.0349

   -0.0631

   -0.2050

    1.4897

    1.4172

   -1.2075

    1.6302

    1.0347

   -0.3034

   -0.7873

   -1.1471

   -0.8095
```



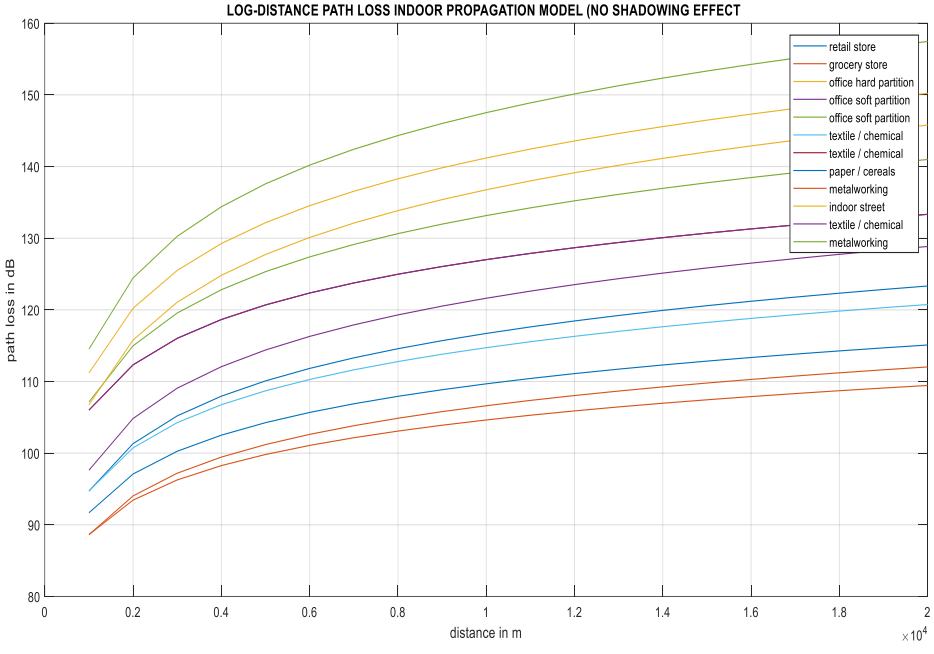
%LOG-DISTANCE PATH LOSS INDOOR PROPAGATION MODEL

```
clc
clear all;
close all;
d0=input('enter the reference distance:');
d=1000:1000:20000;
n= [2.2 1.8 3.0 2.4 2.6 2.0 2.1 1.8 1.6 3.0 2.1 3.3];
f= [914 914 1500 900 1900 1300 4000 1300 1300 900 4000 1300];
X= 0;
for i=1:12
lambda(i)=3e8/(f(i)*10^6);
PL_d0(i)=-10*log10((lambda(i)^2)/((4*pi*d0)^2))+X;
end
disp('PL_d0->d(dB)=');
for i=1:12
for j=1:20
PL(i,j)= PL_d0(i)+10*n(i)*log10(d(j)/d0);
end
disp(PL(i));
end
```

```
%DISTANCE VS PATH LOSS

plot (d, PL);
legend ('retail store', 'grocery store', 'office hard partition', 'office soft partition', 'office soft partition', 'textile / chemical', 'textile / chemical', 'paper / cereals', 'metalworking', 'indoor street', 'textile / chemical', 'metalworking');
xlabel ('distance in m');
ylabel ('path loss in dB');
title ('LOG-DISTANCE PATH LOSS INDOOR PROPAGATION MODEL (NO SHADOWING EFFECT');
grid on;
```

Output:

```
enter the reference distance:30

    1.1174

    0.0326

    1.1006

    0.0859

   -0.7423

    2.3505

    0.7481

    0.8886

   -1.4023

    0.4882

   -0.1961

    0.2916
```


LOG-DISTANCE PATH LOSS INDOOR PROPAGATION MODEL (NO SHADOWING EFFECT
path loss in dB
distance in m
×10^4
retail store
grocery store
office hard partition
office soft partition
office soft partition
textile / chemical
textile / chemical
paper / cereals
metalworking
indoor street
textile / chemical
metalworking

Experiment 4

Study of HATA propagation model.

Aim:

a) Write a matlab program to calculate the pathloss for HATA outdoor propagation model
b) Simulate the Hata path loss model using matlab
c) Obtain the graphical representation by varying various parameters and by considering the various terrains.

Theory: Hata pathloss model computes the path loss as a function of the transmits and receive antenna heights, path distance,radio frequency and type of clusters. We measure the path loss in db in different areas like rural, urban and suburban with the help of propagation path loss model.one of the most widely used outdoor radio propagation model for signal prediction in urban areas is Hata model. This model is applicable for frequencies in the range of 150MHz to 1500 MHz and the distance between 1km to 20 km. It can be used for base station antenna heights ranging from 30 to 200 mtrs. The standard formulas for medium path loss in urban area is given by

$L50(urban)(db)=69.55+26.16*\log_{10}(f_c)-13.82*\log_{10}(hte)-a(hre)+(44.9-6.55*\log10(hte))*\log10(d);$

Where fc is the frequency in hz
d is the path distance in kms.
hte is the effective transmitter(base station) height in mtrs
hre is the effective receiver(mobile) antenna height in mtrs

Table 2.1: parameters of mobile antenna height correction facors for different types of areas in Hata model

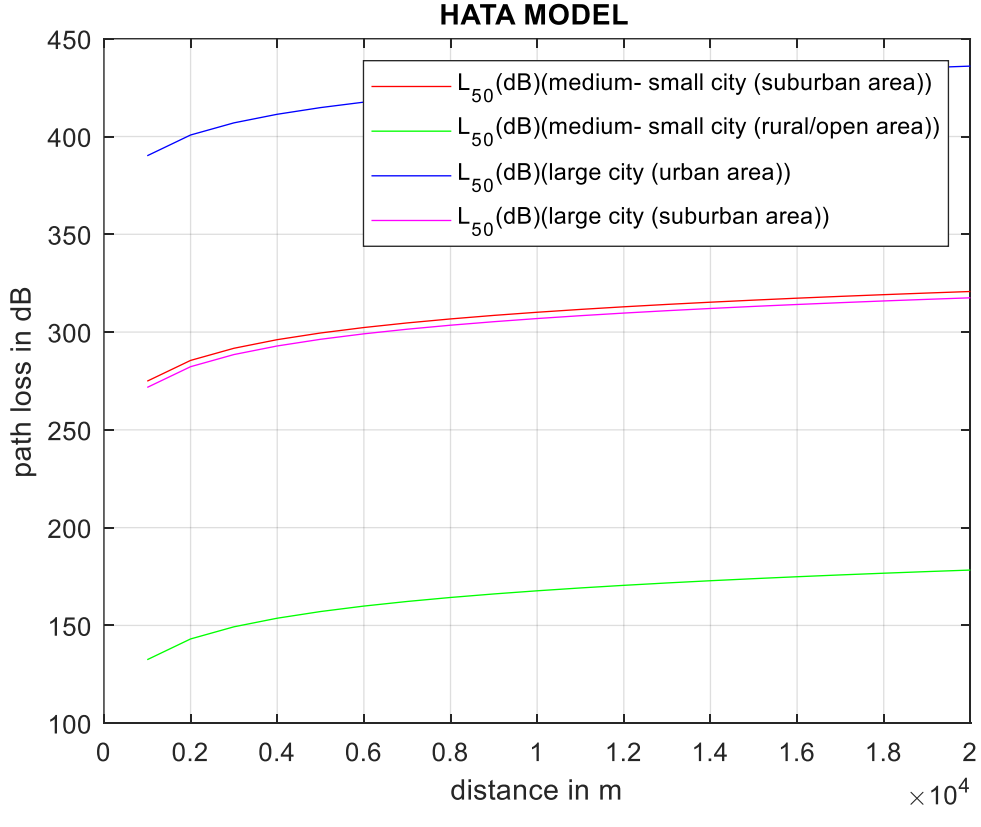
a(hre)	Type of area
ahre =(1.1*log10(f)-0.7)*hre-(1.56*log10(f)-0.8)	Medium small city
ahre=8.29*(log10(1.54*hre)).*(log10(1.54*hre))-1.1	Large city fc<300M Hz

L50 db	Type of area
2*log10(f/28)*log10(f/28))-5.4	Suburban area

