1. Introduction

The Nyquist sampling theorem suggests that a continuous-time signal must be sampled at a rate that is at least twice that of its fundamental frequency in order for that signal to be fully reconstructed from its samples. This minimum frequency is referred to as the Nyquist frequency or rate. Here, full reconstruction refers to the interpolation of the signal between its samples with total accuracy relative to the original continuous-time signal. The fundamental frequency is defined as the lowest multiple of the frequencies at which a periodic signal oscillates.

Undersampling, oversampling, and critical sampling refer to the sampling of a continuous-time signal above, below, and at the Nyquist frequency, respectively. The sampled signal will suffer from aliasing when undersampled. Aliasing refers to the phenomenon that occurs when the interpolation between samples does not yield the original continuous-time signal as a result of the samples being too far apart in time.

When observed in the frequency domain, a sampled signal appears as a periodic version of its own frequency spectrum. When aliasing occurs, each of these periods overlaps with another, as will be demonstrated in this project.

2. Description of Experiments

The goal of this project is to demonstrate the Nyquist sampling theorem for a continuous-time sinusoidal signal of unit amplitude. The function, **nyquist_demo(function_type, bandwidth)**, accepts two arguments:

• function_type (string) specifies the type of time-domain function for which the Nyquist rate is demonstrated; can be 'sine' or 'sinc'
• bandwidth (integer) specifies the bandwidth of the time-domain function in kHz

The first experiment will demonstrate the cases of oversampling, critical sampling, and undersampling for a sine function. The second experiment will repeat the first for a sinc function.

Experiment 1:

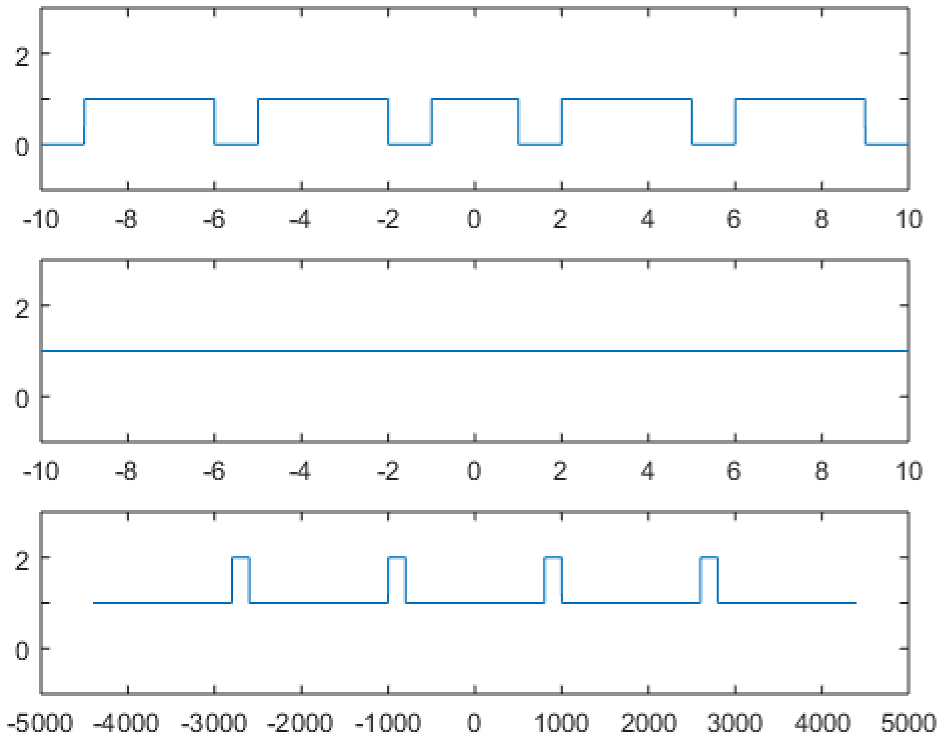
The range frequencies of the spectrum that will be plotted are defined by f_range and f. They are set to a length at which five total instances of the sine wave's frequency spectrum will be visible. Because the sine wave's amplitude is expected to be unity and its frequency spectrum is expected to be a rectangle, the frequency spectra defined by the matrices to be plotted are concatenations of ones produced by the ones( ) function in MATLAB.

In the oversampled matrix, zeros are placed between each square to show that the frequency spectra do not overlap, and thus no aliasing occurs. Similarly, no aliasing occurs in the case of critical sampling (the matrix named critsampling), where no zeros appear because the frequency spectra lie at the boundary of overlap. In the case of undersampling, aliasing is observed in the overlap of the frequency spectra, where the overlap cause two instances of the frequency spectrum to add to an amplitude of 2.

