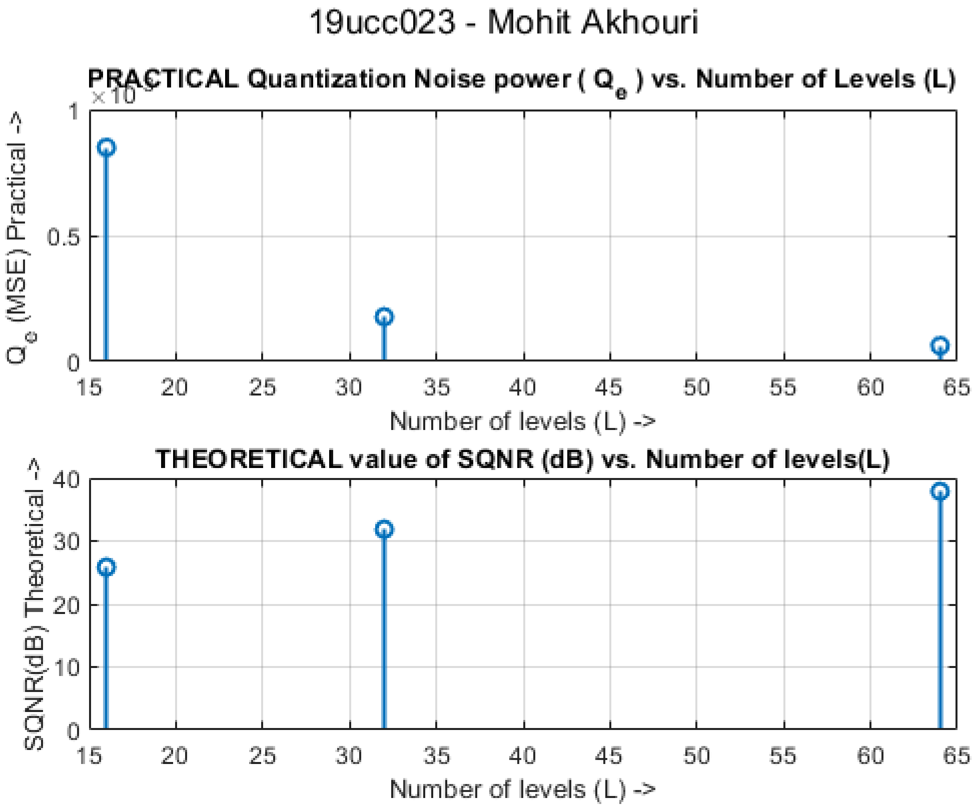
```
% 19ucc023
% Mohit Akhouri
% Experiment 2 - Observation 2

clc;
clear all;
close all;

A = 1; % defining Amplitude
n_cycles = 5; % defining number of cycles
f = 3000; % defining message signal frequency

fs_sampled = 8000; % defining sampling frequency
n_sampled = 0:1:floor(n_cycles*(fs_sampled/f))-1; % defining range of
 n
x_sampled = A*cos(2*pi*f*n_sampled*(1/fs_sampled)); % defining sampled
 signal x_sampled

L = [16 32 64]; % defining array for number of levels
Quantization_Noise_Power = zeros(1,3); % to store the SQNR practical
for i=1:length(L)
    y = myquantizer(x_sampled,L(i));
    Quantization_Noise_Power(i) = mean((y-x_sampled).*(y-
x_sampled)); % SQNR practical calculation
end

% plotting SQNR practical vs. number of levels
figure;
subplot(2,1,1);
stem(L,Quantization_Noise_Power,'Linewidth',1.5);
xlabel('Number of levels (L) ->');
ylabel('Q_{e} (MSE) Practical ->');
title('PRACTICAL Quantization Noise power ( Q_{e} ) vs. Number of
 Levels (L)');
grid on;

% calculating theoretical SQNR in dB
SQNR_db = zeros(1,3);
b16 = 4; % number of bits for L=16
b32 = 5; % number of bits for L=32
b64 = 6; % number of bits for L=64
SQNR_db(1) = 1.76 + 6.02*b16;
SQNR_db(2) = 1.76 + 6.02*b32;
SQNR_db(3) = 1.76 + 6.02*b64;

% plotting SQNR (theoretical) vs. Number of levels
subplot(2,1,2);
stem(L,SQNR_db,'Linewidth',1.5);
xlabel('Number of levels (L) ->');
ylabel('SQNR(dB) Theoretical ->');
title('THEORETICAL value of SQNR (dB) vs. Number of levels(L)');
grid on;
```

```
sgtitle('19ucc023 - Mohit Akhouri');
```



Published with MATLAB® R2020b