Bangladesh University of Engineering and Technology

Course Number: EEE 312
Course Name: Digital Signal Processing I Sessional

Experiment No: 1
Name of The Experiment: Study of Sampling, Quantization and Encoding

Name: MD Atiq Shahriar
ID: 1706090
Department: EEE
Section: B1

Date of Performance: 9/3/2021
Date of Submission: 14/3/2021

Part A

Task: Study of sampling, quantization and encoding of the following signal.

$$y = \sin 2\pi 10t + \sin 2\pi 50t + \sin 2\pi 100t$$

Matlab Code

```matlab
clc;
clear all;
close all;

f=@(t) sin(2*pi*10*t)+sin(2*pi*50*t)+sin(2*pi*100*t);

t=0:.0001:.2;
original=f(t);%Original signal


Fs=[200 500];%sampling rate
bit=[2 4 6];

sqnr=zeros(length(Fs),length(bit));

for i=1:length(Fs)
    n=0:Fs(i)*t(end);
    y=f(n/Fs(i));%sampling


    for j=1:length(bit)

        %Quantization Part
        drange=max(y)-min(y);
        delta=drange/(2^bit(j)-1);
        y_quant_level=round((y-min(y))/delta);
        y_quant=round((y-min(y))/delta)*delta+min(y);%Quantization of signal
        level=min(y)+(0:(2^bit(j)-1))*delta;

        %Reconstruction Part
        y_interp=interp1(n/Fs(i),y_quant,t,'spline');

        err=y_quant-y;%Quantization Error
        sqnr(i,j)=(sum(y.^2)/(length(y)-1))/(sum(err.^2)/(length(err)-1));

        %Plotting Part
        figure
        subplot(4,1,1)
        plot(t,original);
        title('Original Signal');
        xlabel('Time');
        ylabel('Amplitude');

        subplot(4,1,2)
        stem(n/Fs(i),y,'filled');
        title(['Sampled signal at ',num2str(Fs(i)),' Hz']);
```

```
        xlabel('Time');
        ylabel('Amplitude');

        subplot(4,1,3)
        stairs(n/Fs(i),y_quant)
        title(['Quantization at ',num2str(bit(j)),' bits']);
        xlabel('Time');
        ylabel('Quantized Signal');
        subplot(4,1,4)
        plot(t,original,t,y_interp)
        title('Reconstruction');
        xlabel('Time');
        ylabel('Amplitude');
        legend('Original','Reconstructed');

        %Encoding Part
        encoded_signal=dec2bin(y_quant_level)';
        encoded_signal=encoded_signal(:)';
        disp(['Encoded signal at ',num2str(Fs(i)),' Hz and ',
    num2str(bit(j)),' bit:'])
        disp(encoded_signal);
    end

end
disp('SQNR values in dB:');
disp(10*log10(sqnr));
```

Output

The original signal is sampled at two different rates (200 Hz and 500 Hz) and quantized using three different bit values (2 bit, 4 bit and 6 bit).Outputs for different combination of sampling rate and bit values are shown below:

