```
clc;
clear all;
close all;

% Wait for a key press to start simulation
disp('--------------------');
disp('|  Lab1 - PCM    |');
disp('--------------------');
disp(' ');
% input('Press Enter to start simulation');
savepath = 'pdf/Lab1_PCM/';

% Define signal properties
F = 1 * 10^3;
Fs = 40 * 10^3;
tStart = 0;
tStop = 0.01;
sigSNR = 20;
x_range = [-5 5];

% Define channel properties
SNR = 0 : 30;
record_SNR = 20;

% Define PCM properties
nBits = 1 : 48;
record_nBits = 8;

% initialize the average error power matrix
y_avgerr_an = zeros(length(SNR), length(nBits));
y_avgerr_dg = zeros(length(SNR), length(nBits));

% Run for all SNR and nBits
for iSNR = 1 : length(SNR)
 for inBits = 1 : length(nBits)
  % get the original signal
  [x_sig, t] = sig_lab1(F, Fs, sigSNR, tStart, tStop);
  % pass it through AWGN channel to reciever side
  y_arcv = awgn(x_sig, SNR(iSNR));
  % get the error power
  y_aerr = (y_arcv - x_sig) .^ 2;
  % get the average error power
  y_avgerr_an(iSNR, inBits) = sum(y_aerr) / length(y_aerr);
  % quantize the original signal
  x_qua = sig_quantize(x_sig, x_range, 2^nBits(inBits));
  % encode the quantized signal
  x_enc = sig_encode(x_qua, nBits(inBits));
  % pass the encoded signal through AWGN channel to reciever side
  y_drcv = awgn(x_enc, SNR(iSNR));
  % decode the recieved signal
  y_dec = sig_decode(y_drcv, nBits(inBits));
  % dequantize the recieved signal
```

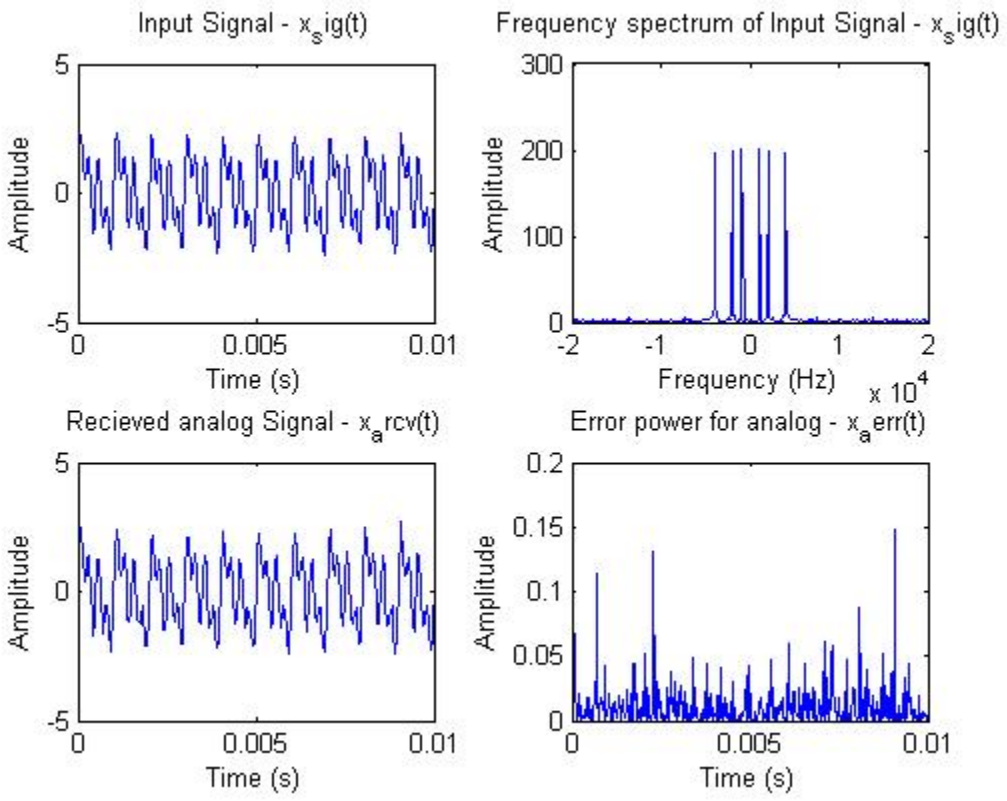
```
y_deq = sig_dequantize(y_dec, x_range, 2^nBits(inBits));
% get the error power
y_derr = (y_deq - x_sig) .^ 2;
% get the average error power
y_avgerr_dg(iSNR, inBits) = sum(y_derr) / length(y_derr);
if(record_SNR == SNR(iSNR) && record_nBits == nBits(inBits))
 % draw the original signal
 figure;
 % get the time-domain plot of input signal
 subplot(2, 2, 1);
 plot(t, x_sig);
 xlim([tStart tStop]);
 ylim(x_range);
 title('Input Signal - x_sig(t)');
 xlabel('Time (s)');
 ylabel('Amplitude');
 % Get the frequency spectrum of input signal
 [Y_sigf, YF] = sig_freq(x_sig, Fs);
 subplot(2, 2, 2);
 plot(YF, abs(Y_sigf));
 title('Frequency spectrum of Input Signal - x_sig(t)');
 xlabel('Frequency (Hz)');
 ylabel('Amplitude');
 % plot the recieved signal
 % the recieved signal
 subplot(2, 2, 3);
 plot(t, y_arcv);
 xlim([tStart tStop]);
 ylim(x_range);
 title('Recieved analog Signal - x_arcv(t)');
 xlabel('Time (s)');
 ylabel('Amplitude');
 % the error power signal
 subplot(2, 2, 4);
 plot(t, y_aerr);
 xlim([tStart tStop]);
 title('Error power for analog - x_aerr(t)');
 xlabel('Time (s)');
 ylabel('Amplitude');
 % plot the pcm encoding signals
 figure;
 % the quantized signal
 subplot(2, 2, 1);
 plot(t, x_qua);
 xlim([tStart tStop]);
 ylim([-1 2^nBits(inBits)]);
 title('Quantized PCM Signal - x_qua(t)');
 xlabel('Time (s)');
 ylabel('Amplitude');
 % the Encoded signal
 subplot(2, 2, 2);
 plot(x_enc);
 ylim([-1 2]);
 title('Encoded PCM Signal - x_enc(t)');
```