```
%HATA Model for distance varied
clc
clear all;
close all;
hte=input('enter the tx height:');
hre=input('enter the rx height');
d=1000:1000:20000;
f=input('enter the frequency:');
% MEDIUM-SMALL CITY (SUB URBAN AREA)
ahre_1=(1.1*log10(f)-0.7)*hre-(1.56*log10(f)-0.8);
L50_1=69.55+26.16*log10(f)-13.82*log10(hte)+(44.9-6.55*log10(hte))*log10(d)-ahre_1;
L50_dB_1=L50_1-(2*log10(f/28)*log10(f/28))-5.4;
% MEDIUM SMALL CITY RURAL / OPEN AREA
ahre_2=(1.1*log10(f)-0.7)*hre-(1.56*log10(f)-0.8);
L50_2=69.55+26.16*log10(f)-13.82*log10(hte)+(44.9-6.55*log10(hte))*log10(d)-ahre_2;
L50_dB_2=L50_2-(4.78*log10(f)*log10(f))+18.33*log10(f)-40.98;
% LARGE CITY URBAN AND SUB-URBAN AREA
```

```
if f <3000000000
ahre_3=8.29*(log10(1.54*hre))*(log10(1.54*hre))-1.1;
else
ahre_3=3.2*(log10(11.75*hre))*(log10(11.75*hre))-4.97;
end
L50_3=69.55+26.16*log10(f)-13.82*log10(hte)+(44.9-6.55*log10(hte))*log10(d)-ahre_3;
L50_dB_3=L50_3-(2*log10(f/28)*log10(f/28))-5.4;

% Distance vs path loss
figure
plot (d,L50_dB_1,'r',d,L50_dB_2,'g',d,L50_3,'b',d,L50_dB_3,'m');
legend ('L_5_0(dB)(medium- small city (suburban area))', 'L_5_0(dB)(medium- small city (rural/open
area))', 'L_5_0(dB)(large city (urban area))', 'L_5_0(dB)(large city (suburban area))');
xlabel ('distance in m');
ylabel ('path loss in dB');
title ('HATA MODEL');
grid on;
```

output:
enter the tx height:30
enter the rx height1
enter the frequency:925000000

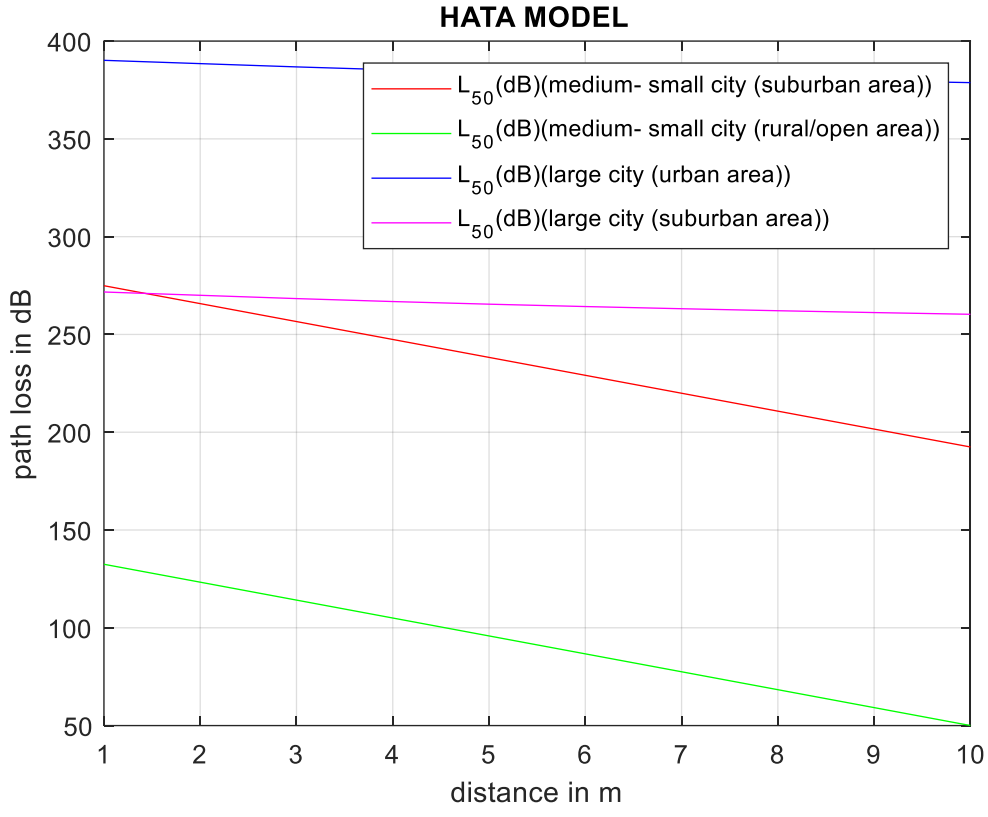



