AMERICAN INTERNATIONAL UNIVERSITY–BANGLADESH (AIUB)

DATA COMMUNICATION

Spring 2024-2025

Section: D

LAB REPORT ON

Study of signal frequency, spectrum, bandwidth and quantization using MATLAB

EXPERIMENT NO

02

Supervised By

DR. MD. HUMAYUN KABIR

sSubmitted By:

Name	**ID**
S.M. Rasel	**22-48039-2**

Date of Submission: **15/03/2025**

ID- 22-48039-2
S.M. Rasel

Objective: The primary objects are given_

1. To understand the use of MATLAB for solvin[g] communication enjuewing problems.

2. To develop an understanding of the MATLAB environment, commands, and systax.

3. To study the concepts of signal frequency, stetum, bantwidth, and quantization.

4. Generate the analyne signals in both time and frequency domains using MATLA[B]

5. To peform quantization an analog signa[l] and observe the offeets of different qua[ntization] levels.

Software Tools: MATLAB.

S.M. Rasel
22-48039-2

Working Principle:

1) Start
2) Initialize MATLAB Environment
   - Clear variables and close figures
   - Define sampling frequency and time duration
3) Generate signals
   - Define frequencies and amplitudes
   - Create individual sinusoidal signals
   - Sum them to form a composite signal
4) Plot Time & Frequency Domain Representation
   - Compute FFT
   - Plot time-domain signal
   - Plot frequency-domain signal using FFT
5) Quantization (4 Levels using quantiz)
   - Define quantization levels
   - Apply quantization
   - Plot original vs quantized signal

