**THE NELSON MANDELA**

**AFRICAN INSTITUTION OF SCIENCE AND TECHNOLOGY**

**(NM-AIST)**



**ASSIGNMENT NUMBER 3:**

**UNDERSTAND AND GIVE A SUMMARY OF THE GIVEN CODE**

|  |  |
| --- | --- |
| **COURSE NAME:** | MOBILE TELECOMMUNICATION TECHNOLOGY |
| **COURSE CODE:** | EMoS 6222 |
| **LECTURER :** | DR. MUSSA Ally Dida |

**Assignment submitted by:**

|  |  |
| --- | --- |
| Carmel NKESHIMANA | MOBILE SYSTEMS (M068/BI21) |

**Assignment Content**

% Understand each line,

%briefly explain each line

% then write a summary of the whole

%code

clc; % clear all the text in command line   
clear all; % clear all the variable from the current work space it is the same like clearvars  
close all; % it is used to close all open figures   
%..............................................................  
% 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  
  
data = randsrc(1, no\_of\_data\_bits, 0:M-1); %this line will generate a 1-by-64 matrix with entry

% which are between 0 and 3

figure(1) %create a figure with propriety 1

stem(data); %stem function will plot the data sequence “data” as stems that extend from a %baseline. The data values are indicated by cycle terminating each stem. And we know “data” is % a vector, this means the x-axis will scale the range from 1 to length(data)

grid on; %this function help to display the major grid lines for the current axes returned by the %gca command and the major grid lines extend from each tick mark.

xlabel('Data Points'); %help to add label on the x-axis , and here the name is “Data Points”

ylabel('Amplitude'); %help to add label on the y-axis, and here the name is “Amplitude”  
title('Original Data '); %this function title insert title page on the beginning of the plot , and here %the title is “Original data”

% Perform QPSK modulation on the input source data

qpsk\_modulated\_data = pskmod(data, M); %the pksmod function modulate the given input %signal “data” (which is the vector obtain from last code) using M-Ary phase shift keying (M-%PSK). And M(4) specify the modulation order

figure(2) %create a figure with propriety 2

stem(qpsk\_modulated\_data); %stem function will plot the data sequence “qpsk\_modulated\_data” % as stems that extend from a baseline. The data values are indicated by cycle terminating each %stem. And we know “qpsk\_modulated\_data” is a vector, this means the x-axis will scale the range from 1 to %length(qpsk\_modulated\_data)

title('QPSK Modulation ') %this function title insert title page on the beginning of the plot, and here %the title is “QPSK Modulation”  
%............................................................  
  
S2P = reshape(qpsk\_modulated\_data, no\_of\_data\_bits/M,M) % reshape function reshapes the %array qpsk\_modulated\_data into a no\_of\_data\_bits/M-by-M array where no\_of\_data\_bits/M and %M indicates the size of each array

%converting the series data stream into four parallel data stream to form using S2P function for all row element and for each column 1,2,3,4

% those four subcarriers are Sub\_carrier1, Sub\_carrier2, Sub\_carrier3, Sub\_carrier4  
Sub\_carrier1 = S2P(:,1)   
Sub\_carrier2 = S2P(:,2)  
Sub\_carrier3 = S2P(:,3)  
Sub\_carrier4 = S2P(:,4)

figure(3) %creation of a figure with propriety 3

subplot(4, 1,1) % divide the current figure into a 4-by-1 grid and create axes in the position %specified by 1

stem(Sub\_carrier1) %stem function will plot the data sequence “Sub\_carrier1” as stems that %extend from a baseline. The data values are indicated by cycle terminating each stem. And we %know “Sub\_carrier1” is a vector, this means the x-axis will scale the range from 1 to %length(Sub\_carrier1)

title('Subcarrier1') ) %this function title insert title page on the beginning of the plot, and here %the title is “'Subcarrier1'”

grid on; %this function help to display the major grid lines for the current axes returned by the %gca command and the major grid lines extend from each tick mark.

