Department of Electronic and Telecommunication Engineering

University of Moratuwa, Sri Lanka

EN2073 - Analog and Digital Communications

Assignment 01

Submitted by

Thalagala B.P. 180631J

Submitted on

May 3, 2021

** `MATLAB` Code used to generate the plots are attached at the end.*

Q1: Plot of the signal $y(t) = A.cos(2.\pi.f.t)$

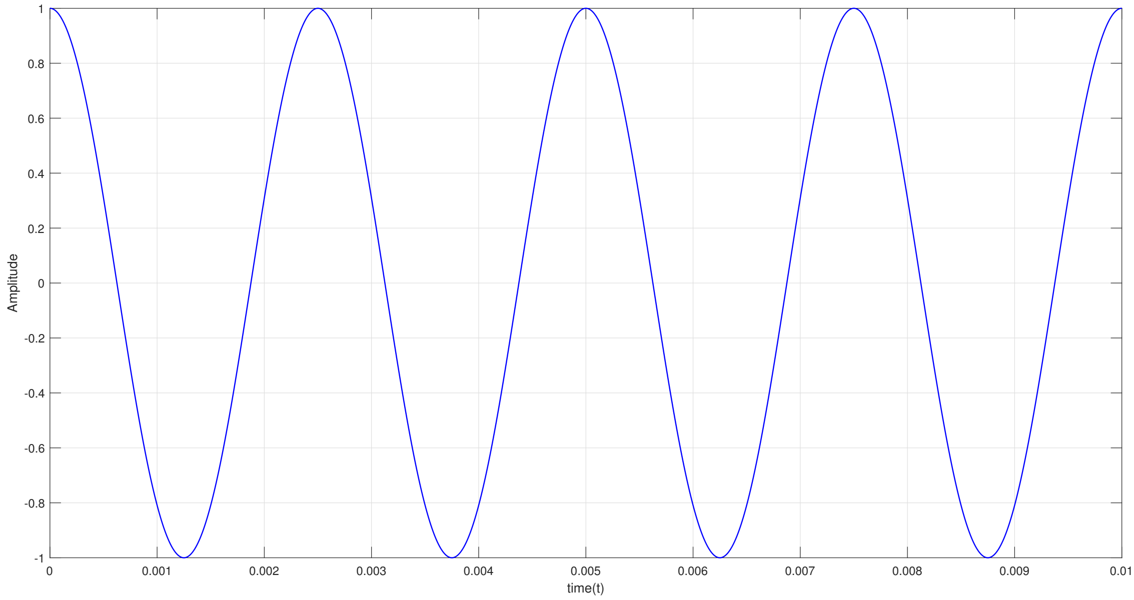


Figure 1: Plot of the signal $y(t) = A.cos(2.\pi.f.t)$, Where $A = 1$ and $f = 400\ Hz$

Q2: Nyquist sampling frequency (f_{nq})

A function containing no frequency higher than f Hz, is completely determined by sampling at $2f$ Hz. This frequency is called Nyquist rate or Nyquist sampling frequency.

$$\therefore Nyquist\ sampling\ frequency\ of\ above\ signal = 2.f = 2 \times 400 = 800\ Hz$$

Q3: Sampled signal at Nyquist sampling frequency (f_{nq})

