GUC
German University in Cairo
الجامعة الألمانية بالقاهرة

Communication Theory Project Report

Made By:

Abdelrahman Walid Ahmed 58-2089

Cladora Magdy 58-5329

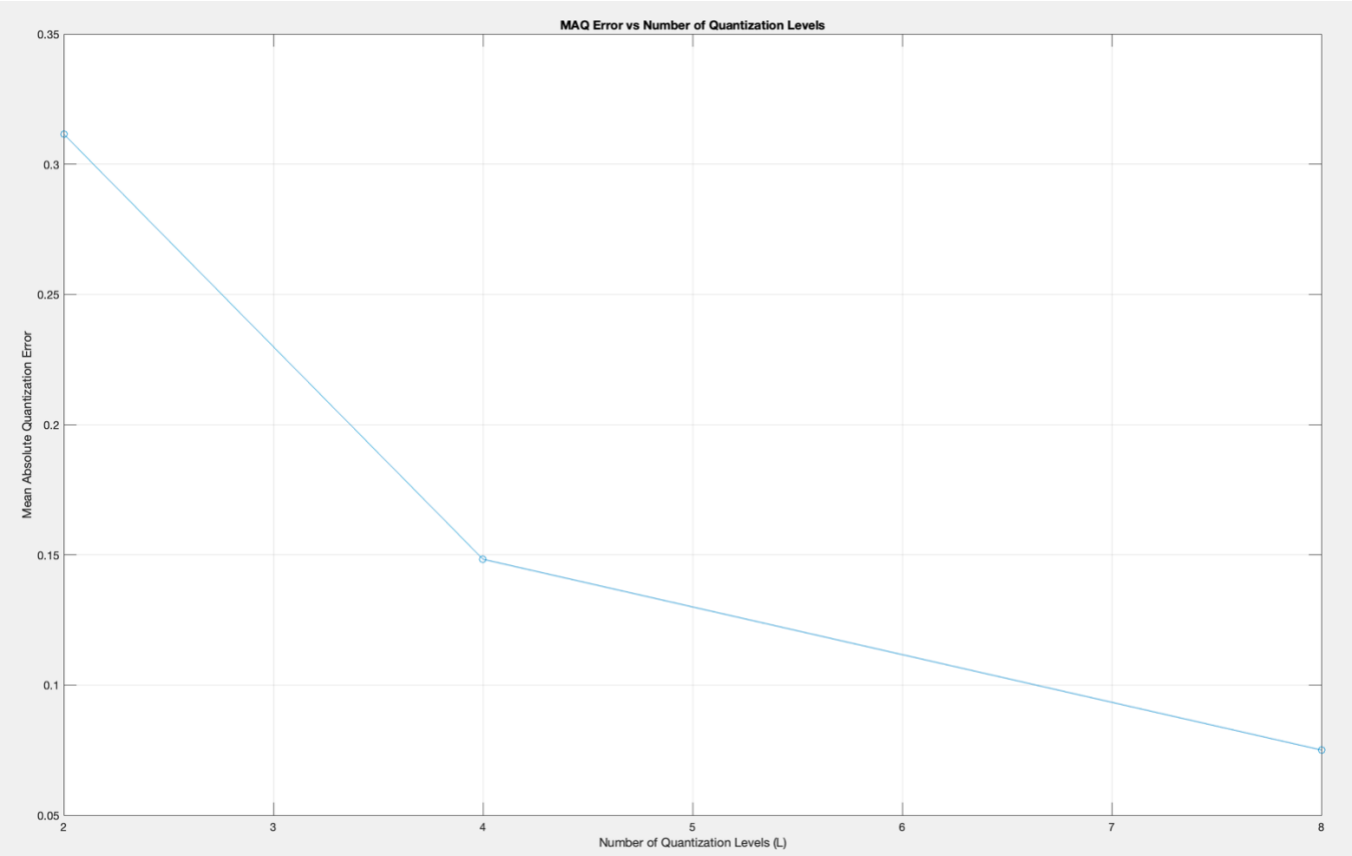
Yehia Mohamed Nader 58-18085

Supervised by:

Prof. Dr. Ahmed El-Mahdy

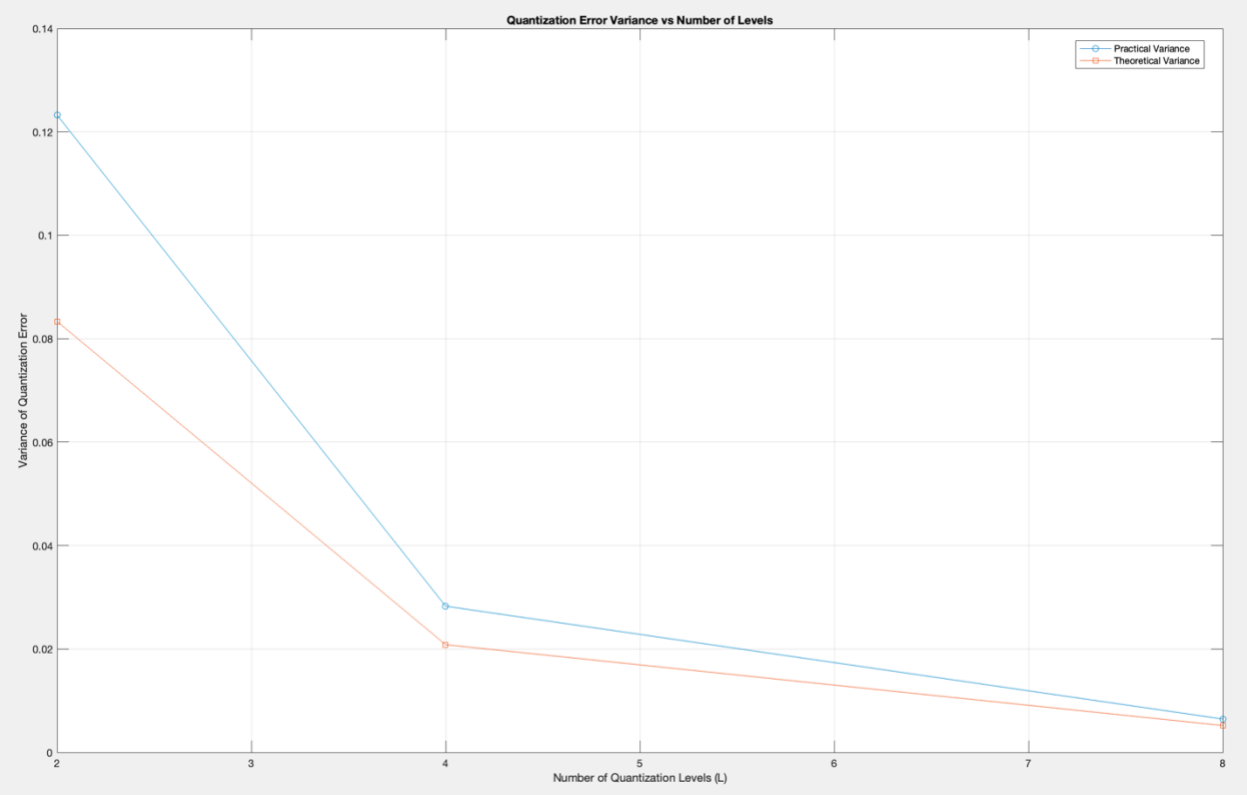
TA. Eng. Ghadir Mostafa Mahmoud

Task 3:



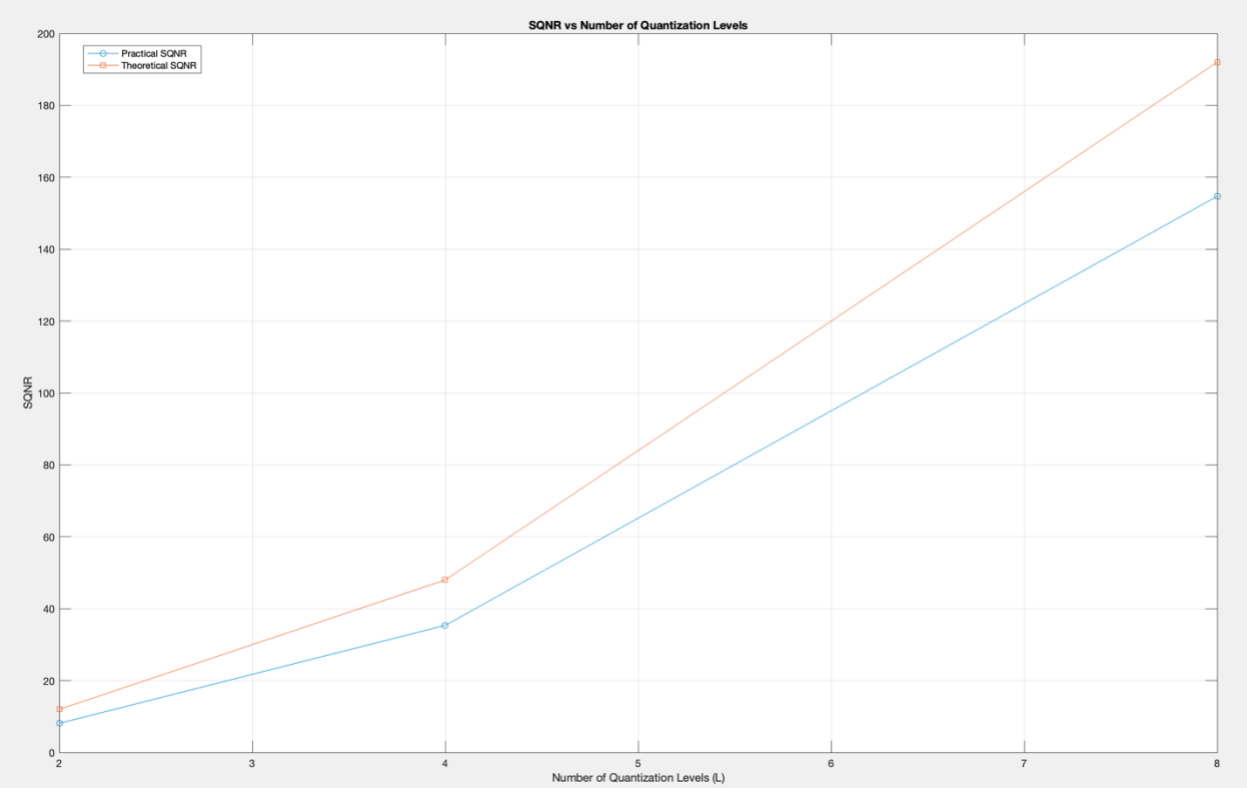
There is an inverse relationship between the number of quantization levels and the mean absolute quantization error: as the number of quantization levels increases, the mean absolute quantization error decreases. A significant drop in the mean absolute quantization error is observed between levels 2 and 4.

Task 4:



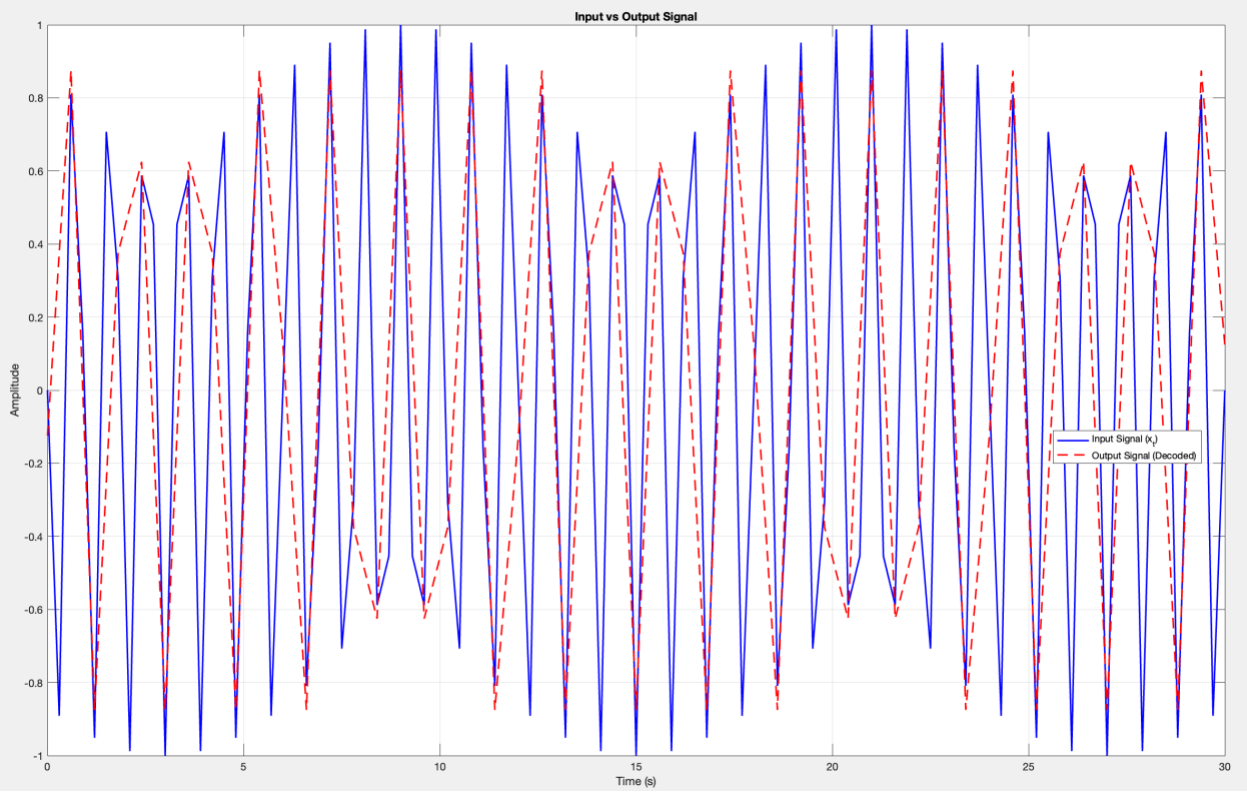
The variance of quantization error decreases as the number of quantization levels increases, demonstrating a negative relationship. The theoretical variance is consistently lower than the practical variance, with the gap narrowing as the number of quantization levels grows. A sharp decline in variance is observed between levels 2 and 4, followed by a slower reduction at higher levels, indicating diminishing returns in error reduction as quantization levels increase.

Task 5:



There is a positive relationship between practical SQNR and theoretical SQNR as the number of quantization levels increases. The theoretical SQNR consistently exceeds the practical SQNR, with the gap between them widening as the number of quantization levels grows. Additionally, both SQNR values experience a rapid increase at lower quantization levels before showing more gradual growth at higher levels.

Task 12:



To improve the accuracy of the output signal, the number of samples can be increased, capturing more detail from the input signal. Increasing the number of quantization levels reduces the absolute error, though it comes at the cost of using more bits for encoding. Alternatively, non-uniform quantization can be employed, which allocates levels more effectively based on the signal's characteristics, improving approximation while maintaining efficiency.

There a difference between input and output mainly due to sampling, quantization and decoding. Sampling captures the signal at specific time intervals, which reduces the level of detail in the signal. If the sampling rate is too low, some parts of the signal might overlap and cause distortion. Quantization rounds the sampled values to a fixed set of levels, which introduces small errors depending on how many levels (L) are used. When the signal is reconstructed using these rounded values, it doesn't perfectly match the original signal.