subplot(4,1,2) ) % divide the current figure into a 4-by-1 grid and create axes in the position %specified by 2

stem(Sub\_carrier2) %stem function will plot the data sequence “Sub\_ carrier2” as stems that %extend from a baseline. The data values are indicated by cycle terminating each stem. And we %know “Sub\_ carrier2” is a vector, this means the x-axis will scale the range from 1 to %length(Sub\_ carrier2)

title('Subcarrier2') %this function title insert title page on the beginning of the plot, and here %the title is “'Subcarrier2'”

grid on; %this function help to display the major grid lines for the current axes returned by the %gca command and the major grid lines extend from each tick mark.

subplot(4,1,3) % divide the current figure into a 4-by-1 grid and create axes in the position %specified by 3

stem(Sub\_carrier3) %stem function will plot the data sequence “Sub\_ carrier3” as stems that %extend from a baseline. The data values are indicated by cycle terminating each stem. And we %know “Sub\_ carrier3” is a vector, this means the x-axis will scale the range from 1 to %length(Sub\_ carrier3)

title('Subcarrier3') %this function title insert title page on the beginning of the plot, and here %the title is “'Subcarrier3'”

grid on; %this function help to display the major grid lines for the current axes returned by the %gca command and the major grid lines extend from each tick mark.

subplot(4,1,4) % divide the current figure into a 4-by-1 grid and create axes in the position %specified by 4

stem(Sub\_carrier4) %stem function will plot the data sequence “Sub\_ carrier4” as stems that %extend from a baseline. The data values are indicated by cycle terminating each stem. And we %know “Sub\_ carrier4” is a vector, this means the x-axis will scale the range from 1 to %length(Sub\_ carrier4)

title('Subcarrier4') ) %this function title insert title page on the beginning of the plot, and here %the title is “'Subcarrier4'”

grid on; %this function help to display the major grid lines for the current axes returned by the %gca command and the major grid lines extend from each tick mark.

%..................................................................  
%..................................................................  
% IFFT OF FOUR SUB\_CARRIERS  
%.................................................................  
%..............................................................  
number\_of\_subcarriers=4; %declaration of number of subcarriers which are 4  
cp\_start=block\_size-cp\_len; %our cp\_start variable equal to Size of each OFDM block minus %Length of the cyclic prefix

ifft\_Subcarrier1 = ifft(Sub\_carrier1) % compute the inverse discrete Fourier transform of %Sub\_carrier1 using fast Fourier transform algorithm . ifft\_Subcarrier1 is the same size as %Sub\_carrier1  
ifft\_Subcarrier2 = ifft(Sub\_carrier2) % compute the inverse discrete Fourier transform of %Sub\_carrier1 using fast Fourier transform algorithm . ifft\_ Subcarrier2 is the same size as %Sub\_ carrier2  
  
ifft\_Subcarrier3 = ifft(Sub\_carrier3) % compute the inverse discrete Fourier transform of %Sub\_carrier3 using fast Fourier transform algorithm . ifft\_Subcarrier3 is the same size as %Sub\_carrier3  
  
ifft\_Subcarrier4 = ifft(Sub\_carrier4) % compute the inverse discrete Fourier transform of %Sub\_carrier4 using fast Fourier transform algorithm . ifft\_Subcarrier4 is the same size as %Sub\_ carrier4  
  
figure(4) %creation of a figure with propriety 4

subplot(4,1,1) % divide the current figure into a 4-by-1 grid and create axes in the position %specified by 1

plot(real(ifft\_Subcarrier1),'r') % real function returns the part of each element in array %ifft\_Subcarrier1 and the results act like the input for the plot function which will draw a graph %of red color (specified with the ‘r’ :second parameter of the function)

title('IFFT on all the sub-carriers') %this function title insert title page on the beginning of the plot, and here %the title is “IFFT on all the sub-carriers”

subplot(4,1,2) % divide the current figure into a 4-by-1 grid and create axes in the position %specified by 2

plot(real(ifft\_Subcarrier2),'c') % real function returns the part of each element in array %ifft\_Subcarrier2 and the results act like the input for the plot function which will draw a graph %of cyan color (specified with the ‘c’ :second parameter of the function)

subplot(4,1,3) % divide the current figure into a 4-by-1 grid and create axes in the position %specified by 3

plot(real(ifft\_Subcarrier3),'b') % real function returns the part of each element in array %ifft\_Subcarrier3 and the results act like the input for the plot function which will draw a graph %of blue color (specified with the ‘b’ :second parameter of the function)

subplot(4,1,4) ) % divide the current figure into a 4-by-1 grid and create axes in the position %specified by 4

plot(real(ifft\_Subcarrier4),'g') % real function returns the part of each element in array %ifft\_Subcarrier3 and the results act like the input for the plot function which will draw a graph %of green color (specified with the ‘g’ :second parameter of the function)

