180574K
Sauranga H.W.C.

EN2072 – Assignment 01

Through this assignment, we have attempted to demonstrate the process of converting a continuous time signal in to the discrete time domain successfully using the sampling theorem and quantization theories.

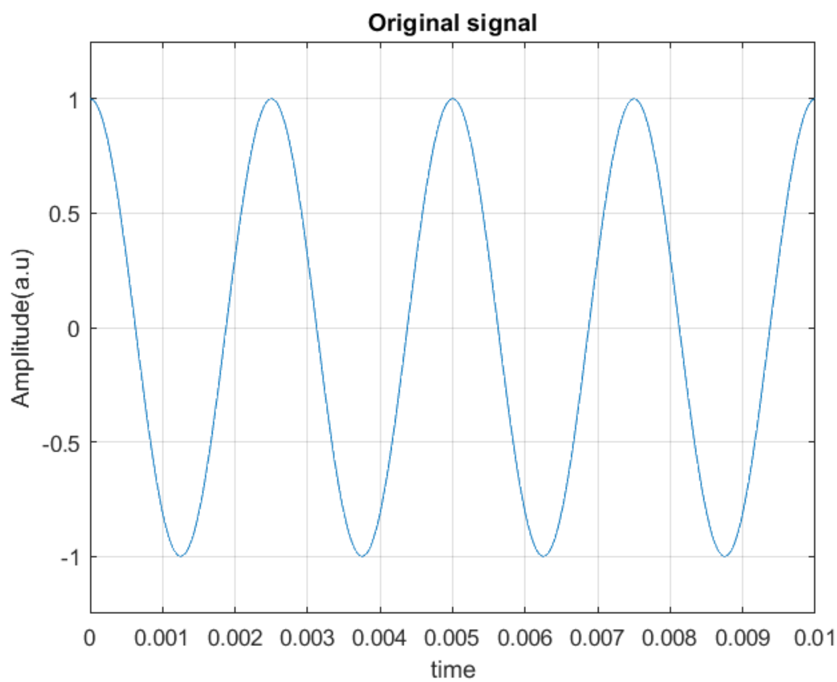
The original continuous time signal used in the assignment as the reference is

$$y(t) = A\cos(2\pi ft)$$

where,

$$A = 1 \text{ a.u}$$
$$f = 400 \text{ Hz}$$

This signal is plotted for a duration of 10 ms as follows.



Calculating the Nyquist sampling frequency

In accordance with the sampling theorem, Nyquist frequency (f_{nq}) is twice as the highest frequency in the signal. Since our original signal contains only one frequency component, we take twice of that value as our Nyquist frequency.

