Cairo University

Faculty of Engineering

Credit Hours System

EECE306

Communications-1

Fall 2023

Project 2 Report

PCM Modulation

<u>Presented to:</u>

Dr. Nora Ahmed

Eng. Hussein Ahmed

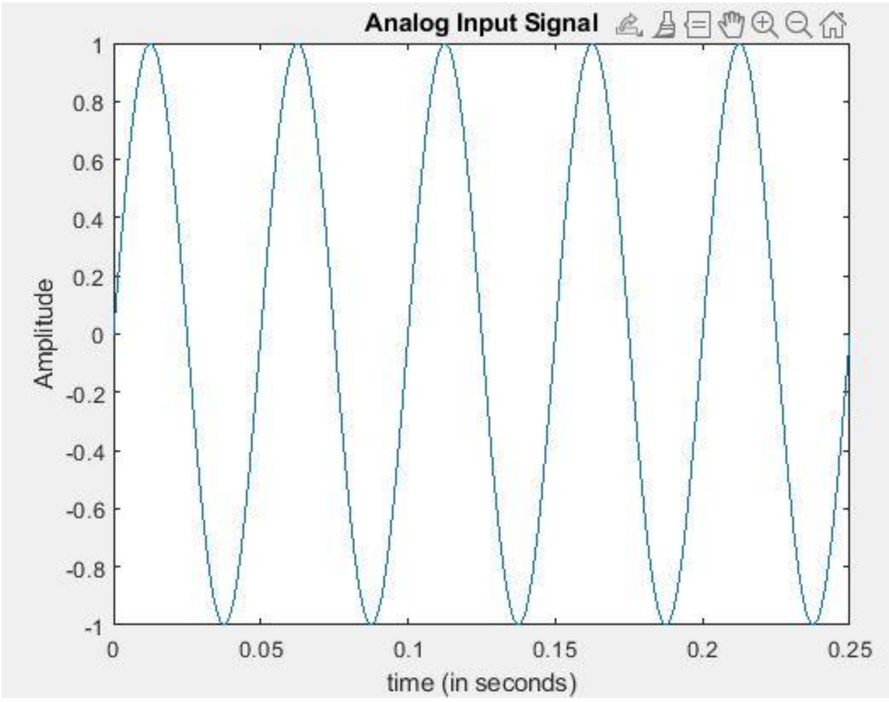
<u>Submitted by:</u>

Alhussein Gamal Hussein Ali 1200399

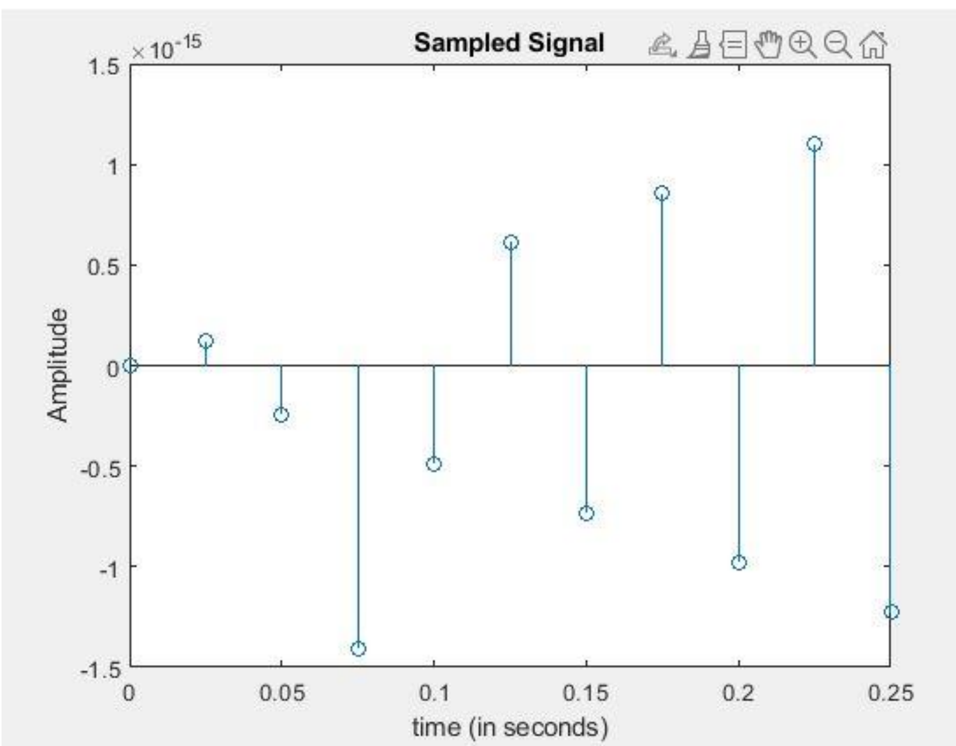
Mohamed Gamal Hussein Ali 1200435

Requirement 1

Input sinusoidal signal with a sine reference and a 1V amplitude.



