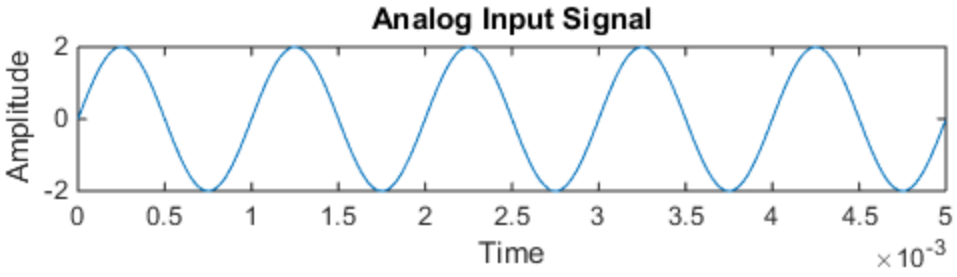
```
% HARSHIT RAI
% 2017152

%Answer

% A) If quantization levels are increased, then the quantization error decreases.
% Because they are inversely proportional.
% Error=(2*Amplitude) / (L)

% B) If sampling frequency is increased, then the quantization error decreases.
% Because now more number of samples are recorded
% and therefore error also gets decreased.
```

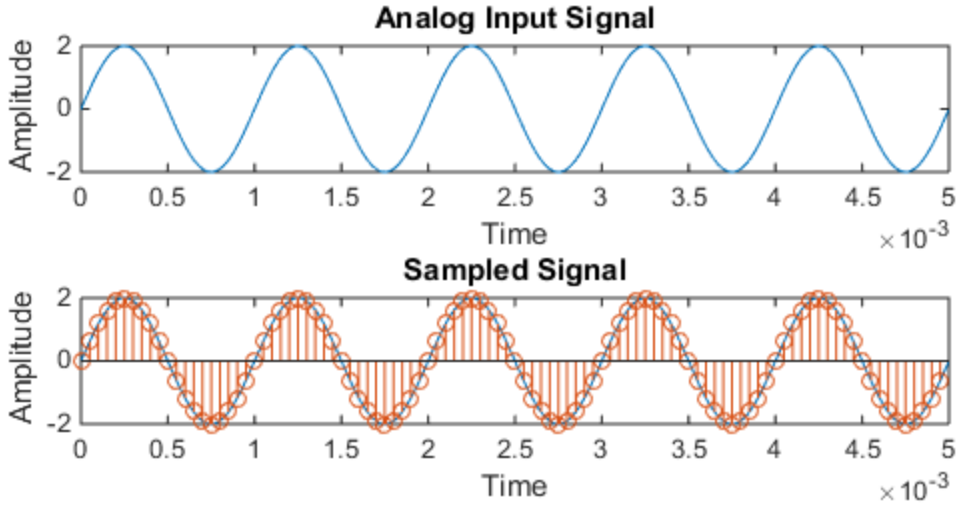
Input Sinusoid

```
clear all;
clc;

fm=1*1000; % Message signal frequency
tm=1/fm;

fs=250*1000; % Sampling frequency of original signal : almost continuous
ts=1/fs;

n=5; % Generate 5 cycles
a=2; % Amplitude of sinusoid
t= 0 : ts : (n*tm-ts) ; % From 0 to ts in step of (n*tm-ts)

mt=a*sin(2*pi*fm*t);  % Analog Input Signal

subplot(3,1,1);
plot(t,mt);
title('Analog Input Signal');
xlabel('Time');
ylabel('Amplitude');
```



