<u>FDP on WMC</u>

Abhijit Bhowmick

Professor, Dept. of Communication Engineering

SENSE, VIT, Vellore, TN

1.

Received signal at distance *d* for an AWGN channel (h = 1) can be expressed as

$$y = \frac{\sqrt{P_s}\ s\,h}{\sqrt{d^n}} + w \tag{1}$$

Received signal power

$$P_y = |y|^2 \tag{2}$$

```
clc;clear all;%close all;
numb_simul=5000;                                       %Number of user
n=2;  %2, 3
Ps=1;  % Transmit power
d=linspace(1,10,20);
Detect_th=4; %Detection threshold in dB
Threshold=10^(Detect_th/10);

for i=1:length(d)
    count=0;
    sig_pow=0;
    for k=1:numb_simul
           %% Genration of Sigmal
           signal=randsrc(1,1);      %Transmitted Data
           %% Genration of Noise
           var=1;                      %Taking as variance
for the AWGN noise
           std1=sqrt(var);                  %Std deviation of
noise
           noise = normrnd(0,std1,1,1);
          %% Channel
           %h1=1;%normrnd(0,1); %AWGN channel
           h1 =1/sqrt(2)*(randn(1,1) + j*randn(1,1));%
Rayleigh channel
           h=abs(h1);
```

```
            % Received Signal
            Rev_sig =(sqrt(Ps)*h.*
signal)/(sqrt(d(i)^n))+noise;
            % Received signal power
             signal_power=abs((Rev_sig).^2);
            % Detection signal power
             if signal_power>=Threshold
                 count=count+1
             end
        %sig_pow=sig_pow+signal_power;
      end
 Pd_sim(i)=(count/numb_simul);
 end
plot(d,Pd_sim,'v-r'); hold on;
 %plot(d,Pd_sim,'ob',d,Pd_th,'-b'); hold on;
```

2. Link Capacity Measurement

Capacity of the channel

$$C = \log_2\left(1 + SNR_{Rx}\right)$$
$$= \log_2\left(1 + \frac{P_s |h|^2}{\sigma_w^2}\right)$$

```
clc;clear all;%close all;
numb_simul=5000;                          %Number of user
%Ps=1;  % Transmit power 1W
SNR=linspace(1,10,20); %Transmitting SNR in dB
snr=10.^(SNR/10);
B=3; %1, 2,3
for i=1:length(SNR)
    %Simulation
    Cap=0;
    for k=1:numb_simul
           h1 =1/sqrt(2)*(randn(1,1) + j*randn(1,1));%
Rayleigh channel
```

```
            h=abs(h1);
            h_mod_sqr=(abs(h))^2;
            snr_link=snr(i)*h_mod_sqr
            C=B*log2(1+snr_link);
            Cap=Cap+C;
    end
 Cavg_simu(i)=(Cap/numb_simul);
end
plot(SNR,Cavg_simu,'o-b'); hold on;
```

3. BER for BPSK transmission through AWGN channel:

BER of BPSK signal for AWGN channel can be written as,

$$P_b = \frac{1}{2} erfc\left(\sqrt{\frac{E_b}{N_0}}\right).$$

```
clc;clear; close all
N = 10^6 % number of bits or symbols
% Transmitter
ip = rand(1,N)>0.5; % generating 0,1 with equal
probability
s = 2*ip-1; % BPSK modulation 0 -> -1; 1 -> 1

n = 1/sqrt(2)*[randn(1,N) + j*randn(1,N)]; %
white gaussian noise, 0dB variance
SNR_dB = [-3:10]; % multiple Eb/N0 values

for i = 1:length(SNR_dB)

   % received signal  y=h*s+n; for AWGN h=1;
```

```
    y = s + sqrt(10^(-SNR_dB(i)/10))*n; % 
additive white gaussian noise SNR = Ps/No; for 
Ps=1, N0=SNR^-1
    % receiver - hard decision decoding
    ip_estm = real(y)>0;

    % counting the errors
    No_bit_err(i) = size(find(ip- ip_estm),2);

end

simBer = No_bit_err/N; % simulated ber
theoryBer = 0.5*erfc(sqrt(10.^(SNR_dB/10))); % 
theoretical ber

semilogy(SNR_dB,theoryBer,'o b');hold on
semilogy(SNR_dB,simBer,'*--r'); hold on
axis([-3 10 10^-5 0.5])
grid on
legend('BER for AWGN theory', 'BER for AWGN 
simulation');
xlabel('Eb/No, dB');
ylabel('Bit Error Rate');
title('Bit error probability curve for BPSK 
modulation');
```

3. Comparison of received SNR for SISO and SIMO transmissions

Received SNR for SISO is $SNR_{rx} = \frac{P_s|h|^2}{\sigma_n^2}$

Received SNR for SIMO is $SNR_{rx} = \frac{P_s\left||h|\right|^2}{\sigma_n^2}$ (for MRC)

$$\left||h|\right|^2 = |h_1|^2 + |h_2|^2 + \cdots.$$

```
clc;clear all;%close all;

numb_simul=5000;

n=2;  %2, 3 Pathloss exponent
Ps=1;  % Transmit power 10, 5
d=linspace(5,50,20);

for i=1:length(d)
    C1=0;
    C2=0;

      for k=1:numb_simul
           %% Genration of Signal
           %Transmitted BPSK Sig
           signal=randsrc(1,1);
           %Channel generation
           h1 =1/sqrt(2)*(randn(1,1) + j*randn(1,1));%
Rayleigh channel
           h2 =1/sqrt(2)*(randn(1,1) + j*randn(1,1));%
Rayleigh channel
           h3 =1/sqrt(2)*(randn(1,1) + j*randn(1,1));%
Rayleigh channel
           h4 =1/sqrt(2)*(randn(1,1) + j*randn(1,1));%
Rayleigh channel
           % Noise variance (noise power)
           Noise_var=10^-3;
```

```matlab
            %% SISO
            % Received signal
            h11=abs(h1);
            y_siso1 =(sqrt(Ps)*h11* signal)/(sqrt(d(i)^n));
            % Received SNR  SISO
            SNR_siso=(y_siso1)^2/Noise_var;
            SNR_siso=SNR_siso+C1;

            %% SIMO and MRC
            % Received signal
            y1 =(sqrt(Ps)*h1.* signal)/(sqrt(d(i)^n));
            y2 =(sqrt(Ps)*h2.* signal)/(sqrt(d(i)^n));
            y3 =(sqrt(Ps)*h3.* signal)/(sqrt(d(i)^n));
            y4 =(sqrt(Ps)*h4.* signal)/(sqrt(d(i)^n));

            % Generation of weight vectors for MRC
            h_norm=(sqrt(abs(h1)^2+abs(h2)^2));
            w1=conj(h1)/h_norm;
            w2=conj(h2)/h_norm;
            w3=conj(h3)/h_norm;
            w4=conj(h4)/h_norm;
            % Received combined signal for MRC
            y_mrc=w1*y1+w2*y2+w3*y3+w4*y4;
            SNR_mrc=(y_mrc)^2/Noise_var;
            SNR_mrc=SNR_mrc+C2;

       end
     SNR_siso_avg(i)=(SNR_siso/numb_simul);
     SNR_mrc_avg(i)=(SNR_mrc/numb_simul);

  end
 plot(d,10*log10(SNR_siso_avg),'*-r'); hold on;
 plot(d,10*log10(SNR_mrc_avg),'v-b'); hold on;
```