American International University- Bangladesh
Department of Computer Engineering
COE 3201: Data Communication Laboratory

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

Abstract:

This experiment is designed to-

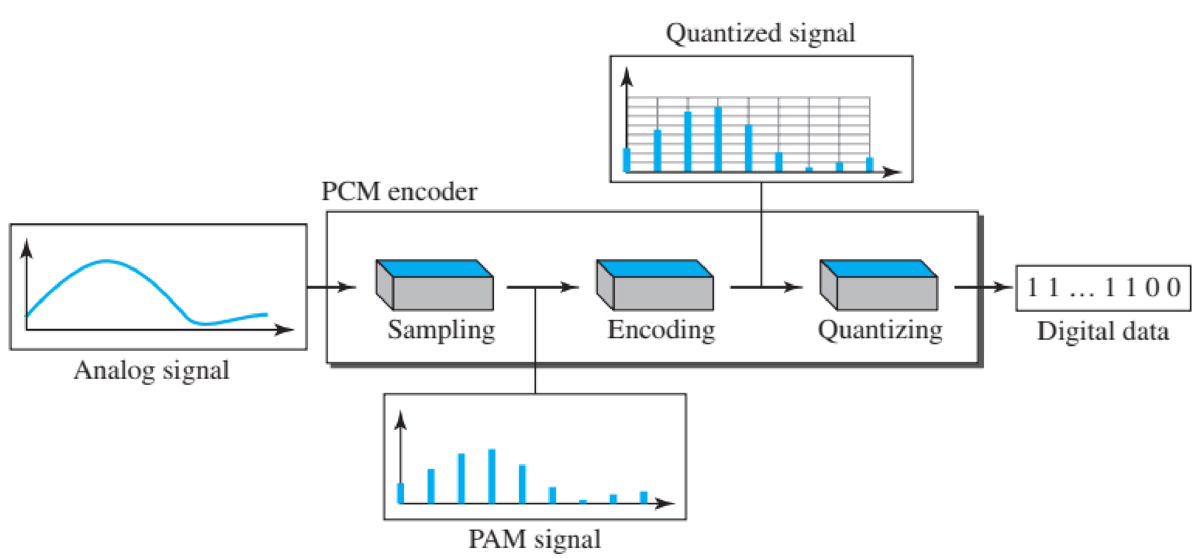
1.To understand the use of MATLAB for solving communication engineering problems.
2.To develop understanding of Digital to Analog conversion using MATLAB.

Introduction:

I. ANALOG TO DIGITAL CONVERSION: Digital signal is superior to analog signal. The tendency today is to change an analog signal to digital data. In this section we describe two techniques, pulse code modulation and delta modulation. After the digital data are created (digitization), we can use one of the line coding techniques to convert the digital data to a digital signal.

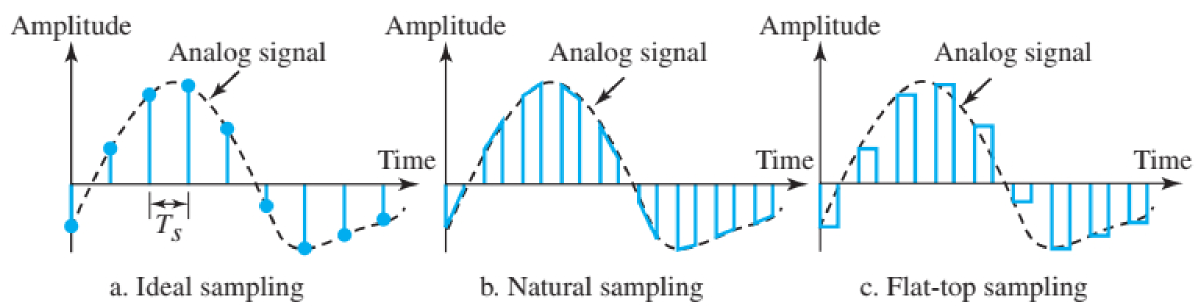
II. PULSE CODE MODULATION (PCM): The most common technique to change an analog signal to digital data (digitization) is called pulse code modulation (PCM). A PCM encoder has three processes, as shown in the following figure.
   a. The analog signal is sampled.
   b. The sampled signal is quantized.
   c. The quantized values are encoded as streams of bits

Figure 1 *Components of PCM encoder*



III. SAMPLING: The first step in PCM is sampling. The analog signal is sampled every T_s s, where T_s is the sample interval or period. The inverse of the sampling interval is called the sampling rate or sampling frequency and denoted by f_s, where $f_s = 1/T_s$. There are three sampling methods—ideal, natural, and flat-top—as shown in the following figure.

