**Lab 08**

**Investigating CDMA transceiver implementation for multiple users**

Objectives

* Implementation of CDMA transceiver for multiple users in MATLAB.

Code Division Multiple Access (CDMA) was an innovative use of direct sequence spread spectrum technology used to provide a multiple access scheme for mobile telecommunications (e.g., 3G) and other wireless systems.

In CDMA, different users can transmit using the same frequency at the same time with different Codes (by using the concept of DSSS).

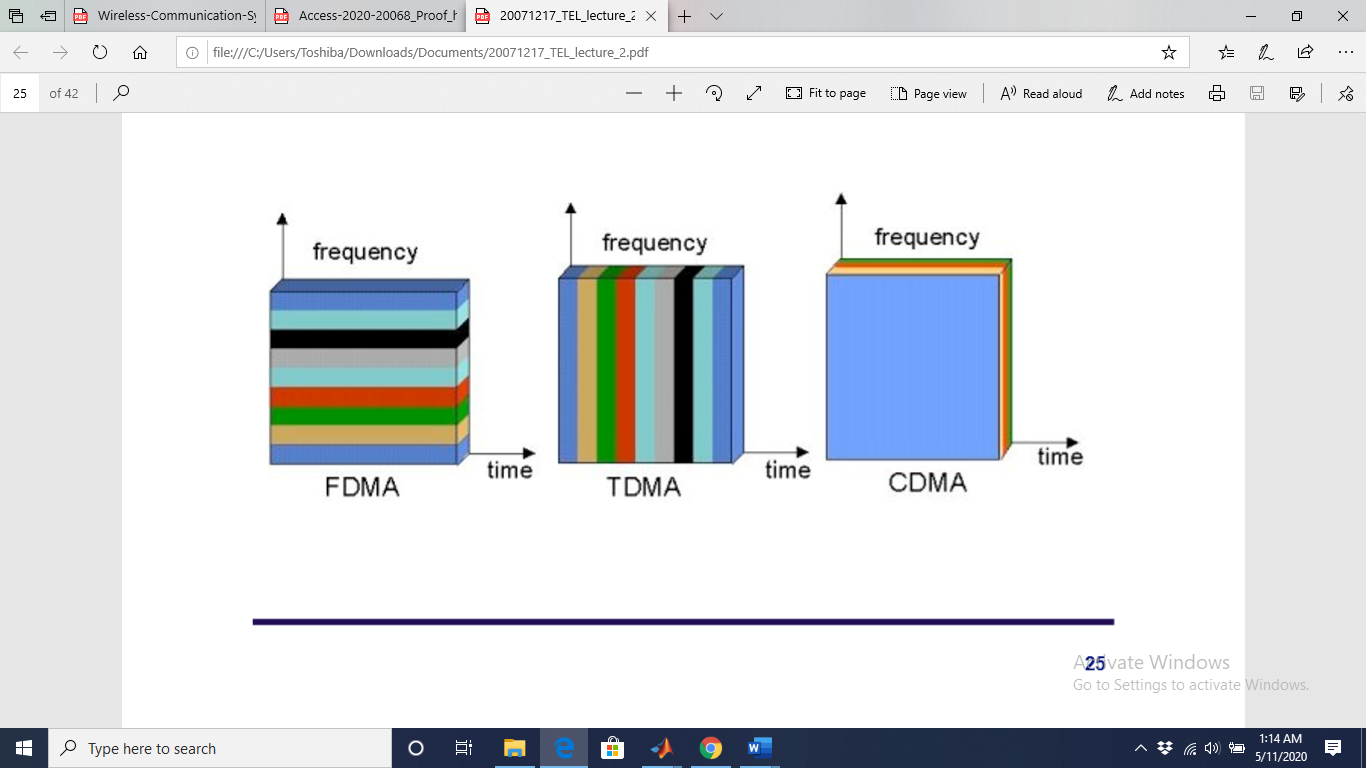


Figure: Different multiple access schemes

We will implement a CDMA MODEM in which different users are transmitting simultaneously using the same frequency with Gaussian noise. Each user will receive a composed signal consisting of the signals corresponding to different users and the Gaussian noise.

Each user at the receiver side will receive this noisy signal, and after multiplying it with the pseudo code, will be able to decode the data correctly.

In the simulation we are going to perform the following steps:

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

% Description of program

%

% 1. Convert input bits to bipolar bits. 1 to 1 and 0 to -1 for user1 and user2.

% 2. Take 100 samples per bit for both user1 and user2 and then plot base band signal which is in bipolar NRZ (non-return-to-zero) format.

