

Dynamic Measurement and Signal Processing

Lab 2: Quantization

Guilherme Aramizo Ribeiro
PhD Student

Patcharapol Gorgitrattanakul
Teaching Assistant

Dr. Jason R Blough
Instructor

December 9, 2015

Abstract

A physical signal was measured through a data acquisition system in order to observe the effects of quantization. A deeper analysis on the quantization error was explored with simulated signals on MATLAB, verifying that the signal to noise ratio is dependent on the number of bits and input voltage range of the acquisition system.

$$R = \frac{V_{high} - V_{low}}{2^Q} \quad (1)$$

The quantization error is an inherent source of noise, and therefore, sets a minimum noise floor for any measured signal. So the Signal to Noise Ratio (SNR) will be limited by the ADC conversion precision and can be calculated as, in decibels:

$$SNR_{ADC} = 20 \log_{10} 2^Q \quad (2)$$

1 Background and Objectives

The objective of the laboratory activity is to advance the students' knowledge and practical expertise in quantization on acquired data. This section will review the background about quantization characteristics of Analog to Digital Converters (ADC) and signal to noise ratio (SNR).

As discussed by [smith], quantization relates to the amplitude of digitized signals after the acquisition. It cannot be confused with sampling, which is in turn related to the sampling time characteristics.

When it comes to quantization values, a data acquisition device is characterized by input voltage range, $\{V_{low}; V_{high}\}$, and the number of bits the acquired data is stored, Q . It follows that the minimum voltage change the system can detect, called resolution R , is calculated as

2 Apparatus

The material used in the lab activity is listed below:

- Oscilloscope 54621A;
- Data Acquisition NI cDAQ 9172;
- Data Acquisition NI 9234;
- Signal Generator AFG310;
- Desktop computer with MATLAB, LabVIEW and device drivers installed ...

3 Experimental Procedures

The lab activity contains two components: an in-lab experimental procedure and a simulation performed

with the MATLAB software. Both activities are described in detail in the lab manual, so this section describes the exceptions found in the activity.

It wasn't possible to decrease the signal generator output amplitude such that it would be mixed in the noise floor. Because the amplitude value is adjusted in discrete steps and the minimal value is well above the noise floor.

4 Data Summary

A sinusoidal signals with frequency of 356 Hz and amplitude of 1.24 V_{RMS} were acquired during the laboratory section. The time history of the experimental and MATLAB generated data was added to the Appendix, Figure 2, 3 and 4. The experiment run using a range of $\pm 5 V$ is presented in the frequency domain in the Figure 1.



Figure 1: Overlaid plot of experimental and simulated data in the frequency domain for $\pm 5 V$ of range and 8, 12 and 24 number of bits.

5 Interpretation and Analysis

Equation 1 demonstrates that a broad input range or low number of bits decrease the resolution. The time history of the simulation data confirms this statement: Figure 2 and 3 both show that the samples are closer to the nominal curve as the number of bits

increase, and this distance is smaller in the experiment with the smaller range, $\pm 5 V$. Also, voltage saturation was observed in the Figure 4, where the simulated signals saturates on $\pm 0.75 V$.

Another point from Figure 1 is that the noise floor of the experimental data is higher than the simulated 24 bits counterpart, even though the acquisition system also has a 24 bits precision. The experimental data also accounts with noises in the environment and in the DAQ electronics.

The oscilloscope's experimental SNR analysis was not possible because the signal generator couldn't reduce the sine amplitude low enough. But according to their respective data-sheet, the oscilloscope has a 8 bit resolution, while the acquisition board has 24 bits. And both systems have $\pm 5 V$ as the narrower acquisition range. So according to Equation 1, the minimum signal amplitude detectable is 19 mV and 298 ηV , respectively. This follows that the NI system has higher SNR.