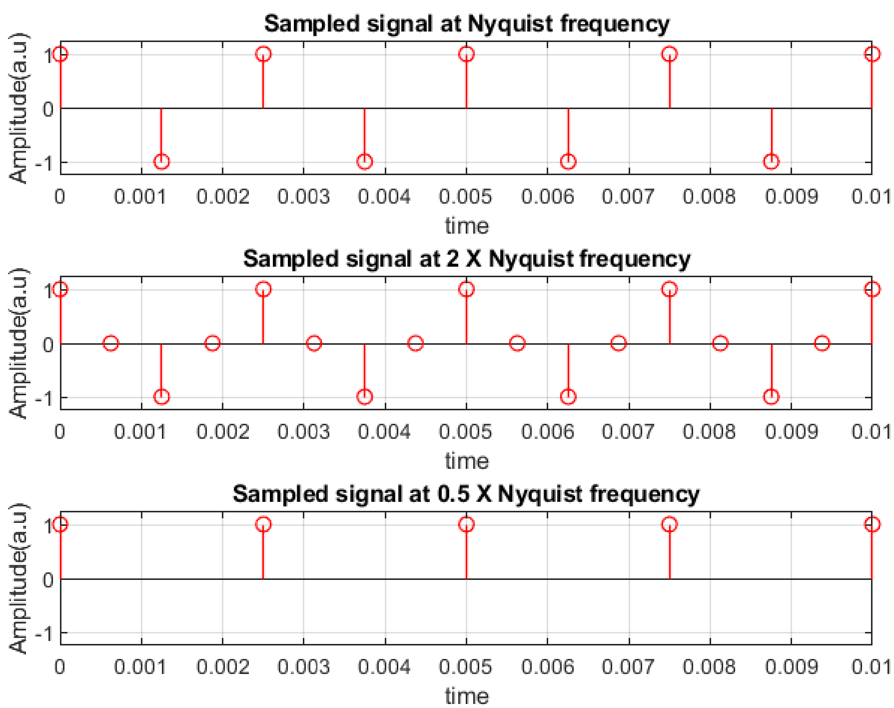
Highest frequency component = 400 Hz

Nyquist frequency (f_{nq}) $= 2 \times 400$ Hz $= 800$ Hz

Plotting the sampled signal

Under the assumption of ideal sampling, the original signal can be sampled at Nyquist frequency (f_{nq}) and plotted as follows.

Signals sampled at $2f_{nq}$ and $0.5f_{nq}$ are also shown alongside for comparison.



Quantization levels.

It is required to maintain the SNR level above 25 dB.

The quantization noise power can be given as,

$$\overline{q^2} = \int_{-\frac{\Delta v}{2}}^{\frac{\Delta v}{2}} q^2 . p(q)\, dq$$

where Δv is the size of a sub interval and $p(q)$ is the probability function of quantization noise (q).

Assuming equally likely quantization error, we can simplify the above expression.

$$\overline{q^2} = \frac{(\Delta v)^2}{12}$$

but $\Delta v = \frac{2m_p}{L}$ where $m_p(= 1)$ is the peak amplitude of the signal and L is the number of equally spaced quantization levels.

Therefore,

$$\overline{q^2} = \frac{1}{3L^2}$$

Now we can calculate the number of quantization levels required based on the SNR specified.