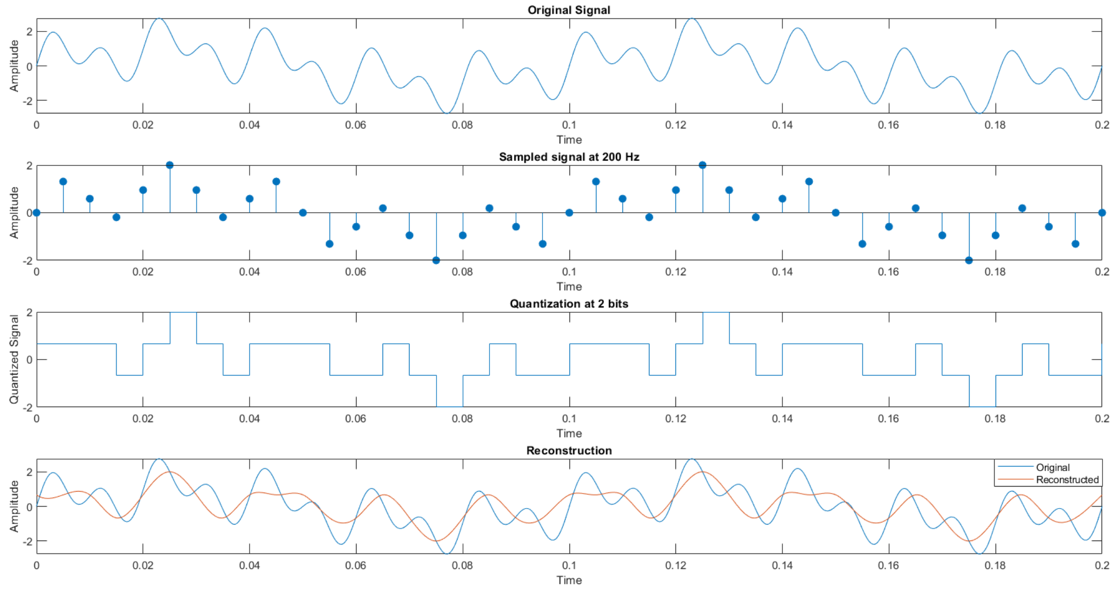

Figure 1: 2 Bit Quantization at 200 Hz

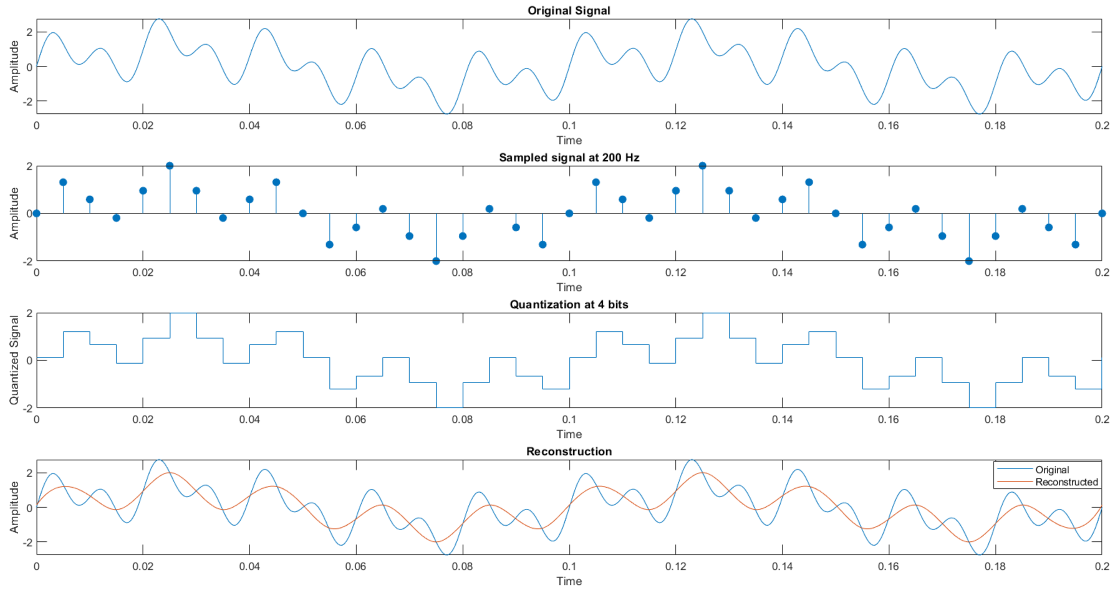

Figure 2: 4 Bit Quantization at 200 Hz

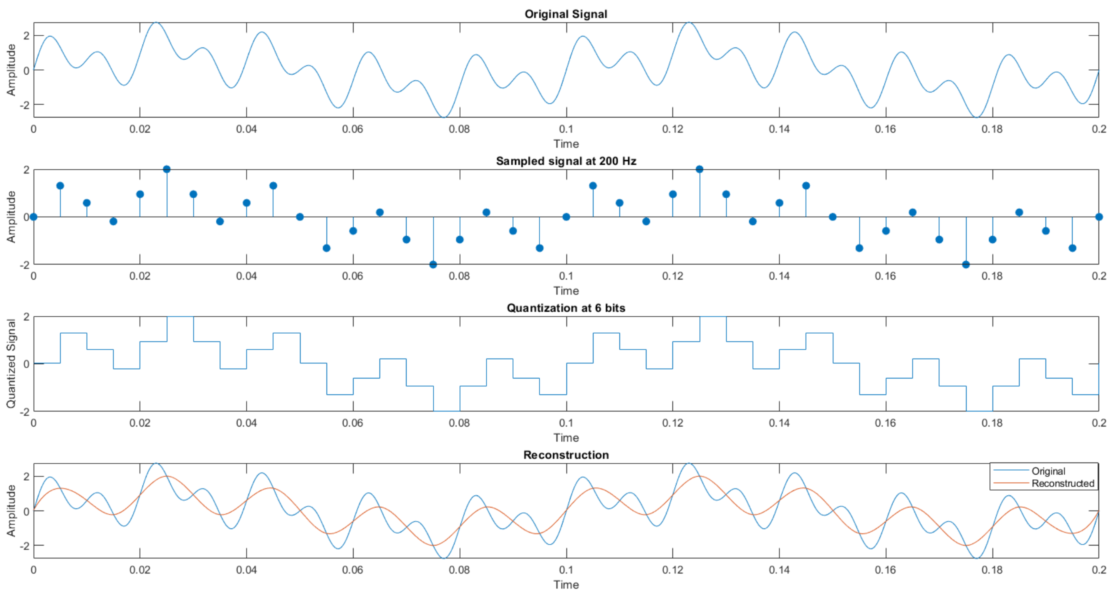

Figure 3: 6 Bit Quantization at 200 Hz

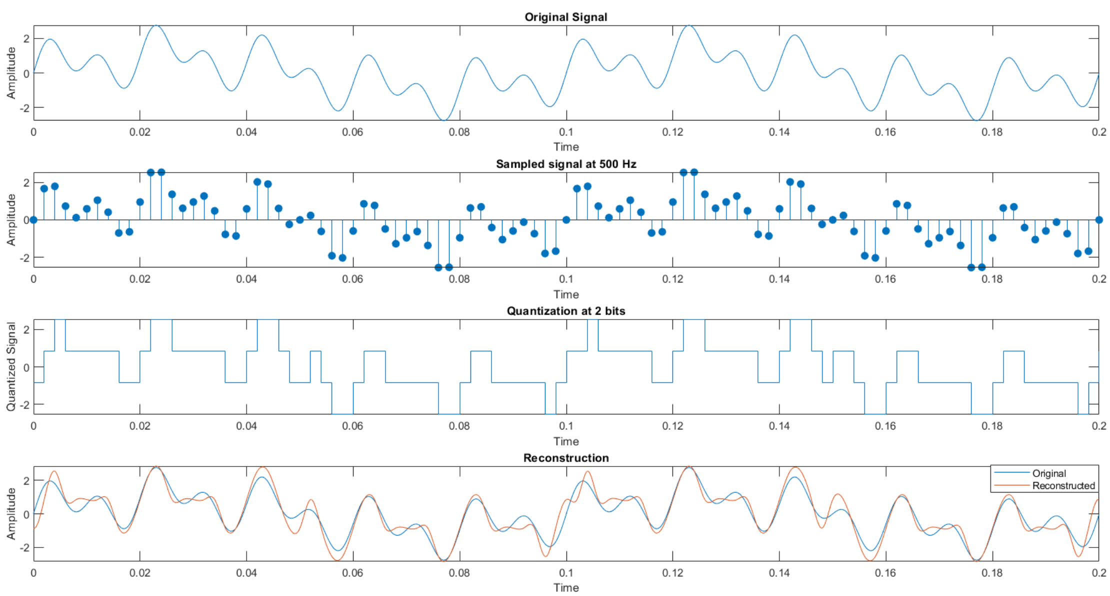

Figure 4: 2 Bit Quantization at 500 Hz

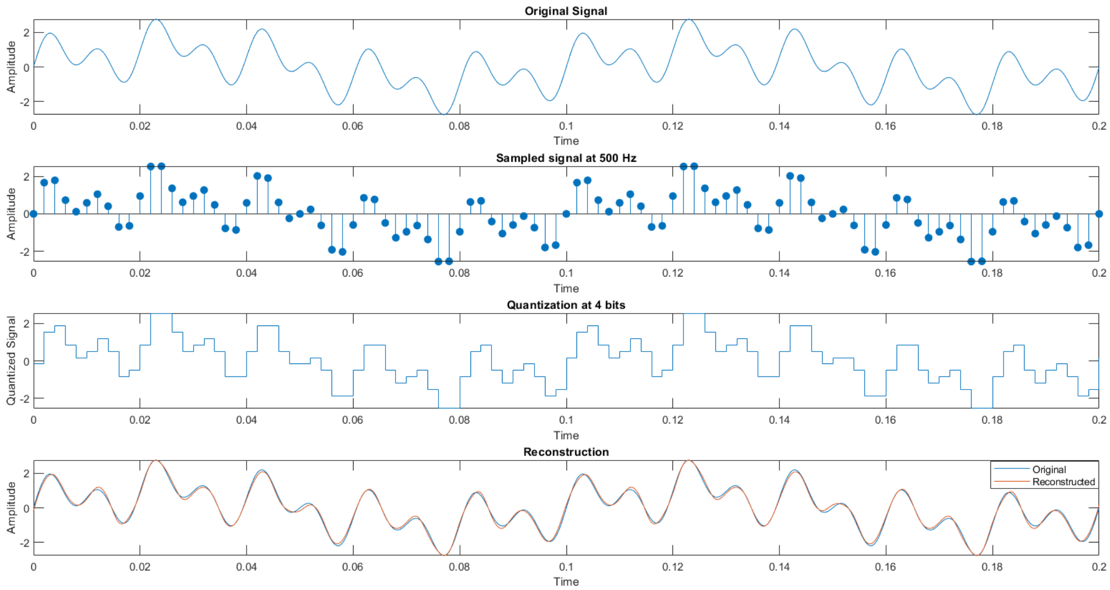

Figure 5: 4 Bit Quantization at 500 Hz

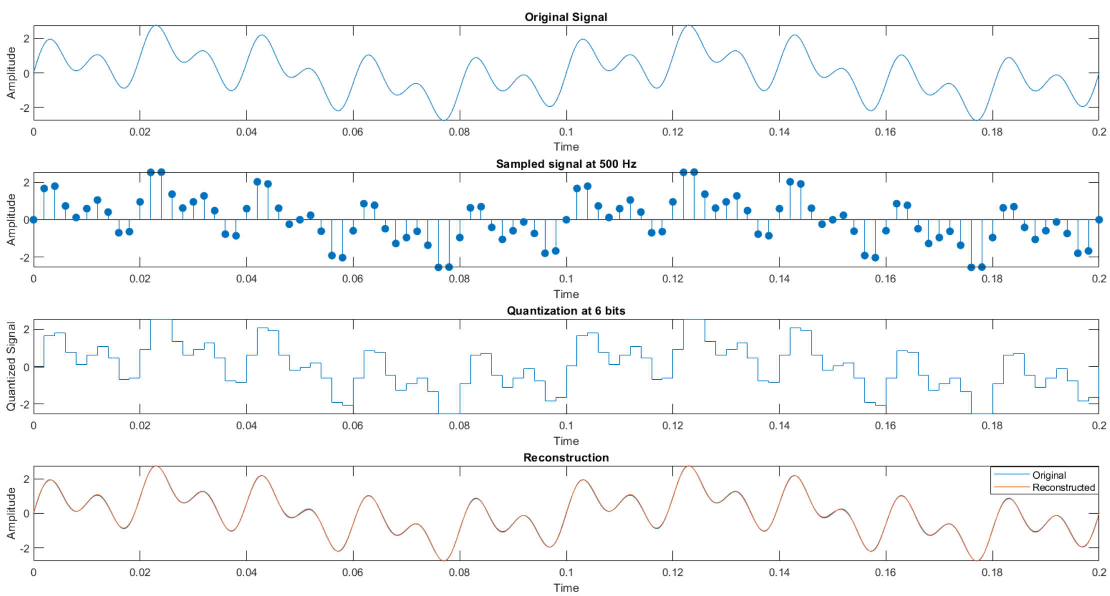

Figure 6: 6 Bit Quantization at 500 Hz

Analysis: When the signal is sampled at lower values the reconstructed signal doesn't quite follow the original signal. Even using higher bit value for quantization couldn't make the reconstructed signal any better.But when the original signal is sampled at higher rate the reconstructed signal is very close to the original one even in low bit quantization. So we can clearly note that sampling rate plays the key role in reconstructing the signal. Quantization level doesn't seem to have a great effect on signal reconstruction.

SQNR:

SQNR(in dB) for different combination of sampling rate and quantization bits are listed as follows:

Sampling Rate	2 Bits	4 Bits	6 Bits
200 Hz	6.9734	21.8236	32.4822
