Tribhuvan University

Institute of Engineering

Thapathali Campus, Thapathali

Telecommunication Lab 2

Submitted By:

Name: Raman Bhattarai

Roll No.: THA077BEI033

Submitted To:

Department of Electronics and Computer Engineering

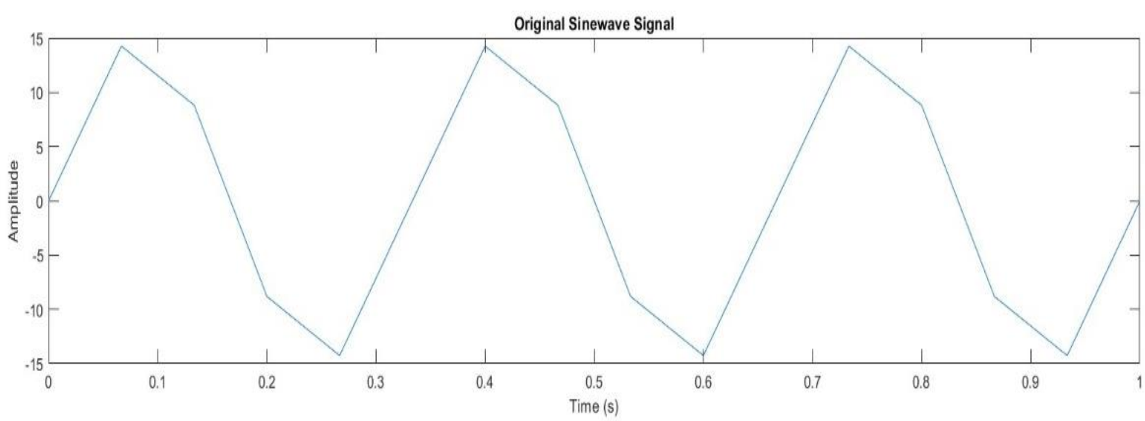
Date: Feb 28, 2025

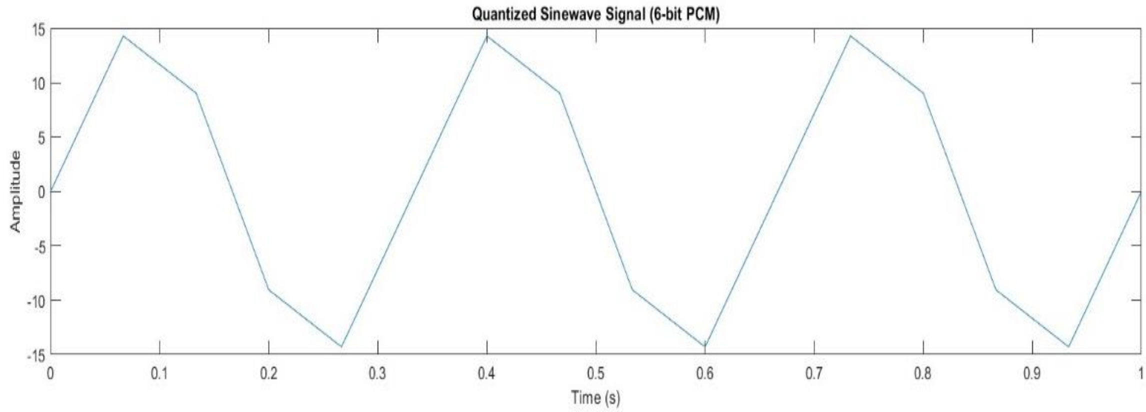
1. Design a 6-bit PCM scheme in MATLAB for a Sinewave signal that is being sampled at a frequency of 15 HZ. The amplitude of the signal is 15 and the sinusoidal frequency is 3 Hz.