Figure 2 *Three different sampling methods for PCM*



In ideal sampling, pulses from the analog signal are sampled. This is an ideal sampling method and cannot be easily implemented. In natural sampling, a high-speed switch is turned on for only the small period of time when the sampling occurs. The result is a sequence of samples that retains the shape of the analog signal. The most common sampling method, called sample and hold, however, creates flat-top samples by using a circuit. The sampling process is sometimes referred to as pulse amplitude modulation (PAM). We need to remember, however, that the result is still an analog signal with nonintegral values. According to the Nyquist theorem, the sampling rate must be at least twice the highest frequency contained in the signal.

$$f_s = 2 \times f_{max}$$

IV. QUANTIZATION: The following are the steps in quantization:

a. We assume that the original analog signal has instantaneous amplitudes 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. As a simple example, assume that we have a sampled signal and the sample amplitudes are between −20 V and +20 V. We decide to have eight levels (L = 8). This means that Δ = 5 V

QUANTIZATION LEVELS:

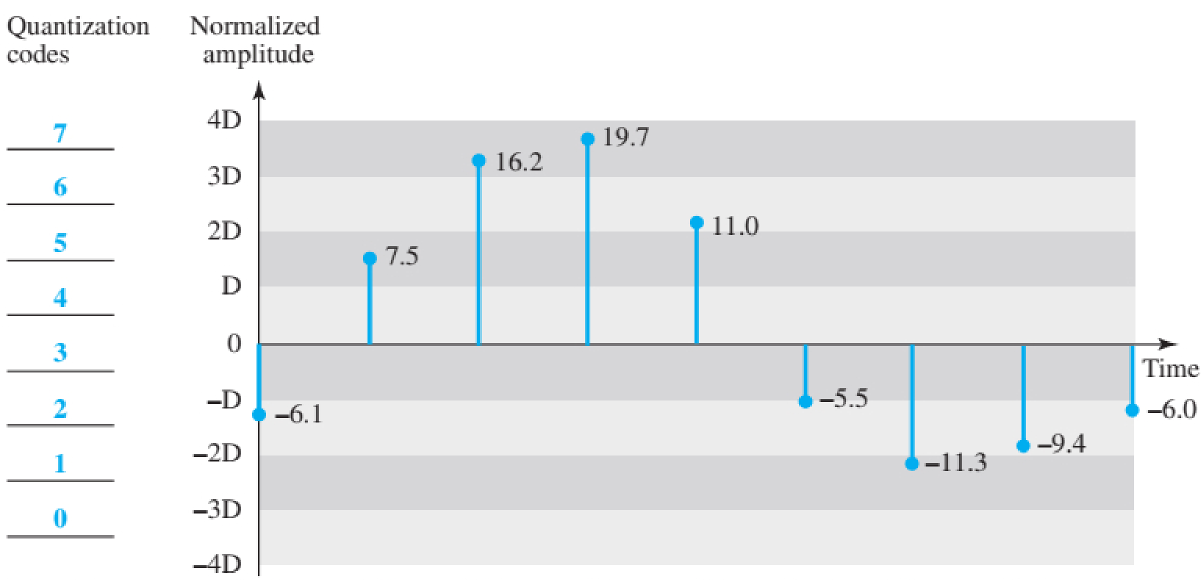
The choice of L, the number of levels, depends on the range of the amplitudes of the analog signal and how accurately we need to recover the signal.

QUANTIZATION ERROR:

The contribution of the quantization error to the SNR_{dB} of the signal depends on the number of quantization levels L, or the bits per sample n_b, as shown in the following formula:

$$SNR_{dB} = 6.02n_b + 1.76 \text{ dB}$$

Figure 3 *Quantization and encoding of a sampled signal*



Normalized PAM values	–1.22	1.50	3.24	3.94	2.20	–1.10	–2.26	–1.88	–1.20
Normalized quantized values	–1.50	1.50	3.50	3.50	2.50	–1.50	–2.50	–1.50	–1.50
Normalized error	–0.38	0	+0.26	–0.44	+0.30	–0.40	–0.24	+0.38	–0.30
Quantization code	2	5	7	7	6	2	1	2	2
Encoded words	010	101	111	111	110	010	001	010	010