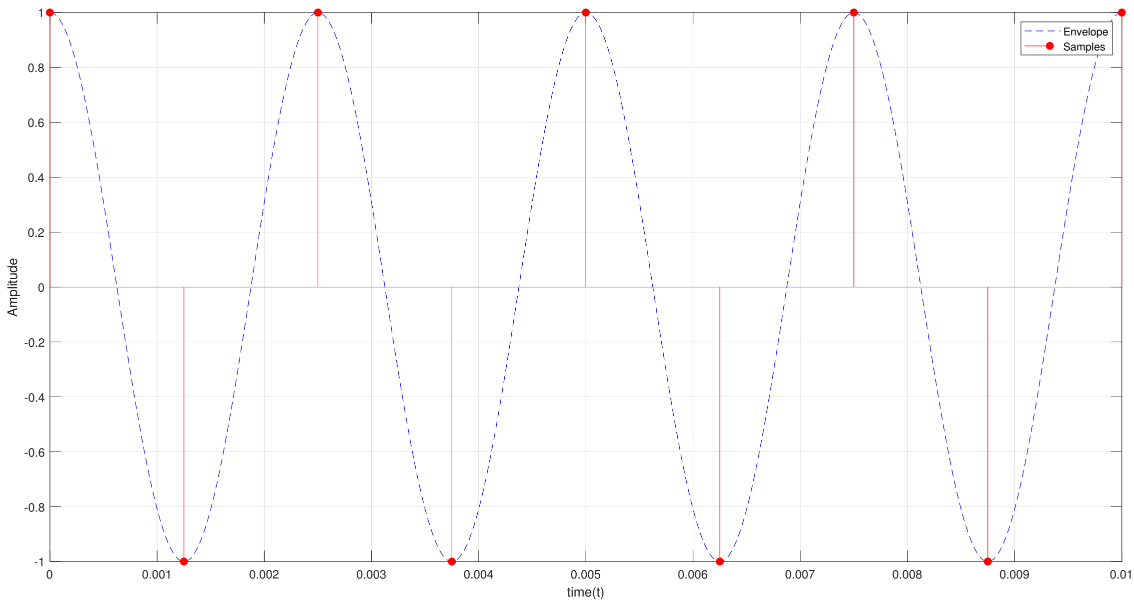


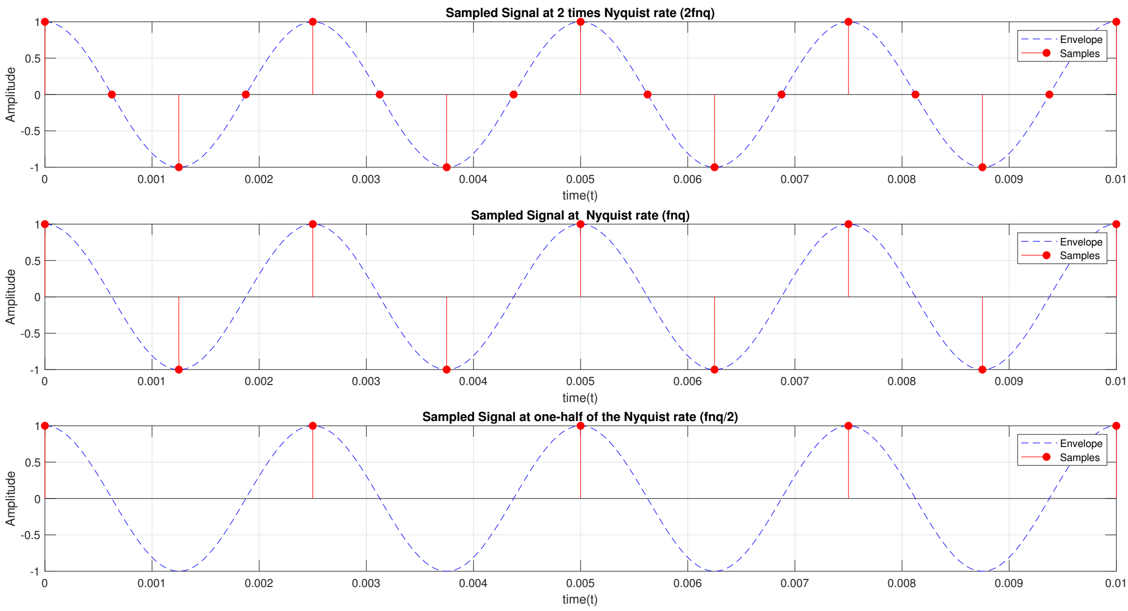
Figure 2: Sampled signal at Nyquist sampling frequency $(f_{nq} = 800\ Hz)$

