Faculty of Engineering

Cairo University

# OFDM Project

Submitted to. Eng. Alaa Khairallah & prof.\Mohamed Khairy

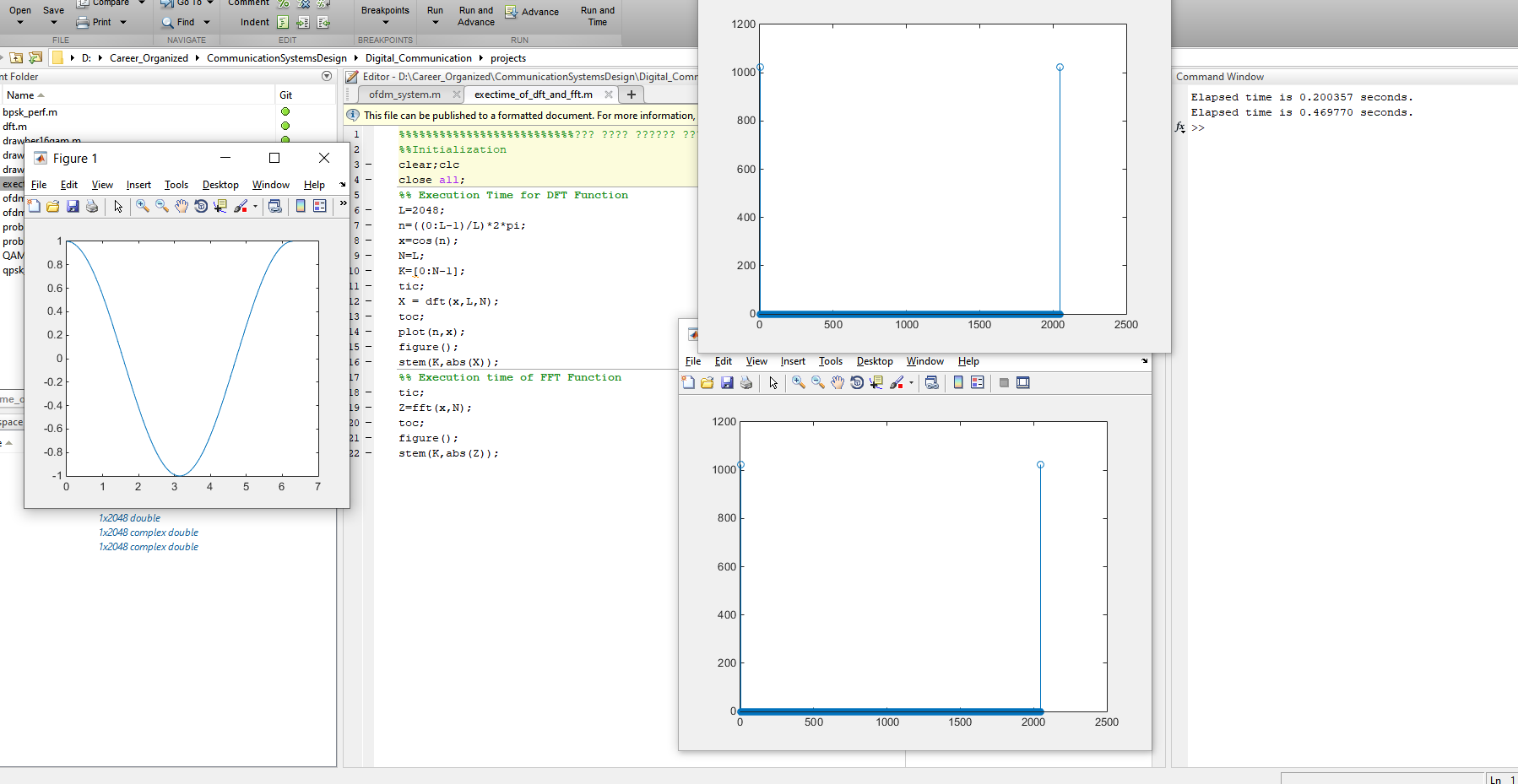
|  |  |  |
| --- | --- | --- |
| Name | Sec | BN |
| Hisham Mohamed Abdallah Naguib | 4 | 51 |