Sampled Signal

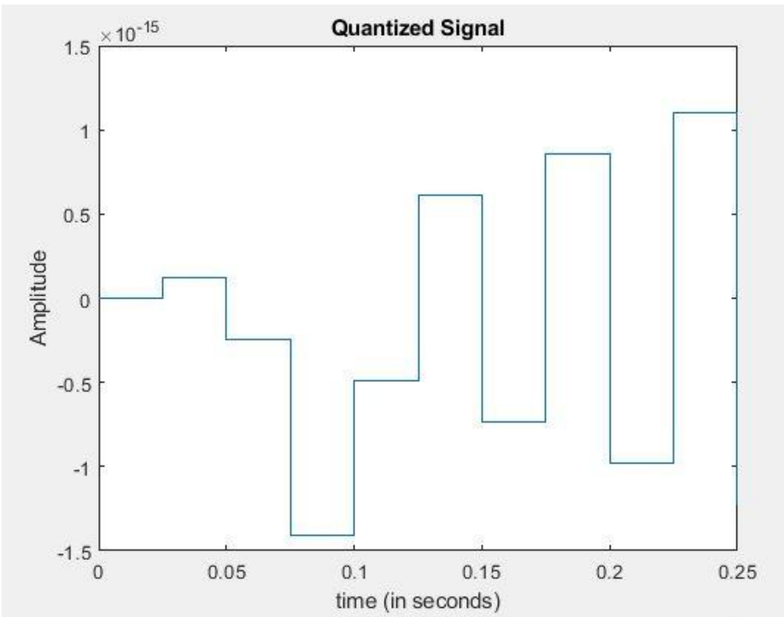


Comment:

Using the Nyquist rate, we should get samples after each half of the sampling period.

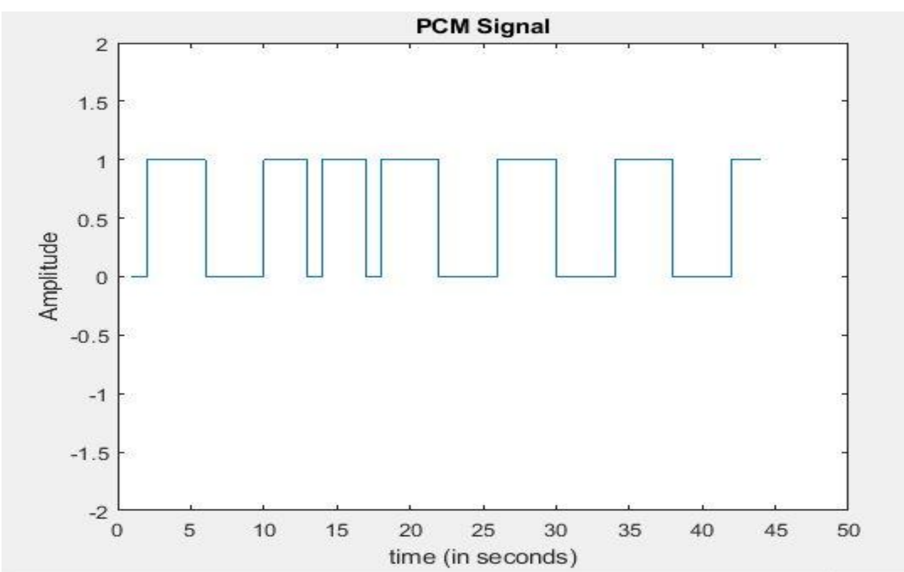
In an ideal situation, the time after each half of a period is 0 using m(t) and the reference carrier, but in MATLAB we had to discretize the signal, so it results in small error. Ideally this is zero.