% 3. Then modulate the signal using BPSK. Take care that sampling rate of sinusoidal carrier matches the sampling rate per bit.

Here it is 100 samples per carrier and then plot the BPSK signal.

% 4. perform the function of exclusive or gate and shift registers.

5. Multiply the BPSK modulated signal with the PN code. Here again we should ensure that the no. of chip per bit\* no of samples per chip = no of samples per bit of BPSK modulated signal.

% 6. Same procedure is carried out for user2 bits.

% 7. The channel is AWGN channel with SNR 5 dB. In channel the signal from user1 is added to signal from user2 and to white Gaussian noise.

% 8. At receiver end, first received signal is multiplied with PN then BPSK demodulated by multiplying with the carrier.

% 9. then the samples over 1 bit interval is summed. And if the sum is greater than 0 then the received bit is 1 else rxbit is 0. Summation is used in place of integration because it is a discrete time system.

% 10. Same procedure is repeated for user2.

**Let us start the implementation step-by-step:**

1) Set Configuration parameters

clc

close all

clear all

%% Initial parameters

SNR = 5; % Signal to Noise ratio in dB

Fc = 1; % carrier frequency

Eb = 2; % energy per bit

Tb = 1; % bit duration

%% Consider two mobile users in the system

mobileUser1 = [1 0 0 1 1 0]; % data transmitted by user 1

mobileUser2 = [1 1 0 1 0 0]; % data transmitted by user 2

length\_mobileUser1 = length(mobileUser1);

length\_mobileUser2 = length(mobileUser2);

%% Convert data sequence into NRZ format

for i = 1:length\_mobileUser1

if mobileUser1(i )== 0

mobileUser1(i) = -1;

end

end

for i = 1:length\_mobileUser2

if mobileUser2(i) == 0

mobileUser2(i) = -1;

end

end

2) CDMA Transmitter Section

USER 1

user1basebandsig1 = [];

for i = 1:length\_mobileUser1

for j = 0.01:0.01:Tb % 100 samples per bit

if mobileUser1(i) == 1

user1basebandsig1 = [user1basebandsig1 1];

else

user1basebandsig1 = [user1basebandsig1 -1];

end

end

end

% BPSK MODULATION FOR USER 1

user1bpskmod1 = [];

for i = 1:length\_mobileUser1

for j = 0.01:0.01:Tb % 100 samples per bit

user1bpskmod1 = [user1bpskmod1 sqrt(2\*Eb)\*mobileUser1(i)\*cos(2\*pi\*Fc\*j)];

end

end

figure(1);

subplot(211)

plot(user1basebandsig1);

axis([0 100\*length\_mobileUser1 -1.5 1.5]);

title('Original Binary Sequence for User1 is')

grid on

subplot(212)

plot(user1bpskmod1);

title(' BPSK Signal for User 1 is');

grid on

%% PN generator for user1

%% PN generator for user1

% let initial seed for user1 is 1010

seed1 = [1 -1 1 -1];

pn1\_user\_1 = [];

% The following for loop performs the function of exclusive or gate and shift registers (with respect to the CDMA standard).

for i = 1:length\_mobileUser1

for j = 1:10 %chip rate is 10 times the bit rate

pn1\_user\_1 = [pn1\_user\_1 seed1(4)];

if seed1(4) == seed1(3)

temp = -1;

else

temp = 1;

end

seed1(4) = seed1(3);

seed1(3) = seed1(2);

seed1(2) = seed1(1);

seed1(1) = temp;

end

end

pnupsampled1\_User\_1 = [];

len\_pn1\_user\_1 = length(pn1\_user\_1);

for i = 1:len\_pn1\_user\_1

for j = 0.1:0.1:Tb

if pn1\_user\_1(i) == 1

pnupsampled1\_User\_1 = [pnupsampled1\_User\_1 1];

else

pnupsampled1\_User\_1 = [pnupsampled1\_User\_1 -1];

end

end

end

sigtx1\_User\_1 = user1bpskmod1.\*pnupsampled1\_User\_1;

figure(2)

subplot(311)

stem(pn1\_user\_1);

axis([0,length(pn1\_user\_1),-1.2,1.2])

title('PN sequence for user1')

subplot(312)

stem(pnupsampled1\_User\_1);

axis([0,length(pnupsampled1\_User\_1),-1.2,1.2])

title('PN sequence for user1 upsampled');