```
%HATA MODEL Receiver height varied
clc
clear all;
close all;
hte = input ('enter the tx height:');
hre = 1:10;
```

```
d=input ('enter the distance between tx and rx :');
f= input('enter the frequency:');

% MEDIUM-SMALL CITY (SUB URBAN AREA)

ahre_1=(1.1*log10(f)-0.7)*hre-(1.56*log10(f)-0.8);
L50_1=69.55+26.16*log10(f)-13.82*log10(hte)+(44.9-6.55*log10(hte))*log10(d)-ahre_1;
L50_dB_1=L50_1-(2*log10(f/28)*log10(f/28))-5.4;

% MEDIUM SMALL CITY RURAL / OPEN AREA
ahre_2=(1.1*log10(f)-0.7)*hre-(1.56*log10(f)-0.8);
L50_2=69.55+26.16*log10(f)-13.82*log10(hte)+(44.9-6.55*log10(hte))*log10(d)-ahre_2;
L50_dB_2=L50_2-(4.78*log10(f)*log10(f))+18.33*log10(f)-40.98;

% LARGE CITY URBAN AND SUB-URBAN AREA
if f <3000000000
ahre_3=8.29*(log10(1.54*hre)).*(log10(1.54*hre))-1.1;
else
ahre_3=3.2*(log10(11.75*hre)).*(log10(11.75*hre))-4.97;
end
L50_3=69.55+26.16*log10(f)-13.82*log10(hte)+(44.9-6.55*log10(hte))*log10(d)-ahre_3;
L50_dB_3=L50_3-(2*log10(f/28)*log10(f/28))-5.4;

% rx height vs path loss
figure
plot (hre, L50_dB_1,'r', hre, L50_dB_2,'g',hre, L50_3,'b',hre, L50_dB_3,'m');
legend ('L_5_0(dB)(medium- small city (suburban area))', 'L_5_0(dB)(medium- small city (rural/open
area))', 'L_5_0(dB)(large city (urban area))', 'L_5_0(dB)(large city (suburban area))');
xlabel ('distance in m');
ylabel ('path loss in dB');
title ('HATA MODEL');
grid on;
```

Output:

```
enter the tx height:30
enter the distance between tx and rx :1000
enter the frequency:925000000
```

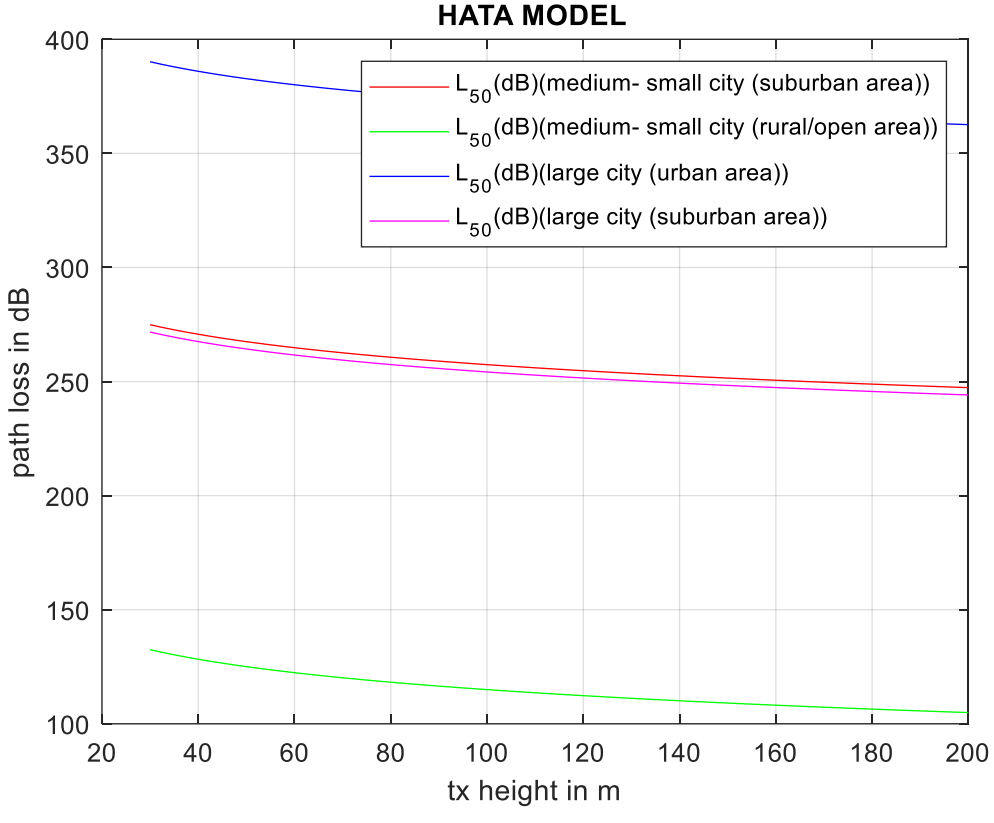



