# **Lab 11**

# **Performance Evaluation of Orthogonal Frequency Division Multiplexing (OFDM) under Fading and AWGN Channels**

**Objectives**

1. Implementation of a basic OFDM transceiver under multipath fading channel.
2. Performance comparison between OFDM transceiver over Fading and AWGN channels.

Launch the MATLAB and write the following script:

clc

close all

clear all

%% Setting Parameters

Subcarriers = 64; % total number of subcarrier (IFFT length equal to Subcarriers)

M = 16; % number of constellations

k = log2(M); % number of bits per constellation

numOfSym = 10^4; % number of OFDM Symbols

GI = 1/4; % Guard Interval or Cyclic Prefix, normaly 25 of the entire OFDM symbols

snr = 15; % Signal to noise ratio in dB

variance = 0.5; % variance of the rayleigh distribution

%% --------------------- TRANSMITER --------------------------------------

%%------------------------------------------------------------------------

%% Generate Data to be modulated on the subcarriers

TxData = randi([0, 1], Subcarriers, numOfSym);

%% Implement QAM modulation

TxData\_Modulated = qammod(TxData', M, 'InputType','bit');

%% Perform IFFT

TxData\_IFFT = ifft(TxData\_Modulated');

%% Adding cyclic Prefix

TxData\_GI = [TxData\_IFFT((1-GI)\*...

Subcarriers+1:end,:);TxData\_IFFT];

%% Plotting OFDM signal in time domain

[row , col] = size(TxData\_GI);

len = row\*col;

ofdm\_signal = reshape(TxData\_GI, 1, len);

figure(1);

plot(real(ofdm\_signal));

xlabel('Time');

ylabel('Amplitude');

title('OFDM Signal');

grid on;

%% ------------------------ GHANNEL ---------------------------------------

%%-------------------------------------------------------------------------

%% 1) using Rayleigh channel:

N0 = 1/10^(snr/10);

for ii = 1:size(TxData\_GI,1)

ray = sqrt((variance)\*((randn(1,length(TxData\_GI(ii,:)))).^2+(randn(1,length(TxData\_GI(ii,:)))).^2));

% include the fading

rx = TxData\_GI(ii,:).\*ray;

% fading + gaussian noise

Receive\_Channel\_rayleigh(ii,:) = rx + sqrt(N0/2)\*(randn(1,length(TxData\_GI(ii,:)))+1i\*randn(1,length(TxData\_GI(ii,:))));

end

%% --------------------- RECEIVER ----------------------------------------

%%------------------------------------------------------------------------

%% CP removal

Recieve\_GIremoved = Receive\_Channel\_rayleigh(GI\*Subcarriers+1 : Subcarriers+GI\*Subcarriers, :);

%% FFT operation

RecieveData\_FFT = fft(Recieve\_GIremoved);

Signal\_Magnitude = real(RecieveData\_FFT);

Signal\_Phase = imag(RecieveData\_FFT);

%% plot the received constellations for a specific subcarrier

n = 4; % selected subcarrier

scatterplot(RecieveData\_FFT(n,:));

title('FFT Output 16-QAM');

%% Demodulation

RecieveData = qamdemod(RecieveData\_FFT',M, 'OutputType','bit');

%% Number of Bit Errors and Bit Error Rate computation

[num , BER] = biterr(TxData, RecieveData');

1. Execute the code and observe the output.
2. Add in the same script the AWGN channel using the following:

%% 2) using AWGN channel

Receive\_Channel\_awgn = awgn(TxData\_GI ,snr,'measured');

In this way the generated signal will pass through the Rayleigh fading channel and the AWGN channel. After the channel you will have two copies of the generated signal one related to the fading channel (called “Receive\_Channel\_rayleigh”) and the other related to the awgn channel (called “Receive\_Channel\_awgn”). At the receiver side perform the CP removal, FFT and demodulation separately for each signal (among Receive\_Channel\_rayleigh and Receive\_Channel\_awgn).

Exercise A

Add a for loop to use a vector of different SNR values.

Comment all the figures to avoid compilation timeout.

Let SNR values vary from -20dB to +20 dB with a step size of 2dB saved in a vector called ‘snrV’.