Source Code:

```
fs = 15;
t = 0:1/fs:1;
f = 3;
A = 15;
x = A * sin(2*pi*f*t);

N = 2^6;
q = (2^6 - 1) / 2;
x_q = round(x / A * q) / q * A;

figure;
subplot(2,1,1);
plot(t, x);
title('Original Sinewave Signal');
xlabel('Time (s)');
ylabel('Amplitude');

subplot(2,1,2);
plot(t, x_q);
title('Quantized Sinewave Signal (6-bit PCM)');
xlabel('Time (s)');
ylabel('Amplitude');
```

Output:





Conclusion:

This code demonstrates the process of quantizing a sinewave signal using a 6-bit Pulse Code Modulation (PCM) scheme, sampled at 15 Hz. It shows how the original continuous sinewave is converted into discrete values, with the quantization process rounding the signal to 64 possible levels. By visualizing both the original and quantized signals, the code highlights the effect of quantization on signal accuracy, with a visible reduction in precision due to the 6-bit resolution.

2. Explain how the choice of a 2-bit PCM scheme and 5- bit PCM scheme differently affects the output.

The choice of a 2-bit PCM scheme versus a 5-bit PCM scheme significantly affects the quantization levels and signal fidelity of the output.

i. 2-bit PCM Scheme:
   a. A 2-bit PCM scheme has only 4 quantization levels (\(2^2 = 4\)), which means the signal can only take one of four discrete values. This coarse quantization leads to a large quantization error because the continuous signal is mapped to very few levels, resulting in significant distortion. The output will have visible steps or jumps, and the accuracy of the representation will be poor, especially for signals with fine variations.
   b. The signal will lose much of its smoothness, and the resulting waveform may look jagged or blocky due to the limited number of levels.

ii. 5-bit PCM Scheme:
   a. A 5-bit PCM scheme offers 32 quantization levels (\(2^5 = 32\)), providing a finer representation of the original signal. This results in a smaller quantization error and better approximation of the original continuous signal. The output will have much more precision compared to the 2-bit scheme, with smoother transitions between levels and less noticeable distortion.
   b. While still a relatively low-bit scheme compared to higher resolutions, the 5-bit PCM representation will better preserve the signal's shape and provide a more accurate reproduction of the original waveform.

Key Differences:

a. Resolution: The 5-bit scheme provides 32 levels, offering a finer resolution and smoother output, while the 2-bit scheme, with only 4 levels, results in a coarser and more distorted signal.
b. Signal Fidelity: The 5-bit scheme will preserve more of the original signal's details, whereas the 2-bit scheme will result in significant loss of information, making the output more approximate.
c. Visual and Auditory Quality: In a 2-bit PCM scheme, you will observe visible steps in the waveform, and in an auditory context, the sound will be rougher. In contrast, the 5-bit PCM scheme provides smoother transitions and a clearer representation both visually and audibly.

3. What does the following MATLAB fucntions do?

1. stairs:
   The stairs function is used to create a stair-step plot of a given set of data. Unlike plot, which connects points with straight lines, stairs creates a piecewise constant graph where each value is held until the next step.Commonly used for discrete-time signals and quantized signals in digital communication.
2. dec2bin:
   The dec2bin function converts a decimal number into its binary (base-2) representation as a string.Useful in digital signal processing (DSP), binary data representation, and bit-level operations.
3. round:
   The round function rounds a given number to the nearest integer or a specified number of decimal places.Used in quantization, error correction, and situations requiring precise rounding of numerical values.
4. hold on:
   The hold on command allows multiple plots to be drawn on the same figure without erasing previous plots. It retains the current graph while adding new elements.Useful for overlaying multiple plots, comparing signals, and visualizing multiple data sets on the same graph.