Simulation tools: MATLABR2016a

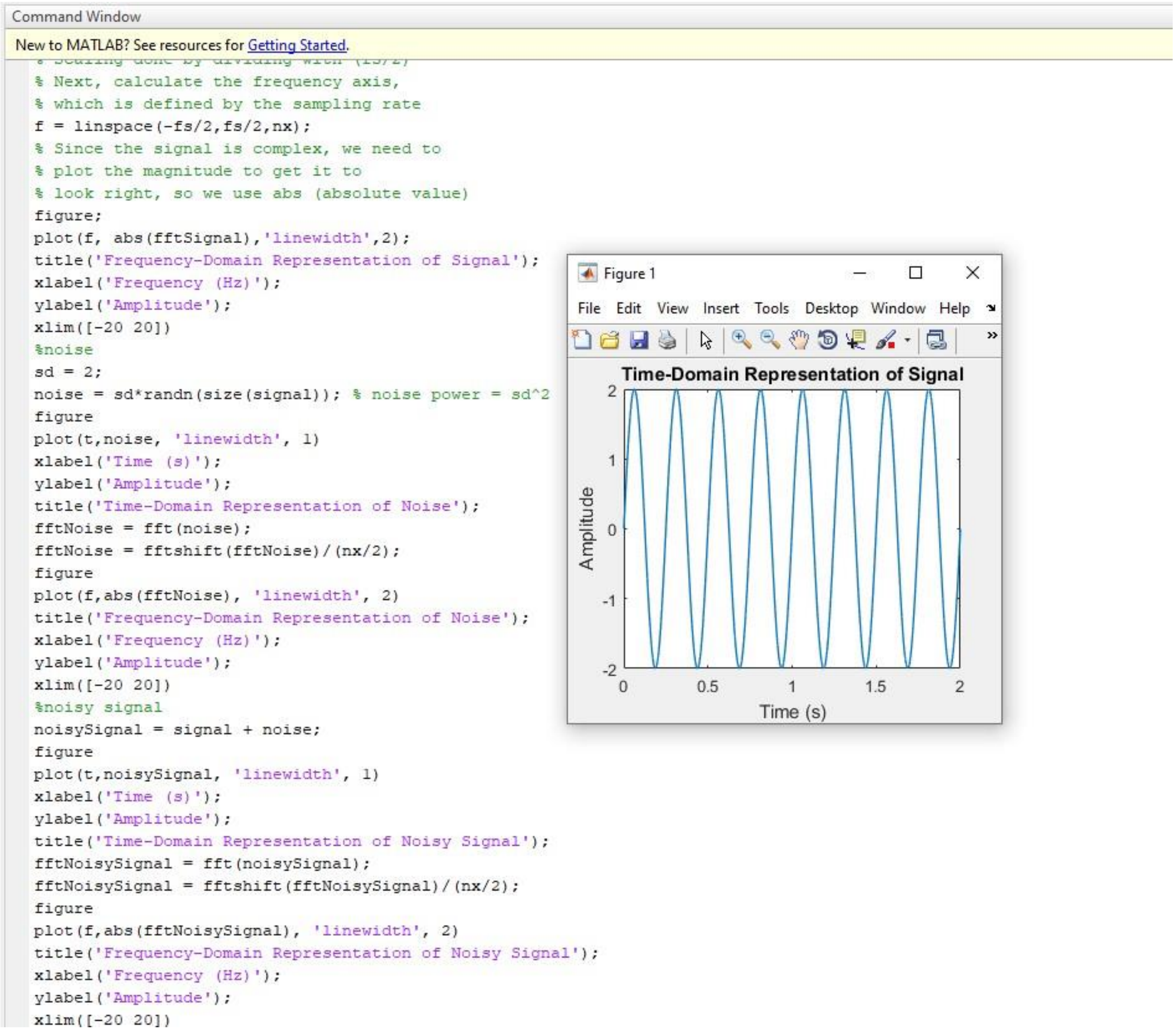
MATLAB Code Here:

```
>> %Define number of samples to take

fs = 8000;

f = 4; %Hz

%Define signal

t = 0:1/fs:2;

signal = 2*sin(2*pi*f*t);

nx = length(t); % Total number of samples

%Plot to illustrate that it is a sine wave

plot(t, signal,'linewidth',1);

title('Time-Domain Representation of Signal');

xlabel('Time (s)');

ylabel('Amplitude');

% Take fourier transform

fftSignal = fft(signal);

% Apply fftshift to put it in the form

% we are used to (see documentation)

fftSignal = fftshift(fftSignal)/(nx/2);

% Scaling done by dividing with (fs/2)

% Next, calculate the frequency axis,

% which is defined by the sampling rate

f = linspace(-fs/2,fs/2,nx);
```

```
% Since the signal is complex, we need to

% plot the magnitude to get it to

% look right, so we use abs (absolute value)

figure;

plot(f, abs(fftSignal),'linewidth',2);

title('Frequency-Domain Representation of Signal');

xlabel('Frequency (Hz)');

ylabel('Amplitude');

xlim([-20 20])

%noise

sd = 2;

noise = sd*randn(size(signal)); % noise power = sd^2

figure

plot(t,noise, 'linewidth', 1)

xlabel('Time (s)');

ylabel('Amplitude');

title('Time-Domain Representation of Noise');

fftNoise = fft(noise);

fftNoise = fftshift(fftNoise)/(nx/2);

figure

plot(f,abs(fftNoise), 'linewidth', 2)

title('Frequency-Domain Representation of Noise');

xlabel('Frequency (Hz)');

ylabel('Amplitude');
```

```
xlim([-20 20])

%noisy signal

noisySignal = signal + noise;

figure

plot(t,noisySignal, 'linewidth', 1)

xlabel('Time (s)');

ylabel('Amplitude');

title('Time-Domain Representation of Noisy Signal');

fftNoisySignal = fft(noisySignal);

fftNoisySignal = fftshift(fftNoisySignal)/(nx/2);

figure

plot(f,abs(fftNoisySignal), 'linewidth', 2)

title('Frequency-Domain Representation of Noisy Signal');

xlabel('Frequency (Hz)');

ylabel('Amplitude');

xlim([-20 20])
```

Simulation with diagram:



Command Window

New to MATLAB? See resources for Getting Started.

```
% Next, calculate the frequency axis,
% which is defined by the sampling rate
f = linspace(-fs/2,fs/2,nx);
% Since the signal is complex, we need to
% plot the magnitude to get it to
% look right, so we use abs (absolute value)
figure;
plot(f, abs(fftSignal),'linewidth',2);
title('Frequency-Domain Representation of Signal');
xlabel('Frequency (Hz)');
ylabel('Amplitude');
xlim([-20 20])
%noise
sd = 2;
noise = sd*randn(size(signal)); % noise power = sd^2
figure
plot(t,noise, 'linewidth', 1)
xlabel('Time (s)');
ylabel('Amplitude');
title('Time-Domain Representation of Noise');
fftNoise = fft(noise);
fftNoise = fftshift(fftNoise)/(nx/2);
figure
plot(f,abs(fftNoise), 'linewidth', 2)
title('Frequency-Domain Representation of Noise');
xlabel('Frequency (Hz)');
ylabel('Amplitude');
xlim([-20 20])
%noisy signal
noisySignal = signal + noise;
figure
plot(t,noisySignal, 'linewidth', 1)
xlabel('Time (s)');
ylabel('Amplitude');
title('Time-Domain Representation of Noisy Signal');
fftNoisySignal = fft(noisySignal);
fftNoisySignal = fftshift(fftNoisySignal)/(nx/2);
figure
plot(f,abs(fftNoisySignal), 'linewidth', 2)
title('Frequency-Domain Representation of Noisy Signal');
xlabel('Frequency (Hz)');
ylabel('Amplitude');
xlim([-20 20])
```


Figure-01: Time-Domain Representation of Signal

Command Window

New to MATLAB? See resources for Getting Started.

```
% Scaling done by dividing with (fs/2)
% Next, calculate the frequency axis,
% which is defined by the sampling rate
f = linspace(-fs/2,fs/2,nx);
% Since the signal is complex, we need to
% plot the magnitude to get it to
% look right, so we use abs (absolute value)
figure;
plot(f, abs(fftSignal),'linewidth',2);
title('Frequency-Domain Representation of Signal');
xlabel('Frequency (Hz)');
ylabel('Amplitude');
xlim([-20 20])
%noise
sd = 2;
noise = sd*randn(size(signal)); % noise power = sd^2
figure
plot(t,noise, 'linewidth', 1)
xlabel('Time (s)');
ylabel('Amplitude');
title('Time-Domain Representation of Noise');
fftNoise = fft(noise);
fftNoise = fftshift(fftNoise)/(nx/2);
figure
plot(f,abs(fftNoise), 'linewidth', 2)
title('Frequency-Domain Representation of Noise');
xlabel('Frequency (Hz)');
ylabel('Amplitude');
xlim([-20 20])
%noisy signal
noisySignal = signal + noise;
figure
plot(t,noisySignal, 'linewidth', 1)
xlabel('Time (s)');
ylabel('Amplitude');
title('Time-Domain Representation of Noisy Signal');
fftNoisySignal = fft(noisySignal);
fftNoisySignal = fftshift(fftNoisySignal)/(nx/2);
figure
plot(f,abs(fftNoisySignal), 'linewidth', 2)
title('Frequency-Domain Representation of Noisy Signal');
xlabel('Frequency (Hz)');
ylabel('Amplitude');
xlim([-20 20])
```