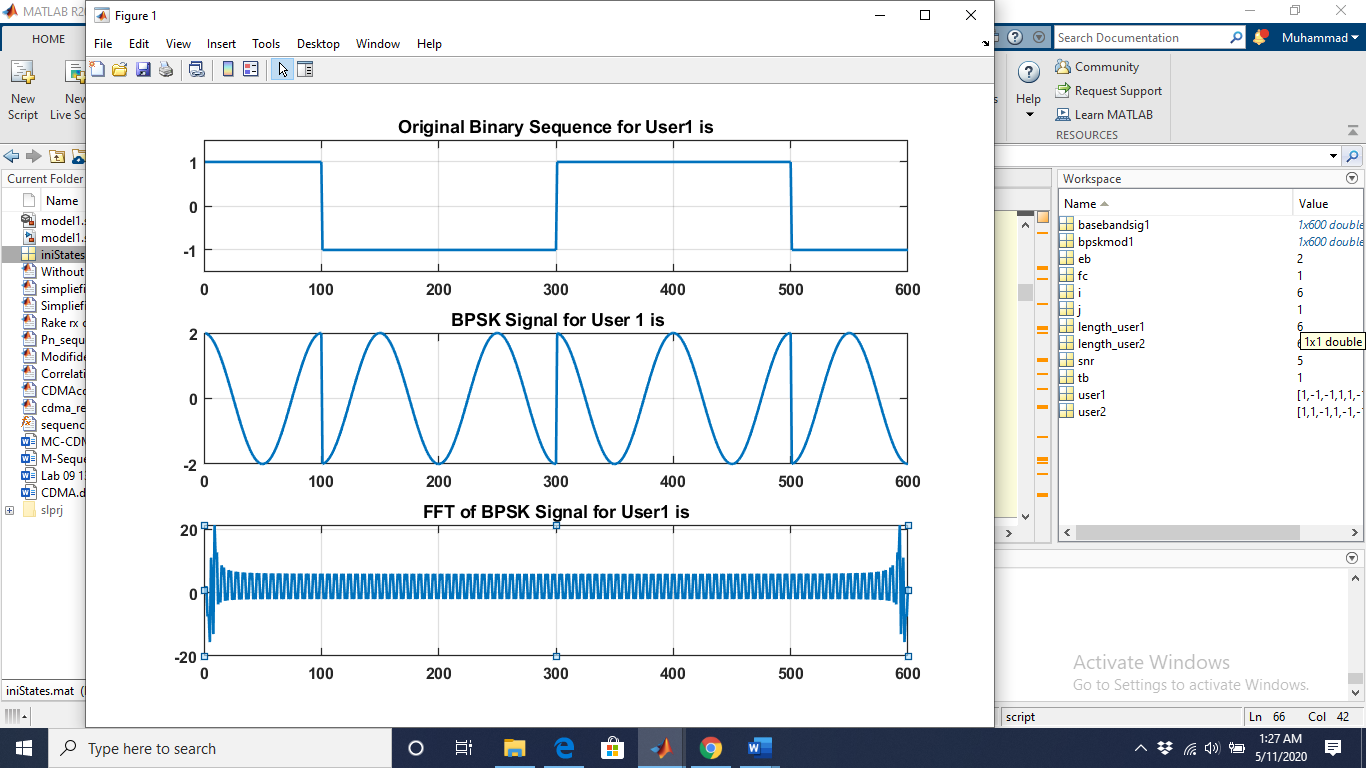
subplot(313)

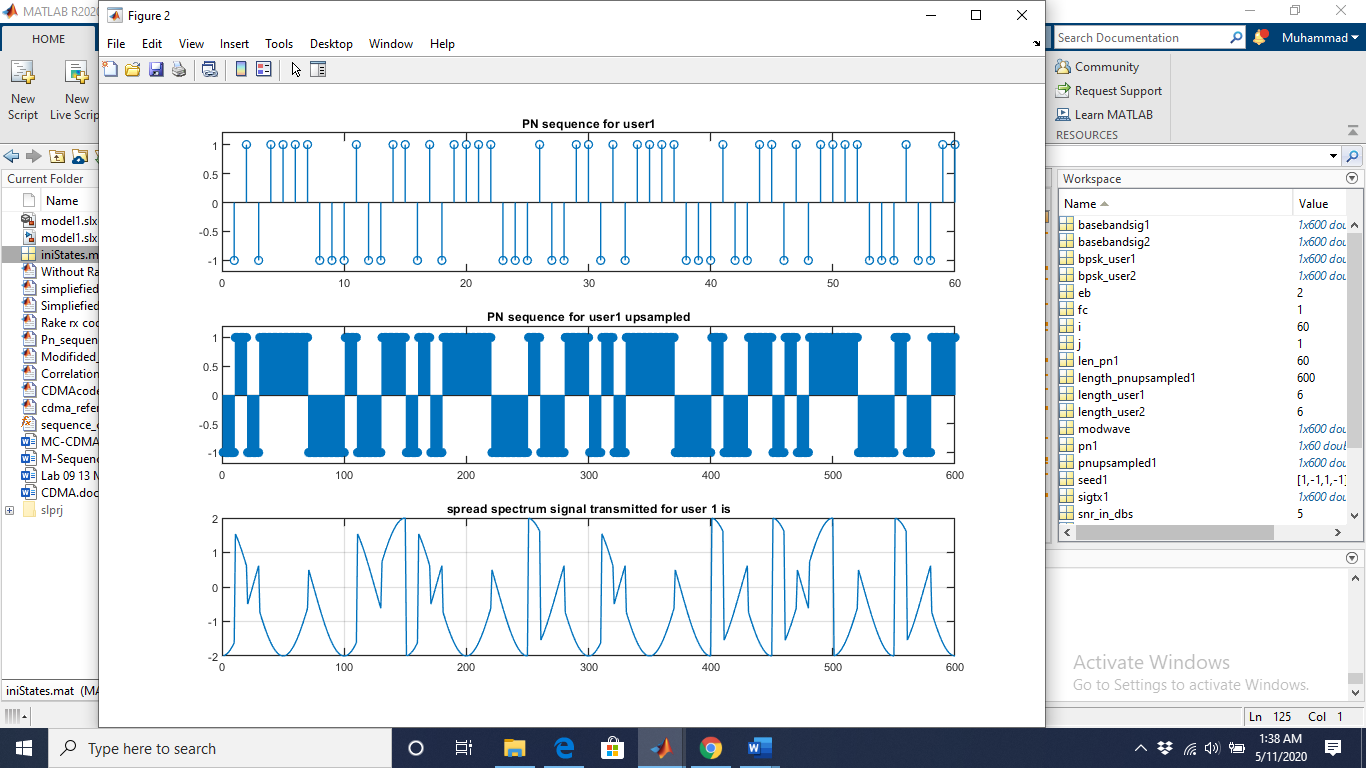
plot(sigtx1\_User\_1);

axis([0 100\*length\_mobileUser1 -2 2]);

title('spread spectrum signal transmitted for user 1 is');

grid on





**User 2**

%% CDMA transmitter user2

user2basebandsig2 = [];

for i = 1:length\_mobileUser2

for j = 0.01:0.01:Tb

if mobileUser2(i) == 1

user2basebandsig2 = [user2basebandsig2 1];

else

user2basebandsig2 = [user2basebandsig2 -1];

end

end

end

% BPSK MODULATION FOR USER 2

user2bpskmod2 = [];

for i = 1:length\_mobileUser2

for j = 0.01:0.01:Tb

user2bpskmod2 = [user2bpskmod2 sqrt(2\*Eb)\*mobileUser2(i)\*cos(2\*pi\*Fc\*j)];

end

end

figure(3);

subplot(211)

plot(user2basebandsig2, 'r');

axis([0 100\*length\_mobileUser2 -1.5 1.5]);

title('Original Binary Sequence for User2 is')

grid on

subplot(212)

plot(user2bpskmod2, 'r');

title(' BPSK Signal for User 2 is');

grid on

%% PN generator for user2

seed2 = [-1 1 -1 1]; %convert it into bipolar NRZ format

pn2\_user\_2 = [];

for i = 1:length\_mobileUser2

for j = 1:10 %chip rate is 10 times the bit rate

pn2\_user\_2 = [pn2\_user\_2 seed2(4)];

if seed2(4) == seed2(3)

temp = -1;

else

temp = 1;

end

seed2(4) = seed2(3);

seed2(3) = seed2(2);

seed2(2) = seed2(1);

seed2(1) = temp;

end

end

pnupsampled2\_User\_2 = [];

len\_pn2\_user\_2 = length(pn2\_user\_2);

for i = 1:len\_pn2\_user\_2

for j = 0.1:0.1:Tb

if pn2\_user\_2(i) == 1

pnupsampled2\_User\_2 = [pnupsampled2\_User\_2 1];

else

pnupsampled2\_User\_2 = [pnupsampled2\_User\_2 -1];

end

end

end

length\_pnupsampled2\_User\_2 = length(pnupsampled2\_User\_2);

sigtx2\_User\_2 = user2bpskmod2.\*pnupsampled2\_User\_2;

figure(4)

subplot(311)

stem(pn2\_user\_2, 'r');

axis([0,length(pn2\_user\_2),-1.2,1.2])

title('PN sequence for user2')

subplot(312)

stem(pnupsampled2\_User\_2, 'r');

axis([0,length\_pnupsampled2\_User\_2,-1.2,1.2])

title('PN sequence for user2 upsampled');

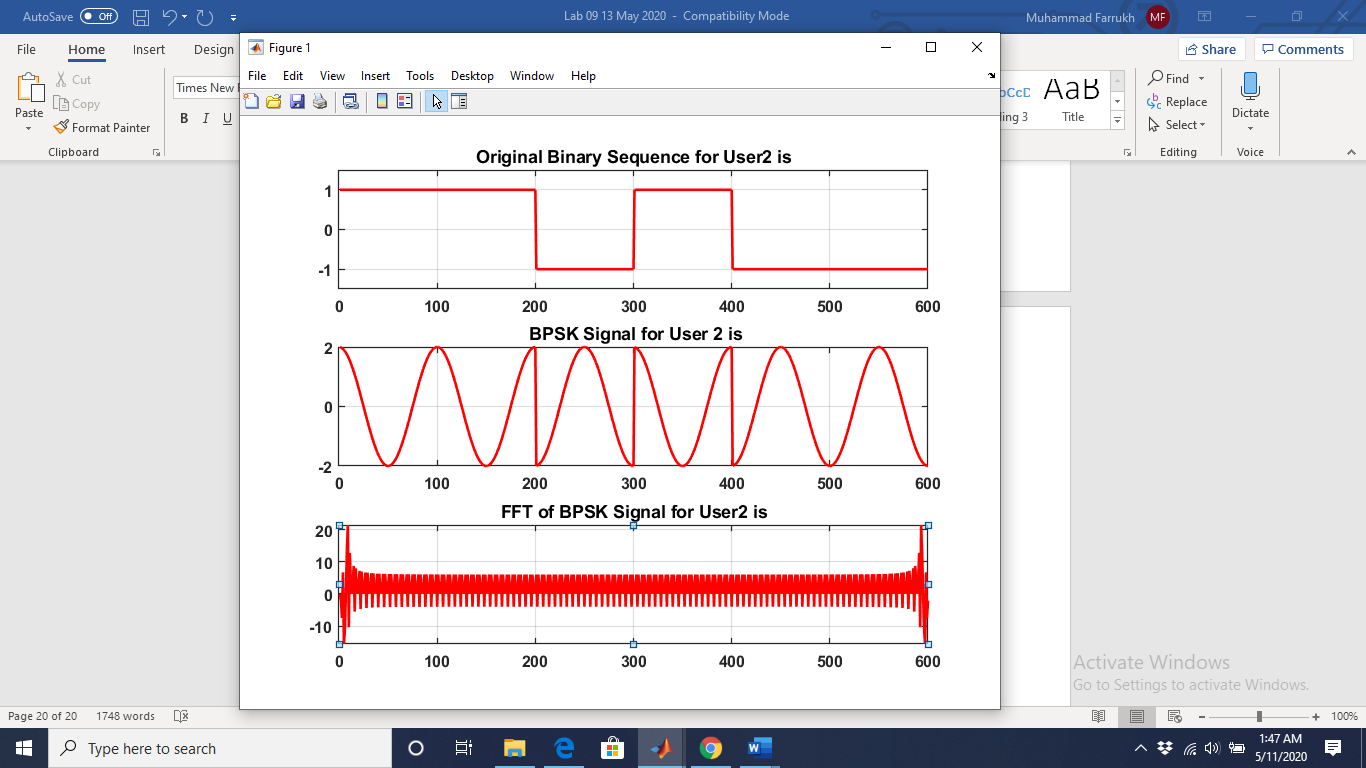
subplot(313)

plot(sigtx2\_User\_2, 'r');

axis([0 100\*length\_mobileUser2 -2 2]);

title('spread spectrum signal transmitted for user 2 is');

grid on





%%%%%%%%%%%%% **PLOTING TRANSMISTTED SIGNALS** %%%%%%%%%%

figure(5)

subplot(2,1,1);

plot(sigtx1\_User\_1);

grid on

title('spread spectrum signal txd for user 1');

subplot(2,1,2);

plot(sigtx2\_User\_2, 'r');

grid on

title('spread spectrum signal txd for user 2');



%%%%%%%%%%%%% **Channel** %%%%%%%%%%

%% AWGN Channel:

trasnmitted\_signal = sigtx1\_User\_1 + sigtx2\_User\_2;

composite\_signal = awgn(trasnmitted\_signal,SNR);

figure(6)

subplot(211)

plot(sigtx1\_User\_1+sigtx2\_User\_2);

title('Composite signal sigtx1+sigtx2');

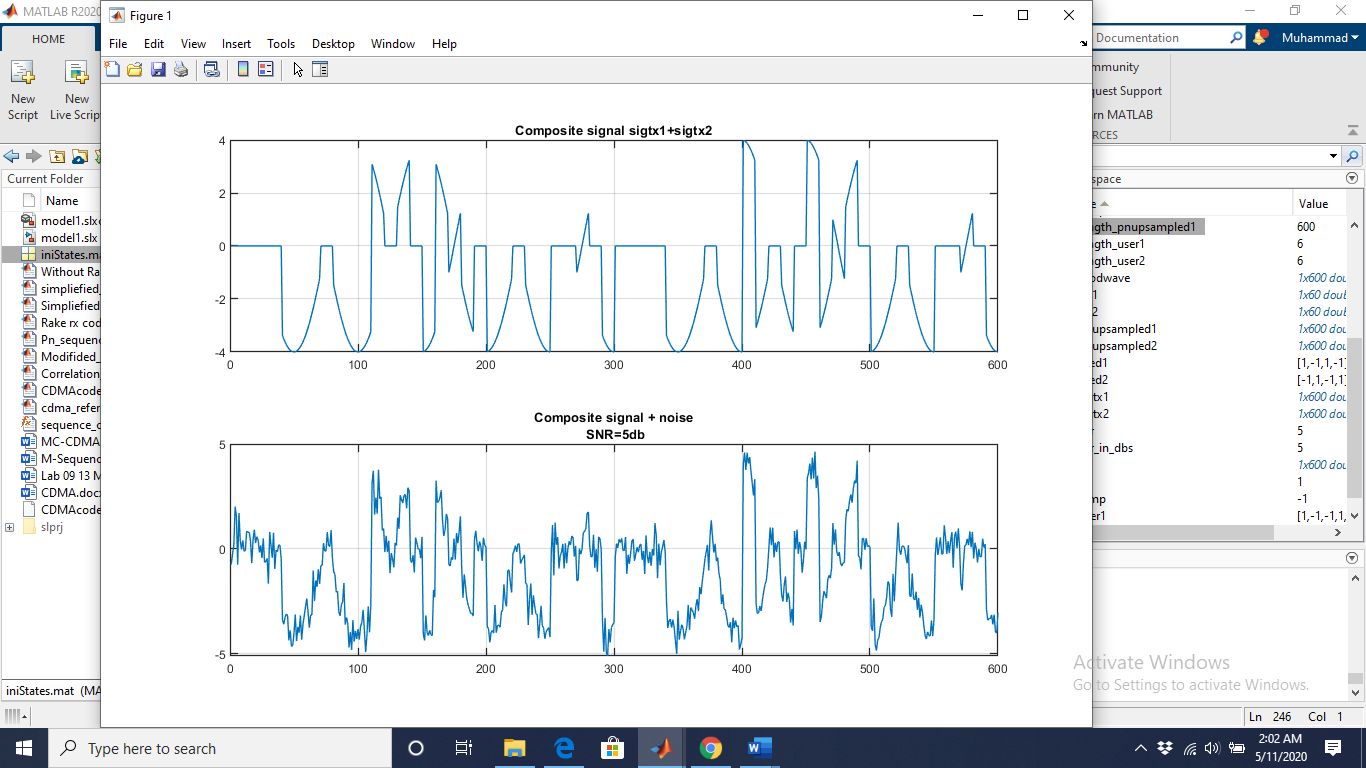
grid on

subplot(212)

plot(composite\_signal);

title(sprintf('Composite signal + noise\n SNR=%ddb',SNR));

grid on



%%%%%%%%%%%%% **Demodulation** %%%%%%%%%%

%% DMODULATION USER 1 %%%%%%%%%%%%%%%%%%%%

rx1 = composite\_signal.\*pnupsampled1\_User\_1;

%%%% BPSK DEMODULATION FOR USER 1

demodcar1 = [];

for i = 1:length\_mobileUser1

for j = 0.01:0.01:Tb

demodcar1 = [demodcar1 sqrt(2\*Eb)\*cos(2\*pi\*Fc\*j)];

end

end

bpskdemod1 = rx1.\*demodcar1;

len\_dmod1 = length(bpskdemod1);

sum = zeros(1,len\_dmod1/100);

for i = 1:len\_dmod1/100

for j = (i-1)\*100+1:i\*100

sum(i) = sum(i)+bpskdemod1(j);

end

end

sum;

rxbits1 = [];

for i = 1:length\_mobileUser1

if sum(i)>0

rxbits1 = [rxbits1 1];

else

rxbits1 = [rxbits1 0];

end

end

rxbits1=rectpulse(rxbits1,100);

figure(7)

subplot(211)

plot(user1basebandsig1)

title('Transmitted bits of user1 data')

grid on

subplot(212)

plot(rxbits1)

axis([0 600 -0.2 1.2]);

title('Received bits of user1 data')

grid on

%% DMODULATION USER 2 %%%%%%%%%%%%%%%%%%%%

rx2 = composite\_signal.\*pnupsampled2\_User\_2;

% BPSK DEMODULATION FOR USER 2

demodcar2 = [];

for i = 1:length\_mobileUser2

for j = 0.01:0.01:Tb

demodcar2 = [demodcar2 sqrt(2\*Eb)\*cos(2\*pi\*Fc\*j)];

end

end

bpskdemod2 = rx2.\*demodcar2;

len\_dmod2 = length(bpskdemod2);

sum = zeros(1,len\_dmod1/100);

for i = 1:len\_dmod2/100

for j = (i-1)\*100+1:i\*100

sum(i) = sum(i)+bpskdemod2(j);

end

end

sum;

rxbits2 = [];

for i = 1:length\_mobileUser2

if sum(i)>0

rxbits2 = [rxbits2 1];

else

rxbits2 = [rxbits2 0];

end

end

rxbits2=rectpulse(rxbits2,100);

figure(8)

subplot(211)

plot(user2basebandsig2)

title('Transmitted bits of user2 data')

grid on

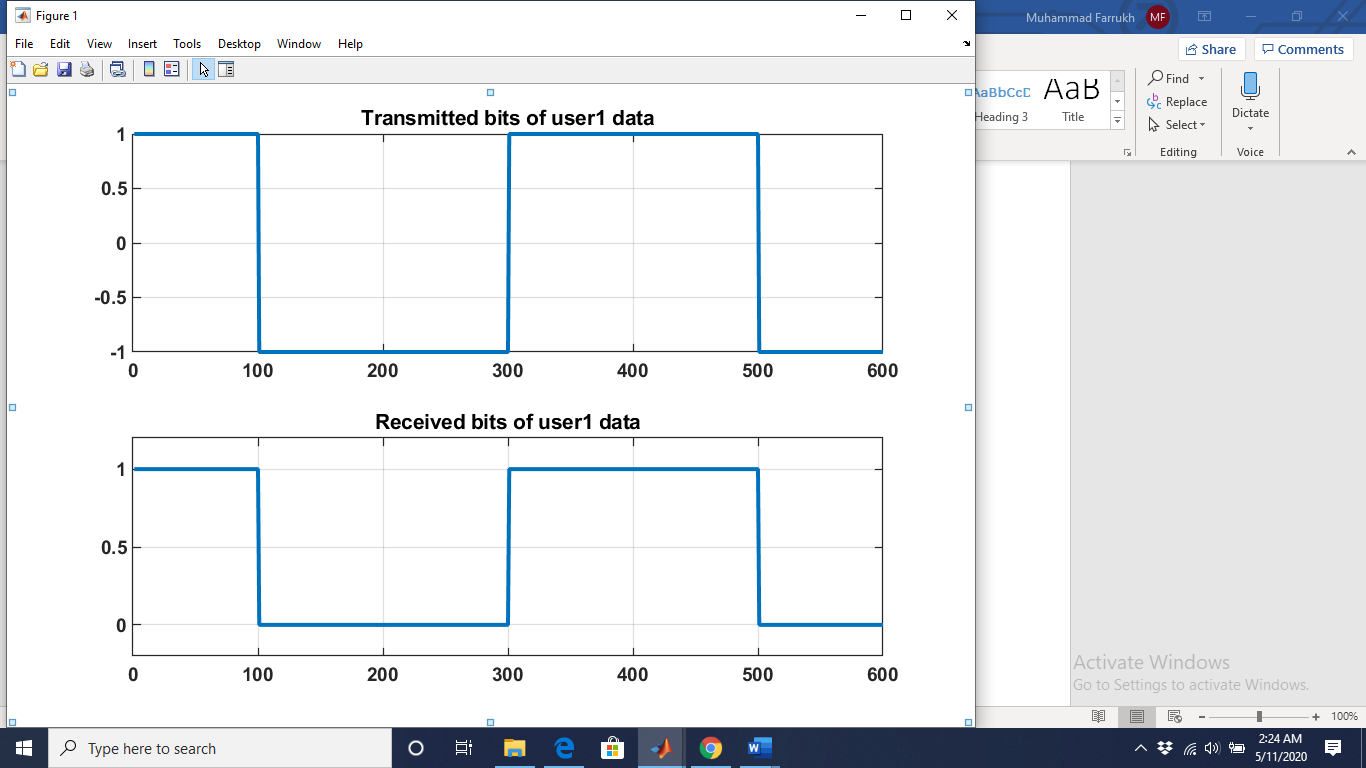
subplot(212)

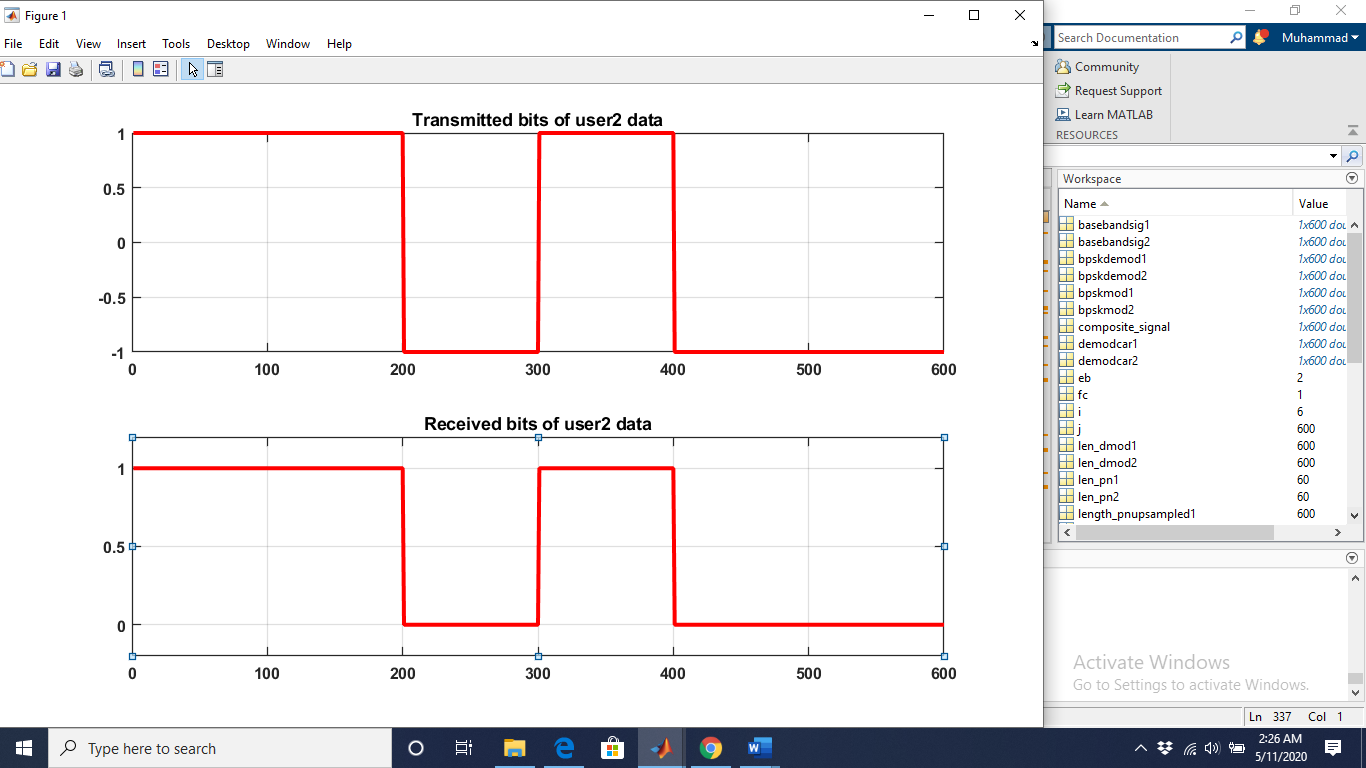
plot(rxbits2)

axis([0 600 -0.2 1.2]);

title('Received bits of user2 data')

grid on





You can compare between the received bits (rxbits1 and rxbits1) and the transmitted bits (user1 and user2). If they are equal this means that the receiver decoded the signal correctly.

**Exercise A**

Modify the program by adding another 2 users to the system that are willing to transmit this data:

mobileUser3 = [1 1 1 0 0 0];

mobileUser4 = [0 0 0 1 1 1];

So now we have a total of 4 users that are transmitting simultaneously using the same frequency.

During the PN generator use the following seeds for user 3 and user 4:

seed3 = [-1 -1 -1 -1];

seed4 = [1 1 1 1];

You can upload your reports on our channel on Teams:

LAB - 2024/2025 Physical Layer for software radio Files Lab08 student reports

Please try to compress your MATLAB scripts (**only Exercise A**) into one file called: Lab08\_yourName