%...........................................................  
%...........................................................  
% ADD-CYCLIC PREFIX

%..........................................................  
%............................................................

for i=1:number\_of\_subcarriers, % this is a for loop which starts from 1 to the %number\_of\_subcarriers (number of subcarriers equal to 4)

ifft\_Subcarrier(:,i) = ifft((S2P(:,i)),16); % we use the matrix ifft\_Subcarrier for all the elements %from each row at column number i, we put the compute the inverse discrete Fourier transform %(of conversion of the series data stream into a parallel data stream to form using S2P function %for all row elements and for each column i) using fast Fourier transform algorithm by padding %SP2(:,i) with trailing zeros to length 16

for j=1:cp\_len, % this is a for loop which starts from 1 to the %cp\_len

cyclic\_prefix(j,i) = ifft\_Subcarrier(j+cp\_start,i) %for the matrice cyclic\_prefix at the column i %and column j we add the value obtained from ifft\_Subcarrier at j+cp\_start row and column i

end

Append\_prefix(:,i) = vertcat( cyclic\_prefix(:,i), ifft\_Subcarrier(:,i)) %Append\_prefix matrix for %all row’s elements we add the concatenates ifft\_Subcarrier(:,i) vertically to the end of %cyclic\_prefix(:,i) when cyclic\_prefix(:,i) and ifft\_Subcarrier(:,i) have a compatible sizes.

% Appends prefix to each subcarriers  
end

% Appends prefix to each subcarrier from 1 to 4

A1=Append\_prefix(:,1);  
A2=Append\_prefix(:,2);  
A3=Append\_prefix(:,3);  
A4=Append\_prefix(:,4);

figure(5) %creation of a figure with propriety 5

subplot(4,1,1) % divide the current figure into a 4-by-1 grid and create axes in the position %specified by 1

plot(real(A1),'r') % real function returns the part of each element in array A1 and the results %act like the input for the plot function which will draw a graph of red color (specified with the

% ‘r’: second parameter of the function)

title('Cyclic prefix added to all the sub-carriers') %this function title insert title page on the %beginning of the plot, and here the title is “Cyclic prefix added to all the sub-carriers”  
subplot(4,1,2) % divide the current figure into a 4-by-1 grid and create axes in the position %specified by 2

plot(real(A2),'c') % real function returns the part of each element in array A2 and the results %act like the input for the plot function which will draw a graph of cyan color (specified with the

% ‘c’: second parameter of the function)  
subplot(4,1,3) % divide the current figure into a 4-by-1 grid and create axes in the position %specified by 3

plot(real(A3),'b') % real function returns the part of each element in array A3 and the results %act like the input for the plot function which will draw a graph of blue color (specified with the

% ‘b’: second parameter of the function)  
subplot(4,1,4) % divide the current figure into a 4-by-1 grid and create axes in the position %specified by 4

plot(real(A4),'g') % real function returns the part of each element in array A4 and the results %act like the input for the plot function which will draw a graph of green color (specified with %the ‘g’: second parameter of the function)  
figure(11) %creation of a figure with propriety 11

plot((real(A1)),'r') % real function returns the part of each element in array A4 and the results %act like the input for the plot function which will draw a graph of red color (specified with %the ‘r’: second parameter of the function)

title('Orthogonality') %this function title insert title page on the %beginning of the plot, and here the title is “Orthogonality”

hold on, %retains plots in the current axes so that new plots added to the axes do not delete %existing plots.