A simple method of quantization which can be implemented on MATLAB is given below:

Input Analog Signal = x
Number of Quantization Levels = L
Step Size, $\Delta = (x_{max} - x_{min})/(L-1)$
Quantized Signal, $x_q = x_{min} + \text{round}((x - x_{min})/\Delta)*\Delta$

where, x_{max} and x_{min} are the maximum value and minimum values, respectively, of the analog input signal x. The symbol L denotes the number of quantization levels. The symbol Δ is the step size of the quantizer or the ADC resolution. Finally, x_q indicates the quantization level, and i is an index corresponding to the binary code.

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. Note that the number of bits for each sample is determined from the number of quantization levels. If the number of quantization levels is L, the number of bits is

$$n_b = log_2(L)$$

Index for Encoding,

$$i = round((x - x_{min})/\Delta)$$

Required bit rate (BR) for the encoding scheme can be determined as

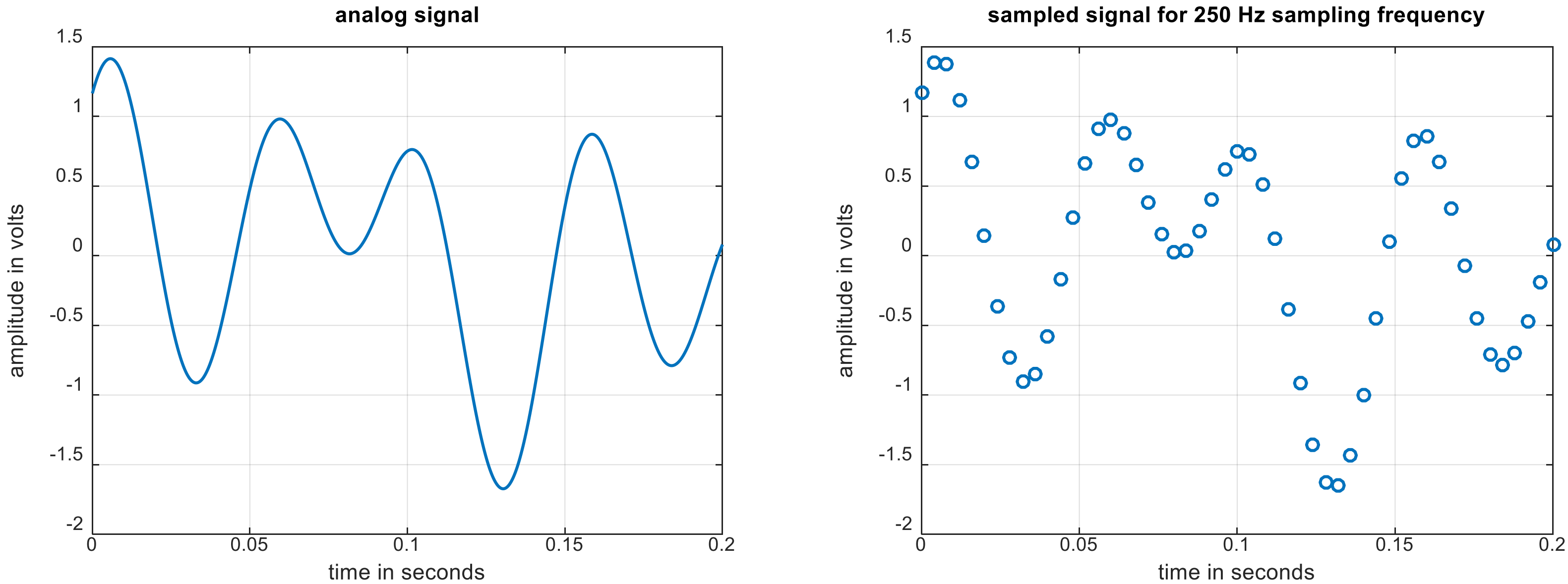
$$BR = f_s \times n_b$$

MATLAB code for Pulse Code Modulation (Sampling, Quantizing, Encoding) of an Analog Signal:

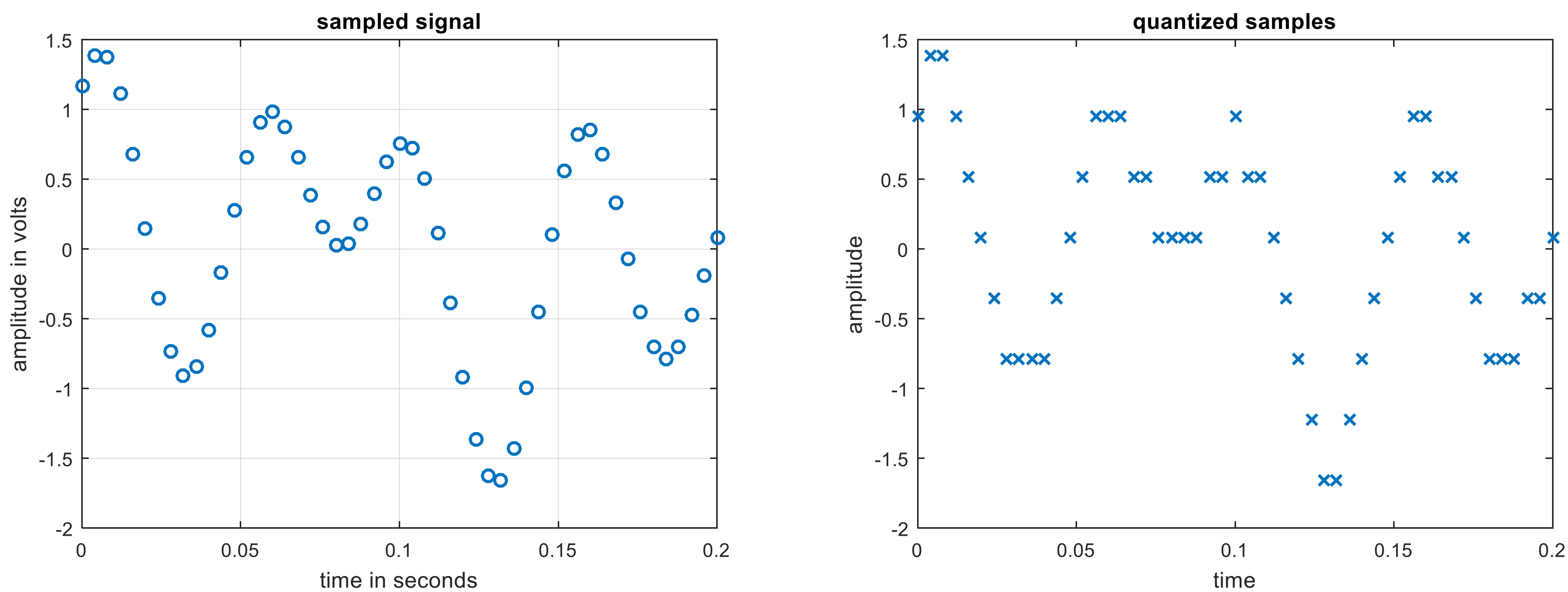
```
clc
clear all
close all

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 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)])
```

Output Figure 1 for Sampling:



Output Figure 2 for Quantizing:



Output for Encoding:

```
The index values for encoding from quantization of the sampled
signal are: 6   7   7   6   5   4   3   2   2   2   2   3   4   5   6   6   6   5
5   4   4   4   4   5   5   6   5   5   4   3   2   1   0   0   1   2   3   4   5   6
6   5   5   4   3   2   2   2   3   3   4

The converted bits from the input analog signal are:  0   1   1   1
1   1   1   1   1   0   1   1   1   0   1   0   0   1   1   1   0   0   1   0   0   1
0   0   1   0   0   1   0   1   1   0   0   0   1   1   0   1   0   1   1   0   1   1
0   1   1   1   0   1   1   0   1   0   0   1   0   0   1   0   0   1   0   0   1   1
0   1   1   0   1   0   1   1   1   0   1   1   0   1   0   0   1   1   1   0   0   1
0   1   0   0   0   0   0   0   0   0   1   0   0   0   1   0   1   1   0   0   0   1
1   0   1   0   1   1   0   1   1   1   0   1   1   0   1   0   0   1   1   1   0   0
1   0   0   1   0   0   1   0   1   1   0   1   1   0   0   0   1
```

Software:
MATLAB2016a

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

Convert the following analog signal into digital data:

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]

a) Show analog signal, sampled signal, and quantized signal.
b) Show the digital data from the analog signal.
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.