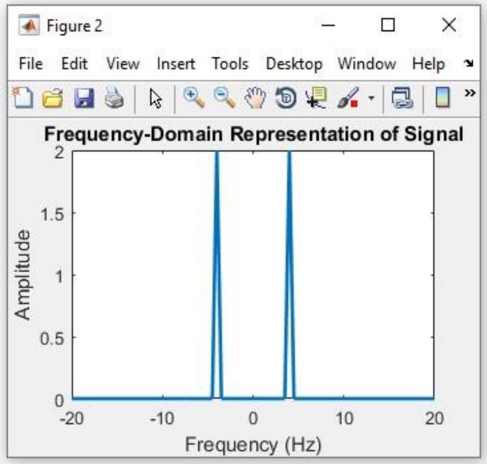


Figure-02: Frequency-Domain Representation of signal

Command Window

New to MATLAB? See resources for Getting Started.

```
% Next, calculate the frequency axis,
% which is defined by the sampling rate
f = linspace(-fs/2,fs/2,nx);
% Since the signal is complex, we need to
% plot the magnitude to get it to
% look right, so we use abs (absolute value)
figure;
plot(f, abs(fftSignal),'linewidth',2);
title('Frequency-Domain Representation of Signal');
xlabel('Frequency (Hz)');
ylabel('Amplitude');
xlim([-20 20])
%noise
sd = 2;
noise = sd*randn(size(signal)); % noise power = sd^2
figure
plot(t,noise, 'linewidth', 1)
xlabel('Time (s)');
ylabel('Amplitude');
title('Time-Domain Representation of Noise');
fftNoise = fft(noise);
fftNoise = fftshift(fftNoise)/(nx/2);
figure
plot(f,abs(fftNoise), 'linewidth', 2)
title('Frequency-Domain Representation of Noise');
xlabel('Frequency (Hz)');
ylabel('Amplitude');
xlim([-20 20])
%noisy signal
noisySignal = signal + noise;
figure
plot(t,noisySignal, 'linewidth', 1)
xlabel('Time (s)');
ylabel('Amplitude');
title('Time-Domain Representation of Noisy Signal');
fftNoisySignal = fft(noisySignal);
fftNoisySignal = fftshift(fftNoisySignal)/(nx/2);
figure
plot(f,abs(fftNoisySignal), 'linewidth', 2)
title('Frequency-Domain Representation of Noisy Signal');
xlabel('Frequency (Hz)');
ylabel('Amplitude');
xlim([-20 20])
```

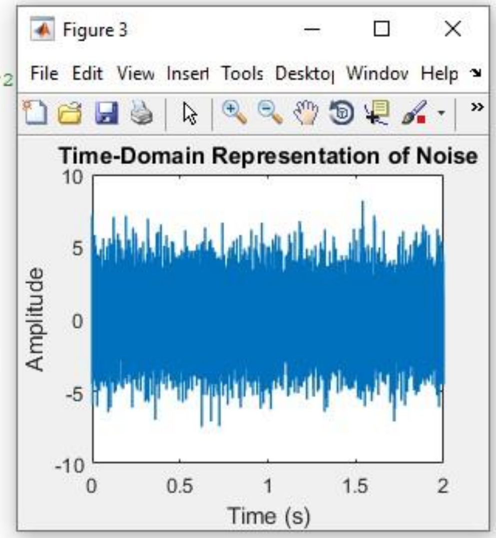


Figure-03: Time-Domain Representation of noise

Command Window

New to MATLAB? See resources for Getting Started.

```
% Scaling done by dividing with (fs/2)
% Next, calculate the frequency axis,
% which is defined by the sampling rate
f = linspace(-fs/2,fs/2,nx);
% Since the signal is complex, we need to
% plot the magnitude to get it to
% look right, so we use abs (absolute value)
figure;
plot(f, abs(fftSignal),'linewidth',2);
title('Frequency-Domain Representation of Signal');
xlabel('Frequency (Hz)');
ylabel('Amplitude');
xlim([-20 20])
%noise
sd = 2;
noise = sd*randn(size(signal)); % noise power = sd^2
figure
plot(t,noise, 'linewidth', 1)
xlabel('Time (s)');
ylabel('Amplitude');
title('Time-Domain Representation of Noise');
fftNoise = fft(noise);
fftNoise = fftshift(fftNoise)/(nx/2);
figure
plot(f,abs(fftNoise), 'linewidth', 2)
title('Frequency-Domain Representation of Noise');
xlabel('Frequency (Hz)');
ylabel('Amplitude');
xlim([-20 20])
%noisy signal
noisySignal = signal + noise;
figure
plot(t,noisySignal, 'linewidth', 1)
xlabel('Time (s)');
ylabel('Amplitude');
title('Time-Domain Representation of Noisy Signal');
fftNoisySignal = fft(noisySignal);
fftNoisySignal = fftshift(fftNoisySignal)/(nx/2);
figure
plot(f,abs(fftNoisySignal), 'linewidth', 2)
title('Frequency-Domain Representation of Noisy Signal');
xlabel('Frequency (Hz)');
ylabel('Amplitude');
xlim([-20 20])
```