Code:

```
a = 1;
b = 9;
t = 0:0.3:30;
x_t = a * sin(0.5*pi*t*b);

% Sampling Task 1
Ns = 2;
ts = t(1 : Ns : end);
xs = x_t(1 : Ns : end);

%Quantization Task 2
function xq = uniform_quantizer(xs , L)
vmax = max(xs);
vmin = min(xs);
delta = (vmax - vmin)/L;
quantized_levels = vmin+(delta/2) : delta : vmax-(delta/2);
for i = 1 : length(xs)
    abs_error = abs(xs(i) - quantized_levels);
    l = find(abs_error == min(abs_error));
    xq(i) = quantized_levels(l(1));
end
end

%Plot Mean Absolute Quantization Error vs Number of Levels Task 3
L = [2 4 8];
mean_abs_error = zeros(size(L));
for i = 1 : length(L)
    xq = uniform_quantizer(xs , L(i));
    mean_abs_error(i) = mean(abs(xs - xq));
end
figure;
plot(L , mean_abs_error, '-o');
xlabel('Number of Quantization Levels (L)');
ylabel('Mean Absolute Quantization Error');
```

```
title('MAQ Error vs Number of Quantization Levels');
grid on;

% Plot Practical and Theoretical Varience vs Number of levels Task 4
pratical_varience = zeros(size(L));
theoretical_variance = zeros(size(L));
for i = 1:length(L)
    xq = uniform_quantizer(xs , L(i));
    pratical_varience(i) = var(xs - xq);
    vmax = max(xs);
    vmin = min(xs);
    delta = (vmax - vmin)/L(i);
    theoretical_variance(i) = (delta^2)/12;
end
figure;
plot(L, pratical_varience, '-o', 'DisplayName', 'Practical Variance');
hold on;
plot(L, theoretical_variance, '-s', 'DisplayName', 'Theoretical Variance');
xlabel('Number of Quantization Levels (L)');
ylabel('Variance of Quantization Error');
title('Quantization Error Variance vs Number of Levels');
legend('Location', 'best');
grid on;
hold off;

% Plot Practical and Theoretical SQNR vs Number of levels Task 5
practical_SQNR = zeros(size(L));
theoretical_SQNR = zeros(size(L));
for i = 1:length(L)
    practical_SQNR(i) = (a^2) / pratical_varience(i);
    theoretical_SQNR(i) = 3 * L(i)^2;
end
%Convert SQNR into dB
% practical_SQNR_dB = 10 * log10(practical_SQNR);
% theoretical_SQNR_dB = 10 * log10(theoretical_SQNR);
figure;
plot(L, practical_SQNR, '-o', 'DisplayName', 'Practical SQNR');
```

```
hold on;
plot(L, theoretical_SQNR, '-s', 'DisplayName', 'Theoretical SQNR');
xlabel('Number of Quantization Levels (L)');
ylabel('SQNR');
title('SQNR vs Number of Quantization Levels');
legend('Location', 'best');
grid on;
hold off;

L = 8;
xq = uniform_quantizer(xs , L);
delta = (vmax - vmin)/L;

% Encode quantized levels Task 6
uniqueQuantizedLevels = vmin+(delta/2) : delta : vmax-(delta/2);%Unique
quantized values in ascending order
encoded = zeros(1,length(xq));
for i = 1:length(xq)
    for j = 1:length(uniqueQuantizedLevels)
        if xq(i) == uniqueQuantizedLevels(j)
            encoded(i) = j-1;
        end
    end
end

% Huffman encoder Task 7
unique_values = unique(encoded);
[counts, edges] = histcounts(encoded, [unique_values - 0.5,
unique_values(end) + 0.5]);
probabilities = counts / length(encoded);
dictionary = huffmandict(unique(encoded),probabilities);
sourceEncoded = huffmanenco(encoded,dictionary);

% Noiseless channel Task 8
noise = 0;
sourceEncoded = sourceEncoded + noise;
```

```
% Huffman decoder Task 10
sourceDecoded = huffmandeco(sourceEncoded,dictionary);

% Decoder Task 9
uniqueDecodedLevels = zeros(1,L);
uniqueDecodedLevels(1) = vmin + (delta/2);
uniqueDecodedLevels(L) = vmax - (delta/2);
for i=2:(L-1)
   uniqueDecodedLevels(i) = uniqueDecodedLevels(i-1) + delta;
end
decodedSequence = zeros(1,length(sourceDecoded));
for i = 1:length(sourceDecoded)
   decodedSequence(i) = uniqueDecodedLevels(sourceDecoded(i) + 1);
end

% Plot Input and Output Signals Task 11
figure;
plot(t, x_t, 'b-', 'LineWidth', 1.5, 'DisplayName', 'Input Signal (x_t)');
hold on;
plot(ts, decodedSequence, 'r--', 'LineWidth', 1.5, 'DisplayName', 'Output
Signal (Decoded)');
xlabel('Time (s)');
ylabel('Amplitude');
title('Input vs Output Signal');
legend('Location', 'best');
grid on;
hold off;

% Efficiency and CR Task 13 and 14
I = -log2(probabilities);
H = sum(probabilities .* I);
L = sum(probabilities .* ceil(I));
eff = H / L;
CR = log2(length(unique(encoded))) / L;
disp(['Compression Efficiency: ', num2str(eff)]);
disp(['Compression Rate (CR): ', num2str(CR)]);
```