Quantized Signal



PCM Signal

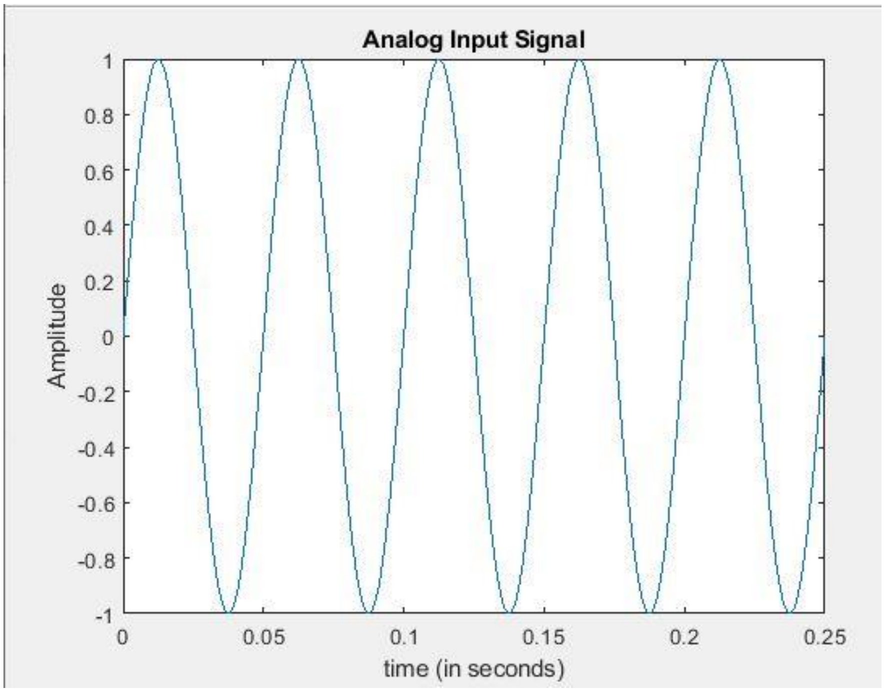
An uniform Mid-Tread quantizer is used with a step size = 2 * 5 / 24 = 0.625



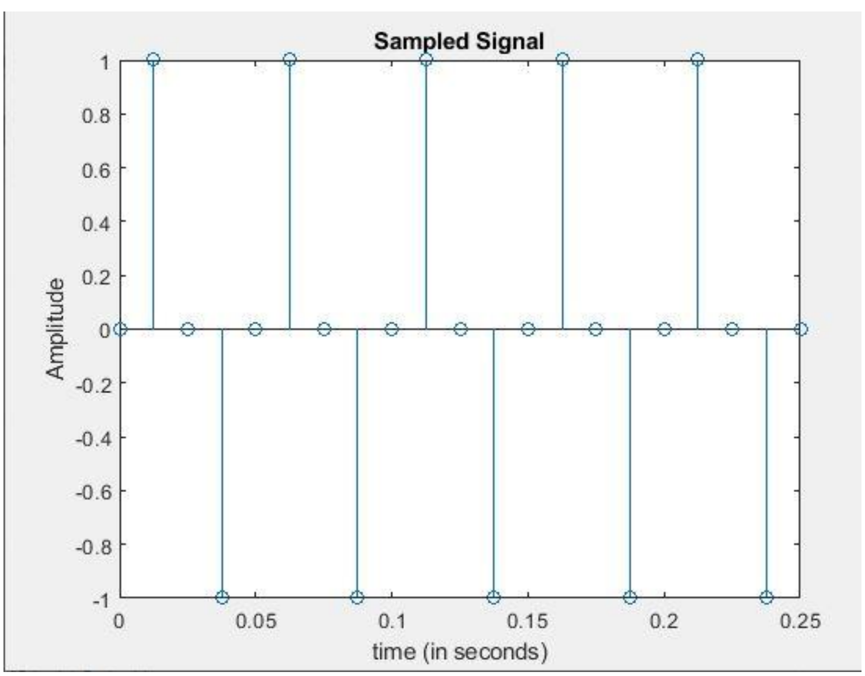
Requirement 2

Repeat requirement 1 except that the sampling rate is double the Nyquist rate.

