AMERICAN INTERNATIONAL UNIVERSITY–BANGLADESH (AIUB)

FACULTY OF ENGINEERING

Course name: Data Communication

Course code: COE 3201

Section: H

Semester: Spring 2023-24

Name: MD. ABU TOWSIF

ID: 22-47019-1

Instructor name: Dr. Muhammad Morshed Alam

Experiment no: 03

Experiment name: Analog Signal quantization using MATLAB

Submission date: Feb 28th, 2024

ID: AB-CDEFG-H

Write a MATLAB code that can generate an approximated quantized signal for the following analog function:

$$x_1(t) = A_1 \cos(2\pi(CDE * 100)t)$$

(a) Define the amplitude $A_1 = GD$, sampling frequency, define the time domain t for function $x_1(t)$ that gives at least 3 complete cycles.
(b) Define the number of quantization levels, step size or resolution, then find the quantized signal x_q.
(c) Obtain the absolute quantization error, $err = abs(x_1 - x_q)$

Finally, use 2x2 subplot to plot analog signal $x_1(t)$, sampling signal of $x_1(t)$, quantized signal $x_q(t)$, and quantized error signal err.

ANSWER:

(a) Define the amplitude $A_1 = GD$, sampling frequency, define the time domain t for function $x_1(t)$ that gives at least 3 complete cycles.

My id:

ID = 22-47019-1
A1 = GD = 97; //Amplitude
F = CDE = 470; // Frequency

So,

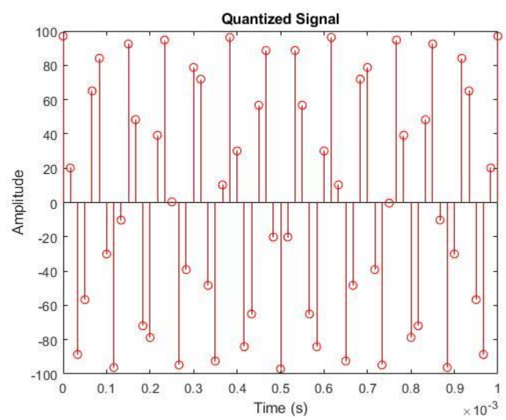
$$x_1(t) = A_1 \cos(2\pi(CDE * 100)t)$$

(b) Define the number of quantization levels, step size or resolution, then find the quantized signal x_q.

<table>
<tr><th>MATLAB Code</th><th>Output Figure</th></tr>
<tr><td>

```
A1 = 97;
Sampling_Frequency = 60e3;
CDE = 470;
Num_Quantization_Levels = 8;
Duration = 3;
Resolution = (2 * A1) /
(2^Num_Quantization_Levels);
t=0:1/Sampling_Frequency:0.001;

x1 = A1 * cos(2 * pi * (CDE *
100) * t);

L=(2^Num_Quantization_Levels)-
1;

delta=(max(x1)-min(x1))/L;

Quantized_Signal =
min(x1)+(round((x1-
min(x1))/delta)).*delta;

stem(t, Quantized_Signal, 'r')
title('Quantized Signal')
xlabel('Time (s)')
ylabel('Amplitude')
```

</td><td>



</td></tr>
</table>

(C) Obtain the absolute quantization error, $err = abs(x_1 - x_q)$

<table>
<tr><th>MATLAB Code</th><th>Output Figure</th></tr>
<tr><td>
A1 = 97;
Sampling_Frequency = 60e3;
CDE = 470;
Num_Quantization_Levels = 8;
Duration = 3;
Resolution = (2 * A1) / (2^Num_Quantization_Levels);
t=0:1/Sampling_Frequency:0.001;

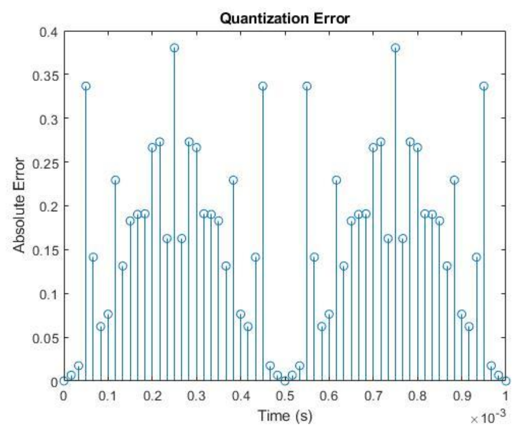
x1 = A1 * cos(2 * pi * (CDE * 100) * t);

L=(2^Num_Quantization_Levels)-1;

delta=(max(x1)-min(x1))/L;

Quantized_Signal = min(x1)+(round((x1-min(x1))/delta)).*delta; %xq

err = abs(x1 - Quantized_Signal);

stem(t, err)
title('Quantization Error')
xlabel('Time (s)')
ylabel('Absolute Error')
</td><td>


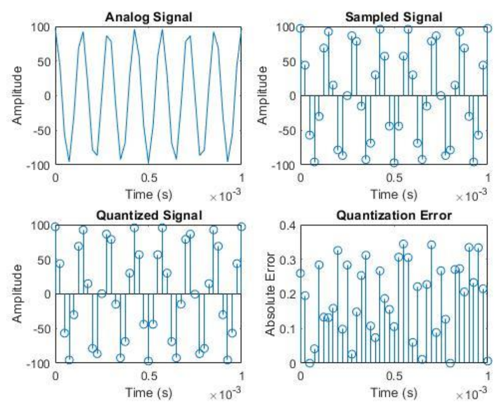
</td></tr>
</table>

(d) Finally, use 2x2 subplot to plot analog signal $x_1(t)$, sampling signal of $x_1(t)$, quantized signal $x_q(t)$, and quantized error signal err.

<table>
<tr><th>MATLAB Code</th><th>Output Figure</th></tr>
<tr><td>

```
A1 = 97;
Sampling_Frequency = 40e3;
CDE = 470;
Num_Quantization_Levels = 8;
Duration = 3;
Resolution = (2 * A1) /
(2^Num_Quantization_Levels);
t=0:1/Sampling_Frequency:0.001;

x1 = A1 * cos(2 * pi * (CDE *
100) * t);

L=(2^Num_Quantization_Levels)-
1;

delta=(max(x1)-min(x1))/L;

Quantized_Signal =
min(x1)+(round((x1-
min(x1))/delta)).*delta;


subplot(2,2,1)
plot(t, x1)
title('Analog Signal')
xlabel('Time (s)')
ylabel('Amplitude')

subplot(2,2,2)
stem(t, x1)
title('Sampled Signal')
xlabel('Time (s)')
ylabel('Amplitude')

subplot(2,2,3)
stem(t, Quantized_Signal)
title('Quantized Signal')
xlabel('Time (s)')
ylabel('Amplitude')
```

</td><td>



</td></tr>
</table>

`subplot(2,2,4)`
`stem(t, Quantization_Error)`
`title('Quantization Error')`
`xlabel('Time (s)')`
`ylabel('Absolute Error')`	