Nama : Muhammad Iqbal Hanif   
Kelas : 2 D4 Telekomunikasi A  
NRP : 1210161020

**Tugas Dasar Sistem Komunikasi**

**Analisa Sinyal BPSK dan AWGN pada Aplikasi Matlab**

1. **BPSK Kanal Ideal (AWGN)**

- Source Code

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Creative Commons

% Attribution-Noncommercial 2.5 India

% You are free:

% to Share ??•h to copy, distribute and transmit the work

% to Remix ??•h to adapt the work

% Under the following conditions:

% Attribution. You must attribute the work in the manner

% specified by the author or licensor (but not in any way

% that suggests that they endorse you or your use of the work).

% Noncommercial. You may not use this work for commercial purposes.

% For any reuse or distribution, you must make clear to others the

% license terms of this work. The best way to do this is with a

% link to this web page.

% Any of the above conditions can be waived if you get permission

% from the copyright holder.

% Nothing in this license impairs or restricts the author's moral rights.

% http://creativecommons.org/licenses/by-nc/2.5/in/

% Script for simulating binary phase shift keyed transmission and

% reception and compare the simulated and theoretical bit error

% probability

% Checked for proper operation with Octave Version 3.0.0

% Author : Krishna

% Email : krishna@dsplog.com

% Version : 1.0

% Date : 5 August 2007

% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear

N = 10^6 % number of bits or symbols

rand('state',100); % initializing the rand() function

randn('state',200); % initializing the randn() function

% 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

Eb\_N0\_dB = [-3:10]; % multiple Eb/N0 values

for ii = 1:length(Eb\_N0\_dB)

% Noise addition

y = s + 10^(-Eb\_N0\_dB(ii)/20)\*n; % additive white gaussian noise

% receiver - hard decision decoding

ipHat = real(y)>0;

% counting the errors

nErr(ii) = size(find([ip- ipHat]),2);

end

simBer = nErr/N; % simulated ber

theoryBer = 0.5\*erfc(sqrt(10.^(Eb\_N0\_dB/10))); % theoretical ber

% plot

close all

figure

semilogy(Eb\_N0\_dB,theoryBer,'b.-');

hold on

semilogy(Eb\_N0\_dB,simBer,'mx-');

axis([-3 10 10^-5 0.5])

grid on

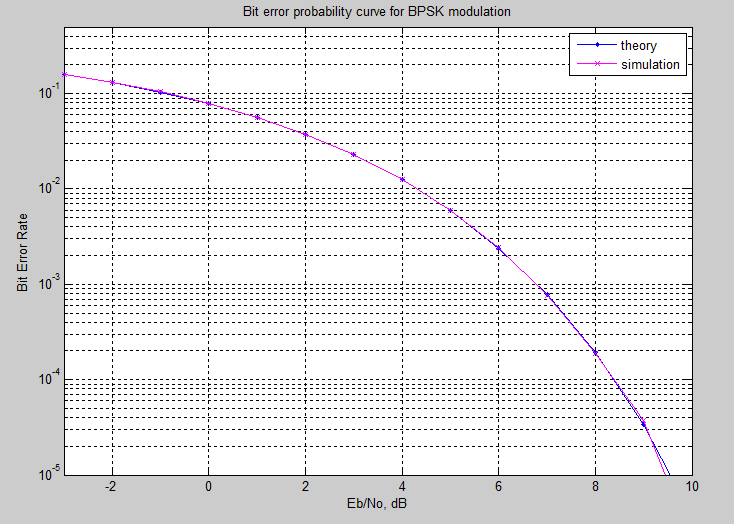
legend('theory', 'simulation');

xlabel('Eb/No, dB');

ylabel('Bit Error Rate');

title('Bit error probability curve for BPSK modulation');

* Hasil Output



* **Analisa**

AWGN (Additive White Gaussian Noise) dapat didefinisikan sebagai noise kelas basic yang merupakan tiruan dari noise alami. Additive artinya ditambahkan. Gaussian berarti mengikuti distribusi Gaussian atau kadang juga disebut distribusi normal. Sedangkan noise ini disebut white karena terdiri dari seluruh frekuensi dalam spektralnya sebagai cahaya putih. White noise ini sebagai WSS noise yang memiliki rapat spektral daya yang konstan.

Jika digambarkan grafik maka dapati terlihat nilai BER nya semakin menurun apabila semakin berjalannya nilai Eb/No. Ini menandakan nilai BER akan semakin berkurang apabila nilai Eb/No semakin besar.

