THE NELSON MANDELA AFRICAN INSTITUTION OF SCIENCE AND TECHNOLOGY (NM-AIST)

COURSE TITLE: EMoS 6222, MOBILE TELECOMMUNICATION AND TECHNOLOGY

LECTURER: Dr.Ally Mussa Dida

STUDENT: Salama Ndayisaba

Specialization: Embedded Sytems

Reg. Number: M083/BI19

Assignment: Matlab Codes

Submitted on 3rd October 2022

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 variables is the same like workspace
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 code has to generate a 1-by-64 matrix with entry

% which are between 0 and 3

figure(1) %creation of a figure with propriety 1

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

grid on; %this function helps the displaying of 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'); %this code helps to add a label on the x-axis, and here the name is “Data Points”

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

title('Original Data '); %this function title inserts 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 code modulates the given input %signal “data” (which is the vector obtained from the last code) using M-Ary phase shift keying (M- %PSK). And M(4) specifies the modulation order

figure(2) %create a figure with propriety 2

stem(qpsk_modulated_data); %stem function plots the data sequence “qpsk_modulated_data” % as stems that extend from a baseline. The data values are indicated by the cycle terminating each %stem. And we know “qpsk_modulated_data” is a vector, this means that 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) %this code reshapes 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) %is to 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 the 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)) % this has to 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 the 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 has to 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) %this has to 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 the 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 inserts 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 the 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; %this code is to declare the 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) %this code gives 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) %This function gives 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) %this code is to give the inverse discrete Fourier transform of %Sub_carrier3 using a fast Fourier transform algorithm. ifft_Subcarrier3 is the same size as %Sub_carrier3

ifft_Subcarrier4 = ifft(Sub_carrier4) %this give 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); % this code is used to 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, %this retains plots in the current axes so that new plots added to the axes do not delete %existing plots.

plot((real(A3)),'b') %this 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') % this 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, %this 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 the array ofdm_signal and the results act like the input for the plot function which will draw a graph

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

ylabel('Amplitude'); %help to add a 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 different 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 the array recvd_signal and the results act like the input for the plot function which will draw a graph

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

ylabel('Amplitude'); %help to add a 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 especially uses 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 especially uses 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 especially uses 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 especially uses the fourth column only.

figure(8), %creation of a figure with propriety 8

plot((imag(R1)),'r'), % real function returns the 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 subcarriers”

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 especially uses 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 especially uses 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 especially uses 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 the 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 “Received Signal with error”
```

The summary of all the work

As is known, OFDM in the full sentence is the Orthogonal Frequency-Division Multiple (OFDM). This scheme is 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 main advantage of OFDM is the making efficient use of the spectrum by allowing overlap. And by dividing the channel into narrowband flat fading subchannels, OFDM is more resistant to frequency selective fading than single carrier systems are. 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 analog 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.

So, Matlab is a powerful and useful tool for OFDM because it manages time.