Q4: Sampled signal at $(2f_{nq})$, (f_{nq}) and $(f_{nq}/2)$

For the given signal there is only one frequency component and therefore, there is no point in sampling it in the rate, higher than the Nyquist rate. Because of there are no other higher frequency components than 400 Hz, signals sampled at both $2f_{nq} = 1600\ Hz$ and $f_{nq} = 800\ Hz$ illustrated in top and middle subplots respectively, will be eventually reconstructed in to the same signal as there is ***no contribution from the samples with zero amplitudes*** at the reconstruction stage.

Therefore, it will be good enough if we sample at the *Nyquist frequency*$(f_{nq} = 800\ Hz)$ as illustrated in the middle subplot and it is possible to reconstruct the original signal without loosing the important information.

Consider the signal sampled at the *half of the Nyquist frequency*$(f_{nq}/2 = 400\ Hz)$, that is the sampling rate is equal to the highest frequency of the signal and bottom subplot represents the effect of this. Because there is no change in the amplitudes of the samples it will be seen as a straight line or more accurately a sinusoidal with a lower frequency when we reconstruct the signal. Which indicates signal carries no useful information if we sample it in this frequency as we have lost all the higher frequency components and they are now seen as lower frequency components. This phenomenon is known as the "***Aliasing or Spectral folding***".



Q5: Minimum number of bits (n_b) required per a sample and number of minimum quantization levels (L) to have a SN_qR ratio greater than 25dB

SN_qR in dB is defined as follows, where S_0 and N_q represents the received signal power and the strength of noise power due to quantization.

$$SN_qR = 10\log_{10}\left(\frac{S_0}{N_q}\right)$$

Size of quantization interval can be written as follows *for uniform quantization* where L is the number of quantization levels and m_p is the maximum amplitude of the signal. In our case which equals to 1.

$$Size\ of\ quantization\ interval\ (\Delta V) = \frac{2.m_p}{L}$$

Then N_q and S_0 can be written as,

$$S_0 = \frac{m_p^2}{2} \qquad \begin{aligned} N_q &= \frac{\Delta V^2}{12} \\ &= \frac{(2.m_p/L)^2}{12} \\ &= \frac{m_p^2}{3L^2} \end{aligned} \qquad \therefore SN_qR = 10\log_{10}\left(\frac{3L^2}{2}\right)$$

According to the requirements following calculations can be done.

$$\begin{aligned} SN_qR &= 10\log_{10}\left(\frac{3L^2}{2}\right) \\ 25 &< 10\log_{10}\left(\frac{3L^2}{2}\right) \\ 10^{2.5} &< \frac{3L^2}{2} \\ \frac{10^{2.5}\times 2}{3} &< L^2 \\ 14.519 &< L \end{aligned} \qquad \therefore L_{min} = 16\ (\because L\ must\ be\ a\ power\ of\ 2)$$

Therefore number of minimum quantization levels (L) = 16 and minimum number of bits (n_b) required per a sample is $\log_2(L) = \log_2(16) = 4$.

Q6: Quantization of Sampled output values

Following `MATLAB` function implements a method to quantize a given sampled signal. Its functionality can be briefly described as follows.