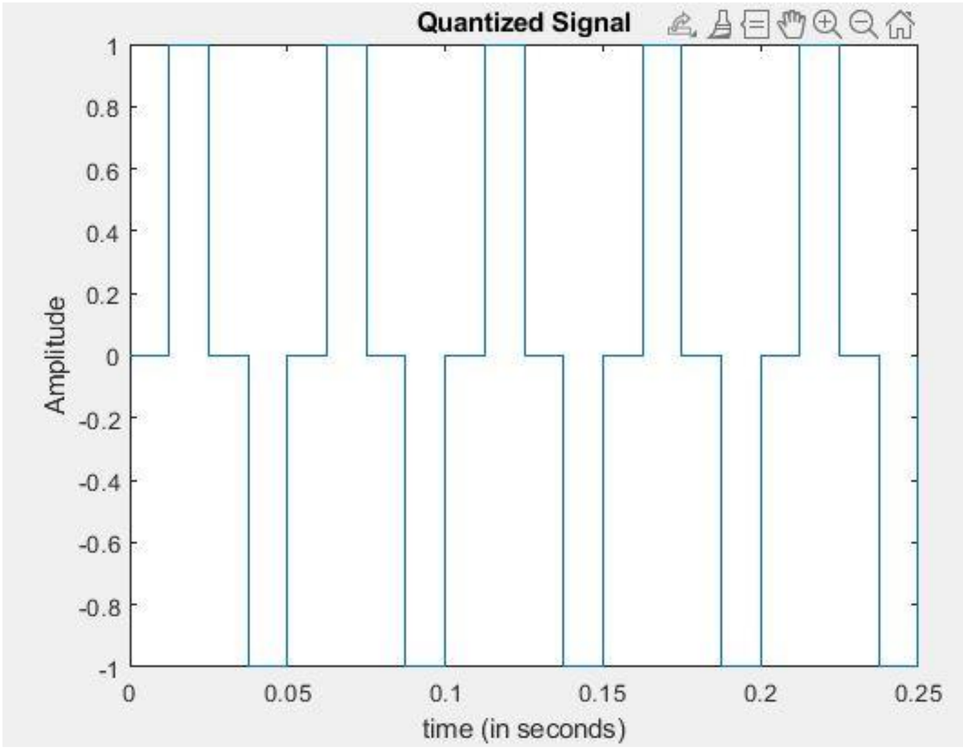
Input sinusoidal signal with a sine reference and a 1V amplitude.