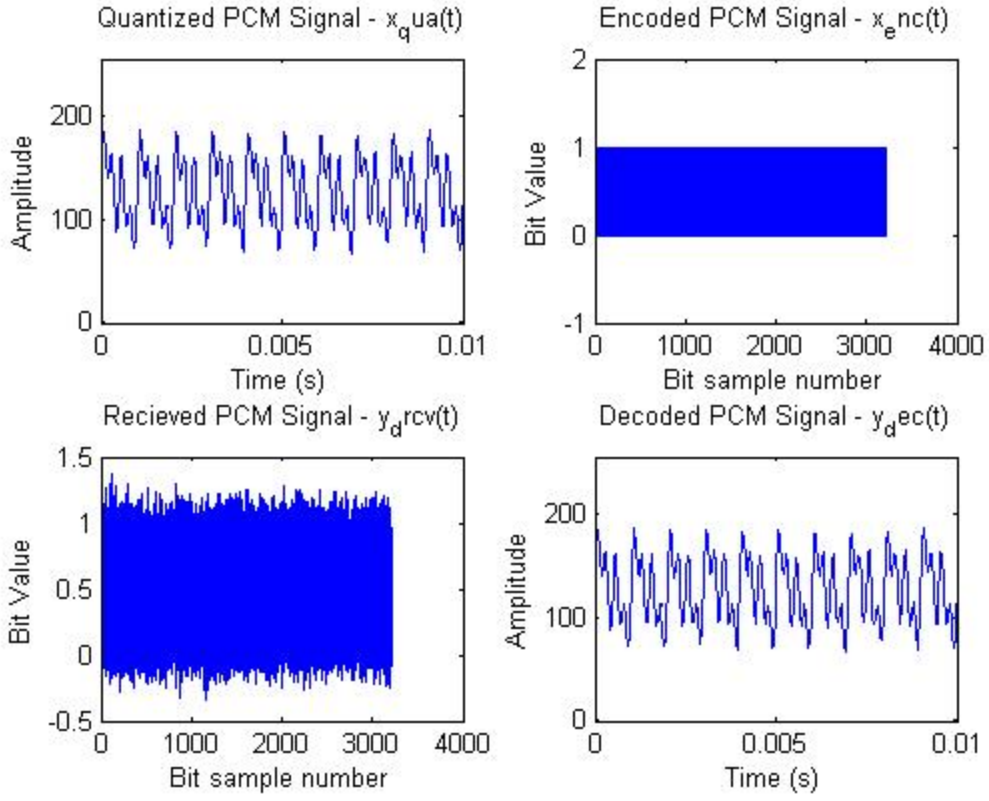
```
    xlabel('Bit sample number');
    ylabel('Bit Value');
    % the Recieved signal
    subplot(2, 2, 3);
    plot(y_drcv);
    title('Recieved PCM Signal - y_drcv(t)');
    xlabel('Bit sample number');
    ylabel('Bit Value');
    % the decoded signal
    subplot(2, 2, 4);
    plot(t, y_dec);
    ylim([-1 2^nBits(inBits)]);
    title('Decoded PCM Signal - y_dec(t)');
    xlabel('Time (s)');
    ylabel('Amplitude');
    % contd recieveing figure
    figure;
    % Get the quantized signal
    subplot(2, 1, 1);
    plot(t, y_deq);
    xlim([tStart tStop]);
    ylim(x_range);
    title('Dequantized PCM Signal - y_deq(t)');
    xlabel('Time (s)');
    ylabel('Amplitude');
    % Get the error signal
    subplot(2, 1, 2);
    plot(t, y_derr);
    title('Error power for PCM - y_derr(t)');
    xlabel('Time (s)');
    ylabel('Amplitude');
    xlim([tStart tStop]);
    drawnow;
   end
  end
end
fig = figure;
for i = 1 : size(y_avgerr_an, 2)
 semilogy(SNR, y_avgerr_an(:, i), 'r');
 hold on;
 semilogy(SNR, y_avgerr_dg(:, i), 'b');
 ttl = sprintf('Bits - Variation of avg. error power (%f bits)', nBits(i));
 title(ttl);
 xlabel('SNR');
 ylabel('Avg. error power - log10');
 xlim([SNR(1) SNR(end)]);
 saveas(fig, [savepath ttl '.jpg']);
 clf(fig);
end
for i = 1 : size(y_avgerr_an, 1)
 semilogy(nBits, y_avgerr_an(i, :), 'r');
 hold on;
 semilogy(nBits, y_avgerr_dg(i, :), 'b');
 ttl = sprintf('SNR - Variation of avg. error power (%f SNR)', SNR(i));
```