1. First the three special case of samples are considered.
   - If the sample's amplitude equals to the maximum amplitude of the signal, then quantized value is set to be $m_p - \Delta V/2$.
   - If the sample's amplitude equals to the minimum amplitude of the signal, then quantized value is set to be $m_p + \Delta V/2$.
   - If the sample's amplitude equals to zero, then there is nothing to quantize and the quantized value is set to be 0.
2. Then the other general values of the samples are considered. These values are again divided into two classes.
   - If a sample is exactly equal to a value of a Quantization level, one of the following situations is possible and the quantization level is determined as follows with the aim of minimizing the effect of noise at the transmission.
     - Negative samples are quantized towards negative infinity
     - Positive samples are quantized towards positive infinity
   - If a sample lies between two Quantization levels it is quantized towards positive infinity.

```matlab
function quantized = quantizeSample(sample, qlevels,maxamp)

DeltaV = 2*maxamp/qlevels;      % Quantiation interval size
if sample == maxamp                         % Positive extreme
    quantized = sample - DeltaV/2;
    return
elseif sample == -1*maxamp                  % Negative extreme
    quantized = sample + DeltaV/2;
    return
elseif abs(sample) == 0 % zero means no sample to quantize
    quantized = 0;
    return
    % If the sample value does not belongs to any of the above cases
else
    % Iterate through Quantization levels
    for level = -1*maxamp:DeltaV:maxamp
        if level == sample % If a sample is exactly equal to a q-level
            if sample <0
                % Negative samples are quantized towards negative infinity
                quantized = level - DeltaV/2;
                return
            else
                % Positive samples are quantized towards positive infinity
                quantized = level + DeltaV/2;
                return
            end
        elseif level > sample         % If a sample lies between two levels
            quantized = level -DeltaV/2;
            return
        end
    end
end
end
```

Q7: Quantization of the Signal sampled at 8 times Nyquist rate ($8f_{nq}$) with 16 Quantization Levels

Observe the following figures while paying attention to the grids which makes it easy to identify the change in amplitudes after the quantization. Corresponding Sampled Values and Quantized Values are also given below and it is seen that both `1.0000 and 0.9239` are quantized on to the `0.9375` quantization level and other values are quantized onto the corresponding discrete levels.

```
Sampled Values:
1.0000    0.9239    0.7071    0.3827    0.0000   -0.3827   -0.7071   -0.9239   -1.0000
Quantized Values:
0.9375    0.9375    0.6875    0.4375         0   -0.4375   -0.6875   -0.9375   -0.9375
```

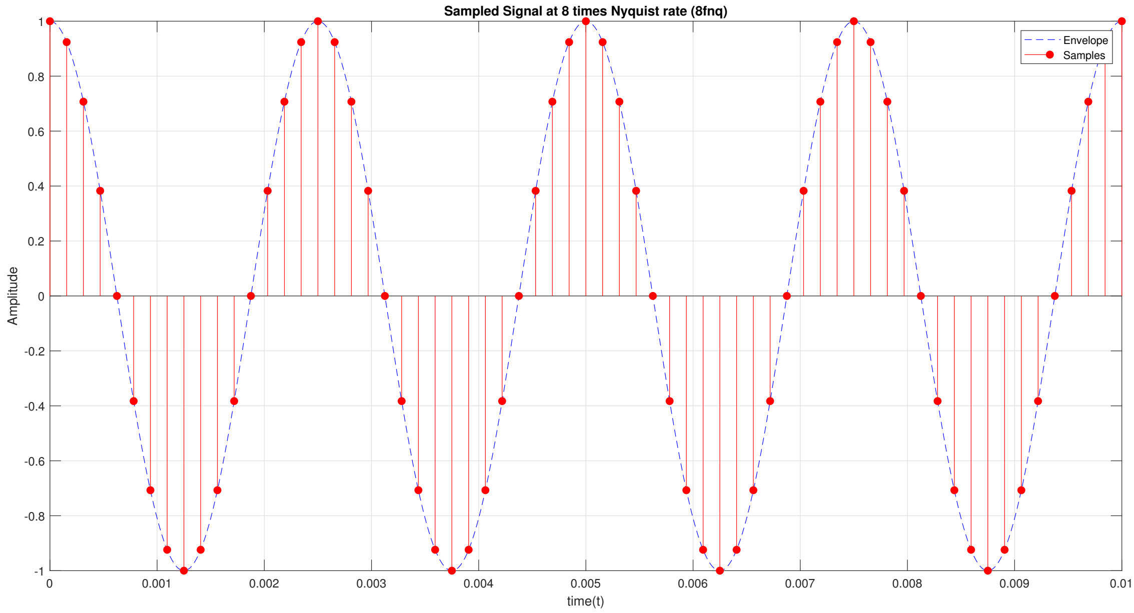


Figure 3: Sampled Signal at 8 times Nyquist rate ($8f_{nq}$)