AWGN atau biasa disebut Additive White Gaussian Noise adalah noise kelas basic yang merupakan tiruan dari noise alami. Teori yang telah dijelaskan seperti diatas dapat dibuktikan pada gambar di atas. Dari grafik didapatkan bahwa hasil secara teori dan praktek menghasilkan grafik yang nyari sama bentuknya.

1. **BPSK Kanal Non Ideal (Rayleigh)**

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% All rights reserved by Krishna Pillai, http://www.dsplog.com

% The file may not be re-distributed without explicit authorization

% from Krishna Pillai.

% Checked for proper operation with Octave Version 3.0.0

% Author : Krishna Pillai

% Email : krishna@dsplog.com

% Version : 1.0

% Date : 8 August 2008

% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Script for computing the BER for BPSK modulation in a

% Rayleigh fading channel

clear

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 -> 0

Eb\_N0\_dB = [-3:35]; % multiple Eb/N0 values

for ii = 1:length(Eb\_N0\_dB)

n = 1/sqrt(2)\*[randn(1,N) + j\*randn(1,N)]; % white gaussian noise, 0dB variance

h = 1/sqrt(2)\*[randn(1,N) + j\*randn(1,N)]; % Rayleigh channel

% Channel and noise Noise addition

y = h.\*s + 10^(-Eb\_N0\_dB(ii)/20)\*n;

% equalization

yHat = y./h;

% receiver - hard decision decoding

ipHat = real(yHat)>0;

% counting the errors

nErr(ii) = size(find([ip- ipHat]),2);

end

simBer = nErr/N; % simulated ber

theoryBerAWGN = 0.5\*erfc(sqrt(10.^(Eb\_N0\_dB/10))); % theoretical ber

EbN0Lin = 10.^(Eb\_N0\_dB/10);

theoryBer = 0.5.\*(1-sqrt(EbN0Lin./(EbN0Lin+1)));

% plot

close all

figure

semilogy(Eb\_N0\_dB,theoryBerAWGN,'cd-','LineWidth',2);

hold on

semilogy(Eb\_N0\_dB,theoryBer,'bp-','LineWidth',2);

semilogy(Eb\_N0\_dB,simBer,'mx-','LineWidth',2);

axis([-3 35 10^-5 0.5])

grid on

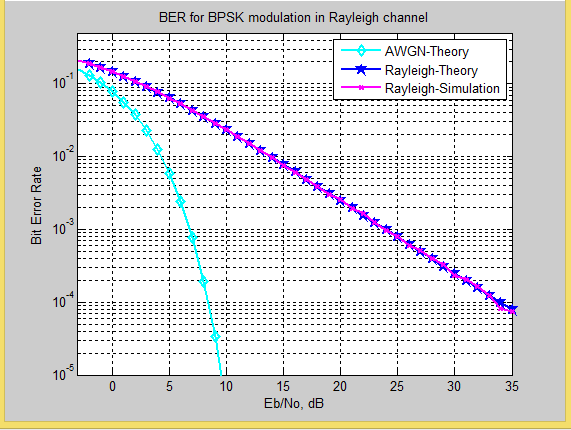
legend('AWGN-Theory','Rayleigh-Theory', 'Rayleigh-Simulation');

xlabel('Eb/No, dB');

ylabel('Bit Error Rate');

title('BER for BPSK modulation in Rayleigh channel');

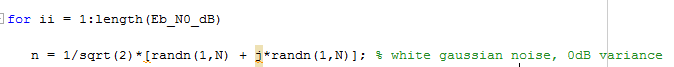
* **Output**



* **Analisa**

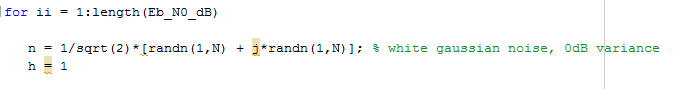
Rayleight Fading adalah bentuk kanal non ideal yang juga dapat didefinisikan sebagai model yang mengasumsikan besaran dari sinyal yang telah melewati media transmisi akan secara random mengalami fading berdasarkan Distribusi Rayleigh. Secara teori, perbedaan kanal AWGN dank anal Rayleigh terdapat pada banyak nya BER yang dihasilkan pada masing masing kanal. Kanal Raileigh lebih banyak menghasilkan BER dibandingkan dengan kanal AWGN.