plot((real(A2)),'c') % real function returns the part of each element in array A2 and the results %act like the input for the plot function which will draw a graph of cyan color (specified with %the ‘c’: second parameter of the function)

hold on, %retains plots in the current axes so that new plots added to the axes do not delete %existing plots.  
plot((real(A3)),'b') % real function returns the part of each element in array A4 and the results %act like the input for the plot function which will draw a graph of blue color (specified with %the ‘b’: second parameter of the function)

hold on, %retains plots in the current axes so that new plots added to the axes do not delete %existing plots.

plot((real(A4)),'g') % real function returns the part of each element in array A4 and the results %act like the input for the plot function which will draw a graph of green color (specified with %the ‘g’: second parameter of the function)

hold on, %retains plots in the current axes so that new plots added to the axes do not delete %existing plots.

grid on %this function help to display the major grid lines for the current axes returned by the %gca command and the major grid lines extend from each tick mark.

%Convert to serial stream for transmission  
[rows\_Append\_prefix cols\_Append\_prefix]=size(Append\_prefix) %size function return the %lengths of the queried dimensions of Append\_prefix separately and put the row length in %rows\_Append\_prefix and column length in cols\_Append\_prefix

len\_ofdm\_data = rows\_Append\_prefix\*cols\_Append\_prefix % for to get the length of the %OFDM data we use the results obtained by multiplication of rows\_Append\_prefix and %cols\_Append\_prefix

% OFDM signal to be transmitted  
ofdm\_signal = reshape(Append\_prefix, 1, len\_ofdm\_data); ) % reshape function reshapes the %array Append\_prefix into a 1-by- len\_ofdm\_data array where 1 and len\_ofdm\_data indicates %the size of each array

figure(6) %creation of a figure with propriety 6

plot(real(ofdm\_signal)); % real function returns the part of each element in array ofdm\_signal and the results act like the input for the plot function which will draw a graph

xlabel('Time'); %help to add label on the x-axis, and here the name is “Times”

ylabel('Amplitude'); %help to add label on the y-axis, and here the name is “Amplitude”

title('OFDM Signal'); %this function title insert title page on the %beginning of the plot, and here the title is “OFDM Signal”

grid on; %like recently said this function help to display the major grid lines for the current axes returned by the gca command and the major grid lines extend from each tick mark.  
%...............................................................

**%Passing time domain data through channel and AWGN**

%.............................................................  
channel = randn(1,2) + sqrt(-1)\*randn(1,2);

%here we have 2 differents functions which are randn and sqrt

%randn(1,2) returns a 1-by-2 array(or matrix) of random numbers where 1 and 2 indicate size of each dimension

%sqrt(-1) return the square root of -1

%this will help to compute the channel size by adding to randn(1,2) the product result of sqrt(-1) %and randn(1,2)

after\_channel = filter(channel, 1, ofdm\_signal); %filter function filters the input data ofdm\_signal using a rational transfer function defined by the numerator and denominator coefficients 1 and channel

awgn\_noise = awgn(zeros(1,length(after\_channel)),0); %zeros(1,length(after\_channel) function %returns an 1-by-length(after\_channel) array(or matrix) where 1 and length(after\_channel) %indicate the size of each dimension. The result obtain become a parameter with 0 for the %function awgn which adds white Gaussian to the vector signal zeros(1,length(after\_channel))

recvd\_signal = awgn\_noise+after\_channel; % the received signal equal to the result given by adding awgn\_noise and after\_channel value  
figure(7) %creation of a figure with propriety 7

plot(real(recvd\_signal)) % real function returns the part of each element in array recvd\_signal and the results act like the input for the plot function which will draw a graph

xlabel('Time'); %help to add label on the x-axis, and here the name is “Time”

ylabel('Amplitude'); %help to add label on the y-axis, and here the name is “Amplitude”  
title('OFDM Signal after passing through channel'); %this function title insert title page on the %beginning of the plot, and here the title is “OFDM Signal after passing through channel”

grid on; %like recently said this function help to display the major grid lines for the current axes returned by the gca command and the major grid lines extend from each tick mark.  
  
%...........................................................