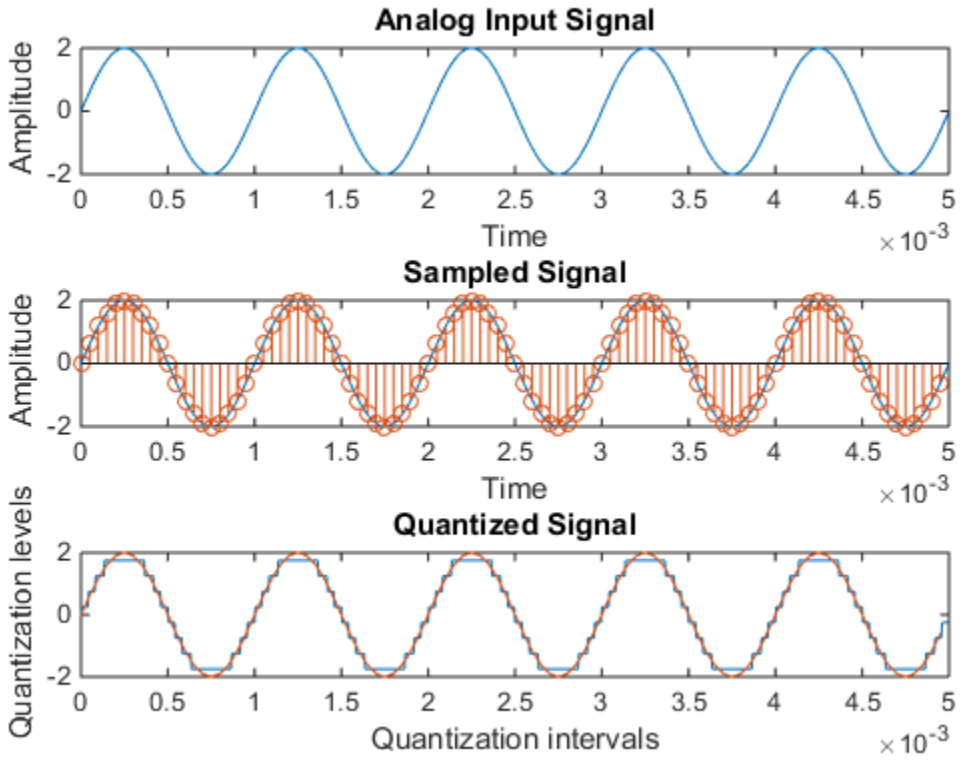

Sampling

```
fs1=20*1000;   % Sampling frequency
ts1=1/fs1;
t1= 0 : ts1 : (n*tm-ts1) ; % Time index
mt1=a*sin(2*pi*fm*t1);

subplot(3,1,2);
plot(t,mt);
title('Sampled Signal');
xlabel('Time');
ylabel('Amplitude');
hold on;
stem(t1,mt1);
```




```
%Quantisation

bit=3; % Quantization bit for 8 levels
l=2^bit;
del=2*a/l; % Step size

samples=length(mt); % Total number of samples
levels=zeros(1,samples); % Array for quantization levels
error=zeros(1,samples);   % Array for quantization error

up=a-del/2; % Maximum voltage
down=-a+del/2; % Minimum voltage

a=transpose(zeros(1,samples));   % For levels
b=transpose(zeros(1,samples));   % For bits corresponding quantization levels
codegenerator=[a b];     % Array for binary code generator

%In the below "for loop" we are mapping the sample
%values to their quantization levels

for h=down:del:up % Iterating from lowest to the highest level
    for r=1:samples % For all samples
        % If the sample value lies within the range -del/2 < sample < del/2
        if(((h-del/2)<mt(r))&&(mt(r)<(h+del/2)))
            levels(r)=h;
            error(r)=mt(r)-h;    % Error= Actual value - apparent value
```

```
            codegenerator(r,1)=h;
        end
    end
end

subplot(3,1,3);
plot(t,levels,t1,mt1);
title('Quantized Signal');
xlabel('Quantization intervals');
ylabel('Quantization levels');
```




Code generator

```
for p=1:samples % For all samples
    if(levels(p)==down+(0*del))
        codegenerator(p,2)=000;
    end
    if(levels(p)==down+(1*del))
        codegenerator(p,2)=001;
    end
    if(levels(p)==down+(2*del))
        codegenerator(p,2)=010;
    end
    if(levels(p)==down+(3*del))
        codegenerator(p,2)=011;
    end
    if(levels(p)==down+(4*del))
```

```
        codegenerator(p,2)=100;
    end
    if(levels(p)==down+(5*del))
        codegenerator(p,2)=101;
    end
    if(levels(p)==down+(6*del))
        codegenerator(p,2)=110;
    end
    if(levels(p)==down+(7*del))
        codegenerator(p,2)=111;
    end

end
```

Quantization error

```
display(rms(error)^2);
display(mean(error));
```

```
ans =

    0.0247


ans =

   2.0000e-04
```

*Published with MATLAB® R2014b*