Pada program matlab di atas, white noise gaussian/AWGN di hasilkan pada script berikut ini:

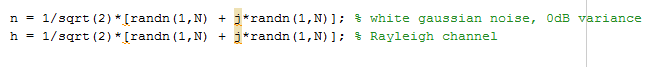


Percobaan di atas memrepresentasikan 2 kanal yang digunakan yaitu Kanal AWGN dan Kanal Rayleigh

Script untuk Kanal AWGN



Script Untuk Kanal Rayleigh:



Perbedaan paling mendasar antara kedua kanal adalah bahwa pada kanal AWGN,nilai h=1,dimana berarti ini Kanal AWGN adalah kanal ideal.Sebaliknya pada saat Kanal Rayleigh,nilai h pada kanal tidak ideal melainkan dibuat dengan berdasarkan kanal yang mengikuti distribusi Rayleigh. Kanal Rayleigh ini biasanya digunakan untuk kanal yang tidak LOS (Line-of-Sight) atau juga sering disebut NLOS (non-Line-of-sight).

Karena hal ini akibatnya apabila terjadi suatu fading pada sinyal yang ditransmisikan, maka BER yang terjadi akan cukup banyak apabila nilai Eb/No tidak cukup tinggi untuk mengurangi BER yang terjadi. Grafik terbukti secara teori dan praktek untuk nilai – nilai pada Kanal Rayleigh.

1. **OFDM Frame Data Mode**

* **Source Code**

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% All rights reserved by Krishna Pillai, http://www.dsplog.com

% The file may not be re-distributed without explicit authorization

% from Krishna Pillai.

% Checked for proper operation with Octave Version 3.0.0

% Author : Krishna Pillai

% Email : krishna@dsplog.com

% Version : 1.0

% Date : 05 June 2008

% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Script for computing the Bit Error probability using OFDM modulation

clear all

nFFT = 64; % fft size

nDSC = 52; % number of data subcarriers

nBitPerSym = 52; % number of bits per OFDM symbol (same as the number of subcarriers for BPSK)

nSym = 10^4; % number of symbols

EbN0dB = [0:10]; % bit to noise ratio

EsN0dB = EbN0dB + 10\*log10(nDSC/nFFT) + 10\*log10(64/80); % converting to symbol to noise ratio

for ii = 1:length(EbN0dB)

% Transmitter

ipBit = rand(1,nBitPerSym\*nSym) > 0.5; % random 1's and 0's

ipMod = 2\*ipBit-1; % BPSK modulation 0 --> -1, 1 --> +1

ipMod = reshape(ipMod,nBitPerSym,nSym).'; % grouping into multiple symbolsa

% Assigning modulated symbols to subcarriers from [-26 to -1, +1 to +26]

xF = [zeros(nSym,6) ipMod(:,[1:nBitPerSym/2]) zeros(nSym,1) ipMod(:,[nBitPerSym/2+1:nBitPerSym]) zeros(nSym,5)] ;

% Taking FFT, the term (nFFT/sqrt(nDSC)) is for normalizing the power of transmit symbol to 1

xt = (nFFT/sqrt(nDSC))\*ifft(fftshift(xF.')).';

% Appending cylic prefix

nSym\*80) + j\*randn(1,nSym\*80)];

% Adding noise, the term sqrt(80/64) is to account for the wasted energy due to cyclic prefix

yt = sqrt(80/64)\*xt + 10^(-EsN0dB(ii)/20)\*nt;

% Receiver

yt = reshape(yt.',80,nSym).'; % formatting the received vector into symbols

yt = yt(:,[17:80]); % removing cyclic prefix

% converting to frequency domain

yF = (sqrt(nDSC)/nFFT)\*fftshift(fft(yt.')).';

yMod = yF(:,[6+[1:nBitPerSym/2] 7+[nBitPerSym/2+1:nBitPerSym] ]);

% BPSK demodulation

% +ve value --> 1, -ve value --> -1

ipModHat = 2\*floor(real(yMod/2)) + 1;

ipModHat(find(ipModHat>1)) = +1;

ipModHat(find(ipModHat<-1)) = -1;

semilogy(EbN0dB,theoryBer,'bs-','LineWidth',2);

hold on

semilogy(EbN0dB,simBer,'mx-','LineWidth',2);

axis([0 10 10^-5 1])

grid on

legend('theory', 'simulation');

xlabel('Eb/No, dB')

ylabel('Bit Error Rate')

title('Bit error probability curve for BPSK using OFDM')

% Concatenating multiple symbols to form a long vector

xt = reshape(xt.',1,nSym\*80);

% Gaussian noise of unit variance, 0 mean

nt = 1/sqrt(2)\*[randn(1,nSym\*80) + j\*randn(1,nSym\*80)];