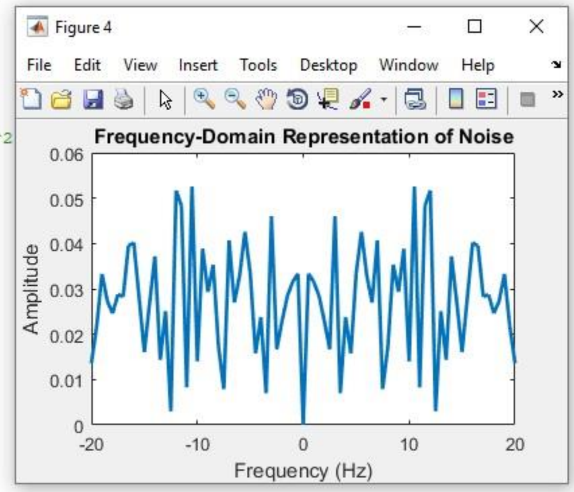


Figure-04: Frequency-Domain Representation of Noise

Command Window

New to MATLAB? See resources for Getting Started.

```
% Scaling done by dividing with (fs/2)
% Next, calculate the frequency axis,
% which is defined by the sampling rate
f = linspace(-fs/2,fs/2,nx);
% Since the signal is complex, we need to
% plot the magnitude to get it to
% look right, so we use abs (absolute value)
figure;
plot(f, abs(fftSignal),'linewidth',2);
title('Frequency-Domain Representation of Signal');
xlabel('Frequency (Hz)');
ylabel('Amplitude');
xlim([-20 20])
%noise
sd = 2;
noise = sd*randn(size(signal)); % noise power = sd^2
figure
plot(t,noise, 'linewidth', 1)
xlabel('Time (s)');
ylabel('Amplitude');
title('Time-Domain Representation of Noise');
fftNoise = fft(noise);
fftNoise = fftshift(fftNoise)/(nx/2);
figure
plot(f,abs(fftNoise), 'linewidth', 2)
title('Frequency-Domain Representation of Noise');
xlabel('Frequency (Hz)');
ylabel('Amplitude');
xlim([-20 20])
%noisy signal
noisySignal = signal + noise;
figure
plot(t,noisySignal, 'linewidth', 1)
xlabel('Time (s)');
ylabel('Amplitude');
title('Time-Domain Representation of Noisy Signal');
fftNoisySignal = fft(noisySignal);
fftNoisySignal = fftshift(fftNoisySignal)/(nx/2);
figure
plot(f,abs(fftNoisySignal), 'linewidth', 2)
title('Frequency-Domain Representation of Noisy Signal');
xlabel('Frequency (Hz)');
ylabel('Amplitude');
xlim([-20 20])
```