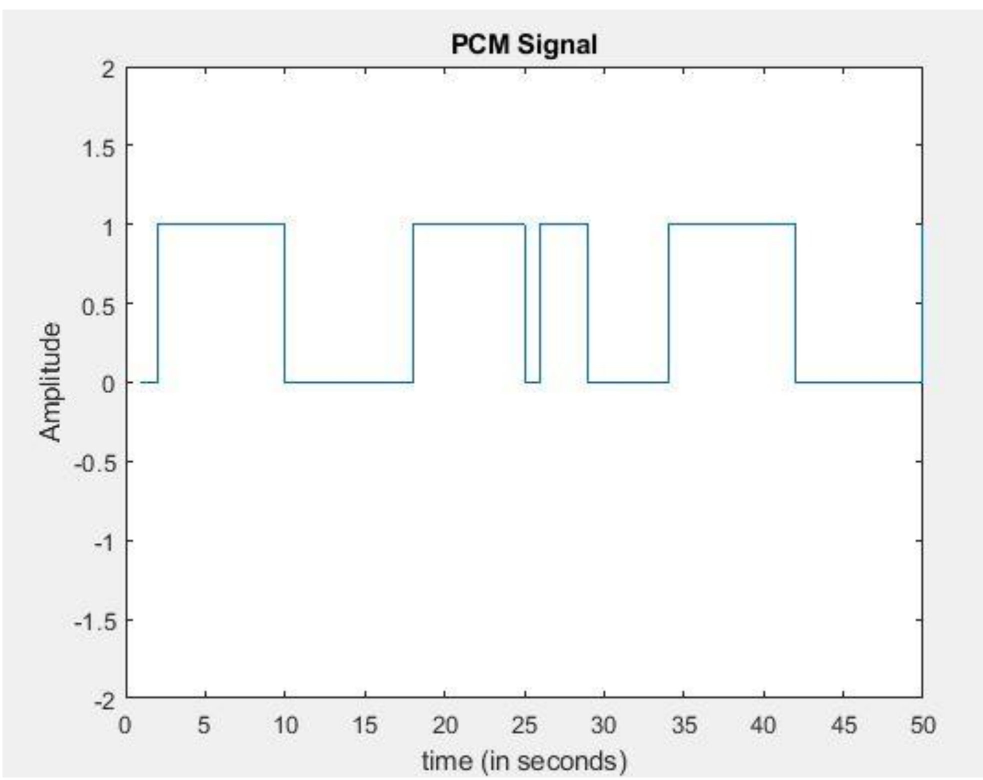
Sampled Signal



Quantized Signal



PCM Signal



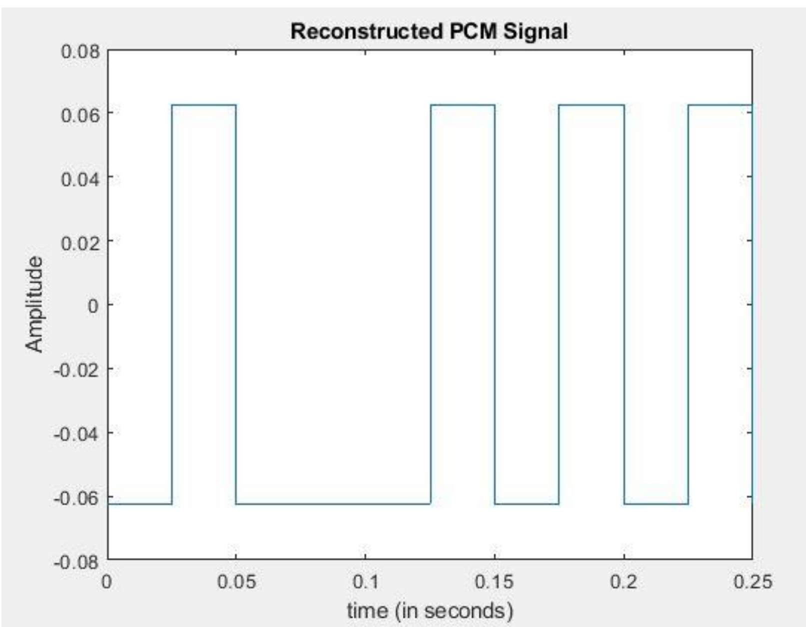
Observing the results, we deduce the following conclusions:

1. The analytical Bit Rate, fs, is (80 Hz) * N (4 bits) = 320 bits.sec^{-1}.
2. By increasing the sampling rate to double the Nyquist rate, the bit rate also increased, which allowed for higher speed in transmission. But it also had a disadvantage, which is using a larger bandwidth, which might be inefficiently used.
3. Using a sine as a reference for the sampling process introduces the inefficiency that, since a sample is taken at every ts/2, we include in our samples every intersection with the x-axis.
4. The higher sampling rate means a greater number of samples per T and thus we maintain the quantized signal closer to the original signal, which means better retrieval of the signal.

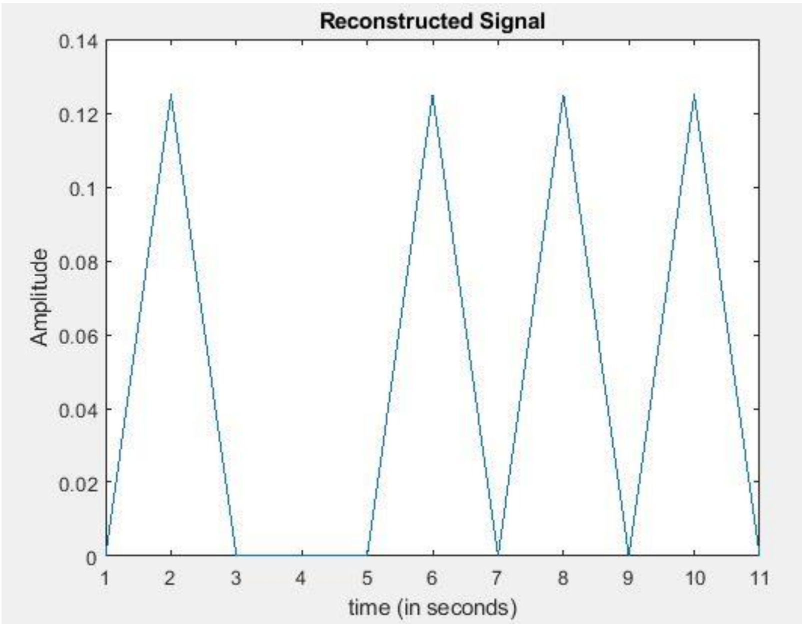
Requirement 3

Try to reconstruct the original waveform from the PCM signal for the above two cases, then comment on your results and justify your answers.