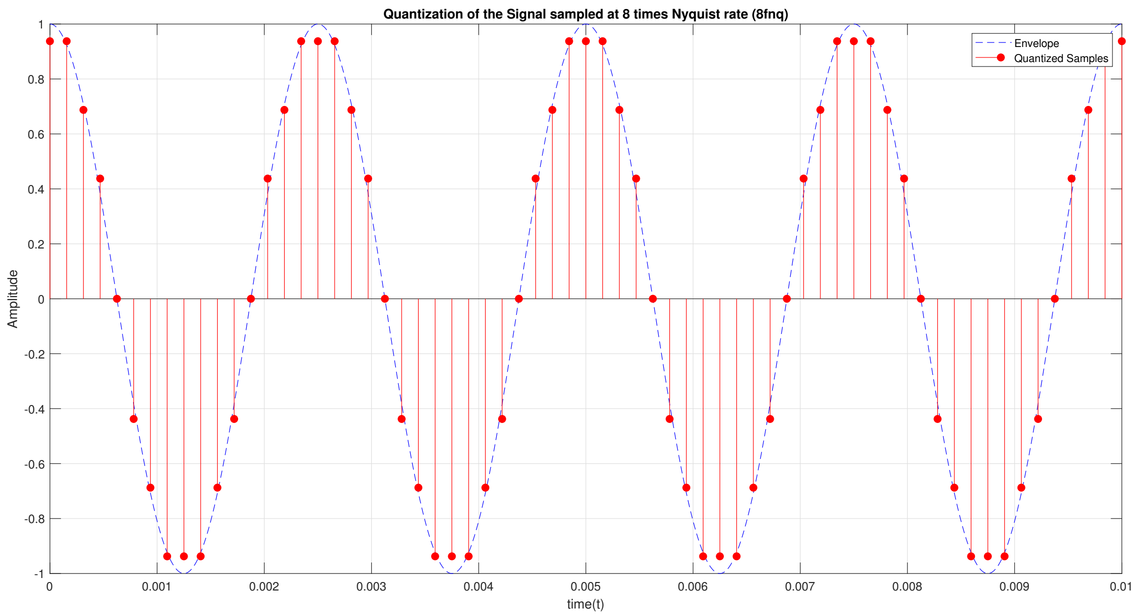


Figure 4: Quantization of the Signal sampled at 8 times Nyquist rate with 16 Quantization Levels

Q8: Quantization of the Signal sampled at 8 times Nyquist rate ($8f_{nq}$) with 32 and 8 Quantization Levels

```
Sampled Values:
1.0000    0.9239    0.7071    0.3827    0.0000   -0.3827   -0.7071   -0.9239   -1.0000
Quantized Values:
0.9688    0.9063    0.7188    0.4063         0   -0.4063   -0.7188   -0.9063   -0.9688
```