```
%HATA MODEL transmitter height varied
clc
clear all;
close all;
hre = input ('enter the rx height:');
hte = 30:200;
d=input ('enter the distance between tx and rx :');
f= input('enter the frequency:');

% MEDIUM-SMALL CITY (SUB URBAN AREA)

ahre_1=(1.1*log10(f)-0.7)*hre-(1.56*log10(f)-0.8);
L50_1=69.55+26.16*log10(f)-13.82*log10(hte)+(44.9-6.55*log10(hte))*log10(d)-ahre_1;
L50_dB_1=L50_1-(2*log10(f/28)*log10(f/28))-5.4;

% MEDIUM SMALL CITY RURAL / OPEN AREA
ahre_2=(1.1*log10(f)-0.7)*hre-(1.56*log10(f)-0.8);
L50_2=69.55+26.16*log10(f)-13.82*log10(hte)+(44.9-6.55*log10(hte))*log10(d)-ahre_2;
L50_dB_2=L50_2-(4.78*log10(f)*log10(f))+18.33*log10(f)-40.98;

% LARGE CITY URBAN AND SUB-URBAN AREA
if f <3000000000
ahre_3=8.29*(log10(1.54*hre)).*(log10(1.54*hre))-1.1;
else
ahre_3=3.2*(log10(11.75*hre)).*(log10(11.75*hre))-4.97;
end
L50_3=69.55+26.16*log10(f)-13.82*log10(hte)+(44.9-6.55*log10(hte))*log10(d)-ahre_3;
L50_dB_3=L50_3-(2*log10(f/28)*log10(f/28))-5.4;
```

```
% rx height vs path loss
figure
plot (hte, L50_dB_1,'r', hte, L50_dB_2,'g',hte, L50_3,'b',hte, L50_dB_3,'m');
legend ('L_5_0(dB)(medium- small city (suburban area))', 'L_5_0(dB)(medium- small city (rural/open
area))', 'L_5_0(dB)(large city (urban area))', 'L_5_0(dB)(large city (suburban area))');
xlabel ('tx height in m');
ylabel ('path loss in dB');
title ('HATA MODEL');
grid on;
```

Output:

```
enter the rx height:1
enter the distance between tx and rx :1000
enter the frequency:925000000
```



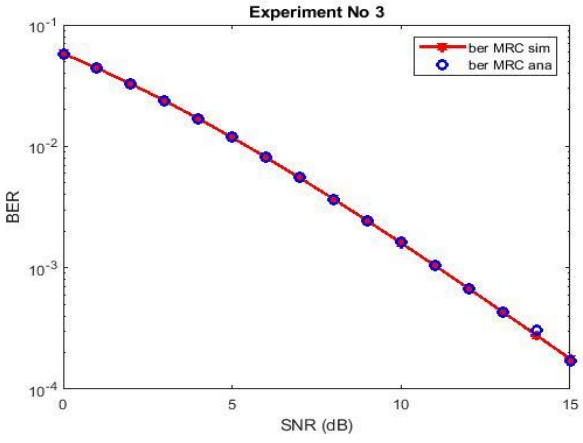
Experiment 5

Bit error rate analysis of digital communication receivers with Maximal Ratio Combining (MRC) receives diversity in frequency-flat and slowly varying fading channel.

Aim

Bit error rate analysis of digital communication receivers with Maximal Ratio Combining (MRC) receive diversity in frequency-flat and slowly varying fading channel.