Reconstructed PCM signal of the signal sampled with the Nyquist rate:

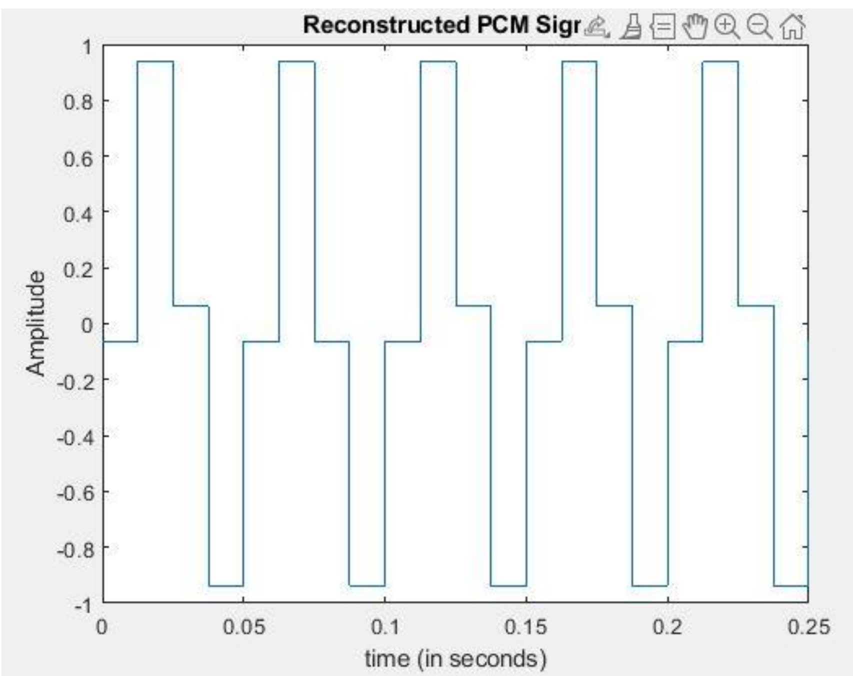


Reconstructed signal of the signal sampled with the Nyquist rate:

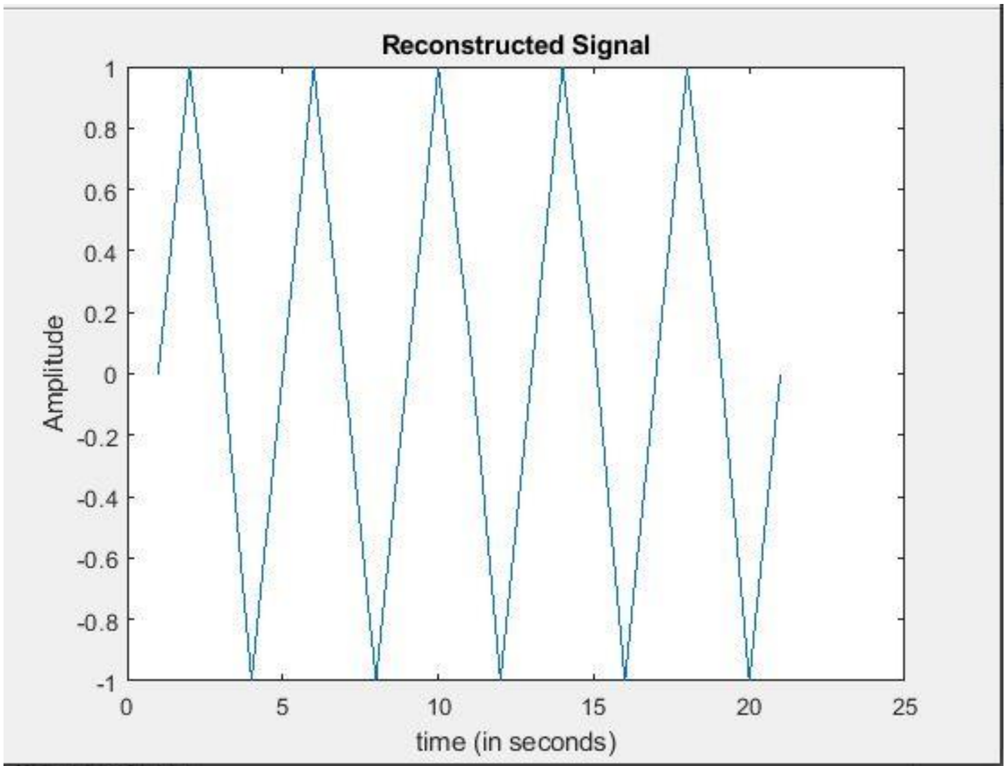


Note the severly attenuated amplitude. It can be seen as all at zero level.

Reconstructed PCM signal of the signal sampled with double of the Nyquist rate:



Reconstructed signal of the signal sampled with double of the Nyquist rate:



Here the amplitude is not attenuated, and the signal is closer to the actual signal.

Conclusions:

1. Sampling using the sine function as a reference results in the complete loss of the signal at the receiving end or the inability to reconstruct the signal, because at transmission, every zero crossing has been sampled.
2. Reconstructing a signal with double the Nyquist rate is of a much higher quality than of a signal with just the Nyquist rate, as we take double the amount of samples at double the Nyquist rate. Additionally lower amplitude attenuation is observed.

MATLAB CODE

```
clc;
clear all;

% input signal
fm = 20;
Tm = 1/fm;
Am = 1;

% quantiaztion data
L = 16;
n = 4;

% sampling data
fs = 2 * fm; % Nyquist Rate


BitRate = fs*n;

display(BitRate);

ti = (0:1/1000:5*Tm);
x = Am * sin(2*pi*fm*ti);

figure(1);
% subplot(3, 1, 1);
plot(ti, x);
xlabel('time (in seconds)');
ylabel('Amplitude');
title('Analog Input Signal');

% Sampling
t = 0:1/fs:5*Tm;
y = Am * sin(2*pi*fm*t);
figure(2);
% subplot(3, 1, 2);
plot(ti, x);
stem(t, y);
xlabel('time (in seconds)');
ylabel('Amplitude');
title('Sampled Signal');

% Quantization
MAX = Am;
MIN = -MAX;
```

```
stepsize = (MAX - MIN)/L;

in = MIN:stepsize:MAX;
out = MIN-stepsize/2:stepsize:MAX+stepsize/2;
[index, z] = quantiz(y, in, out);

for i=1:length(z)
    if(z(1,i) == (MIN-stepsize/2))
        z(1,i) = MIN+stepsize/2;
    end
end
figure(3);
% subplot(3, 1, 2);
plot(t, z);
stairs(t, y);
xlabel('time (in seconds)');
ylabel('Amplitude');
title('Quantized Signal');

% Encoder
for i=1:length(index)
    if(index(i) ~= 0)
        index(i) = index(i)-1;
    end
end
encode = de2bi(index,'left-msb');
cnt = 1;
figure(4)
for i=1:length(index)
    for j=1:4
        binary_encode(cnt) = encode(i,j);
        cnt = cnt+1;
    end
end
stairs(binary_encode);
ylim([-2 2]);
xlim([0 50])
xlabel('time (in seconds)');
ylabel('Amplitude');
title('PCM Signal');

% Decoder
decode = reshape(binary_encode,n,length(binary_encode)/n);
decodeprime = bi2de(decode,'left-msb');
decode_quantized=stepsize*index+MIN+(stepsize/2);
```

```
figure(5);
stairs(t, decode_quantized);
xlabel('time (in seconds)');
ylabel('Amplitude');
title('Reconstructed PCM Signal');

for i=1:length(index)
    if(index(i) == 0)
        decodedsignal(i) = (stepsize * index(i)) - Am;
    else
        decodedsignal(i) = ((stepsize * index(i)) - Am) + stepsize;
    end
end

figure(6);
plot(decodedsignal);
xlabel('time (in seconds)');
ylabel('Amplitude');
title('Reconstructed Signal');
```