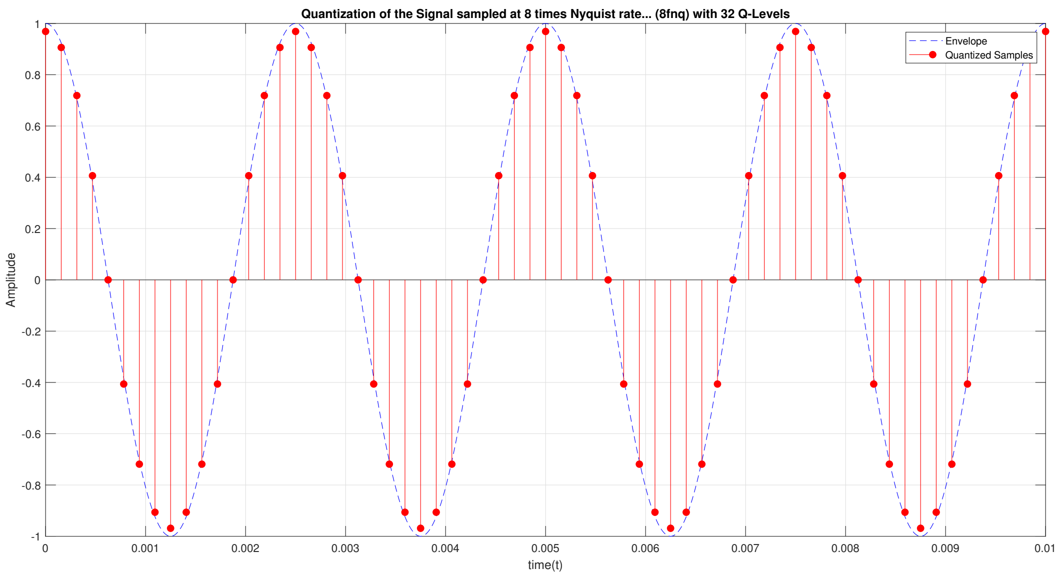


Figure 5: Quantization of the Signal sampled at 8 times Nyquist rate with 32 Quantization Levels

```
Sampled Values:
1.0000    0.9239    0.7071    0.3827    0.0000   -0.3827   -0.7071   -0.9239   -1.0000
Quantized Values:
0.8750    0.8750    0.6250    0.3750         0   -0.3750   -0.6250   -0.8750   -0.8750
```

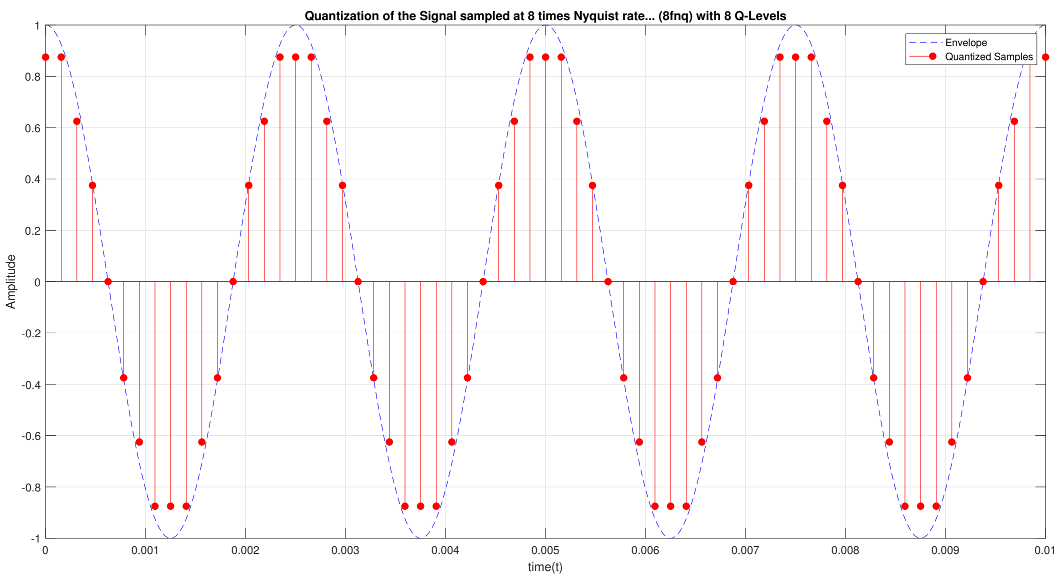


Figure 6: Quantization of the Signal sampled at 8 times Nyquist rate with 8 Quantization Levels

```
Sampled Values:
1.0000    0.9239    0.7071    0.3827    0.0000   -0.3827   -0.7071   -0.9239   -1.0000

Quantized Values with 32 Levels:
0.9688    0.9063    0.7188    0.4063         0   -0.4063   -0.7188   -0.9063   -0.9688

Quantized Values with 16 Levels:
0.9375    0.9375    0.6875    0.4375         0   -0.4375   -0.6875   -0.9375   -0.9375

Quantized Values with 8 Levels:
0.8750    0.8750    0.6250    0.3750         0   -0.3750   -0.6250   -0.8750   -0.8750
```

When observing the above plots and the given values following conclusions can be made.

