AMERICAN INTERNATIONAL UNIVERSITY BANGLADESH

Faculty of Engineering

Laboratory Report Cover Sheet

Students must complete all details except the faculty use part.

Please submit all reports to your subject supervisor or the office of the concerned faculty.

Lab Title: **Study of Analog to Digital Conversion using MATLAB**

Experiment Number: **05** Due Date: **19 /03/2024** Semester: **Spring 2023-2024**

Subject Code: **COE3103** Subject Name: **DATA COMMUNICATION** Section: **E**

Course Instructor: **NOWSHIN ALAM** Degree Program: **B.Sc. CSE**

Declaration and Statement of Authorship:

1. I/we hold a copy of this report, which can be produced if the original is lost/ damaged.
2. This report is my/our original work and no part of it has been copied from any other student's work or from any other source except where due acknowledgement is made.
3. No part of this report has been written for me/us by any other person except where such collaboration has been authorized by the lecturer/teacher concerned and is clearly acknowledged in the report.
4. I/we have not previously submitted or currently submitting this work for any other course/unit.
5. This work may be reproduced, communicated, compared and archived for the purpose of detecting plagiarism.
6. I/we give permission for a copy of my/our marked work to be retained by the school for review and comparison, including review by external examiners.

I/we understand that

7. Plagiarism is the presentation of the work, idea or creation of another person as though it is your own. It is a form of cheating and is a very serious academic offence that may lead to expulsion from the University. Plagiarized material can be drawn from, and presented in, written, graphic and visual form, including electronic data, and oral presentations. Plagiarism occurs when the origin of the material used is not appropriately cited.
8. Enabling plagiarism is the act of assisting or allowing another person to plagiarize or to copy your work.

Group Number (if applicable): 08 Individual Submission ☐ Group Submission ☐

No.	Student Name	Student Number	Student Signature	Date
Submitted by:				
1	**MD. ASHIKUZZAMAN ABIR**	**22-47006-1**		
Group Members:				
2	**MD. SHOHANUR RAHMAN SHOHAN**	**22-46013-1**		
3	**MD. TAHSIN HASIB**	**22-46026-1**		
4	**KHUSHBU ALAM RAHI**	**22-46947-1**		
5	**A. F. M. RAFIUL HASSAN**	**22-47048-1**		

For faculty use only:

Total Marks: _______ **Marks Obtained:** _______

Faculty comments____

Introduction: Analog to Digital (ADC) is a fundamental process in digital signal processing and communications systems, where continuous analog signals are converted into discrete digital representation. Understanding the principles, techniques of ADC is crucial in various fields, including telecommunications, instrumentation, and control system.

In this experiment, we delve into the realm of ADC through the lens of MATLAB, a powerful numerical computing environment widely used for algorithm development and data analysis. Our objective is to gain insights into the conversion process and analyze the performance characteristics of ADC systems. By this experiment we aim to enhance our understanding of analog to digital conversion principles, gain proficiency in MATLAB based simulation techniques and develop insights that are valuable for practical applications in signal processing and communication systems.

Theory:

I) Analog to Digital conversion: An analog signal is superior to a digital signal. Converting an analog signal to digital data is the current trend. We discuss two methods in this section. These are delta modulation and pulse code modulation. After the creation of digital data (digitization), the digital data can be transformed into a digital signal using one of the line coding techniques.

II) Pulse Code Mudulation (PCM): Pulse code modulation (PCM) is the most widely used method for digitizing analog signals into digital data. There are three processes of PCM encoder. These are:

a) The analog signal is sampled.
b) The sample signal is quantized.
c) The quantized values are encoded as streams of bits.

III) Sampling: The first step in PCM is sampling. The analog signal is sampled every T_s s, where T_s is the sample internal or period. The inverse of the sampling interval is called the sampling ~~rate~~ rate or samplig frequency and denoted by f_s, where $f_s = 1/T_s$. There are three sampling methods. Those are, ideal, natural and flat-top. In ideal sampling, pulse from the analog signal are sampled. In natural sampling, a high speed switch in turned on for only the small period of time when the sampling occurs. The most common sampling method, called sample and hold. The sampling process is sometimes referred to as pulse amplitude modulation (PAM).

iv) Quantization: The following steps in quantization:

a) We assume that original analog signal has instantaneous amplitude between V_{min} and V_{max}.

b) We divide the range into L zones, each of height Δ delta

$$\Delta = \frac{V_{max} - V_{min}}{L}$$

c) We assign quantized values of 0 to L-1 to the midpoint of each zone.

d) We approximate the value of the sample amplitude to the quantized values.

v) Encoding: The last step in PCM is encoding. After each sample is quantized and the number of bits per sample is decided, each sample can be changed to an n_b-bit code word.

