AMERICAN INTERNATIONAL UNIVERSITY–BANGLADESH (AIUB)

DATA COMMUNICATION

Spring 2024-2025

Section: D

Study of Analog to Digital Conversion using MATLAB

LAB REPORT ON

Supervised By

DR. MD. HUMAYUN KABIR

Submitted By

Name	ID
S. M. RASEL	22-48039-2

Date of Submission: 14/04/2025

22-48039-2

Title: Study of Analog to Digital Conversion using -MATLAB

Objective:

The objective of this experiment was to understand the use of MATLAB of the solving communication engineering problem. It also developmed and understanding of Anlog to Digital Convesion using MATLAB.

Working principle:

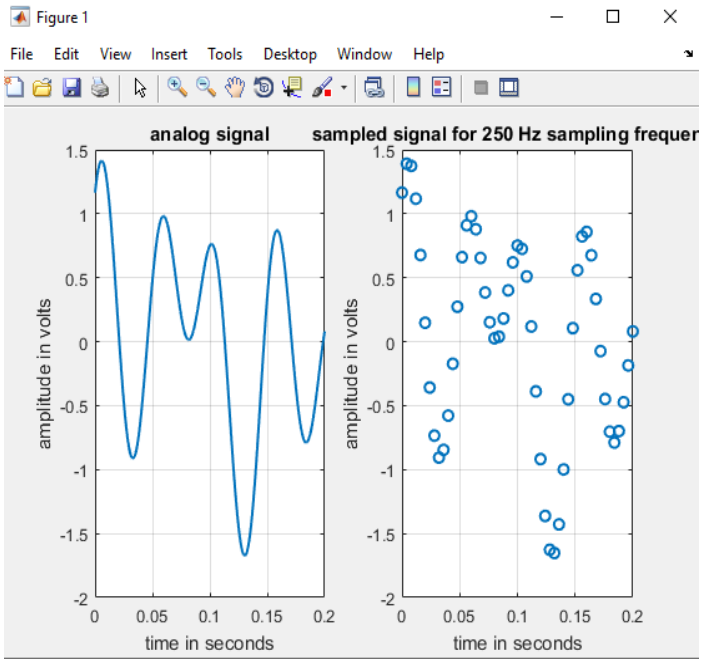
Analog signal → Sampling → Encoding → Quantizaing → → Digital Data (signal)

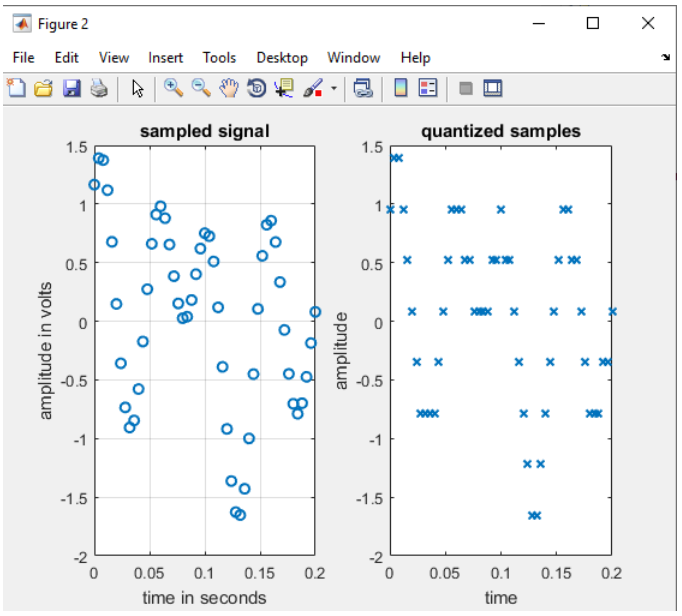
Simulation tools: MATLABR2016a

```
clc
clear all
close all

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Analog to Digital Conversion
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

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 = a(1)*sin(2*pi*f(1)*analog_t) +
a(2)*cos(2*pi*f(2)*analog_t) + a(3)*sin(2*pi*f(3)*analog_t +
pi/4);

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 = a(1)*sin(2*pi*f(1)*samp_t) +
a(2)*cos(2*pi*f(2)*samp_t) + a(3)*sin(2*pi*f(3)*samp_t + pi/4);

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;

%% Quantizing
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
i = round((samp_sig-min(samp_sig))/delta); % index for encoding
dig_data_matrix = de2bi(i,nb); % encoded binary bits are as a
matrix here
dig_data = reshape(dig_data_matrix',1,[]); % encoded binary bits
are as an array here
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)])
```

Simulations & Code Screen Short:





Performance Task:
The selected ID is following:

The selected ID is the following:

2	2	-	4	8	0	3	9	-	2
A	**B**		**C**	**D**	**E**	**F**	**G**		**H**

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)
= 11*sin (2*pi*12*t) + 12*cos (2*pi*14*t) + 11*sin (2*pi*8*t) + 13*sin (2*pi*9*t)

Code:

```
clc
clear all close all
time_duration = 0.2;
a = [0.4 0.6 0.8];
Composite signal f = [5 12 20];
array for composite signal analog_t = 0:0.0001:time_duration;
analog_sig=11*sin (2*pi*12*analog_t)+12*cos(2*pi*14*analog_t) + 11*sin(2*pi*8*analog_t + pi/4); 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')
fs = 250;

ts = 1/fs;
samp_t = 0:1/fs:time_duration;
samp_sig = 9*sin(2*pi*8*samp_t) + 11*cos(2*pi*10*samp_t) + 10*sin(2*pi*4*samp_t + pi/4); 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'])
L = 8;
delta = (max(samp_sig) - min(samp_sig))/(L-1);
quant_sig = min(samp_sig) + round((samp_sig-min(samp_sig))/delta)*delta;
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')
Bits/Sample nb = log2(L);
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)
```