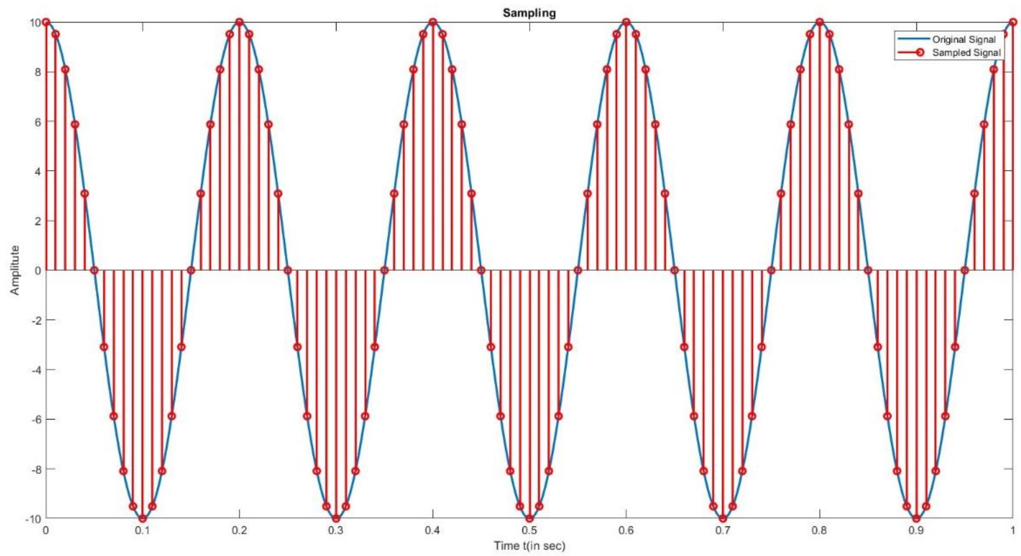
4. Understand the code provided in this lab and write your interpretation in your own words.

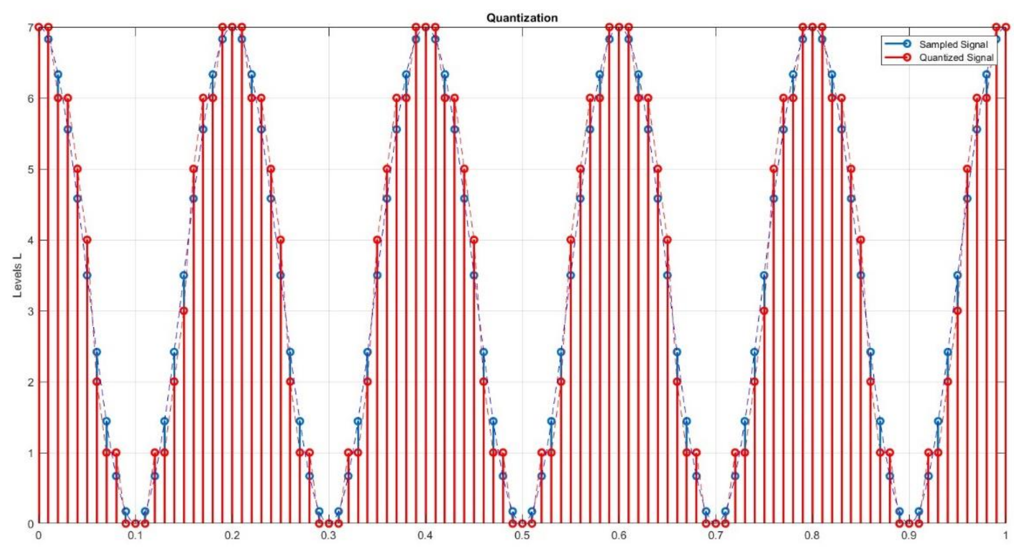
Source Code:

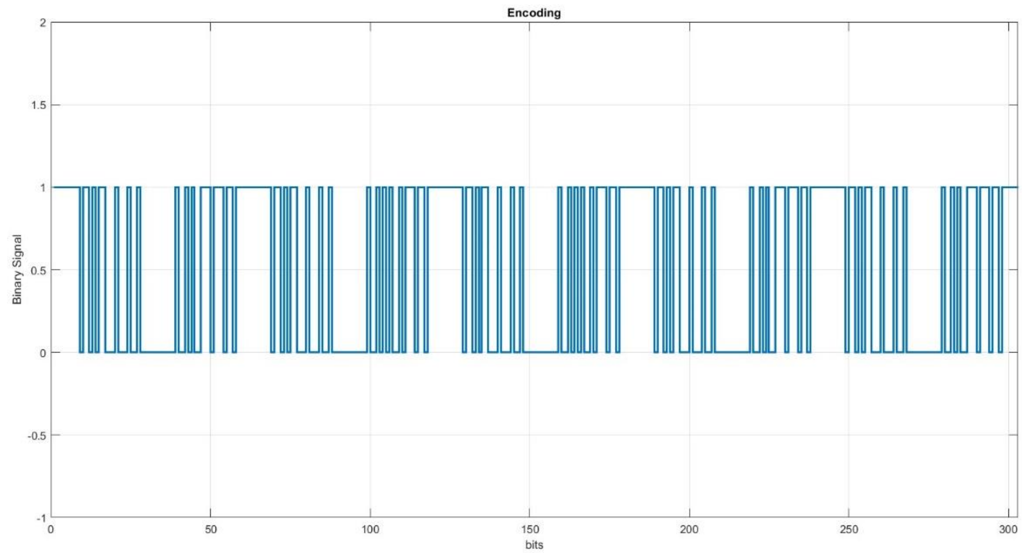
```
function [y Bitrate]=pcm(A,fm,fs,n)

t=0:1/(100*fm):1;
x=A*cos(2*pi*fm*t);

ts=0:1/fs:1;
xs=A*cos(2*pi*fm*ts);

x1=xs+A;
x1=x1/(2*A);
L=(-1+2^n);
x1=L*x1;
xq=round(x1);
r=xq/L;
r=2*A*r;
r=r-A;

y=[];
for i=1:length(xq)
d=dec2bin(xq(i),n);
y=[y double(d)-48];
end
Bitrate=n*fs;

figure(1)
plot(t,x,'linewidth',2)
title('Sampling')
ylabel('Amplitute')
xlabel('Time t(in sec)')
hold on
stem(ts,xs,'r','linewidth',2)
hold off
legend('Original Signal','Sampled Signal');

figure(2)
stem(ts,x1,'linewidth',2)
title('Quantization')
ylabel('Levels L')
hold on
stem(ts,xq,'r','linewidth',2)
plot(ts,xq,'--r')
plot(t,(x+A)*L/(2*A),'--b')
grid
hold off
legend('Sampled Signal','Quantized Signal');
```

```
figure(3)
stairs([y y(length(y))],'linewidth',2)
title('Encoding')
ylabel('Binary Signal')
xlabel('bits')
axis([0 length(y) -1 2])
grid
```

Output:







Conclusion:

This MATLAB function simulates Pulse Code Modulation (PCM) by performing three main steps: sampling, quantization, and encoding. It processes a cosine wave and visualizes the results through three plots.

1. Signal Generation (Original and Sampled Signal)

- A continuous cosine wave is generated with a given amplitude `A` and frequency `fm`.
- The signal is sampled at a frequency `fs`, creating a discrete-time representation.
- This step represents sampling, where the continuous signal is converted into discrete-time samples.

2. Quantization (Mapping Samples to Discrete Levels)

- The sampled values are normalized and scaled to fit within a range of `2^n` discrete levels (`n` is the number of bits per sample).
- The values are then rounded to the nearest quantization level.
- Finally, the quantized values are mapped back to their original amplitude range.
- This step introduces quantization error, as the original values are approximated by discrete levels.

3. Encoding (Binary Representation of Quantized Values)

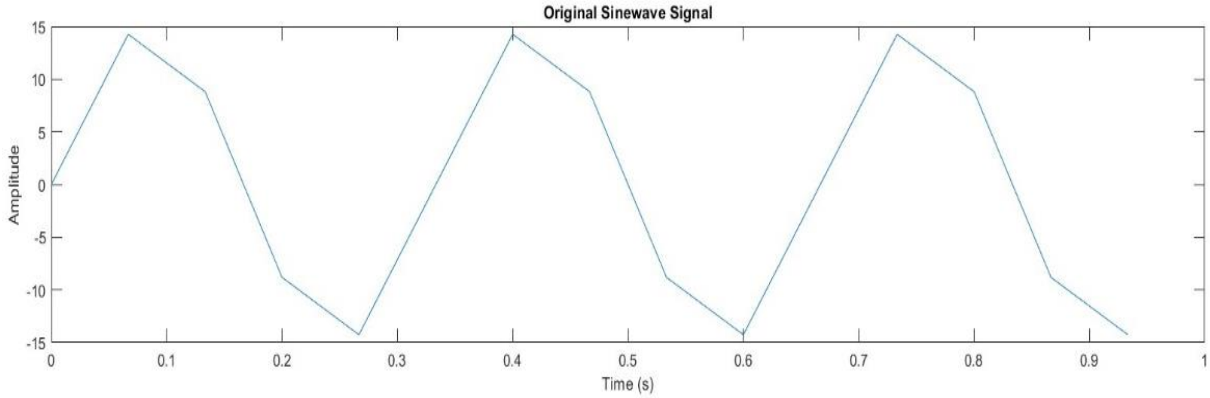
- Each quantized value is converted into an `n`-bit binary representation using `dec2bin()`.
- The bit rate is calculated as `Bitrate = n * fs`, indicating the number of bits transmitted per second.
- This step prepares the signal for digital transmission by encoding it in binary format.