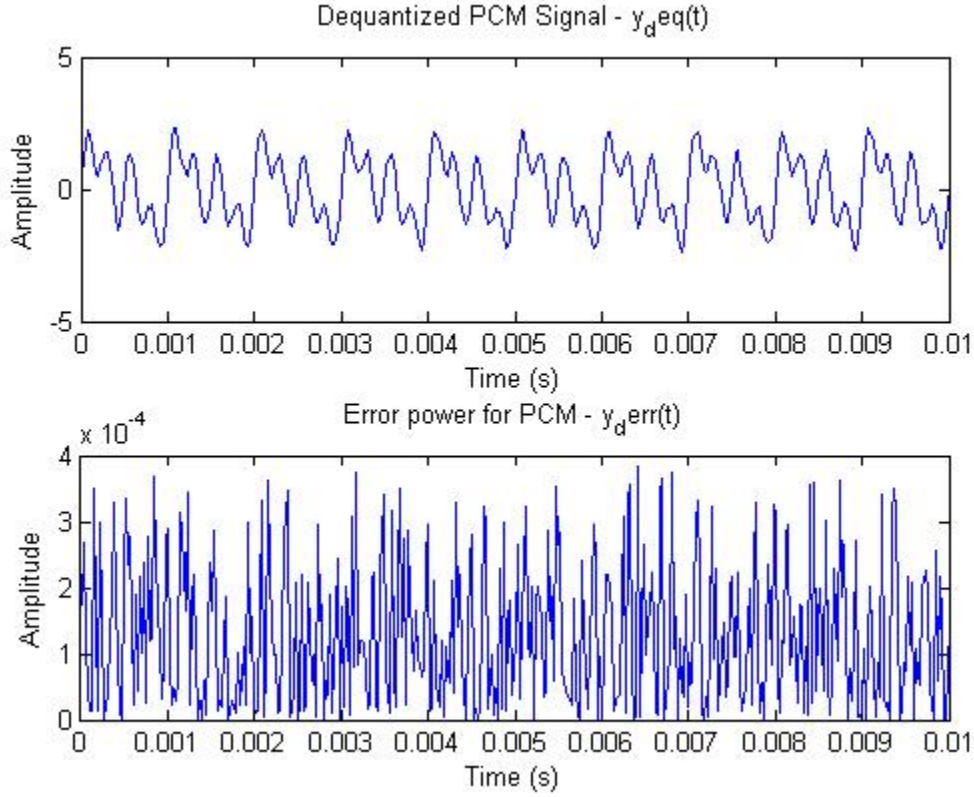
```
 title(ttl);
 xlabel('No. of Bits');
 ylabel('Avg. error power - log10');
 xlim([nBits(1) nBits(end)]);
 saveas(fig, [savepath ttl '.jpg']);
 clf(fig);
end
close(fig);
figure;
surf(log10(y_avgerr_an), zeros(size(y_avgerr_an)));
hold on;
surf(log10(y_avgerr_dg), ones(size(y_avgerr_dg)));
ttl = 'Variation of avg. error power';
title(ttl);
ylabel('SNR');
xlabel('No. of bits');
zlabel('Avg. error power - log10');
```