```
clc;
clear all;
close all;
%%%%%%%%%%%%% Initialization %%%%%%%%%%%%%%%%%%%%%%%%
N=5; % Number of trials
m = 10^6; %Number of bits in each trial
ip = rand(1,m)>0.5; % Generated bits
BPSK = 2*ip-1; % Generated BPSK symbols
snr_dB = 0:1:15; % range of snr values
snr = 10.^(snr_dB/10); % snr value in the normal scale
L=2; % Number of diversity branches
% theoretical BER value for MRC combiner with 2 diversity branches
p_R_MRC = 1/2 - 1/2*(1+1./snr).^(-1/2);
ber_MRC_ana = p_R_MRC.^2.*(1+2*(1-p_R_MRC));
%%%%%%%%%% Receive MRC one by Two System %%%%%%%%%%%%%%%%%%
n_err=zeros(1,length(snr_dB)); % Initialize the bit error counter
for p = 1:N
for q = 1:length(snr_dB)
% Generate white noise samples
No = 1/sqrt(2)*[randn(L,m) + 1j*randn(L,m)];
% Generate channel coefficient
h = 1/sqrt(2)*[randn(L,m) + 1j*randn(L,m)];
symbol = kron(ones(L,1),BPSK); % array of symbols
rec_vector = h.*symbol + 10^(-snr_dB(q)/20)*No;% received symbol
% Decision metric
dec_metric = sum(conj(h).*rec_vector,1)./sum(h.*conj(h),1);
ip_hat = real(dec_metric)>0; % Estimated symbol
n_err(q) = n_err(q)+size(find([ip- ip_hat]),2); % compare input and estimated symbols
end
end
ber_MRC_sim = n_err/(N*m);
semilogy(snr_dB,ber_MRC_ana,'-r*','LineWidth',2)
hold on;
semilogy(snr_dB,ber_MRC_sim,'ob','LineWidth',2)
legend('ber MRC sim', 'ber MRC ana');
xlabel('SNR (dB)');
ylabel('BER');
```


Experiment No 3
ber MRC sim
ber MRC ana
BER
SNR (dB)
10^-1
10^-2
10^-3
10^-4
0
5
10
15

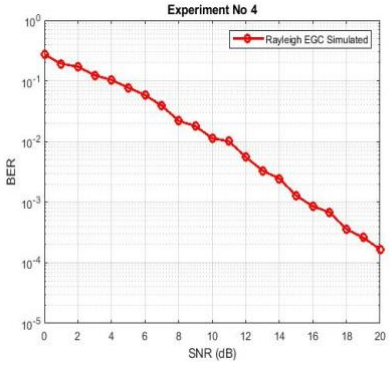
Experiment 6

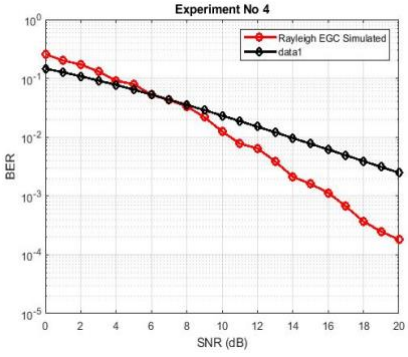
Bit error rate analysis of digital communication receivers with Equal Gain Combining (EGC) receives diversity in frequency-flat and slowly varying fading channel.

Aim

Bit error rate analysis of digital communication receivers with Equal Gain Combining (EGC) receive diversity in frequency-flat and slowly varying fading channel.

```
clc;
close all;
clear all;
% Number of information bits
m= 10^3;
%Range of SNR values
snr_dB = [0:1:20];
for j=1:1:length(snr_dB)
n_err = 0;
n_bits = 0;
while n_err < 100
inf_bits=round(rand(1,m));
% BPSK modulator
x=-2*(inf_bits-0.5);
% Noise variance
N0=1/10^(snr_dB(j)/10);
n1 = sqrt(N0/2)*abs((randn(1,length(x)) + i*randn(1,length(x)))); %noise for the first
n2 = sqrt(N0/2)*abs((randn(1,length(x)) + i*randn(1,length(x)))); %noise for the first
h1 = sqrt(0.5)*abs((randn(1,length(x)) + i*randn(1,length(x)))); %rayleigh amplitude 1
h2 = sqrt(0.5)*abs((randn(1,length(x)) + i*randn(1,length(x)))); %rayleigh amplitude 1
%Equal Gain combining
y1 = h1.*x+n1; % Signal 1
y2 = h2.*x+n2; % Signal 2
y_equal = 0.5*(y1+y2);
% dec_metric=(norm(y_equal- h1*x-h2*x))^2;
% Decision making at the Receiver
est_bits=y_equal < 0;
% Calculate Bit Errors
diff=inf_bits-est_bits;
n_err=n_err+sum(abs(diff));
n_bits=n_bits+length(inf_bits);
end
% Calculate Bit Error Rate
BER(j)=n_err/n_bits;
end
semilogy(snr_dB,BER,'or-','LineWidth',2);
legend('Rayleigh EGC Simulated', 'Rayleigh Theoretical');
axis([0 20 10^-5 1]);
xlabel('SNR (dB)');
ylabel('BER');
grid on;
```


Experiment No 4
Rayleigh EGC Simulated
BER
SNR (dB)


Experiment No 4
Rayleigh EGC Simulated
data1
BER
SNR (dB)

Experiment 7

BER performance of 2x1 MISO under the Rayleigh fading channel

Aim

BER performance of 2x1 MISO under the Rayleigh fading channel in frequency-flat and slowly varying fading channel.

