

DSP

Lab 1 Sampling and Quantization

A. Objectives:

- Introduction to Matlab features for signal analysis.
- Apply Matlab to examine relationship between quantization bits and signal-to-noise ratios (SNR).
- Apply LabVIEW to examine aliasing
- Apply Matlab to investigate non-ideal low-pass filtering for reconstructing a signal from its samples.

B. Background

This laboratory exercise focuses on the relationship between noise and interference resulting from digitizing an analog signal. Digitization results in 2 types of noise or distortion. The first is called **aliasing** and results from sampling a continuous-time (CT) signal at discrete points in time. If the signal is not sampled at a rate higher than twice its highest frequency, then aliasing occurs. The second type of noise results from quantizing or rounding-off the sample amplitudes to discrete levels. This rounding error is typically modeled as an additive noise process and is called *quantization noise*.

Sampling and Aliasing:

A mathematical representation of the sampling operation is given by a sequence of delta Dirac impulse functions in continuous time:

$$s(t) = \sum_{n=-\infty}^{\infty} \delta(t - nT), \quad (1)$$

where the impulses are separated by T seconds (sampling interval) corresponding to a sampling frequency of $F_s = \frac{1}{T}$ Hz. The sampling function $s(t)$ is illustrated in Fig. 1.

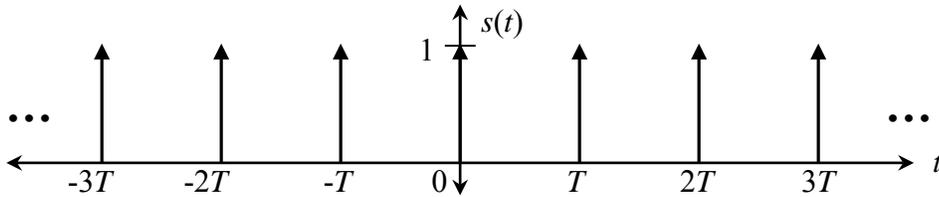


Figure 1. Impulse train to model sampling operation.

It can be shown that the Fourier Transform of $s(t)$ is also an impulse train in the frequency domain with impulses separated by the sampling frequency:

$$\hat{S}(f) = \frac{1}{T} \sum_{k=-\infty}^{\infty} \delta(f - kF_s) \quad (2)$$

This impulse train is illustrated in Fig. 2.

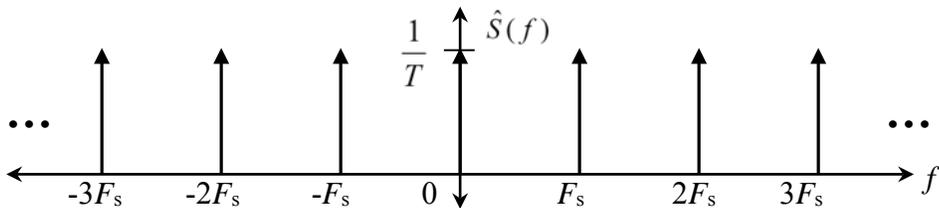


Figure 2. Fourier transform of sampling impulse train that creates aliased spectra.

Now consider a CT signal, $x_c(t)$, that is bandlimited to F_N :

$$X_c(f) = 0 \quad \text{for } |f| > F_N \quad (3)$$

where F_N is referred to as the bandlimit. As an example, consider the Fourier Transform of bandlimited signal presented in Fig. 3.

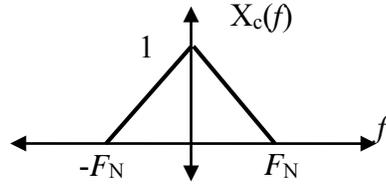


Figure 3. Example of the Fourier Transform of a bandlimited continuous-time signal.