Simulations & Code Screen Short:

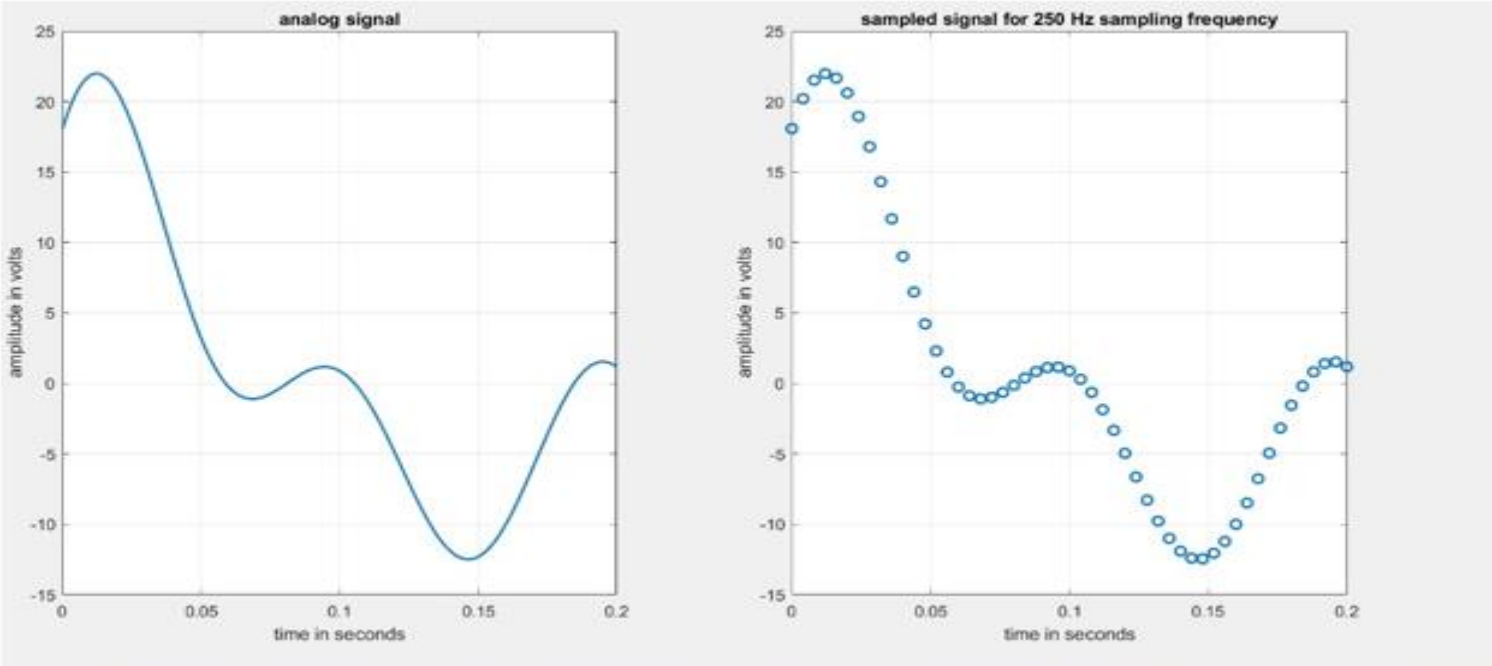


Fig 1: Analog Signal and Sampled signal for 250 Hz.

Data Sheet:

EXP NO: 5
Name : S. M. Rasel
ID : 22-48039-2

a = amplitude
Φ = step size
L = levels

index encoding for Encoding
Φ = round(x − xmin)/Δ

$y = A_{max} \cos(2\pi f t + \phi)$

$y = 0.4 \times \cos(2\pi \times 0.4 + t) + 0.6 \cos(2\pi \times 11 \times t + \pi/2)$

```
clc
clear all
close all
time_deration = 0.2;
a = [0.4  0.6  0.8];
f = [5  12  20];
analog_t = 0:0.0001:time_duration;
analog_sig = a(1)*sin(2*pi*f(1)*analog_t) +
             a(2)*cos(2*pi*f(2)*analog_t) + a(3)
             *sin(2*pi*f(3)*analog_t + pi/4);
a(2)*cos(2*pi*f(2)*analog_t) + a(3)*sin(2*pi*f(3)
*analog_t + pi/4);
figure,
subplot(1,2,1)
plot(analog_t, analog_sig, 'LineWidth', 1.5)
grid on
xlabel('time in seconds')
ylabel('amplitude in volts')
```

Discussion and Conclusion: 22-48039-2

In this experiment, we learnd different methods of Digital to Analog conversion. PULSE CODE MODULATION (PCM) is most common techique to change an analog signal to digital signal or data. We Implemented the processes of PCM using MATLAB. Analog, quantized and sampled signals have been shown in the report.

References:

1. Prakash C. Gupta. "Data communcation", prentice Hall India pvt.
2. Forouzan, B.A "Data Communcation and Network, Tata McGraw" (2005)
3. AIUB Data Com. Lab Manual, Report 5.