Calculate the BER value (based on the simulated data) related to Fading and AWGN channels for each SNR value. So, you need to create two different vectors ber\_awgn and ber\_rayleigh with number of elements equal to the SNR values we are considering.

You will have the following output:



Note: for plotting use the following code:

figure;

semilogy(snrV, ber\_awgn,'-ok', snrV, ber\_ray, '-or');

grid;

ylabel('BER');

xlabel('SNR [dB]');

legend('AWGN', 'Rayleigh');

Exercise B

In this exercise we will compare between OFDM signals under AWGN and fading channels using different modulation schemes.

1. Create a new script.
2. User the following parameters:

%% Setting Parameters

Subcarriers = 128; % total number of subcarrier (IFFT length equal to Subcarriers)

M1 = 16; % number of constellations 16QAM

M2 = 32; % number of constellations 32QAM

M3 = 64; % number of constellations 64QAM

M4 = 256; % number of constellations 256QAM

numOfSym = 21600; % number of OFDM Symbols

numOfSym1 = numOfSym\*2;

numOfSym2 = numOfSym\*2.5;

numOfSym3 = numOfSym\*3;

numOfSym4 = numOfSym\*4;

GI = 1/4; % Guard Interval or Cyclic Prefix, normaly 25 of the entire OFDM symbols

snrV = -20:2:30; % Signal to noise ratio in dB

ber\_awgn = zeros(1, length(snrV)); % vector containing BER values for each SNR

ber\_rayleigh = zeros(1, length(snrV));

The number of symbols for each modulation (numOfSym1, …, numOfSym4) are selected in this way to generate the same number of symbols for fair comparison.

1. Generate 4 different OFDM signals using 16QAM (so all subcarriers in the first signal are modulated according to 16QAM), 32QAM (all subcarriers in the first signal are modulated according to 32QAM), 64QAM (so all subcarriers in the first signal are modulated according to 64QAM) and 256QAM (so all subcarriers in the first signal are modulated according to 256QAM).
2. Perform the IFFT operation 4 times since we have 4 OFDM signals.
3. Use the predefined function in MATLAB to model a fading channel using the following parameters and script:

tx = reshape(TxData\_GI,[],1); TxData is the signal after Adding cyclic Prefix

fs = 3.84e6; % Hz

pathDelays = [0 200 800 1200 2300 3700]\*1e-9; % sec

avgPathGains = [0 -0.9 -4.9 -8 -7.8 -23.9]; % dB

fD = 50; % Hz

rayleighchan = comm.RayleighChannel('SampleRate',fs, ...

'PathDelays',pathDelays, ...

'AveragePathGains',avgPathGains, ...

'MaximumDopplerShift',fD);

faded\_signal = rayleighchan(tx);

faded\_signal\_plus\_noise = awgn(faded\_signal, snr, 'measured');

Recieve\_Channel = reshape(faded\_signal\_plus\_noise, [], numOfSym/2);

Receive\_Channel\_rayleigh = Recieve\_Channel;

1. Use another copy for each signal and pass it through the AWGN channel as we did before.
2. At the receiver side perform 4 times the FFT operation and 4 times the demodulation to recover the original bits using the proper demodulator.
3. You should obtain the following results:





For the report only upload the 2 scripts of Exercise A and Exercise B.

**OFDM application for image transmission:** (not included in the report)

In this lab and the previous one, we explored a simplified implementation of OFDM. However, more advanced implementations are possible. You can find a zip file named "OFDM\_IMAGE" on Teams, which contains an advanced implementation of OFDM following the standard. This implementation demonstrates how we can transmit an image and reconstruct it at the receiver side using OFDM.

Start by running the OFDM\_SIM script and insert the following parameters:

*source data filename: lena.bmp*

*IFFT size: 256*

*Number of carriers: 64*

*Modulation(1=BPSK, 2=QPSK, 4=16PSK, 8=256PSK):*

*Amplitude clipping introduced by communication channel (in dB):1*

*Signal-to-Noise Ratio (SNR) in dB: 30*

After running the code, please check the image named "lena\_OFDM" that was received.

- Run the code again, changing the modulation scheme to 1, 2, 4, and 8, and observe how this affects the reconstruction of the transmitted image and the time it takes in seconds.

- Run the code again, decrease the SNR to 10 dB, and observe how this affects the reconstructed image.