```
clc;
close all;
clear all;
ndata=2;
x=randint(ndata,1,1);
x=[1 2];
y=[x];
% Input data bits
Data_input_bit(1,1)=x(1,1);
Data_input_bit(1,2)=x(1,2);
figure;
plot(Data_input_bit);
title('input data bits');
z=qammod(Data_input_bit,4);
%CHANNEL COEFFICENTS MATRIX
h=[0.3 -.2];
%h11=1; h12=1; h21=1; h22=1;

%NOISE COEFFICENTS
e=[.1 .1];
%e11=1; e12=1; e21=1; e22=1;
out=zeros(10,1);
for i=1;%:ndata-1;
% Symbols at time period T;
out(i,1)=z(i);
out(i+1,1)=z(i+1);
% Symbols at time period T+1;
out(i,2)=-conj(z(i+1));
out(i+1,2)=conj(z(i));
%time_t2(i,1)=-conj(z(i+1));
%time_t2(i+1,1)=conj(z(i));
end
s1=out(i,1);
s2=out(i+1,1);
%for j=1:100
for i=1;
%Recieved data by RX1  Antenna at time interval T
r(1,1)= (h(1,1)*s1) + (h(1,2)*s2) + e(1,1);
```

```
%Recieved data by RX1 Antenna at time interval (T+1)
r(1,2)= ((-h(1,1))*conj(s2)) + (h(1,2)*conj(s1)) + e(1,2);
end

t(1,1)=((conj(h(1,1))*r(1,1)));
t(1,2)=h(1,2)*(conj(r(1,2)));
t(2,1)=((conj(h(1,2)))*r(1,1));
t(2,2)=((h(1,1)*(conj(r(1,2)))));

%Maximum Likelehhod Detection Scehme
s1_e =t(1,1) + t(1,2);
s2_e= t(2,1) - t(2,2);
%s1_e= ((conj(h(1,1))*r(1,1))) + ((h(1,2)*(conj(r(1,2)))+ );
%s2_e= (((conj(h(2,1)))*r(2,1)) + ((h(1,2)*(conj(r(2,2))));
%performing 4 QAM Demodulation
%final output bits
```

Experiment 8

OFDM transmitter and receiver in AWGN channel

Aim

Matlab simulation of OFDM transmitter and receiver in AWGN channel

```
clc;
clear all;
close all;
% Initiation
no_of_data_bits = 64%Number of bits per channel extended to 128
M =4 %Number of subcarrier channel
n=256;%Total number of bits to be transmitted at the transmitter
block_size = 16; %Size of each OFDM block to add cyclic prefix
cp_len = floor(0.1 * block_size); %Length of the cyclic prefix
%  Transmitter
% Source generation and modulation
% Generate random data source to be transmitted of length 64
data = randsrc(1, no_of_data_bits, 0:M-1);
figure(1),stem(data); grid on; xlabel('Data Points'); ylabel('Amplitude')
title('Original Data ')
% Perform QPSK modulation on the input source data
qpsk_modulated_data = pskmod(data, M);
figure(2),stem(qpsk_modulated_data);title('QPSK Modulation ')
% Converting the series data stream into four parallel data stream to form
% four sub carriers
S2P = reshape(qpsk_modulated_data, no_of_data_bits/M,M)
Sub_carrier1 = S2P(:,1)
Sub_carrier2 = S2P(:,2)
Sub_carrier3 = S2P(:,3)
Sub_carrier4 = S2P(:,4)
figure(3), subplot(4,1,1),stem(Sub_carrier1),title('Subcarrier1'),grid on;
subplot(4,1,2),stem(Sub_carrier2),title('Subcarrier2'),grid on;
subplot(4,1,3),stem(Sub_carrier3),title('Subcarrier3'),grid on;
subplot(4,1,4),stem(Sub_carrier4),title('Subcarrier4'),grid on;
% IFFT OF FOUR SUB_CARRIERS
number_of_subcarriers=4;
cp_start=block_size-cp_len;
ifft_Subcarrier1 = ifft(Sub_carrier1)
ifft_Subcarrier2 = ifft(Sub_carrier2)
ifft_Subcarrier3 = ifft(Sub_carrier3)
ifft_Subcarrier4 = ifft(Sub_carrier4)
```

