Digital Signal Processing Laboratory

EXPERIMENT-5

Shorya Sharma
19EE01017

Quantization

Quantization is the process of mapping continuous amplitude (analog) signal into discrete amplitude (digital) signal. During **quantization**, the input amplitude is round off to the nearest quantized level. This rounding off is known as **quantization error**. Quantization error can be reduced by increasing the numbers of quantization levels. Quantization refers to the process of transforming an analog signal, which has a continuous set of values, to a digital signal, which has a discrete set.

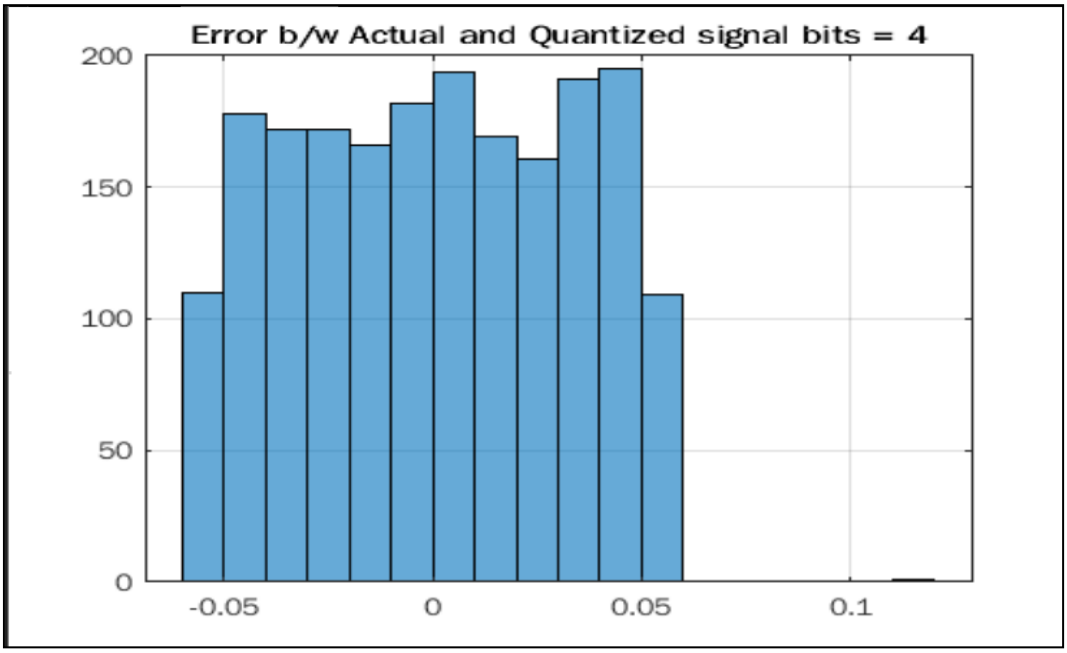
Code

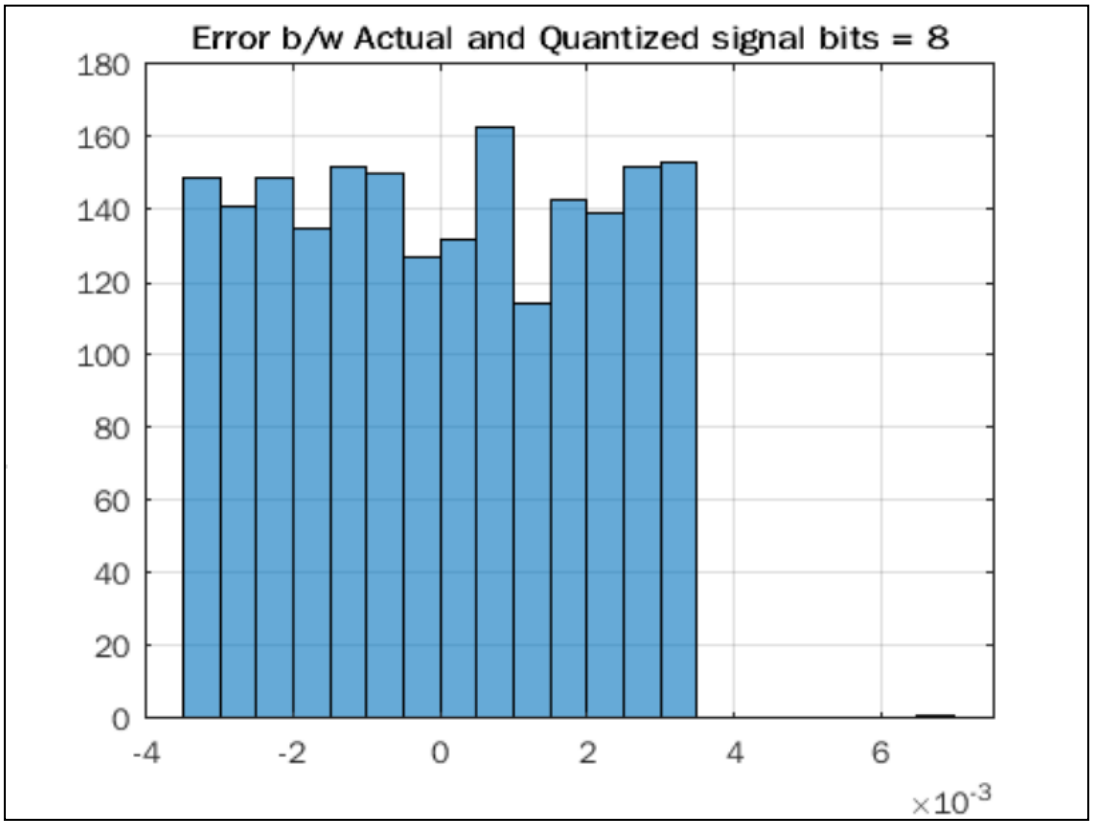
```
clc
clear all
close all

% Code for Quantization

% Random signal
a = -1; b = 1;
x_rand = a + (b-a)*rand(1,2000);
figure
stem(x_rand);
title("Input signal");
bits = [4,8];

for k = 1 : length(bits)
    % N bit processor
    N = bits(k);
    L = pow2(N);
    x_max = max(x_rand);
    x_min = min(x_rand);
    delta = (x_max - x_min)/L;

    % Quantizing the data
    quantized = zeros(1,length(x_rand));

    for i = 1 : length(x_rand)
        ind = floor((x_rand(i) - x_min)/delta);
        lower = x_min + ind * delta;
        upper = lower + delta;

        if upper - x_rand(i) < x_rand(i) - lower
            ind = ind + 1;
        end

        if ind >= L
            ind = L -1;
        end

        quantized(i) = x_min + ind * delta;
    end

    error = zeros(1,length(x_rand));

    for i = 1 : length(x_rand)
        error(i) = x_rand(i) - quantized(i);
    end
    disp("Bits = "+N);
```

```
    disp(quantized);

    figure;
    stem(quantized);
    title("Quantized values Bits = "+N);

    %Plotting histogram
    lower_bound = min(error);
    upper_bound = max(error);
    figure;
    histogram(error);
    grid on;
    title("Error b/w Actual and Quantized signal bits = "+N);

    % Calculating variance
    var = variance(error);
    disp("Varaince");
    disp(var);

end

function [y] = variance(x)
    sum = 0;
    n = length(x);

    for i = 1 : n
        sum = sum + x(i);
    end
    mean = sum/n;
    var = 0;

    for i = 1 : n
        var = var + (mean - x(i))^2;
    end
    var = var / (n-1);
    y = var;
end
```