$$\text{SNR} = \frac{\text{Signal power}}{\text{Noise power}}$$

Signal power = $\frac{A^2}{2} = \frac{1}{2}$

$$\therefore SNR = \frac{\frac{1}{2}}{\frac{1}{3L^2}}$$

$$10^{2.5} = \frac{3L^2}{2}$$

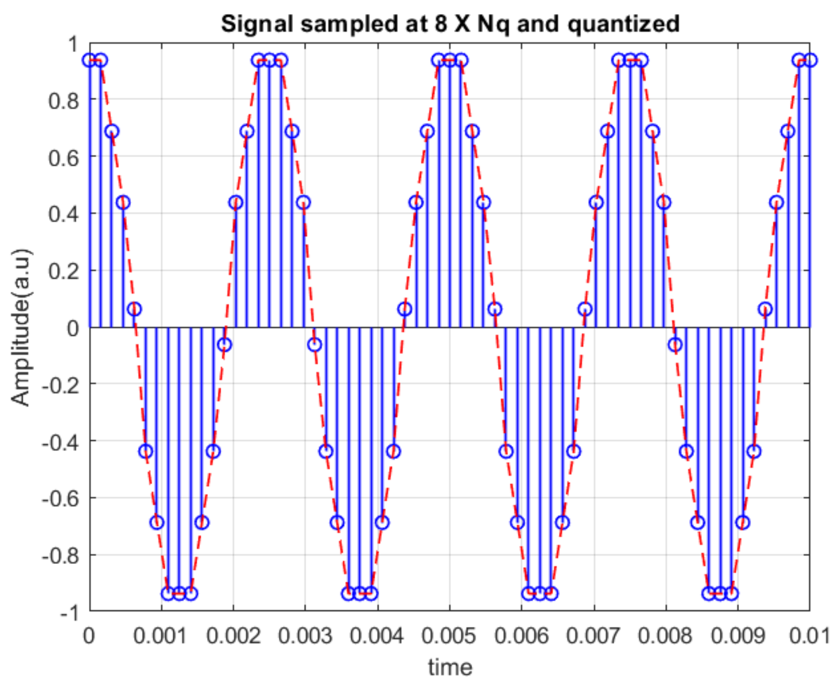
$$L = 14.52$$

no. of minimum quantization levels required **(L) = 15**

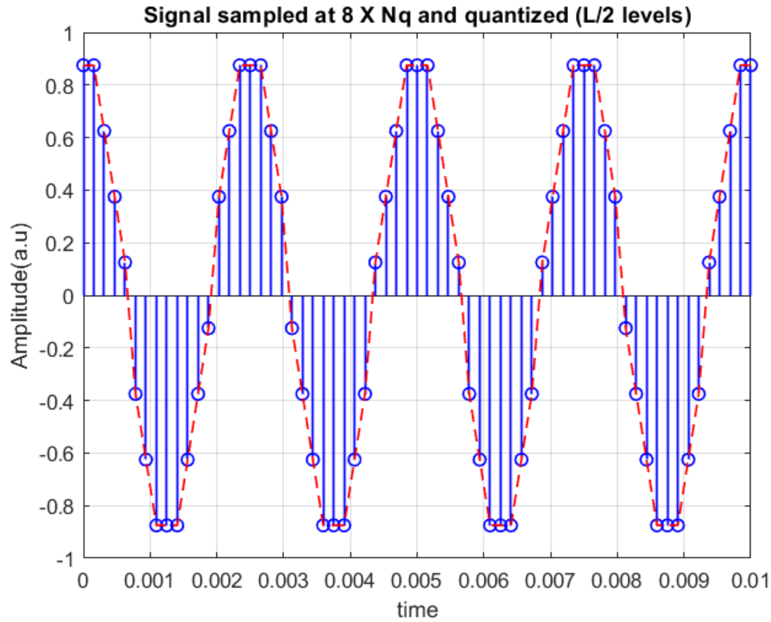
no. of bits per sample $\mathbf{(n_0)}$ **= 4**

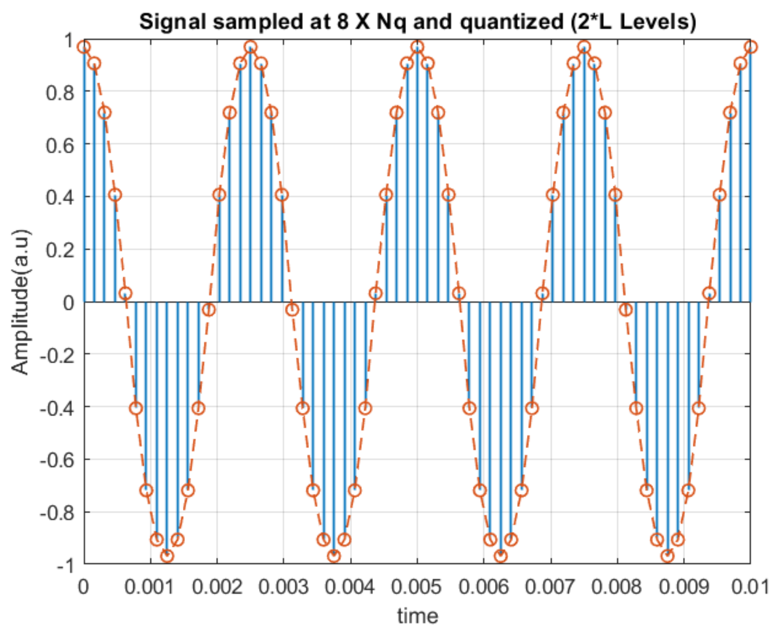
Sampling the signal at $8f_{nq}$ and quantizing

The signal is sampled at $8f_{nq}$ frequency and quantized according to the calculation shown above.



Now the same sampled signal above is quantized with L*2 levels and L/2 levels





Observations

The three graphs with varying number of quantization levels can be used to analyze the significance offered by them.

As the number of quantization levels increases, the resolution of the graph increases. This is clearly evident when we compare the graph with $L/2$ levels with the graph with $2 \times L$ levels. The graph with $2 \times L$ quantization levels bear a much more similar resemblance to the original continuous time signal that was used in the sampling and reconstruction process.