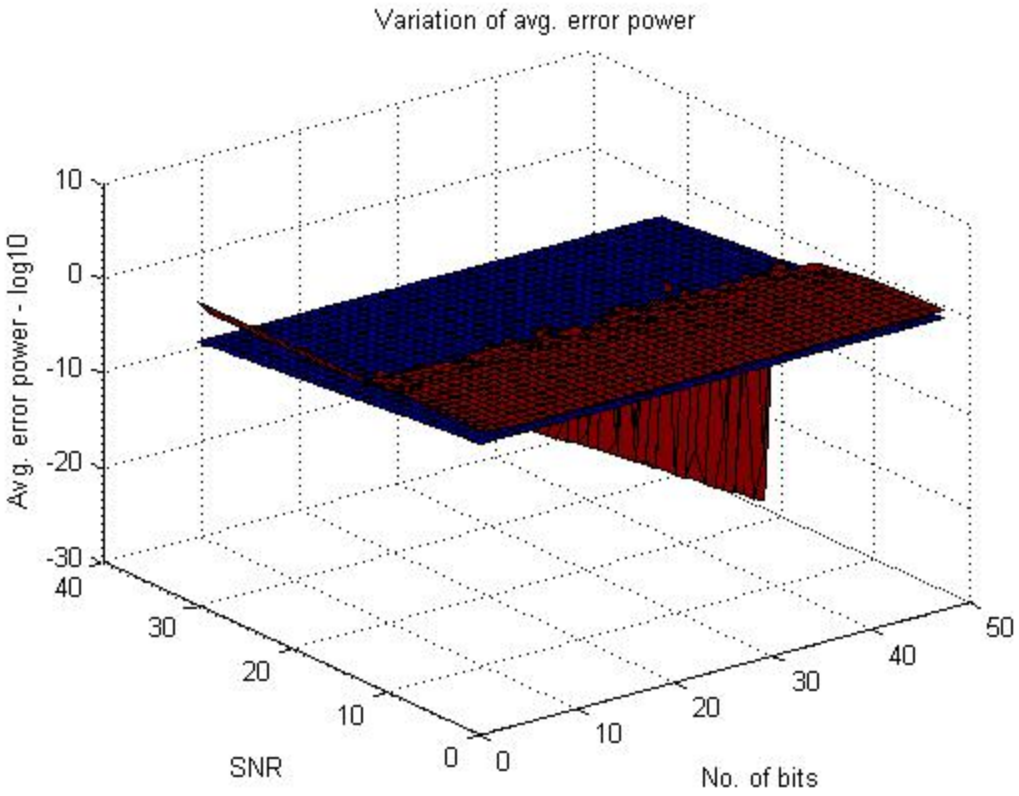
```
--------------------

|  Lab1 - PCM    |
--------------------
```




Quantized PCM Signal - $x_q$ua(t)
Amplitude
Time (s)
Encoded PCM Signal - $x_e$nc(t)
Bit Value
Bit sample number
Recieved PCM Signal - $y_d$rcv(t)
Bit Value
Bit sample number
Decoded PCM Signal - $y_d$ec(t)
Amplitude
Time (s)


Dequantized PCM Signal - $y_d$eq(t)
Amplitude
Time (s)
Error power for PCM - $y_d$err(t)
x 10^-4
Amplitude
Time (s)



Published with MATLAB® 7.14