The number of bit is, $n_b = \log_2 L$.

bit rate, $BR = f_s \times n_b$.

ID = AB-CDEFG-H

ID = 22 – 47006 - 1

sig = a1*sin(2*pi*f1*t) + a2*cos(2*pi*f2*t) + a3*sin(2*pi*f3*t) + a4*sin(2*pi*f4*t)

The values of amplitude & frequency are as follows: a1 = F + 1, a2 = F + 3, a3 = F + 2, a4 = F + 4, f1 = G + 5, f2 = G + 7, f3 = G + 1, f4 = G + 2

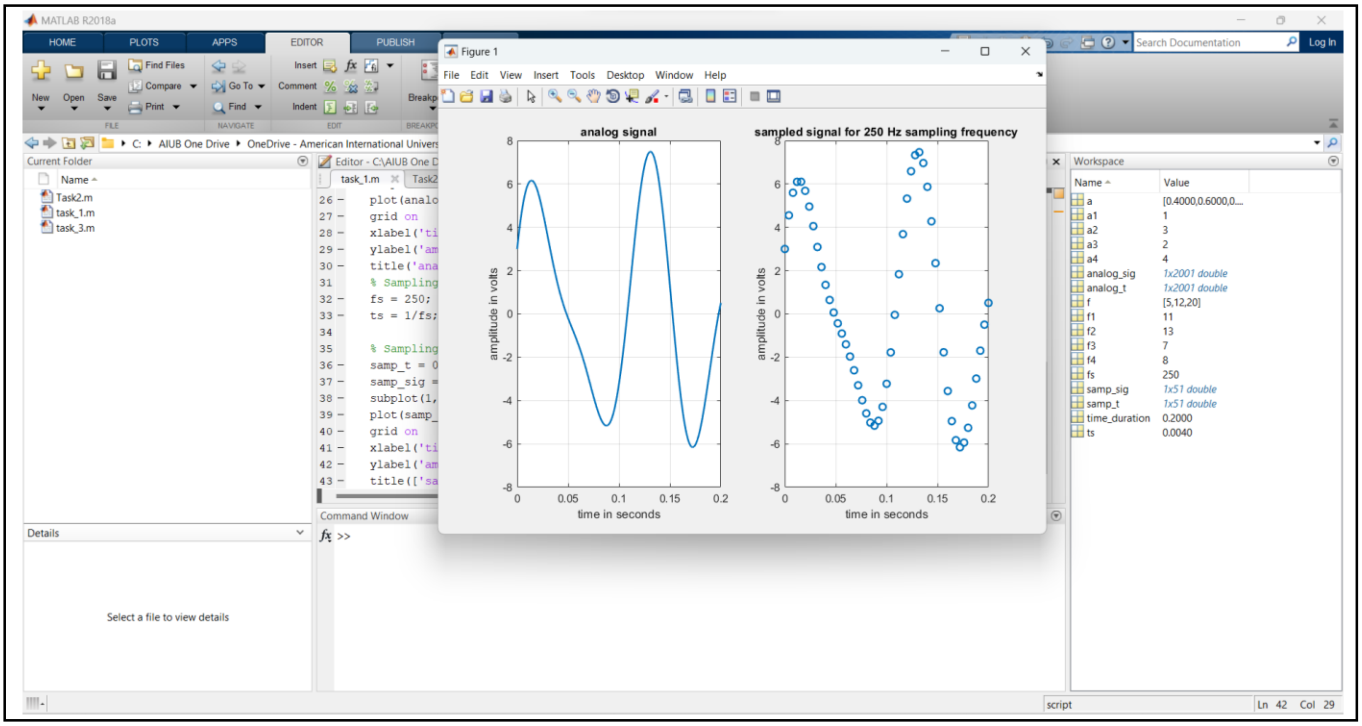
(a)

Show analog signal, sampled signal, and quantized signal.