Appendix

The Matlab code used for the assignment is as follows.

```
clc
close all
```

Original Signal plot

```
%Specifications given
f = 400;
A = 1;
t = 0:0.00001:0.01;
%Defining the orginal signal
y_t = A*cos(2*pi*f*t);
%plotting the original signal
plot(t,y_t)
title("Original signal")
xlabel("time")
ylabel("Amplitude(a.u)")
axis([0 0.01 -1.25 1.25])
grid on
```

Calculating the Nyquist sampling frequency

```
f_s = 2*f;

%time vector for the sampled signal
ts1 = 0:(1/f_s):0.01;
y_ts1 = A*cos(2*pi*f*ts1);
subplot(3,1,1)
plot(ts1,y_ts1,'r','LineWidth',0.75)
title("Sampled signal at Nyquist frequency")
xlabel("time")
ylabel("Amplitude(a.u)")
axis([0 0.01 -1.25 1.25])
grid on
%time vector for the sampled signal (2nq)
ts2 = 0:(1/(2*f_s)):0.01;
y_ts2 = A*cos(2*pi*f*ts2);
subplot(3,1,2)
plot(ts2,y_ts2,'r','LineWidth',0.75)
title("Sampled signal at 2 X Nyquist frequency")
xlabel("time")
ylabel("Amplitude(a.u)")
axis([0 0.01 -1.25 1.25])
grid on
%time vector for the sampled signal (0.5nq)
ts_half = 0:(1/(0.5*f_s)):0.01;
y_ts_half = A*cos(2*pi*f*ts_half);
subplot(3,1,3)
plot(ts_half,y_ts_half,'r','LineWidth',0.75)
```

```
title("Sampled signal at 0.5 X Nyquist frequency")
xlabel("time")
ylabel("Amplitude(a.u)")
axis([0 0.01 -1.25 1.25])
grid on
```

Quantizing the signal sampled at 8 X Nyquist frequency (L levels)

```
%sampling the signal at 8nq
ts8 = 0:(1/(8*f_s)):0.01;
y_ts8 = A*cos(2*pi*f*ts8);
%Defining the quantization levels and bits
bits = 4;
qlevels = 2^bits;
%scaling the signal
scalingFactor = (A-(-A))/qlevels;
y_ts8_q = y_ts8/scalingFactor;
%rounding to the midpoints
roundTargets = [-7.5:1.0:7.5];
y_ts8_q = interp1(roundTargets,roundTargets,y_ts8_q,'nearest','extrap');
%rescaling to the original amplitude
y_ts8_q = (y_ts8_q)*scalingFactor;
%plotting the quantized signal
figure;
stem(ts8,y_ts8_q,'b','LineWidth',1)
hold on
plot(ts8,y_ts8_q,'r--','LineWidth',1)
xlabel ('time')
ylabel('Amplitude(a.u)')
title('Signal sampled at 8 X Nq and quantized')
grid on
```

Quantizing the signal sampled at 8 X Nyquist frequency (L/2 levels)

```
%Defining the quantization levels and bits
bits = 3;
qlevels = 2^bits;
%scaling the signal
scalingFactor = (A-(-A))/qlevels;
y_ts8_q = y_ts8/scalingFactor;
%rounding to the midpoints
roundTargets = [-3.5:1.0:3.5];
y_ts8_q = interp1(roundTargets,roundTargets,y_ts8_q,'nearest','extrap');
%rescaling to the original amplitude
y_ts8_q = (y_ts8_q)*scalingFactor;
%plotting the quantized signal
figure;
stem(ts8,y_ts8_q,'b','LineWidth',1)
hold on
plot(ts8,y_ts8_q,'r--','LineWidth',1)
xlabel ('time')
```

```
ylabel('Amplitude(a.u)')
title('Signal sampled at 8 X Nq and quantized (L/2 levels)')
grid on
```

Quantizing the signal sampled at 8 X Nyquist frequency (L*2 levels)

```
%Defining the quantization levels and bits
bits = 5;
qlevels = 2^bits;
%scaling the signal
scalingFactor = (A-(-A))/qlevels;
y_ts8_q = y_ts8/scalingFactor;
%rounding to the midpoints
roundTargets = [-15.5:1.0:15.5];
y_ts8_q = interp1(roundTargets,roundTargets,y_ts8_q,'nearest','extrap');
%rescaling to the original amplitude
y_ts8_q = (y_ts8_q)*scalingFactor;
%plotting the quantized signal
figure;
stem(ts8,y_ts8_q,'o','LineWidth',1)
hold on
plot(ts8,y_ts8_q,'o--','LineWidth',1)
xlabel ('time')
ylabel('Amplitude(a.u)')
title('Signal sampled at 8 X Nq and quantized (2*L Levels)')
grid on
```