% Adding noise, the term sqrt(80/64) is to account for the wasted energy due to cyclic prefix

yt = sqrt(80/64)\*xt + 10^(-EsN0dB(ii)/20)\*nt;

% Receiver

yt = reshape(yt.',80,nSym).'; % formatting the received vector into symbols

yt = yt(:,[17:80]); % removing cyclic prefix

% converting to frequency domain

yF = (sqrt(nDSC)/nFFT)\*fftshift(fft(yt.')).';

yMod = yF(:,[6+[1:nBitPerSym/2] 7+[nBitPerSym/2+1:nBitPerSym] ]);

% BPSK demodulation

% +ve value --> 1, -ve value --> -1

ipModHat = 2\*floor(real(yMod/2)) + 1;

ipModHat(find(ipModHat>1)) = +1;

ipModHat(find(ipModHat<-1)) = -1;

% converting modulated values into bits

ipBitHat = (ipModHat+1)/2;

ipBitHat = reshape(ipBitHat.',nBitPerSym\*nSym,1).';

% counting the errors

nErr(ii) = size(find(ipBitHat - ipBit),2);

end

simBer = nErr/(nSym\*nBitPerSym);

theoryBer = (1/2)\*erfc(sqrt(10.^(EbN0dB/10)));

close all; figure

semilogy(EbN0dB,theoryBer,'bs-','LineWidth',2);

hold on

semilogy(EbN0dB,simBer,'mx-','LineWidth',2);

axis([0 10 10^-5 1])

grid on

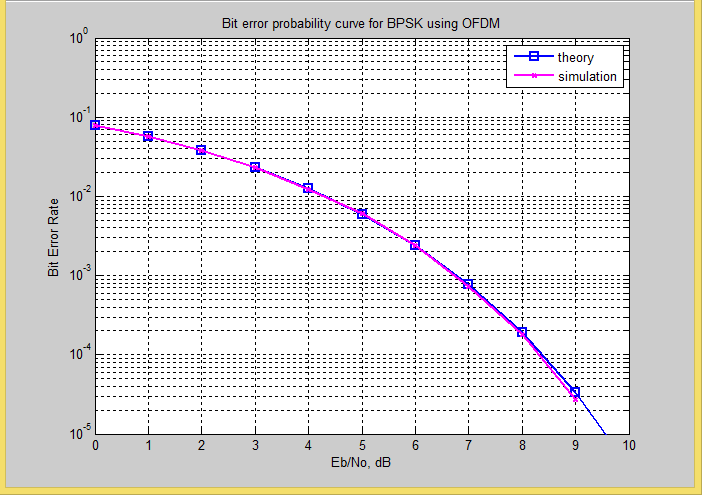
legend('theory', 'simulation');

xlabel('Eb/No, dB')

ylabel('Bit Error Rate')

title('Bit error probability curve for BPSK using OFDM')

* **Output**



* **Analisa**

OFDM atau biasa disebut Orthogonal Frequency Division Multiplexing dapat didefinisikan metode dari encoding data digital pada multiple carrier frequencies. Pada OFDM, Data yang dikirim berbentuk multiple carrier sehingga apabila ada carrier yang hilang atau bermasalah, sinyal pada carrier lain masih dapat sampai pada receiver. Jika dibandingkan dengan kanal yang lain seperi AWGN, BER pada OFDM lebih sedikit jumlahnya. Hal ini terbukti dapat terbukti pada grafik di atas.

Cara kerja dari OFDM adalah sebagai berikut. Deretan data informasi yang akan dikirim dikonversikan kedalam bentuk parallel, sehingga bila bit rate semula adalah *R* , maka bit rate di tiap-tiap jalur parallel adalah *R/M* dimana *M* adalah jumlah jalur parallel (sama dengan jumlah sub-carrier). Setelah itu, modulasi dilakukan pada tiap-tiap sub-carrier. Modulasi ini bisa berupa BPSK, QPSK, QAM atau yang lain, tapi ketiga teknik tersebut sering digunakan pada OFDM. Kemudian sinyal yang telah termodulasi tersebut diaplikasikan ke dalam Inverse Discrete Fourier Transform (IDFT), untuk pembuatan simbol OFDM. Penggunaan IDFT ini memungkinkan pengalokasian frekuensi yang saling. tegak lurus (*orthogonal*), mengenai hal ini akan dijelaskan lebih lanjut. Setelah itu simbol-simbol OFDM dikonversikan lagi kedalam bentuk serial, dan kemudian sinyal dikirim.