Consider sampling $x_c(t)$ by multiplying it with the impulse train of Eq. (1), which effectively zeroes out the information between sampling points:

$$x_s(t) = x_c(t) \sum_{n=-\infty}^{\infty} \delta(t - nT). \quad (4)$$

Now for a linear time-invariant (LTI) system, multiplication in the time domain corresponds to convolution in the frequency domain, given by:

$$\hat{X}_s(f) = \hat{X}_c(f) * \frac{1}{T} \sum_{k=-\infty}^{\infty} \delta(f - kF_s) = \frac{1}{T} \int_{-\infty}^{\infty} \hat{X}_c(f - \lambda) \sum_{k=-\infty}^{\infty} \delta(\lambda - kF_s) d\lambda. \quad (5)$$

Since the convolution of a signal with a shifted delta Dirac function, results in a shifted version of the signal, Eq. (5) shows that sampling replicates the original signal spectrum along the frequency axis shifted by integer multiples of the sampling frequency F_s . The convolution integral of Eq. (5) reduces to:

$$\hat{X}_s(f) = \frac{1}{T} \sum_{k=-\infty}^{\infty} \hat{X}_c(f - kF_s) \quad (6)$$

The aliasing described in Eq. (6) is illustrated in Fig. 4 for two cases. Figure 4a shows the case when the sampling frequency is greater than twice F_N . Figure 4b shows the case when the sampling frequency is less than twice F_N . Note in the latter case the spectrum interferes with images of itself, thereby distorting the information in the overlapping spectral regions.

The **Nyquist frequency** is defined as half of the sampling frequency of a digital processing system. This is also referred to as the **folding frequency**, since frequencies beyond this value fold back onto the non-aliased spectral range. Note that when the sampling theorem conditions are met (*i.e.* Nyquist frequency is greater than F_N), the aliased spectra do not interfere with each other (as illustrated in Fig. 4a) and the original signal can be recovered from its samples with an idea low-pass filter. When overlap between aliased spectra occurs, the interference results in an irreversible noise/distortion process and the original signal cannot be recovered (as illustrated in Fig. 4b). Aliasing in this case irreversibly degrades the signal. Bandlimit F_N is a property of the continuous-time signal and is sometimes referred to as the Nyquist rate. Note: The Nyquist frequency is a property of the processing system while Nyquist rate is a property of the continuous-time signal.