Code & Simulation:

```
clc
clear all
close all

% 22-47006-1
% AB-CDEFG-H
a1= 1;
a2= 3;
a3= 2;
a4= 4;
f1= 11;
f2= 13;
f3= 7;
f4= 8;
% sig = a1*sin(2*pi*f1*t) + a2*cos(2*pi*f2*t) + a3*sin(2*pi*f3*t) +
a4*sin(2*pi*f4*t);
% [a1 = F + 1, a2 = F + 3, a3 = F + 2, a4 = F + 4, f1 = G + 5, f2 = G +
7, f3 = G + 1, f4 = G + 2]

time_duration = 0.2;

a = [0.4 0.6 0.8];
f = [5 12 20];
analog_t = 0:0.0001:time_duration;
analog_sig = a1*sin(2*pi*13*analog_t) + a2*cos(2*pi*15*analog_t) +
a3*sin(2*pi*9*analog_t) + a4*sin (2*pi*10*analog_t);
figure
subplot(1,2,1)
plot(analog_t, analog_sig,'linewidth',1.5)
grid on
xlabel('time in seconds')
ylabel('amplitude in volts')
title('analog signal')
% Sampling Frequency
fs = 250;
ts = 1/fs;

% Sampling
samp_t = 0:1/fs:time_duration;
samp_sig = a1*sin(2*pi*13*samp_t) + a2*cos(2*pi*15*samp_t) +
a3*sin(2*pi*9*samp_t) + a4*sin (2*pi*10*samp_t);
subplot(1,2,2)
plot(samp_t, samp_sig,'o','linewidth',1.5)
grid on
```

```
xlabel('time in seconds')
ylabel('amplitude in volts')
title(['sampled signal for ',num2str(fs),' Hz sampling frequency'])
```



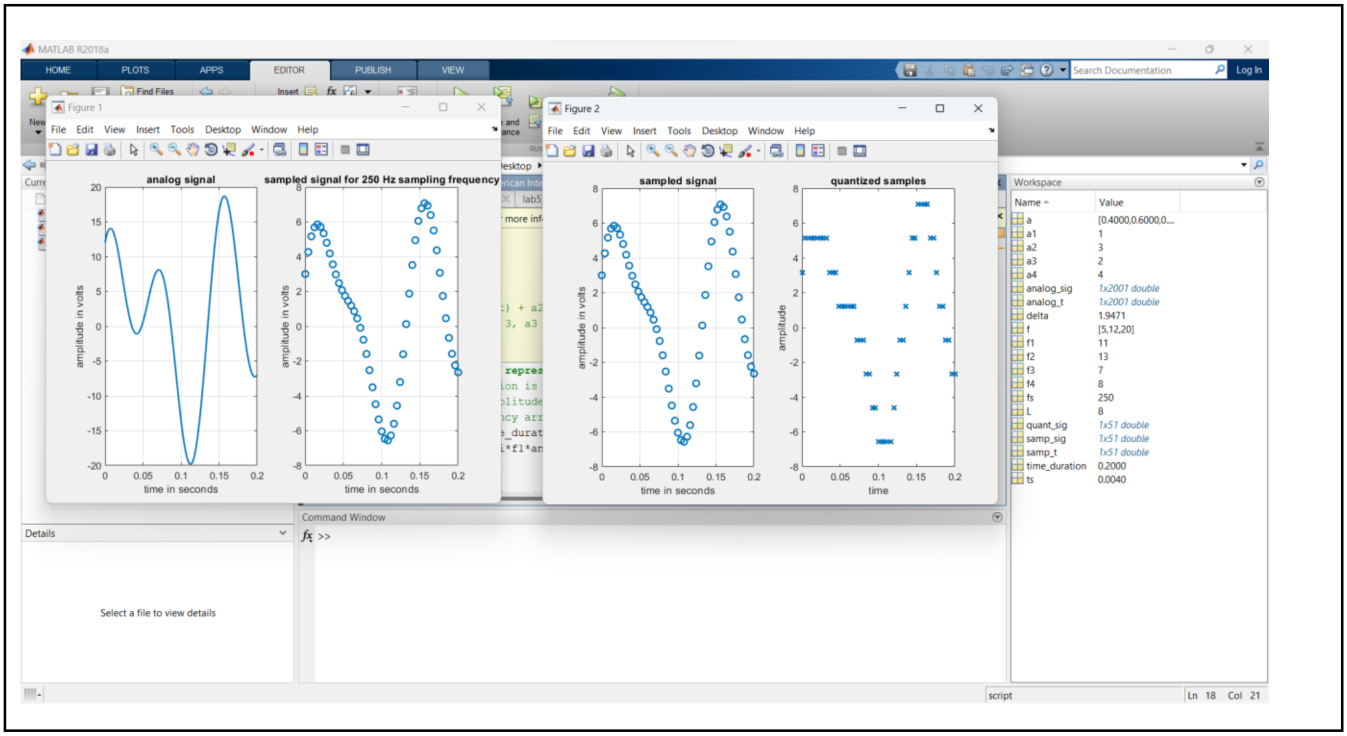
(b)

Show the digital data from the analog signal.