Results

Input = 2000 signals





Variance

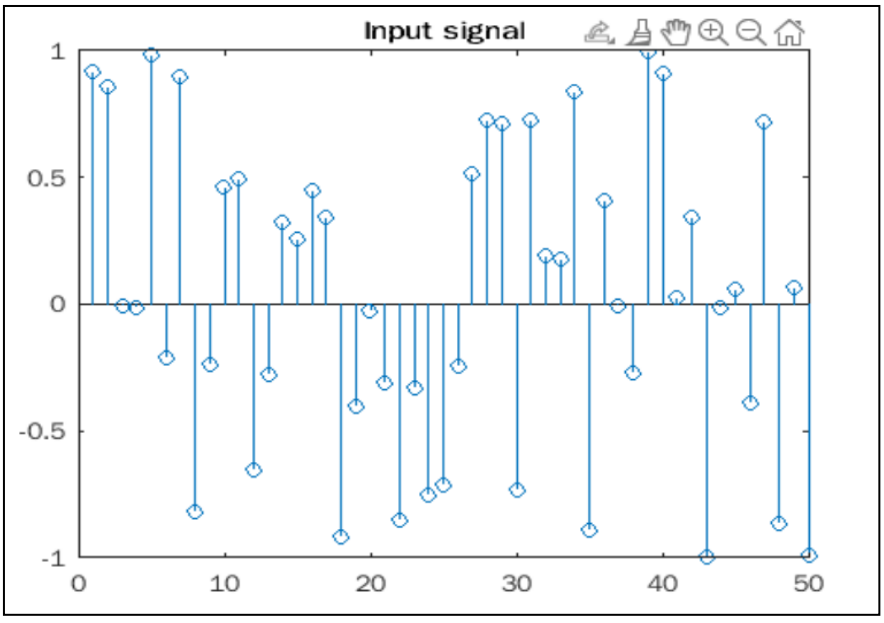
4 Bit processor

`1.062881321097460e-03`

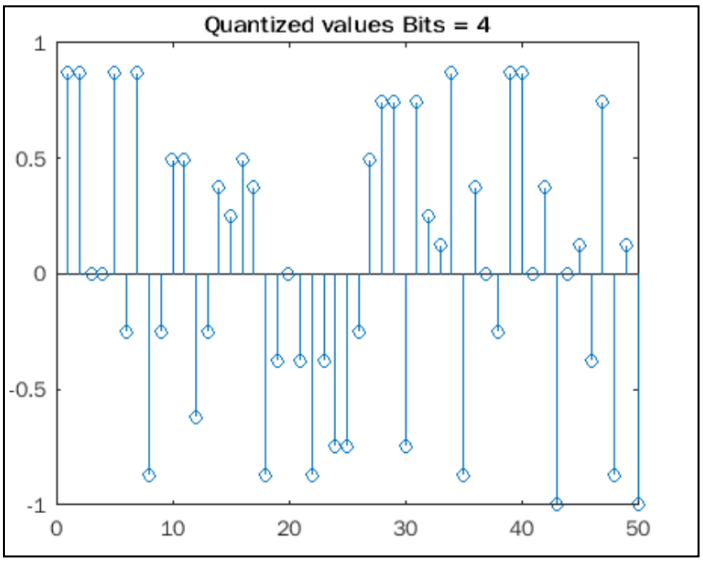
8 Bit processor

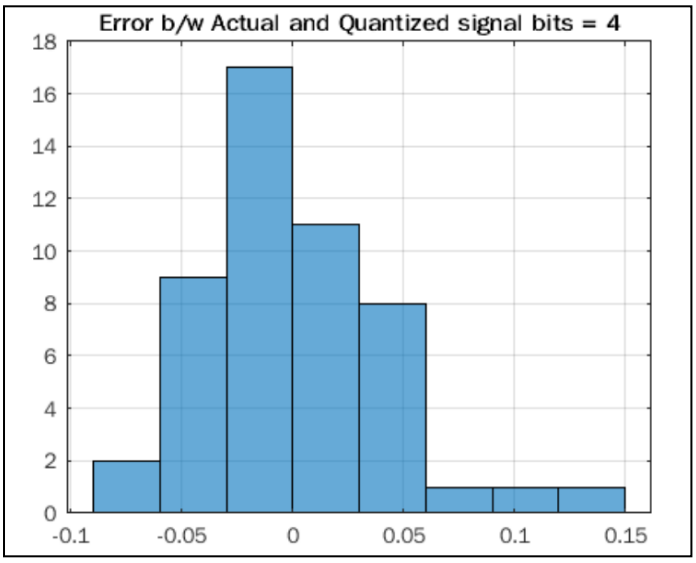
`4.206563674937122e-06`

For understanding the plots clearly let us analyse 50 random input points:



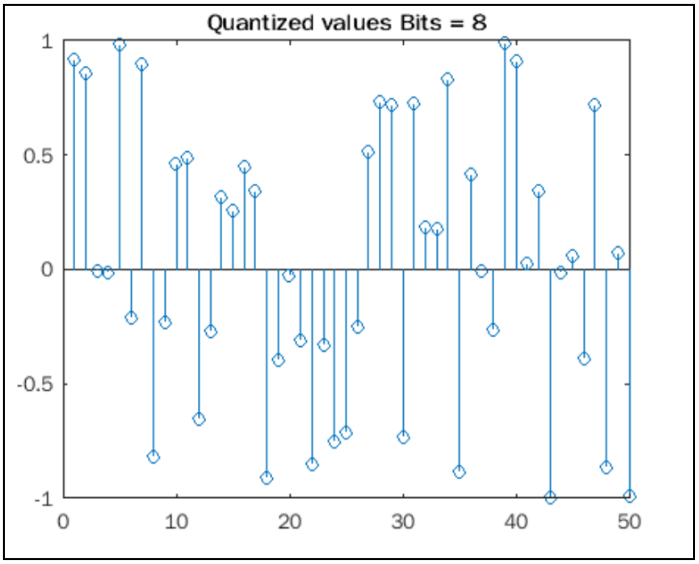
4 Bit Processor

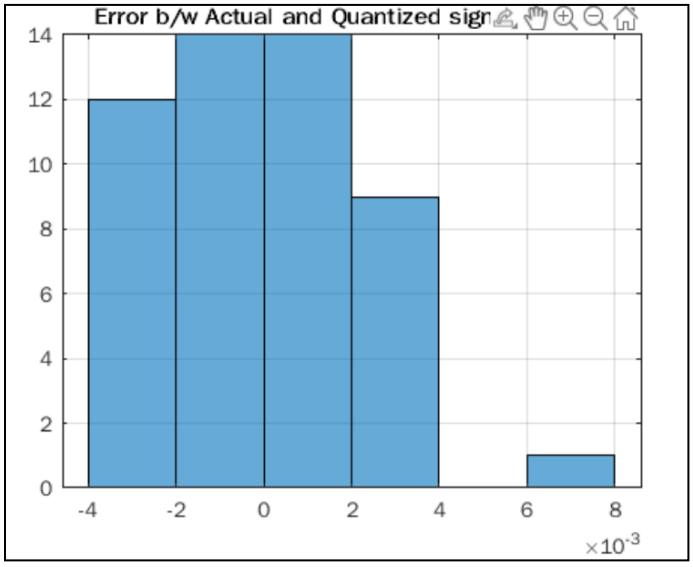





```
Variance: 1.628832921576908e-03
```

8 Bit Processor






```
Variance: 6.106799256861047e-06
```

Conclusion

Quantization is done by digitizing an analog signal by rounding them off to the nearest quantization levels. Here in the above code there are 2^N quantized levels. The minimum value of the input signal is taken as the first quantization level. The next quantization levels are derived by adding delta (max(input) – min (input) / (total quantized levels)). The quantized values are found by rounding it to the nearest quantized level. It was observed that 8Bit processor is more efficient than 4Bit because it has more number of quantized levels which results in less error and more accuracy. The variance of error signal in 8 bit processor is less compared to the 4 Bit processor.