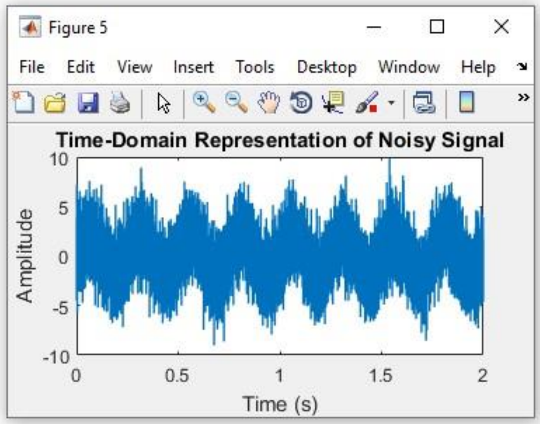


Figure-05: Time-Domain Representation of Noisy Signal

Command Window

New to MATLAB? See resources for Getting Started.

```
% Scaling done by dividing with (fs/2)
% Next, calculate the frequency axis,
% which is defined by the sampling rate
f = linspace(-fs/2,fs/2,nx);
% Since the signal is complex, we need to
% plot the magnitude to get it to
% look right, so we use abs (absolute value)
figure;
plot(f, abs(fftSignal),'linewidth',2);
title('Frequency-Domain Representation of Signal');
xlabel('Frequency (Hz)');
ylabel('Amplitude');
xlim([-20 20])
%noise
sd = 2;
noise = sd*randn(size(signal)); % noise power = sd^2
figure
plot(t,noise, 'linewidth', 1)
xlabel('Time (s)');
ylabel('Amplitude');
title('Time-Domain Representation of Noise');
fftNoise = fft(noise);
fftNoise = fftshift(fftNoise)/(nx/2);
figure
plot(f,abs(fftNoise), 'linewidth', 2)
title('Frequency-Domain Representation of Noise');
xlabel('Frequency (Hz)');
ylabel('Amplitude');
xlim([-20 20])
%noisy signal
noisySignal = signal + noise;
figure
plot(t,noisySignal, 'linewidth', 1)
xlabel('Time (s)');
ylabel('Amplitude');
title('Time-Domain Representation of Noisy Signal');
fftNoisySignal = fft(noisySignal);
fftNoisySignal = fftshift(fftNoisySignal)/(nx/2);
figure
plot(f,abs(fftNoisySignal), 'linewidth', 2)
title('Frequency-Domain Representation of Noisy Signal');
xlabel('Frequency (Hz)');
ylabel('Amplitude');
xlim([-20 20])
```

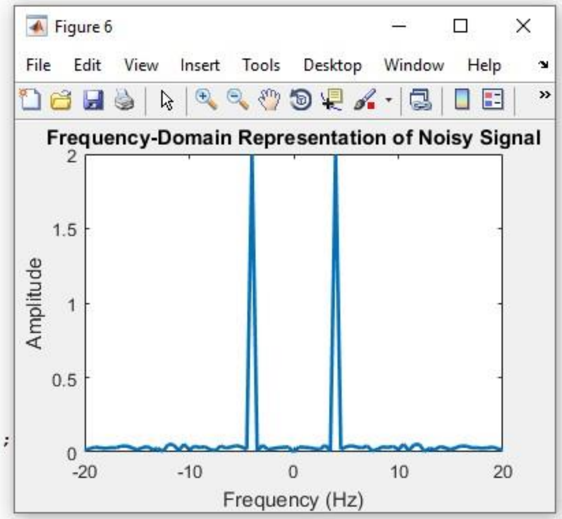


Figure-06: Frequency-Domain Representation of Noisy Signal

Exp. Number : 02
Date : 10/02/2025
Student ID : 22-48039-2
Name : S.M. Rasel

Lab Task Code-4

```
~~%Close all;~~
~~%clc;~~
~~%Define number of Samples to take~~
fs = 8000;
f = 4; % Hz
%Define signal
t = 0:1/fs:2;
singnal = 2*sin(2*pi*f*t);
nx = Length(t); % total number of Samples
%plot to illustrate that it is a sin wave
plot(t,signal,'Linewidth', 1);
title('Time-Domain Represantaion of Signal');
Xlabel('Time (s)');
yLabel ('Amplitude');
% Take fourier transform
fftsignal = fft(signal);
%Apply fftshift to put it in the form
%we are used to (see documentation)
fftsignal = fftshift (fftsignal)/(nx/2);
f = Linspace(-fs/2, fs/2, nx);
figure;
```

Discussion and conclusion:

From the above simulations various functionalities of MATLAB were observe in hand. Various functions that were available on MATLAB were ~~far~~ learned and observed. Using this knowledge, MATLAB software plotted Frequency, spectrum, Bandwidth and Quantization. The quantizing of an analog signal is done by discretizing the signal with a number of Quantization levels. Various formatting on the graph was learned from this experiment as well. Hence, it can be said that all the objectives of this experiment were obtain properly.

References:

• Prakash C. Gupta, “Data communications”, Prentice Hall India Pvt.

• William Stallings, "Data and Computer Communications”, Pearson

• AIUB Data Communication Engineering Lab Manual, Report 02