- As depicted in Fig. 5 when the quantization levels are increases every sampled value is mapped into a unique quantization level but when we decrease the number of quantization levels as in Fig. 4 and Fig. 6 several sample values may be mapped into the same quantization level. This introduces an ambiguity at the signal reconstruction stage as we have lost the information about actual value of the amplitudes.
- In addition to that when the number of quantization levels are decreased the actual value of the sampled signal and the value of the quantized signal differs a lot which again introduces an ambiguity at the signal reconstruction stage. This can be clearly seen when comparing the original signal's envelope drawn in dashed line with the quantized signal.

Therefore in order to minimize the quantization noise we need to increase the number of quantization levels.

Q1 Code Snippet

```
A = 1;          % Amplitude
f = 400;        % Frequency
time = 10e-3;   % Time limit
t = linspace(0,time,1000);
y_t = A*cos(2*pi*f*t); % Input signal
plot(t,y_t,'Color','b','LineStyle','-','LineWidth',1)
grid on; xlabel('time(t)'); ylabel('Amplitude')
```

Q3 Code Snippet

```
fnq = 2*f;                           % Nyquist sampling frequency
delta_t = 1/fnq;                     % Time duration between two samples
sample_at = 0:delta_t:time;          % Sampling moments
samples = A*cos(2*pi*f*sample_at);   % Sampled signal
plot(t,y_t, '--', 'Color','b');      % plotting the envelope
hold on
stem(sample_at,samples,'filled', 'r');% plotting the sampled signal
legend("Envelope","Samples");legend('show')
grid on; xlabel('time(t)'); ylabel('Amplitude')
hold off
```

Q4 Code Snippet

```
% Sampled Signal at 2 times Nyquist rate (2fnq)
subplot(3,1,1)
delta_t_2fnq = 1/(2*fnq);          % Time duration between two samples
sample_at = 0:delta_t_2fnq:time;   % Sampling moments
samples = A*cos(2*pi*f*sample_at); % Sampled signal
plot(t,y_t, '--', 'Color','b');    % plotting the envelope
hold on
stem(sample_at,samples,'filled', 'r'); % plotting the sampled signal
legend("Envelope","Samples");legend('show')
title("Sampled Signal at 2 times Nyquist rate (2fnq)")
grid on; xlabel('time(t)'); ylabel('Amplitude')
hold off

% Sampled Signal at  Nyquist rate (fnq)
subplot(3,1,2)
delta_t_1fnq = 1/(1*fnq);          % Time duration between two samples
sample_at = 0:delta_t_1fnq:time;   % Sampling moments
samples = A*cos(2*pi*f*sample_at); % Sampled signal
plot(t,y_t, '--', 'Color','b');    % plotting the envelope
```

```
hold on
stem(sample_at,samples,'filled', 'r'); % plotting the sampled signal
legend("Envelope","Samples");legend('show')
title("Sampled Signal at  Nyquist rate (fnq)")
grid on; xlabel('time(t)'); ylabel('Amplitude')
hold off

% Sampled Signal at one-half of the Nyquist rate (fnq/2)
subplot(3,1,3)
delta_t_halffnq = 1/(0.5*fnq);      % Time duration between two samples
sample_at = 0:delta_t_halffnq:time; % Sampling moments
samples = A*cos(2*pi*f*sample_at);  % Sampled signal
plot(t,y_t, '--', 'Color','b');     % plotting the envelope
hold on
stem(sample_at,samples,'filled', 'r'); % plotting the sampled signal
legend("Envelope","Samples");legend('show')
title("Sampled Signal at one-half of the Nyquist rate (fnq/2)")
grid on; xlabel('time(t)'); ylabel('Amplitude')
hold off
```