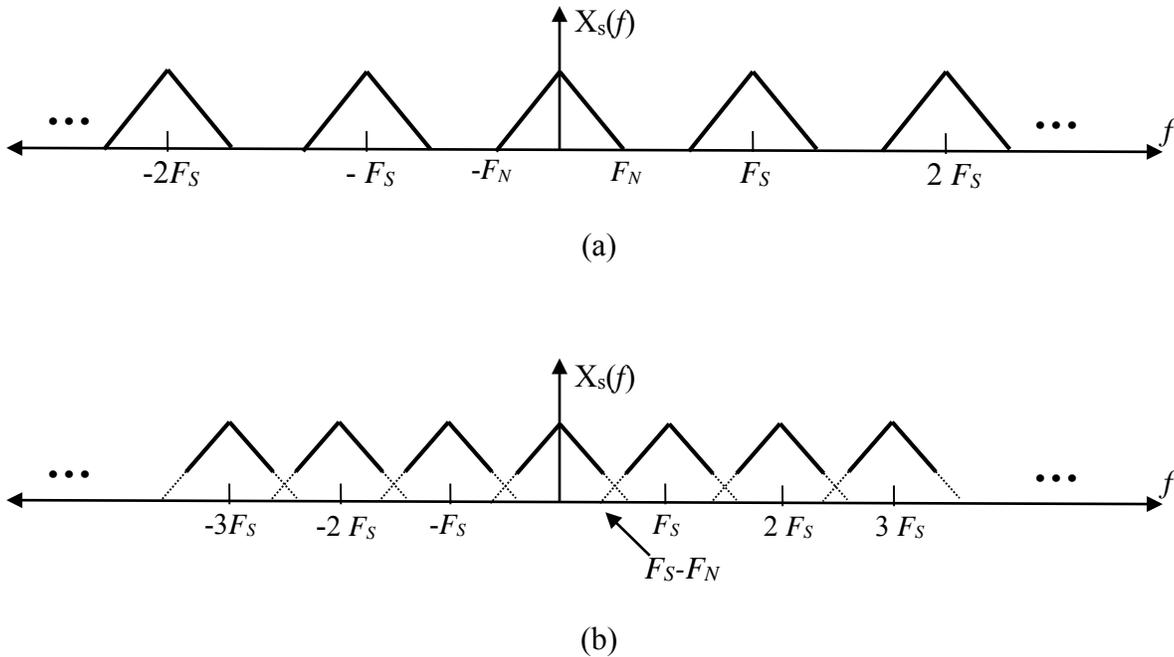


Figure 4. Aliased spectra when sampling rate is (a) greater than twice the bandwidth; $F_s > 2F_N$, (b) less than twice the bandwidth, $F_s < 2F_N$.

The derivation leading to Fig. 4 is the basis for the **sampling theorem, which states that a bandlimited signal can be recovered from its samples if it is sampled at a rate greater than twice its highest frequency**. The recovery can be performed with ideal low-pass filtering or sinc function interpolation. Note in Fig. 4a an ideal low-pass filter can be applied with a cut-off anywhere between F_N and $F_s - F_N$. This way the original spectrum is undistorted and aliased spectra are eliminated. For the sampled signal of Fig. 4b, this filtering is not possible because the original spectrum overlaps with the aliased spectra. A low-pass cutoff frequency does not exist to eliminate the aliased spectra without distorting the original spectrum.

Quantization and Uniformly Distributed Additive Noise.

A discrete-time signal cannot be stored on a computer, since it has continuous amplitudes. Therefore, the amplitudes must also be converted to discrete levels to be stored as binary words in a computer memory. Figure 5 illustrates the quantization process.

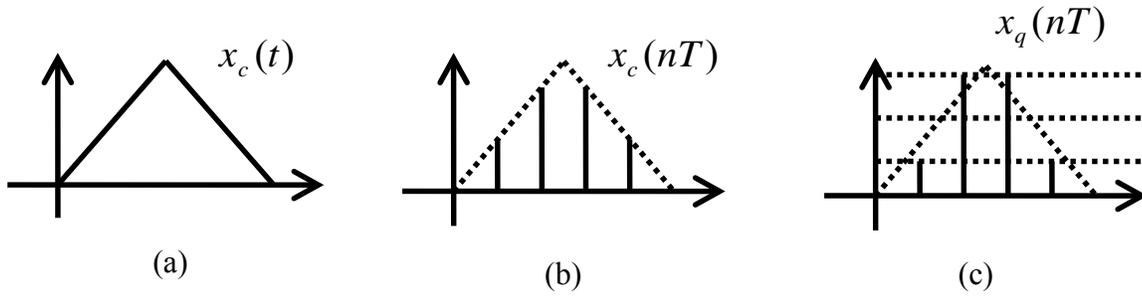


Figure 5. (a) CT signal, (b) corresponding discrete-time signal, (c) digital signal quantized to 4 levels corresponding to a 2 bit word per sample.

For signals that are densely quantized, the error from round-off to the nearest quantization level is effectively a random noise process. The error can be modeled by subtracting the distance from the discrete-time signal amplitude to the nearest quantization level, given by:

$$n_q(nT) = x_c(nT) - x_q(nT), \quad (7)$$

where $x_q(nT)$ is the signal value after rounding off to a quantization level.

Noise processes are modeled as random variables (RV) with probability distributions to describe their amplitude variations. If noise samples are uncorrelated from sample to sample, the process is called **white noise** and has a flat average spectrum (equal energy is expected at all frequencies). The noise amplitudes are modeled with normal distributions for many natural processes. However, quantization noise is typically modeled with a uniform distribution, since the amplitudes of n_q are expected with equal probability between the quantization levels. Let Δ denote the distance between quantization levels. Then the probability density function for RV n_q is given as:

$$p(n_q) = \begin{cases} \frac{1}{\Delta} & \text{for } -\frac{\Delta}{2} < n_q < \frac{\Delta}{2} \\ 0 & \text{Elsewhere} \end{cases} \quad (8)$$

Noise and signal relationships are often described in terms of their power ratios. For a noise process modeled as a zero-mean RV, its power is the same as its variance. In the case of the uniform distribution, the variance or power is given by:

$$\sigma_n^2 = \frac{\Delta^2}{12} \quad (9)$$

and the rms value is equivalent to its standard deviation (i.e. square root of expression in Eq. (9)).