**%OFDM receiver part**

%..........................................................  
recvd\_signal\_paralleled = reshape(recvd\_signal, rows\_Append\_prefix, cols\_Append\_prefix); % reshape function reshapes the array recvd\_signal into a rows\_Append\_prefix -by- %cols\_Append\_prefix array where rows\_Append\_prefix and cols\_Append\_prefix indicates the %size of each array

%........................................................  
%........................................................  
% Remove cyclic Prefix  
%.......................................................  
%......................................................

recvd\_signal\_paralleled(1:cp\_len,:)=[];%this row off code helps to remove value in the matrix %recvd\_signal\_paralleled from the row 1 to cp\_len for all column of the matrix.

R1=recvd\_signal\_paralleled(:,1); %this code helps to put in R1 a vector matrix from the matrix %recvd\_signal\_paralleled for all rows but specially use the first column only.  
R2=recvd\_signal\_paralleled(:,2); %this code helps to put in R2 a vector matrix from the matrix %recvd\_signal\_paralleled for all rows but specially use the second column only.  
R3=recvd\_signal\_paralleled(:,3); %this code helps to put in R3 a vector matrix from the matrix %recvd\_signal\_paralleled for all rows but specially use the third column only.  
R4=recvd\_signal\_paralleled(:,4); %this code helps to put in R4 a vector matrix from the matrix %recvd\_signal\_paralleled for all rows but specially use the fourth column only.  
figure(8), %creation of a figure with propriety 8

plot((imag(R1)),'r'), % real function returns imaginary part of each element in array R1 and the %results act like the input for the plot function which will draw a graph of green color (specified %with the ‘g’: second parameter of the function)

subplot(4,1,1), % divide the current figure into a 4-by-1 grid and create axes in the position %specified by 1

plot(real(R1),'r'), % real function returns the part of each element in array R1 and the results %act like the input for the plot function which will draw a graph of red color (specified with %the ‘r’: second parameter of the function)

title('Cyclic prefix removed from the four sub-carriers') %this function title insert title page on the %beginning of the plot, and here the title is “Cyclic prefix removed from the four sub-carriers”  
subplot(4,1,2), % divide the current figure into a 4-by-1 grid and create axes in the position %specified by 2

plot(real(R2),'c') % real function returns the part of each element in array R2 and the results act %like the input for the plot function which will draw a graph of cyan color (specified with the %‘c’: second parameter of the function)

subplot(4,1,3), % divide the current figure into a 4-by-1 grid and create axes in the position %specified by 3

plot(real(R3),'b') % real function returns the part of each element in array R3 and the results act %like the input for the plot function which will draw a graph of blue color (specified with the %‘b’: second parameter of the function)  
subplot(4,1,4), % divide the current figure into a 4-by-1 grid and create axes in the position %specified by 4

plot(real(R4),'g') % real function returns the part of each element in array R4 and the results act %like the input for the plot function which will draw a graph of green color (specified with the ‘g’: %second parameter of the function)

%...................................................  
%...................................................  
% FFT Of received signal  
for i=1:number\_of\_subcarriers, % this is a for loop which starts from 1 to the %number\_of\_subcarriers  
% FFT  
fft\_data(:,i) = fft(recvd\_signal\_paralleled(:,i),16); % we use the matrix recvd\_signal\_paralleled %for all the elements from each row at column number i, we put the compute the inverse %discrete Fourier transform (of conversion of the series data stream into a parallel data stream %to form using recvd\_signal\_paralleled vector constitute with all row combined with the column %i using fast Fourier transform algorithm by padding recvd\_signal\_paralleled (:,i) with trailing %zeros to length 16  
  
end  
F1=fft\_data(:,1); %this code helps to put in F1 a vector matrix from the matrix fft\_data %paralleled for all rows but specially use the first column only.  
F2=fft\_data(:,2); %this code helps to put in F2 a vector matrix from the matrix fft\_data %paralleled for all rows but specially use the second column only.  
F3=fft\_data(:,3); %this code helps to put in F3 a vector matrix from the matrix fft\_data %paralleled for all rows but specially use the third column only.  
F4=fft\_data(:,4); %this code helps to put in F4 a vector matrix from the matrix fft\_data %paralleled for all rows but specially use the fourth column only.