Code & Simulation:

```
clc
clear all
close all

% 22-47006-1
% AB-CDEFG-H
a1= 1;
a2= 3;
a3= 2;
a4= 4;
f1= 11;
f2= 13;
f3= 7;
f4= 8;
% sig = a1*sin(2*pi*f1*t) + a2*cos(2*pi*f2*t) + a3*sin(2*pi*f3*t) +
a4*sin(2*pi*f4*t);
% [a1 = F + 1, a2 = F + 3, a3 = F + 2, a4 = F + 4, f1 = G + 5, f2 = G +
7, f3 = G + 1, f4 = G + 2]

time_duration = 0.2;
%% Analog-like signal's representation
% Analog signal generation is not possible in MATLAB
a = [0.4 0.6 0.8]; % amplitude array for composite signal
```

```
f = [5 12 20]; % frequency array for composite signal
analog_t = 0:0.0001:time_duration;
analog_sig = a1*sin(2*pi*f1*analog_t) + 12*cos(2*pi*f2*analog_t) +
7*sin(a3*pi*f3*analog_t) + a4*sin (2*pi*f4*analog_t);
figure
subplot(1,2,1)
plot(analog_t, analog_sig,'linewidth',1.5)
grid on
xlabel('time in seconds')
ylabel('amplitude in volts')
title('analog signal')
%% Sampling Frequency
fs = 250;
ts = 1/fs;

samp_t = 0:1/fs:time_duration;
samp_sig = a1*sin(2*pi*f1*samp_t) + a2*cos(2*pi*f2*samp_t) +
a3*sin(2*pi*f3*samp_t) + a4*sin (2*pi*f4*samp_t);
subplot(1,2,2)
plot(samp_t, samp_sig,'o','linewidth',1.5)
grid on
xlabel('time in seconds')
ylabel('amplitude in volts')
title(['sampled signal for ',num2str(fs),' Hz sampling frequency'])

%% Levels for Quantization
L = 8;
delta = (max(samp_sig) - min(samp_sig))/(L-1); % step size
quant_sig = min(samp_sig) + round((samp_sig -
min(samp_sig))/delta)*delta; % quantized signal
figure
subplot(1,2,1)
plot(samp_t, samp_sig,'o','linewidth',1.5)
grid on
xlabel('time in seconds')
ylabel('amplitude in volts')
title('sampled signal')
subplot(1,2,2)
plot(samp_t, quant_sig,'x','linewidth',1.5);
xlabel('time')
ylabel('amplitude')
title('quantized samples')
```



(c)

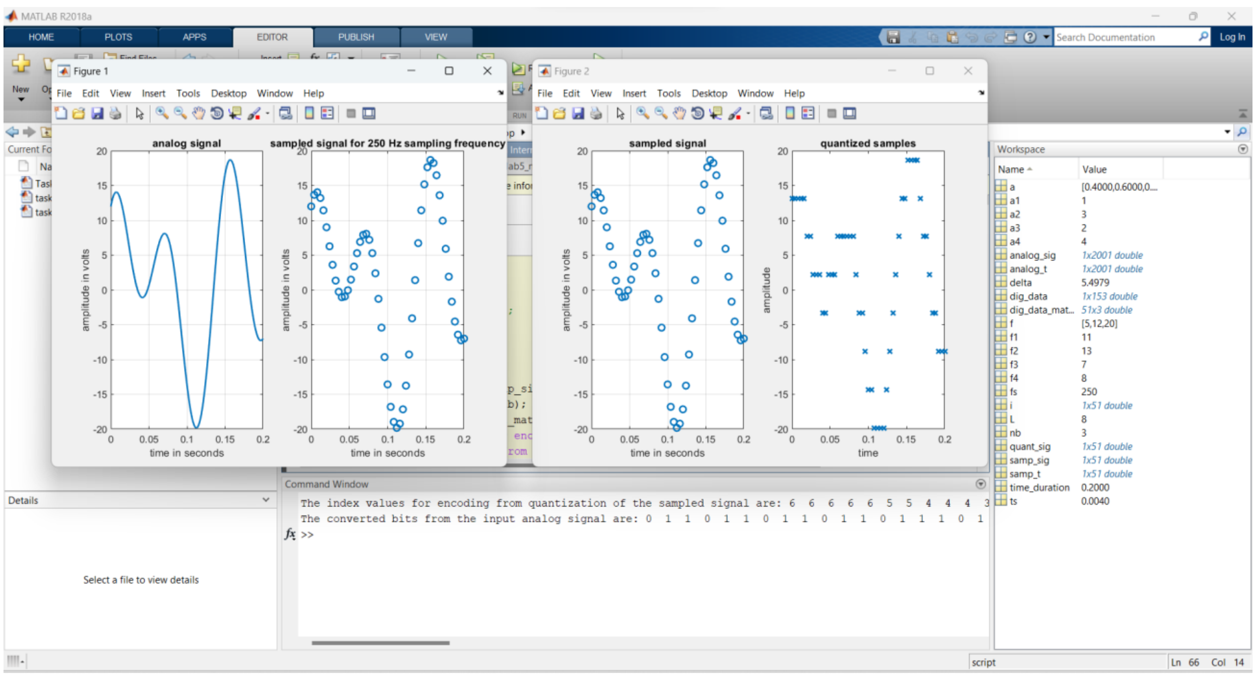
What are the appropriate values of sampling frequency and number of levels of quantization if minimum required SNR and bandwidth of the channel are 25 dB and 150 Hz respectively.