```
figure(4), subplot(4,1,1),plot(real(ifft_Subcarrier1),'r'),
title('IFFT on all the sub-carriers')
subplot(4,1,2),plot(real(ifft_Subcarrier2),'c')
subplot(4,1,3),plot(real(ifft_Subcarrier3),'b')
subplot(4,1,4),plot(real(ifft_Subcarrier4),'g')
% ADD-CYCLIC PREFIX
for i=1:number_of_subcarriers,
ifft_Subcarrier(:,i) = ifft((S2P(:,i)),16)% 16 is the ifft point
for j=1:cp_len,
cyclic_prefix(j,i) = ifft_Subcarrier(j+cp_start,i)
end
Append_prefix(:,i) = vertcat( cyclic_prefix(:,i), ifft_Subcarrier(:,i))
% Appends prefix to each subcarriers
end
A1=Append_prefix(:,1);
A2=Append_prefix(:,2);
A3=Append_prefix(:,3);
A4=Append_prefix(:,4);
figure(5), subplot(4,1,1),plot(real(A1),'r'),title('Cyclic prefix added to all the sub-carriers')
subplot(4,1,2),plot(real(A2),'c')
subplot(4,1,3),plot(real(A3),'b')
subplot(4,1,4),plot(real(A4),'g')
figure(11),plot((real(A1)),'r'),title('Orthogonality'),hold on ,plot((real(A2)),'c'),hold on ,
plot((real(A3)),'b'),hold on ,plot((real(A4)),'g'),hold on ,grid on
% Convert to serial stream for transmission
[rows_Append_prefix cols_Append_prefix]=size(Append_prefix)
len_ofdm_data = rows_Append_prefix*cols_Append_prefix
% OFDM signal to be transmitted
ofdm_signal = reshape(Append_prefix, 1, len_ofdm_data);
figure(6),plot(real(ofdm_signal)); xlabel('Time'); ylabel('Amplitude');
title('OFDM Signal');grid on;

%Passing time domain data through channel and AWGN
channel = randn(1,2) + sqrt(-1)*randn(1,2);
after_channel = filter(channel, 1, ofdm_signal);
awgn_noise = awgn(zeros(1,length(after_channel)),0);
recvd_signal = awgn_noise+after_channel; % With AWGN noise
figure(7),plot(real(recvd_signal)),xlabel('Time'); ylabel('Amplitude');
title('OFDM Signal after passing through channel');grid on;
%OFDM receiver part
recvd_signal_paralleled = reshape(recvd_signal,rows_Append_prefix, cols_Append_prefix);
% Remove cyclic Prefix
recvd_signal_paralleled(1:cp_len,:)=[];
R1=recvd_signal_paralleled(:,1);
R2=recvd_signal_paralleled(:,2);
```

```
R3=recvd_signal_paralleled(:,3);
R4=recvd_signal_paralleled(:,4);
figure(8),plot((imag(R1)),'r'),subplot(4,1,1),plot(real(R1),'r'),
title('Cyclic prefix removed from the four sub-carriers')
subplot(4,1,2),plot(real(R2),'c')
subplot(4,1,3),plot(real(R3),'b')
subplot(4,1,4),plot(real(R4),'g')
% FFT Of recievied signal
for i=1:number_of_subcarriers,
% FFT
fft_data(:,i) = fft(recvd_signal_paralleled(:,i),16);
end
F1=fft_data(:,1);
F2=fft_data(:,2);
F3=fft_data(:,3);
F4=fft_data(:,4);
figure(9), subplot(4,1,1),plot(real(F1),'r'),title('FFT of all the four sub-carriers')
subplot(4,1,2),plot(real(F2),'c')
subplot(4,1,3),plot(real(F3),'b')
subplot(4,1,4),plot(real(F4),'g')
% Signal Reconstructed
% Conversion to serial and demodulationa
recvd_serial_data = reshape(fft_data, 1,(16*4));
qpsk_demodulated_data = pskdemod(recvd_serial_data,4);
figure(10)
stem(data)
hold on
stem(qpsk_demodulated_data,'rx');
grid on;xlabel('Data Points');ylabel('Amplitude');
title('Recieved Signal with error')
```