In a classic derivation of the SNR from quantization noise, the signal is assumed to be a sinusoid with amplitude A such that it extends from the highest to lowest quantization levels. Given the signal

amplitude extends to the full-scale range of the quantizer ($2A$), Δ can be expressed in terms of A for a specified number of bits (R) used for the signal amplitude representation. The SNR for a quantized full-range sinusoid in dB is given as:

$$SNR_q(R) = 1.7609 + 20 \log_{10}(2^R - 1) \text{ dB}. \quad (10)$$

For large R (i.e. greater than 4), the minus 1 is ignored and Eq. (10) is often approximated as:

$$SNR_q(R) \approx 1.7609 + 6R \text{ dB}, \quad (11)$$

which leads to an often quoted rule-of-thumb that 6dB SNR is gained for each additional bit used to encode (quantize) the sampled signal. Note that the SNR_q formula for Eq. (10) only applies to a sinusoid that fills the full range of the quantizer. If the sinusoid does not extend from the highest to lowest quantization level, then the SNR values will be less. This idea is developed in one of the pre-lab exercises.

C. Pre-Laboratory Assignment

1. Consider quantizing the signal $x(t) = 3.5 \cos(2\pi t)$ Volts.
 - a) For one full period, sketch the results of quantizing $x(t)$ uniformly with a 3 bit word, where the highest quantization level is set to 3 volts and the lowest level is set to -4 volts. What is Δ (from Eq. (8)) in this case?
 - b) For one full period, sketch the results of quantizing $x(t)$ with 1 bit, where one quantization level is set to -1.75 volts and the other to 1.75 volts. Would you expect to hear/see quantization *noise* or distortion if a signal was quantized in this way?
2. Consider an 8-bit quantizer with a Full-Scale range from -8 to +8 volts.
 - a) Compute the SNR_q for a non-aliased sinusoid with amplitude 1 volt.
 - b) Describe what occurs when a sinusoid with a 10 volt amplitude is presented to this quantizer. Can the overload error (clipping) best be described as noise or distortion? Provide a reason for your answer.
3. Consider a system with a sampling rate of $f_s = 500$ Hz. Sketch the DTFT magnitudes for each of the CT signals below after sampling over the frequency range of 0 to f_s . Be sure to explicitly label the frequency axis to denote the sinusoids position on the axis.
 - (a) $s(t) = 5 \sin(2\pi 200t)$
 - (b) $s(t) = 5 \sin(2\pi 600t)$
 - (c) $s(t) = 5 \sin(2\pi 1250t)$
 - (d) $s(t) = 5 \sin(2\pi 256t)$

$$(e) \quad t \approx 5 \sin(2\pi 998t)$$

4. Study the scripts *quant_lab1.m* and *sincinterp_lab1.m* to understand what they do and what are the critical parameters that can be changed for the purpose of observing different cases and settings.

D. Laboratory Assignment

This assignment requires the use and modification of a Matlab script called *quant_lab1.m* downloadable from the course materials website: http://kstio.com/index.php?p=1_4_digital-signal-processing

Go to the *mfile* link and save the *quant_lab1.m* mfile file to your computer. Open these files in Matlab's editor to study the comments and modify the code to do the exercises. This also requires additional data files to be downloaded from the course materials website under the *data* link.