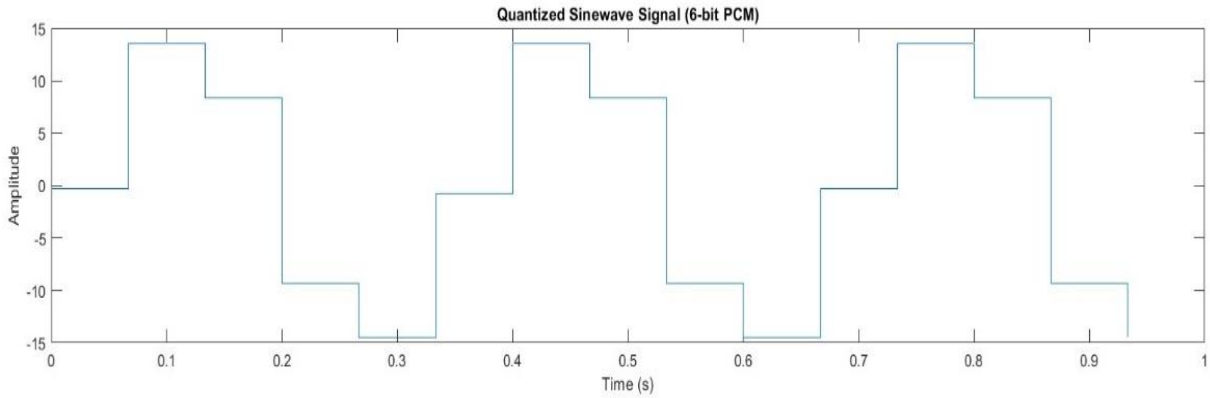
5. Use built-in Functions Quantize and Encode to perform PCM.

Source Code:

```
fs = 15;
t = 0:1/fs:1-1/fs;

f = 3;
A = 15;
x = A * sin(2*pi*f*t);

% PCM Quantization (6-bit)
N = 2^6;
q_levels = linspace(-A, A, N);  %
partition = (q_levels(1:end-1) + q_levels(2:end)) / 2;

index = discretize(x, partition); %
index(isnan(index)) = 1;

x_q = q_levels(index);

% PCM Encoding
x_encoded = dec2bin(index - 1, 6);

% Plotting
figure;
subplot(2,1,1);
plot(t, x);
title('Original Sinewave Signal');
xlabel('Time (s)');
ylabel('Amplitude');

subplot(2,1,2);
stairs(t, x_q);
title('Quantized Sinewave Signal (6-bit PCM)');
xlabel('Time (s)');
ylabel('Amplitude');

disp('PCM Encoded Binary Data:');
disp(x_encoded);
```

Output:





Conclusion:

The given MATLAB code performs sampling, quantization, and encoding of a sinewave signal using a manual implementation of 6-bit PCM (Pulse Code Modulation). First, a 3 Hz sinewave with an amplitude of 15 is sampled at 15 Hz for a duration of 1 second. The signal is then quantized into 64 discrete levels (2^6 = 64) by computing partition boundaries and assigning each sample to its nearest quantization level using the discretize function. The quantized values are then encoded into binary using the dec2bin function, converting each quantized value's index into a 6-bit binary representation. The original sinewave is plotted alongside its quantized version using the stairs function, which visually represents the step-like distortion introduced by quantization. Finally, the encoded PCM binary data is displayed in the console. This implementation demonstrates PCM encoding in MATLAB without relying on built-in Communications Toolbox functions, highlighting the fundamental process of quantization and digital signal representation.