Finally, in respect to the critical thinking section, for a sine wave with period of $T = 56 \mu s$ and RMS voltage of $A_{RMS} = 0.35 V$, the desired sampling rate, F_s , voltage range, L , and number of bits, Q , for the requirements described in the manual are

$$F_s = 524288 \text{ Hz} \approx 2^{\text{nextpow2}(20/T)} \quad (3)$$

$$L = \pm 1 \text{ V} \approx \pm 2 (A_{RMS} \sqrt{2}) \quad (4)$$

$$Q = 8 > \log_2 20 \quad (5)$$

being $n = \text{nextpow2}(x)$ a function that returns the next power of 2 of x , or in other words, an integer n , $2^{n-1} < x \leq 2^n$.

6 Conclusions

On this assignment a physical signal was measured through a data acquisition system in order to observe the effects of quantization. This data was also compared to simulated signals on MATLAB. Finally, the studies of acquisition parameters were explored with an example case.

From the frequency analysis of the discretized sine waves, higher SNR was observed in signals with higher number of bits and higher input voltage band usage. But experimental data has an inherent process noise floor that might be more significant than the quantization error, thus limiting the SNR to a maximum value.

Appendices

A Effects of Number of Bits and Voltage Range



Figure 2: Time history of the experimental data and quantized simulated data for a ADC input range of $\pm 5\text{ V}$



Figure 3: Time history of the experimental data and quantized simulated data for a ADC input range of ± 300 V



Figure 4: Time history of the experimental data and quantized simulated data for a ADC input range of ± 0.75 V

B MATLAB DSP Functions

```

classdef myDSP
    methods(Static)

        function [SIGNAL, nowin] = reshape(signal, win_size, overlap_p)
        %RESHAPE Reshape vector in window blocks, with overlaps
        % Example:
        % Generate data
        % Define dynamic system
        sys = rss(3, 2, 1);
        Fs = -100*min(real(eig(sys.A)));
        N = 10000;

        time = linspace(0, N/Fs, N).';
        u = reshape(repmat(randn(round(N/20),1), [1 20]).', [N 1]);
        y = lsim(sys, u, time);

        winsize = 1000;
        overlap = 0.3;
        yreshaped = myDSP.reshape(y, winsize, overlap);
        ureshaped = myDSP.reshape(u, winsize, overlap);

        win = window(@hann, winsize);
        gain = [1; 2*ones(winsize-1, 1)] / (mean(win)*winsize);

        GY = bsxfun(@times, fft(bsxfun(@times, yreshaped, win)), gain);
        Gyy = squeeze(mean(conj(GY) .* GY, 2));

        GU = bsxfun(@times, fft(bsxfun(@times, ureshaped, win)), gain);
        Guy = squeeze(mean(bsxfun(@times, conj(GU), GY), 2));

        f = linspace(0, Fs, winsize);

        figure
        subplot(311); semilogy(f, sqrt(Gyy)); grid on; grid minor
        xlim([0 Fs/2]); xlabel('frequency [Hz]')
        subplot(312); semilogy(f, sqrt(abs(Guy))); grid on; grid minor
        xlim([0 Fs/2]); xlabel('frequency [Hz]')
        subplot(313); plot(time, y); xlabel('time [s]')

        len = size(signal, 1); % signal length
        nocha = size(signal, 2); % number of channels

        if win_size > len
    
```

```

        error('Window_size_is_greater_than_signal_size')
    end
    if overlap_p > 1 || overlap_p < 0
        error('Overlap_should_range_between_0_and_1')
    end

    overlap = round(overlap_p*win_size);
    nowin = floor((len - win_size)/(win_size - overlap) + 1); % number of windows

    SIGNAL = zeros(win_size, nowin, nocha);
    for k = 1 : nocha
        % make overlap multiple of win_size, percentual to absolute
        idx = repmat((1:win_size).', [1 nowin]) + repmat((0:nowin-1)*(win_size-overlap), [1 nowin]);
        SIGNAL(:, :, k) = reshape(signal(idx, k), [win_size, nowin]);
    end
end

function Y = discretize( U, N, V )
    % Discretize vector U on the range +-V into N bits

    U = uencode( U, N, V, 'signed' );

    U = cast( U, 'double' );
    N = cast( N, 'double' );
    V = cast( V, 'double' );

    Y = interp1( [ -2^(N-1) 2^(N-1)-1 ], [-V V], U );
end

end

end
end

```