Quantization

This is the only part of the lab that will require a written component. The report should have an **abstract, introduction, experimental procedure, results and analysis, and conclusion.**

1. Use the Matlab script *quant_lab1* to create a 2-second sinusoid with unit amplitude, frequency 523.2 Hz, sampled at 16kHz, and quantize the signal using values of "R" bits ranging from 2 to 16.
 - (a) Design an experiment to determine the maximum SNR level required for noise to be heard by humans. Create a table with each entry (row) containing the number of quantization bits, the theoretical SNR from Eq. (10), the experimental SNR computed in the *quant_lab1* script, and an indication of whether quantization noise/distortion was heard (use the *sound* function in Matlab to replay sounds if needed). Decide on a consistent way to determine when the quantization noise is heard and when distortion is heard, i.e. keep the volume constant for all trials, stay approximately the same distance from the speakers (or use headphones). If using multiple people to make a judgment, create appropriate rules (i.e. decide noise was heard if a majority hear it). Don't vary procedures between experiments or quantization bit values or else you will add unnecessary variability into the comparisons and skew results. Also come up with a description of how the sound of noise differs from the sound of distortion that you used in making a judgement.
 - (b) Copy a figure of the average power spectra for a case when the quantization noise was clearly heard and paste to a file so it can be put in the results section of your lab report. Do the same for when it was decided that distortion was heard. **In the discussion section address the question, "The points at which noise and distortion were heard, were there are characteristics patterns in the spectrum that correspond to this?"**
2. Repeat Exercise 1 using a sine wave with frequency 500 Hz. **In the discussion section, describe and explain observed differences with part 1.**

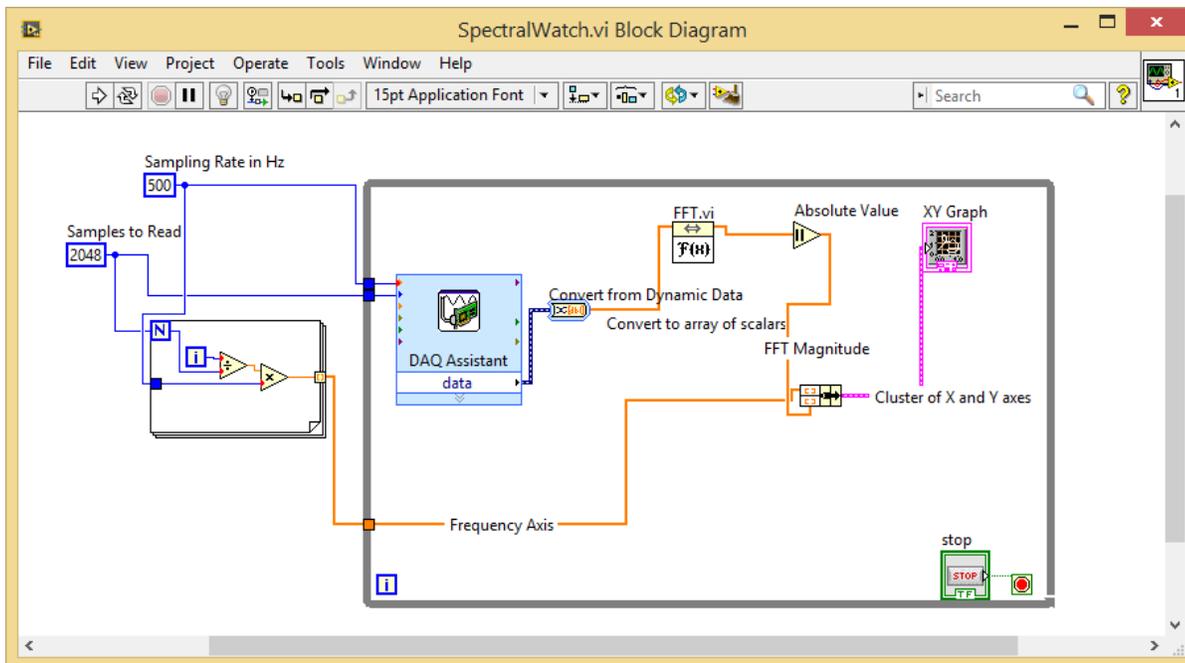
- Repeat Exercise 1 using wave file *mozart-1.wav*. In the discussion section, describe and explain observed differences with part 1.