Code & Simulation:

```
clc
clear all
close all

% 22-47006-1
% AB-CDEFG-H
a1= 1;
a2= 3;
a3= 2;
a4= 4;
f1= 11;
f2= 13;
f3= 7;
f4= 8;
% sig = a1*sin(2*pi*f1*t) + a2*cos(2*pi*f2*t) + a3*sin(2*pi*f3*t) +
a4*sin(2*pi*f4*t);
% [a1 = F + 1, a2 = F + 3, a3 = F + 2, a4 = F + 4, f1 = G + 5, f2 = G +
7, f3 = G + 1, f4 = G + 2]

time_duration = 0.2;

a = [0.4 0.6 0.8];
f = [5 12 20];
analog_t = 0:0.0001:time_duration;
```

```
analog_sig = a1*sin(2*pi*f1*analog_t) + 12*cos(2*pi*f2*analog_t) +
7*sin(a3*pi*f3*analog_t) + a4*sin (2*pi*f4*analog_t);
figure
subplot(121)
plot(analog_t, analog_sig,'linewidth',1.5)
grid on
xlabel('time in seconds')
ylabel('amplitude in volts')
title('analog signal')
%% Sampling Frequency
fs = 250;
ts = 1/fs;

%% Sampling
samp_t = 0:1/fs:time_duration;
samp_sig = a1*sin(2*pi*f1*samp_t) + 12*cos(2*pi*f2*samp_t) +
7*sin(a3*pi*f3*samp_t) + a4*sin (2*pi*f4*samp_t);
subplot(122)
plot(samp_t, samp_sig,'o','linewidth',1.5)
grid on
xlabel('time in seconds')
ylabel('amplitude in volts')
title(['sampled signal for ',num2str(fs),' Hz sampling frequency'])

%% Levels for Quantization
L = 8;

delta = (max(samp_sig) - min(samp_sig))/(L-1); % step size
quant_sig = min(samp_sig) + round((samp_sig -
min(samp_sig))/delta)*delta; % quantized signal
figure
subplot(1,2,1)
plot(samp_t, samp_sig,'o','linewidth',1.5)
grid on
xlabel('time in seconds')
ylabel('amplitude in volts')
title('sampled signal')
subplot(1,2,2)
plot(samp_t, quant_sig,'x','linewidth',1.5);
xlabel('time')
ylabel('amplitude')
title('quantized samples')

%% Number of Bits/Sample
nb = log2(L);
%% Encoding
% SNRdb = 25;

% nb = (SNRdb - 1.76)/6.02);
% L = ceil(2^nb)
```

```
i = round((samp_sig-min(samp_sig))/delta);
dig_data_matrix = de2bi(i,nb);
dig_data = reshape(dig_data_matrix',1,[]);
disp(['The index values for encoding from quantization of the sampled signal are: ',num2str(i)])
disp(['The converted bits from the input analog signal are: ',num2str(dig_data)])
```



Conclusion: The experiment has provided a comprehensive understanding of analog to digital conversion principles and techniques using MATLAB simulations. This experiment was completed properly without any errors.