# Table of Contents:

# Problem 1:

Elapsed time for DFT: 0.469770 seconds

Elapsed time for FFT: 0.200357 seconds



# 

# Problem 2:

## **Transmitter Model:**

Generating BitStream and grey Encode them:

%% 16-QAM Transmitter Model

N=100; % Number of Samples

M=16; % Number bits per symbol

%-------------------Data Generation---------------------------%

data=randi([0 M-1],N,1);

%data=[0:M-1];

%-----------Grey Encoding------------%

[datagrey,mapgrey] = bin2gray(data,'qam',M);

Mapping Symbols to Constellations:

%-------------------16-QAM Modulation----------------------------%

consttable=[-3-3i, -3-1i, -3+3i, -3+1i,...

-1-3i, -1-1i, -1+3i, -1+1i,...

3-3i, 3-1i, 3+3i, 3+1i,...

1-3i, 1-1i, 1+3i, 1+1i];

for k=1:length(datagrey)

tx(k) = consttable(datagrey(k)+1);

end

tx=tx(:);

scatterplot(tx,1,0,'b\*');

for k = 1:16

text(real(tx(k))-0.3,imag(tx(k))+0.3,...

dec2base(data(k),2,4));

text(real(tx(k))-0.3,imag(tx(k))-0.3,...

dec2base(datagrey(k),2,4),'Color',[1 0 0]);

end

title('Transmitted Symbols');

## Fading Channel Model:

%% Rayleigh-Fading Channel Model

hr=normrnd(0,sqrt(0.5),1);

hi=normrnd(0,sqrt(0.5),1);

h=(hr+1i\*hi)\*ones(1,N);

h=h(:);

rx=tx.\*h;

scatterplot(rx)

title('Rayleigh-Fading Channel Effect');

## **AWGN Channel Model:**

%% AWGN Channel

%AWGN Noise

mu=0;

No=0.1;

variance=No/2;

sigma=sqrt(variance);

nc=normrnd(mu,sigma,[1,N]);

ns=normrnd(mu,sigma,[1,N]);

n=nc+1i\*ns;

n=n(:);

yk=rx+n;

scatterplot(yk);

title('AWGN Channel Effect');

## Equalizer:

%Equalizer assuming channel is known

yk=yk/h;

## Correlation and Decision Model:

%% Correlator & Decision Model

consttable=[-3-3i, -3-1i, -3+3i, -3+1i,...

-1-3i, -1-1i, -1+3i, -1+1i,...

3-3i, 3-1i, 3+3i, 3+1i,...

1-3i, 1-1i, 1+3i, 1+1i];

for N = 1:length(yk)

%compute the minimum distance for each symbol

[~, idx] = min(abs(yk(N) - consttable));

datademod(N) = idx-1;

end

[datademodbin,mapbin] = gray2bin(datademod,'qam',M);

datademodbin=datademodbin(:);

## BitError Rate Estimation:

%% BER

BER=biterr(datademodbin,data)/(length(data)\*log2(M));

ALL above procedures are same for different types of modulation we just change params like M for number of levels, constellation tables

**BPSK BER Before & After Repetition:**

**QPSK BER Before & After Repetition:**

**16-QAM BER Before & After Repetition:**

# Problem 3:

## Coding

## Interleaver

## Mapper

## 32-point IFFT

## Add Cyclic Extension

## Channel

## Reciever

## Results:

### QPSK Flat channel & Coding

### QPSK Selective channel & Coding

### QPSK Flat channel & No Coding

### QPSK Selective channel & No Coding

### 16-QAM Flat channel & Coding

### 16-QAM Selective channel & Coding

### 16-QAM Flat channel & No Coding

### 16-QAM Selective channel & No Coding

# Full Code:

## Problem 1:

## Problem 2:

## Problem 3: