AMERICAN INTERNATIONAL UNIVERSITY – BANGLADESH

FACULTY OF SCIENCE & TECHNOLOGY

Course Title: Data Communication

Lab Report-2

Submitted by:

NAME: ABU SHALEH MD. KAIUM

SECTION: G

ID: 20-42475-1

DATE OF SUBMISSION: 20/02/2022

PROGRAM: BSc CSE

COURSE TITLE: Data Communication

Submitted to:

Course Teacher: Tanjil Amin

Class Task:

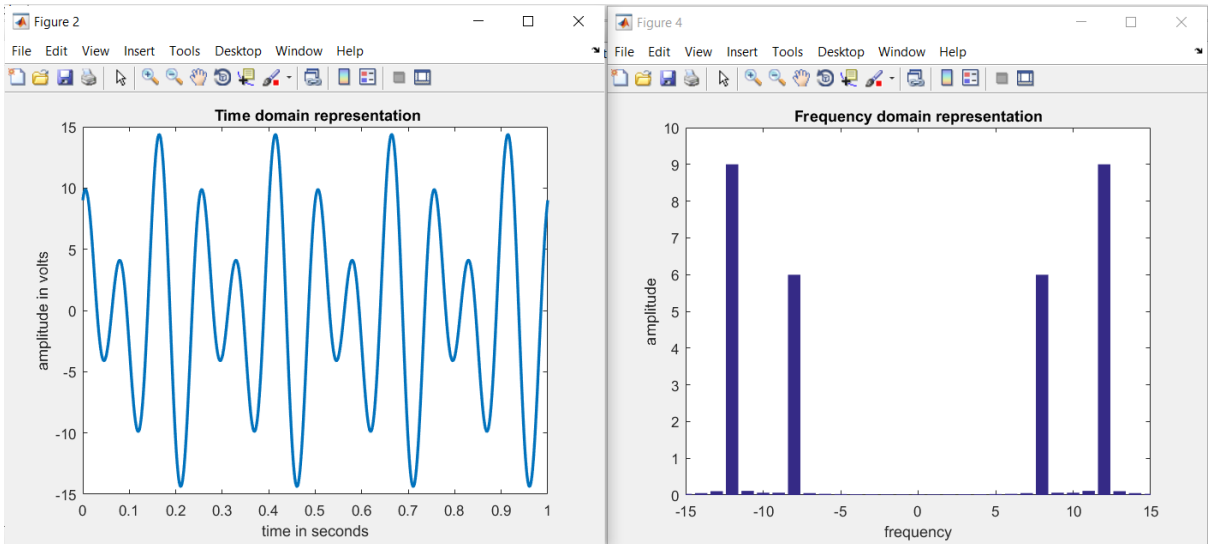
Similar task can be done where we use a composite signal instead of signals x1 and x2. Suppose our composite signal is signal_x = a1*sin(2*pi*f1*t) + a2*cos(2*pi*f2*t); Here, a1 = (B + G + H), a2 = (C + E + H), f1 = (G + H + 2), and f2 = (E + F+ H). [Assume your ID is AB-CDEFG-H]

*****Show this signal both in time domain and frequency domain.

Code:

```
A=2;
B=0;
C=4;
D=2;
E=4;
F=7;
G=5;
H=1;
a1 = (B + G + H);
a2 = (C + E + H);
f1 = (G + H + 2);
f2 = (E + F+ H);
fs= 1000;
t = 0:1/fs:1;
x = a1*sin(2*pi*f1*t) + a2*cos(2*pi*f2*t);
figure
plot(t,x, 'linewidth',2)
xlabel('time in seconds')
ylabel('amplitude in volts')
title(' Time domain representation')
```

```
figure
fx = abs(fftshift(fft(x)))/(length(x)/2);
freq = linspace(-fs/2, fs/2, length(x));
figure
bar(freq, fx,'linewidth',1)
xlim([-15 +15])
xlabel('frequency')
ylabel('amplitude')
title('Frequency domain representation')
```



Performance Task for Lab Report: (your ID = AB-CDEFG-H)

**Generate a composite signal using three simple signals as, x1 = a1*cos(2*pi*f1*t), x2 = a2*sin(2*pi*f2*t), x3 = a3*cos(2*pi*f3*t) signal_x = x1 + x2 + x3

Select the values of the amplitude and frequency as follows: a1 = A + C + 1, a2 = A + D + 2, a3 = A + E + 1, f1 = A + E + 1, f2 = A + D + 2, f3 = A + C + 1.

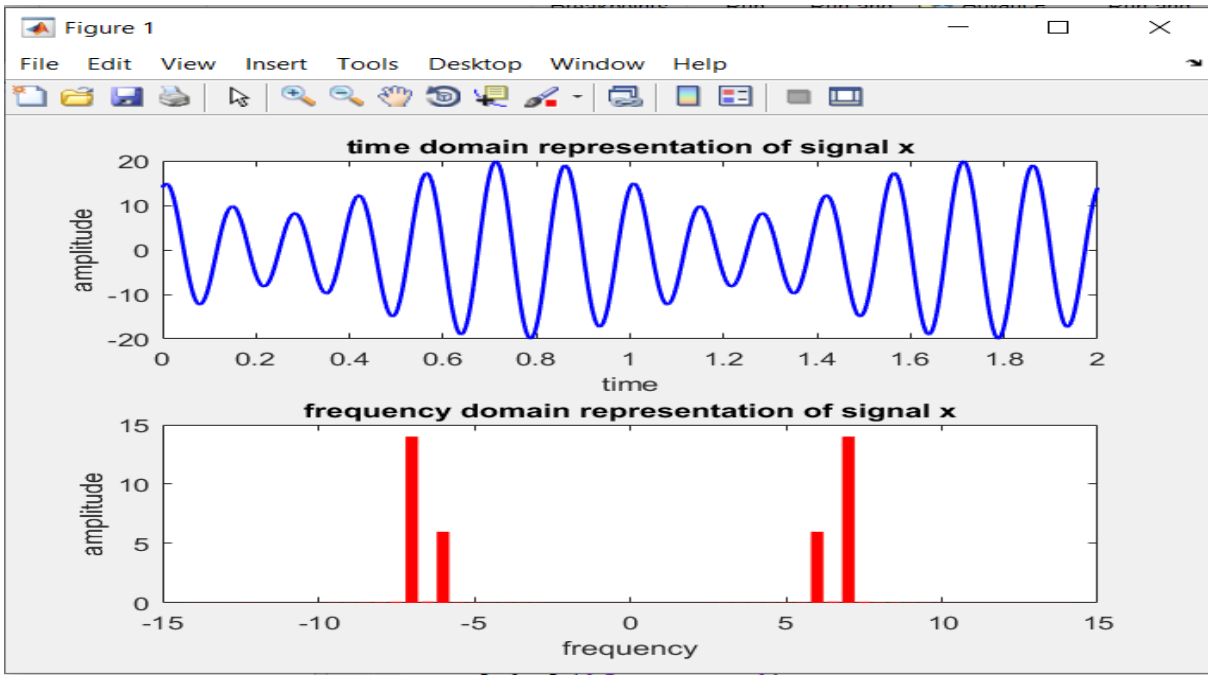
(a) Show time domain and frequency domain representations of signal_x in a single figure window using subplot. Use axis, or xlim, or ylim to appropriately represent the signal.

Code:

```
clc;

clear all;

close all;

A=2;
B=0;
C=4;
D=2;
E=4;
F=7;
G=5;
H=1;
fs = 2000;

t = 0:1/fs:2;

a1 = A + C + 1;

a2 = A + D + 2;

a3 = A + E + 1;

f1 = A + E + 1;

f2 = A + D + 2;

f3 = A + C + 1;

fs = 1000;

fs = 1000;

t = 0:1/fs:2;

x1= a1*cos(2*pi*f1*t);

x2= a2*sin(2*pi*f2*t);

x3 = a3*cos(2*pi*f3*t);

signal_x= x1+x2+x3;


% time domain representation

subplot(2,1,1);
```

```
plot(t,signal_x,'b','linewidth',1.5);
xlabel('time');
ylabel('amplitude');
title('time domain representation of signal x');

% frequency domain representation
fx = abs(fftshift(fft(signal_x)))/(length(signal_x)/2);
freq = linspace(-fs/2, fs/2, length(signal_x));
subplot(2,1,2);
bar(freq, fx,'r','linewidth',1.5);
xlim([-15 +15]);
xlabel('frequency');
ylabel('amplitude');
title('frequency domain representation of signal x');
```

Output:



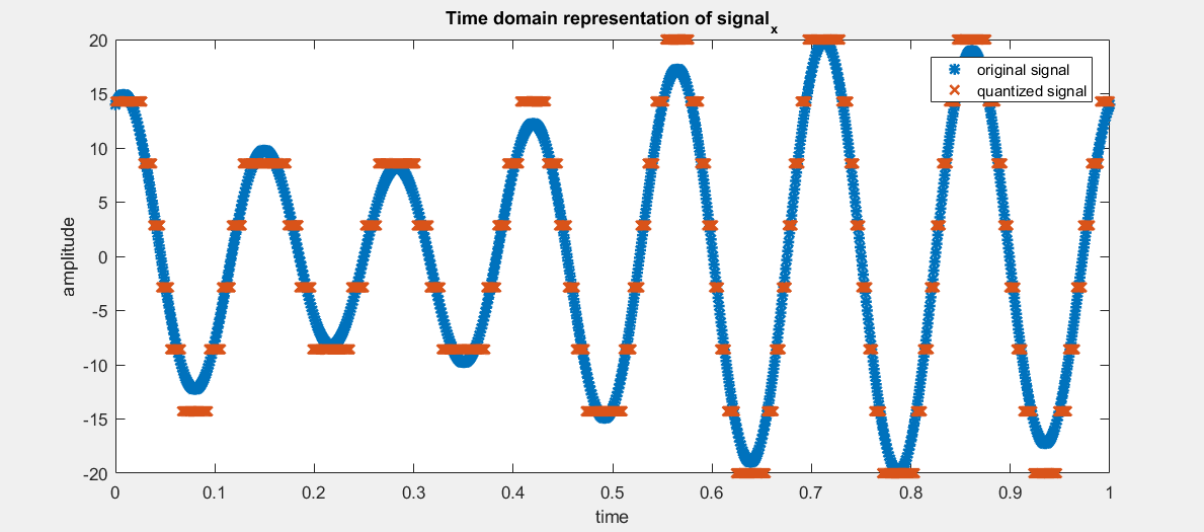
(b) Quantize signal_x in 8 equally distributed levels and provide image for one cycle of the original signal and quantized signal. Use axis, or xlim, or ylim to appropriately represent the signal.

Code:

```
clc;
clear all;
close all;
A=2;
B=0;
C=4;
D=2;
E=4;
F=7;
G=5;
```

```
H=1;
a1= A+C+1;
a2= A+D+2;
a3= A+E+1;
f1= A+E+1;
f2= A+D+2;
f3= A+C+1;
fs = 1000;
t = 0:1/fs:1;
x1= a1*cos(2*pi*f1*t);
x2= a2*sin(2*pi*f2*t);
x3 = a3*cos(2*pi*f3*t);
x= x1+x2+x3;
p= linspace(-15,15,7);
c= linspace(-20,20,8);
[i,q]= quantiz(x,p,c);
plot(t,x,'*',t,q,'x','linewidth',1.5);
xlabel('time');
ylabel('amplitude');
title('Time domain representation of signal_x');
legend('original signal','quantized signal');
```

Output:


Time domain representation of signal x
original signal
quantized signal
amplitude
time
20
15
10
5
0
-5
-10
-15
-20
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1