500 Hz	8.8872	21.3414	33.7825

From the table it's clearly noticeable that SQNR increases significantly with increasing number of bits. Whereas the sampling rate has almost zero effect on SQNR. This fact is quite reasonable as increasing bits significantly reduces the quantization error and hence the power of error becomes small. With increasing sampling rate SQNR increases too but very insignificantly. Sampling rate plays key role in reconstructing the quantized signal as it was seen in earlier analysis.

Encoding:

After quantization every sample element of the signal falls in a certain quantized level.So each sample can be uinquely identified by the level number it falls in. While encoding the signal level number of each sample is converted into binary and placed in a row to make a complete representation of the signal. So knowing the sampling rate,bit value and which quantization level correspond to what value is sufficient to decode the encoded signal and reconstruct the original continuous signal. Encoded signal for different values of sampling rate and quantization bit is as follows:

2 Bit Quantization at 200 Hz:

1010100110111001101010010110010001100101101010011011100110101001011001000110010110

2 Bit Quantization at 500 Hz:

01101110101010100101101111101010101001011011111001011001000001101001010101010000011010010101010100011010111010101010010110111110101010100101101111100110100100000110100101010101000001101001010101010100110

4 Bit Quantization at 200 Hz:

```
1000110010100111101111111011011110101100100000110101100001000000000100
1000010100111000110010100111101111111011011110101100100000110101011000
0100000000100100001010011100
```

4 Bit Quantization at 500 Hz:

```
0111110011011010100010011011100101010110101011111111110010011010101011
1001010101011001110111011001011101111000011000100010011010101010100110
0100010101100011000000000101100110100110010001100111010100100011100
11001101101010001001101110010101011010101111111111001001101010111001
0101010110011101110110010111100010000110001000100110101010100110010
01010110001100000000010110011010011001000110011101010010001110000
```

6 Bit Quantization at 200 Hz:

```
1000001101001010010111001011101111111011100111001010011101001000000000
1011010110100011010001000000001000110001101011000101110000011010010100
0101110010111011111111011100111001010011101001000000001011010110100011
0100010000000100011000110101100010111000000
```

6 Bit Quantization at 500 Hz:

```
0111111101001101101010011000011001111011011001010101110110001010111
1111111111110000100111101011101111100101010110010101100111111001110
1111001110111010111111000100110000001000000110011000101010101001011010
0100000101000110000011110000000000000101001001111010000110100100100
1000011110010110001001001011100000110100110110101001100001100111101
0110010101011101100010101111111111111111000010011110101110111110010
0101100101011001111110011101111100111011101100000100010011000000100000
0110011000101010101001011010010000010100011000000111100000000000000101
0010011110100001101001001001100001111001011000100100101110000
```

Analysis: As it can be seen the total number of bits representing a signal depends both on the sampling rate and quantization bits. Increasing each of these two comprehensively increases the total bits. More memory is needed to store long binary signal. Hence more the sampling rate and quantization bits, device with more memory is needed to process the signal. This makes encoding with high sampling rate and high quantizatiion bits more expensive but it provides more accuracy in retrieving the original signal too.