figure(9) %creation of a figure with propriety 9

subplot(4,1,1) ), % divide the current figure into a 4-by-1 grid and create axes in the position %specified by 1

plot(real(F1),'r') % real function returns the part of each element in array F1 and the results act %like the input for the plot function which will draw a graph of red color (specified with the %‘r’: second parameter of the function)

title('FFT of all the four sub-carriers') %this function title insert title page on the beginning of the %plot, and here the title is “FFT of all the four sub-carriers”  
  
subplot(4,1,2) ), % divide the current figure into a 4-by-1 grid and create axes in the position %specified by 2

plot(real(F2),'c') % real function returns the part of each element in array F2 and the results act %like the input for the plot function which will draw a graph of cyan color (specified with the %‘c’: second parameter of the function)  
subplot(4,1,3) ), % divide the current figure into a 4-by-1 grid and create axes in the position %specified by 3

plot(real(F3),'b') % real function returns the part of each element in array F1 and the results act %like the input for the plot function which will draw a graph of blue color (specified with the %‘b’: second parameter of the function)  
subplot(4,1,4) ), % divide the current figure into a 4-by-1 grid and create axes in the position %specified by 4

plot(real(F4),'g') % real function returns the part of each element in array F4 and the results act %like the input for the plot function which will draw a graph of green color (specified with the %‘g’: second parameter of the function)

%................................  
%..............................  
% Signal Reconstructed  
%..................................  
%..................................  
% Conversion to serial and demodulations

recvd\_serial\_data = reshape(fft\_data, 1,(16\*4)); % reshape function reshapes the array

% fft\_data into a 1-by-(16\*4) array where 1 and (16\*4) indicates the size of each array

qpsk\_demodulated\_data = pskdemod(recvd\_serial\_data,4); %pskdemod demodulates the input M-PSK signals recvd\_serial\_data. 4 specifies the modulation order.   
figure(10) %creation of a figure with propriety 10  
stem(data) %stem function will plot the data sequence “data” as stems that extend from a %baseline. The data values are indicated by cycle terminating each stem. And we know “data” is % a vector, this means the x-axis will scale the range from 1 to length(data)  
hold on, %retains plots in the current axes so that new plots added to the axes do not delete %existing plots.  
stem(qpsk\_demodulated\_data,'rx'); %stem function will plot the data sequence %“qpsk\_demodulated\_data” at values specified by ‘rx’. The qpsk\_demodulated\_data and rx %inputs must be vectors or matrices of the same size.   
grid on; %like recently said this function help to display the major grid lines for the current axes %returned by the gca command and the major grid lines extend from each tick mark.

xlabel('Data Points'); %help to add label on the x-axis, and here the name is “Data Points”

ylabel('Amplitude'); %help to add label on the y-axis, and here the name is “Amplitude”  
title('Recieved Signal with error') %this function title insert title page on the beginning of the %plot, and here the title is “Recieved Signal with error”

**Summary: This code aims to show the design of modulation of OFDM in practice.**

OFDM is a frequency-division multiplexing (FDM) scheme used as a digital multi-carrier modulation method. A large number of closely spaced orthogonal sub-carrier signals are used to carry data on several parallel data streams or channels. Each sub-carrier is modulated with a conventional modulation scheme (such as quadrature amplitude modulation or phase-shift keying) at a low symbol rate, maintaining total data rates similar to conventional single-carrier modulation schemes in the same bandwidth.

The primary advantage of OFDM over single-carrier schemes is its ability to cope with severe channel conditions (for example, attenuation of high frequencies in a long copper wire, BE(EXTC) SEM VII (RCOE) narrowband interference and frequency-selective fading due to multipath) without complex equalization filters. Channel equalization is simplified because OFDM may be viewed as using many slowly modulated narrowband signals rather than one rapidly modulated wideband signal. The low symbol rate makes the use of a guard interval between symbols affordable, making it possible to eliminate intersymbol interference (ISI) and utilize echoes and time-spreading (on analogue TV these are visible as ghosting and blurring, respectively) to achieve a diversity gain, i.e. a signal-to-noise ratio improvement. This mechanism also facilitates the design of single frequency networks (SFNs), where several adjacent transmitters send the same signal simultaneously at the same frequency, as the signals from multiple distant transmitters may be combined constructively, rather than interfering as would typically occur in a traditional single-carrier system.