In the discussion section for preceding Exercises 1, 2, and 3, compare the 3 plots for the average spectra and comment on differences/similarities observed for the quantization noise and signals. Also comment on any differences when the quantization noise became audible or changed in nature/timbre (i.e. not just louder noise, but a change in the nature of sound).

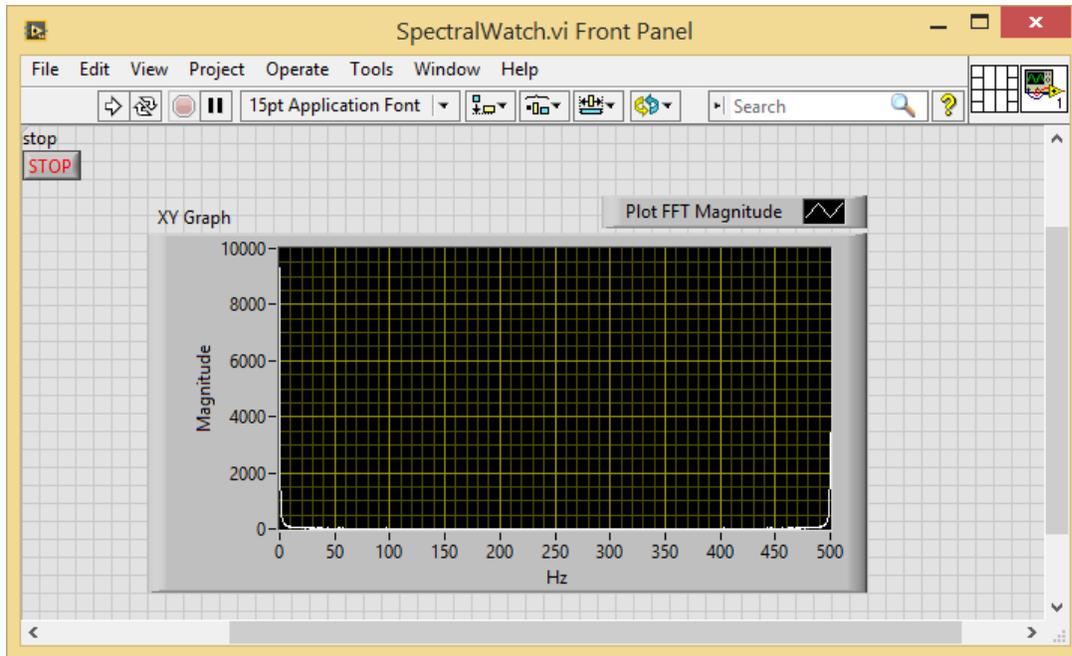
Aliasing

No write-up is required here, just demo to TA

- Build the VI in Figure 6 to read data from the DAQ and display the FFT magnitudes. Connect the function generator to the DAQ input and verify the results computed in the pre-laboratory assignment # 3. You can copy and paste the plot of the graphs for your own record (to copy a graph in LabVIEW, right click on the plot and select “export simple image to clipboard”). Show the plot for the signal of pre-lab problem 3(e) to the TA/Instructor to verify you did the work correctly.



(a)



(b)

Figure 6. VI for computing and plotting FFT Magnitudes from 0 to the sampling frequency. a) the block diagram, b) the control panel with an XY graph.

Reconstruction from discrete samples

No write-up is required here, just demonstrate to the TA

5. In Matlab run the *sincinterp_lab1* script. Observe the difference between the “original” signals and those reconstructed from its samples with a low-pass filter (i.e. convolution with a sinc function or equivalently, sinc function interpolation). Write another script to find the mean square error (MSE) between the reconstructed signal and the “original” signal. Rank the signals based on the errors from smallest to largest. Then compute the FFT magnitudes of each of the “original” signals (convert to dB) and plot them on the same graph for comparison (can just plot the first half of the spectrum, since last half will be mirror image of first). What do you observe in the spectra that would suggest the ranking of the signals based on the MSE reconstruction error? Just show the spectral plots and MSE values to the TA/Instructor and make your arguments orally.