Q7 Code Snippet

```
figure;
delta_t_8fnq = 1/(8*fnq);          % Time duration between two samples
sample_at = 0:delta_t_8fnq:time;   % Sampling moments
samples = A*cos(2*pi*f*sample_at);% Sampled signal
plot(t,y_t, '--', 'Color','b');    % plotting the envelope
hold on
% Quantization the sampled signal using the created function
qlevels = 16;
quantized_samples = zeros(1,length(samples));
for sample = 1: length(samples)
    quantized_samples(sample) = quantizeSample(round(samples(sample),5), qlevels,A);
end

%% Uncomment this part to plot the sampled signal
% stem(sample_at,samples, 'filled', 'r');
% legend("Envelope","Samples");legend('show')
% title("Sampled Signal at 8 times Nyquist rate (8fnq)")

% plotting the quantized signal
stem(sample_at,quantized_samples, 'filled', 'r');
legend("Envelope","Quantized Samples");legend('show')
title("Quantization of the Signal sampled at 8 times Nyquist rate (8fnq) with 16 Q-Levels")
grid on; xlabel('time(t)'); ylabel('Amplitude')
hold off
```

Q8 Code Snippet

Quantized with 2L = 32 Quantization Levels

```
figure;
plot(t,y_t, '--', 'Color','b');         % plotting the envelope
hold on
% Quantization the sampled signal using the created function
qlevels = 32;
quantized_samples = zeros(1,length(samples));
for sample = 1: length(samples)
    quantized_samples(sample) = quantizeSample(round(samples(sample),5), qlevels,A);
end
% plotting the quantized signal
stem(sample_at,quantized_samples, 'filled', 'r');
legend("Envelope","Quantized Samples");legend('show')
title("Quantization of the Signal sampled at 8 times Nyquist rate..." + ...
    " (8fnq) with 32 Q-Levels")
grid on; xlabel('time(t)'); ylabel('Amplitude')
hold off
```

Quantized with L/2 = 8 Quantization Levels

```
figure;
plot(t,y_t, '--', 'Color','b');         % plotting the envelope
hold on
% Quantization the sampled signal using the created function
qlevels = 8;
quantized_samples = zeros(1,length(samples));
for sample = 1: length(samples)
    quantized_samples(sample) = quantizeSample(round(samples(sample),5), qlevels,A);
end
% plotting the quantized signal
stem(sample_at,quantized_samples, 'filled', 'r');
legend("Envelope","Quantized Samples");legend('show')
title("Quantization of the Signal sampled at 8 times Nyquist rate..." + ...
    " (8fnq) with 8 Q-Levels")
grid on; xlabel('time(t)'); ylabel('Amplitude')
hold off
```

Function definition for Quantization

```
function quantized = quantizeSample(sample, qlevels,maxamp)

DeltaV = 2*maxamp/qlevels;      % Quantiation interval size
if sample == maxamp                      % Positive extreme
    quantized = sample - DeltaV/2;
```

```
    return
elseif sample == -1*maxamp               % Negative extreme
    quantized = sample + DeltaV/2;
    return
elseif abs(sample) == 0 % zero means no sample to quantize
    quantized = 0;
    return
    % If the sample value does not belongs to any of the above cases
else
    % Iterate through Quantization levels
    for level = -1*maxamp:DeltaV:maxamp
        if level == sample % If a sample is exactly equal to a q-level
            if sample <0
                % Negative samples are quantized towards negative infinity
                quantized = level - DeltaV/2;
                return
            else
                % Positive samples are quantized towards positive infinity
                quantized = level + DeltaV/2;
                return
            end
        elseif level > sample        % If a sample lies between two levels
            quantized = level -DeltaV/2;
            return
        end
    end
end
end
```