Part B

Task: The following signal is to be sampled, compressed using μ-law and lastly uniformly quantized:

$$y = \sin(2\pi 10t) + \sin(2\pi 50t) + \sin(2\pi 100t)$$

For this task sampling rate and quantization bit are chosen to be 400 Hz and 6 bits respectively.

Matlab Code:

```
clc;
clear all;
close all;

f=@(t) sin(2*pi*10*t)+sin(2*pi*50*t)+sin(2*pi*100*t);

t=0:.0001:.2;
original=f(t);%Original signal


Fs=400;
bit=6;

n=0:Fs*t(end);
y=f(n/Fs);%Sampling

y_norm=(y-min(y))/(max(y)-min(y))*2-1;%Normalization between -1 to
    1
y_compressed=(log(1+255*abs(y_norm))/log(1+255)).*sign(y_norm);%
    Compression using u-law


%Quantization
drange=max(y_compressed)-min(y_compressed);
delta=drange/(2^bit-1);
y_com_quant=round((y_compressed-min(y_compressed))/delta)*delta+min
    (y_compressed);


%Plotting
figure
subplot(4,1,1)
plot(t,original);
title('Original Signal');
xlabel('Time');
ylabel('Amplitude');

subplot(4,1,2)
stem(n/Fs,y);
title(['Sampled signal at ',num2str(Fs),' Hz']);
xlabel('Time');
ylabel('Amplitude');

```

```
subplot(4,1,3)
stem(n/Fs,y_compressed)
title('Compression using u-law');
xlabel('Time');
ylabel('Compressed Signal');
subplot(4,1,4)
stairs(n/Fs,y_com_quant)
title('6 bits Quantization of Compressed Signal');
xlabel('Time');
ylabel('Quantized amplitude');
```

Output:

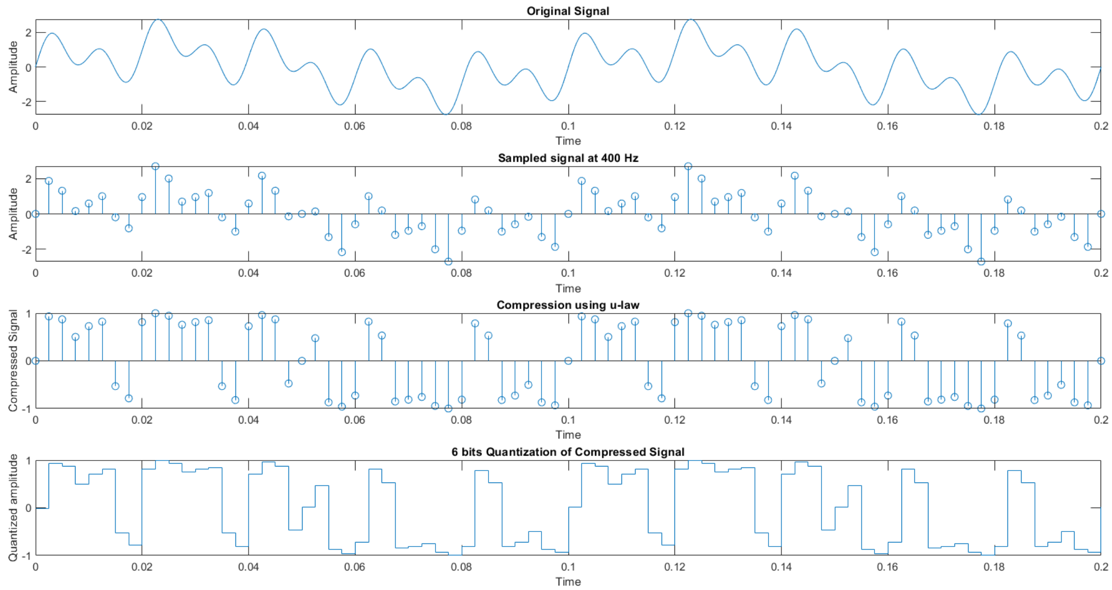

Figure 7: Compression and Quantization