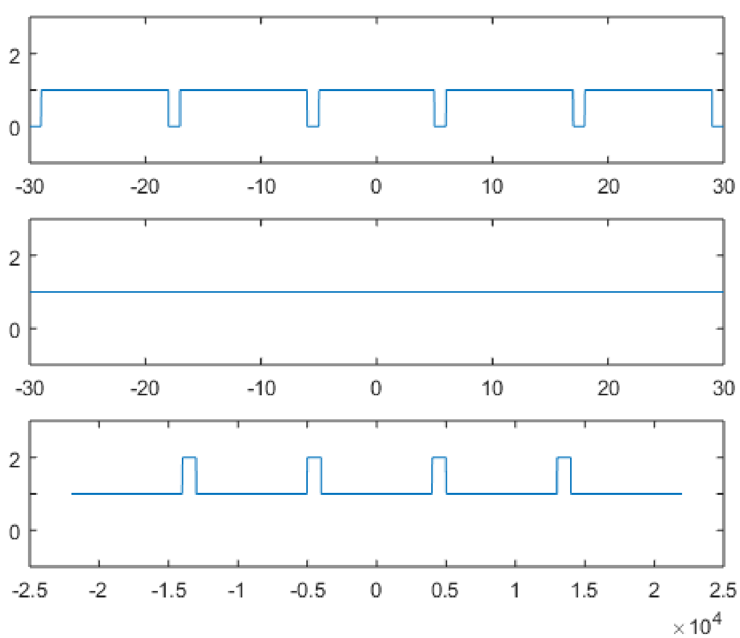
See the end of this section for the MATLAB scripts.

Each figure shows (from top to bottom) the oversampled, critically sampled, and undersampled signal in the frequency domain. The figures demonstrate the experiment for 1 kHz, 5 kHz, 10 kHz, and 100 kHz sinusoids.

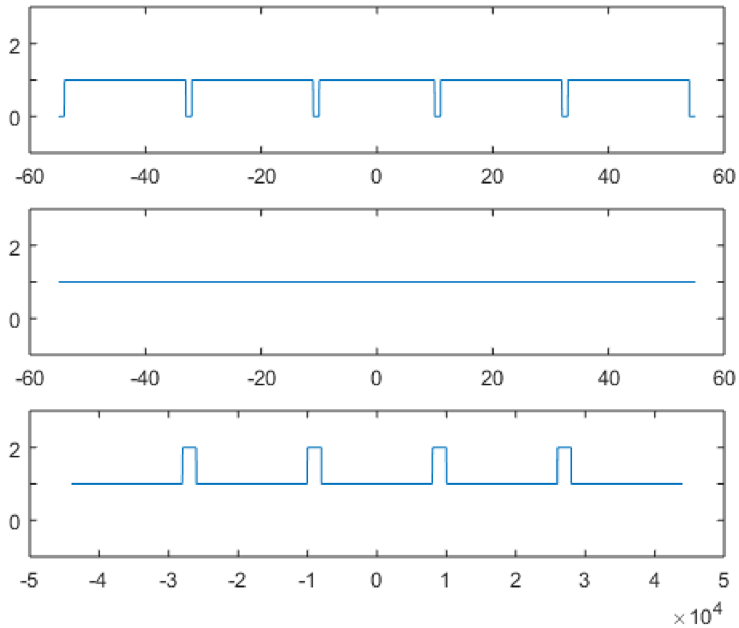
Frequency Spectrum of 1 kHz Sine Wave Sampled Above, Below, and At Nyquist Rate:



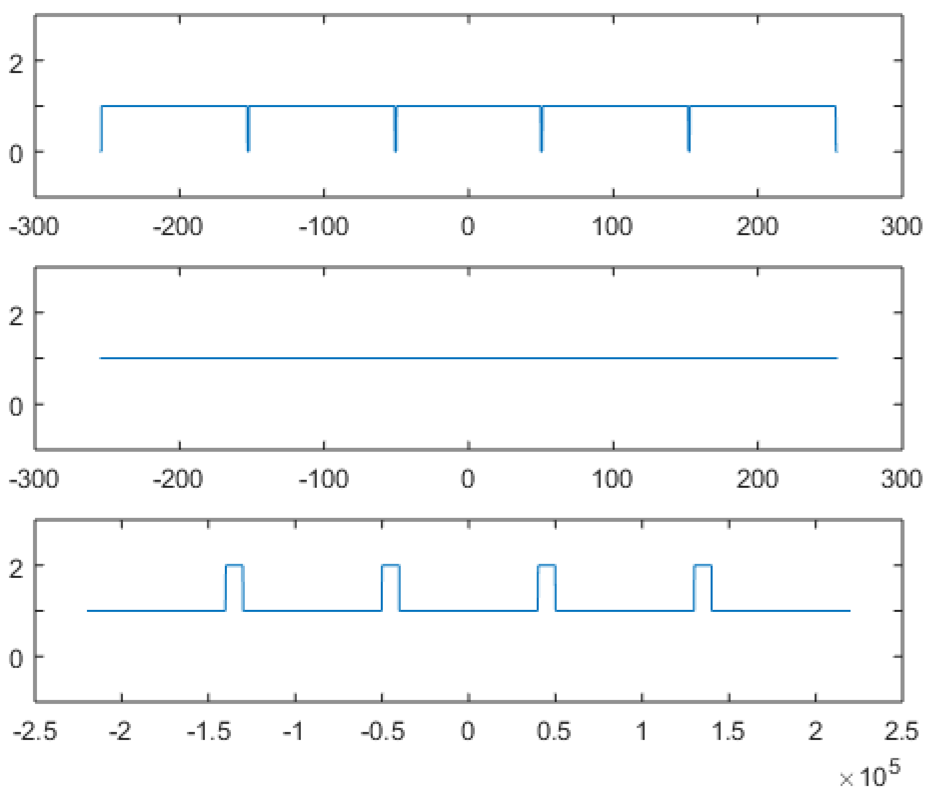
Frequency Spectrum of 5 kHz Sine Wave Sampled Above, Below, and At Nyquist Rate:



Frequency Spectrum of 10 kHz Sine Wave Sampled Above, Below, and At Nyquist Rate:



Frequency Spectrum of 50 kHz Sine Wave Sampled Above, Below, and At Nyquist Rate:

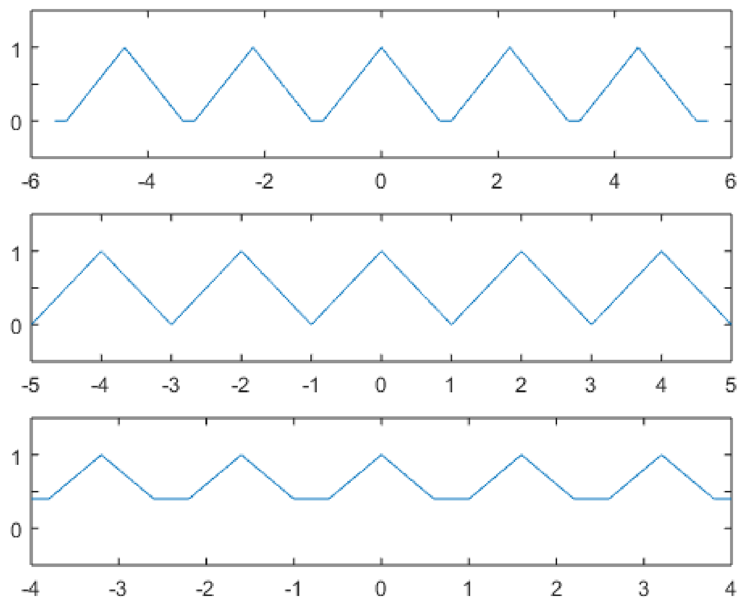


Experiment 2:

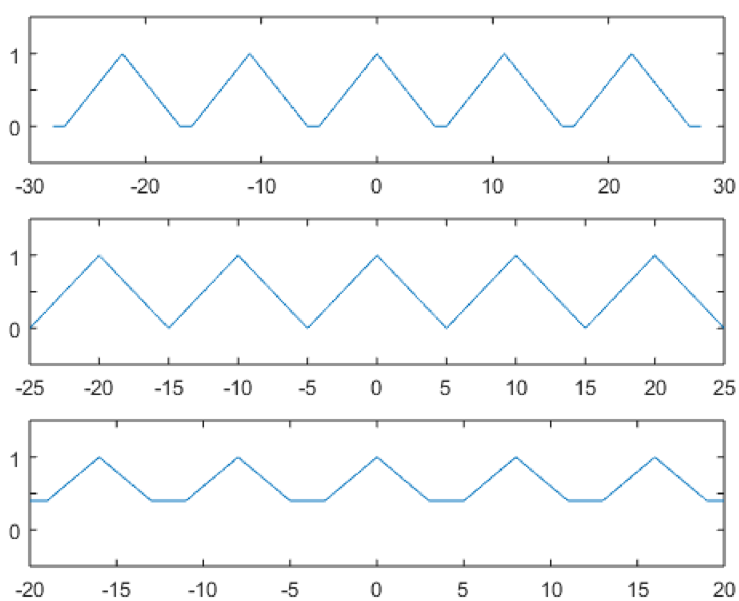
Because the sinc wave's amplitude is expected to be unity and its frequency spectrum is expected to be a triangle, the frequency spectra defined by the matrices to be plotted are concatenations of up ramps and down ramps. Once again, the oversampled case shows no overlap, the critically sampled case shows no overlap as well as no zeros between instances of the spectrum, and the undersampled case shows overlap, causing the triangles to be truncated and nonzero regions to appear between them.

The figures show (from top to bottom) the oversampled, critically sampled, and undersampled signal in the frequency domain.

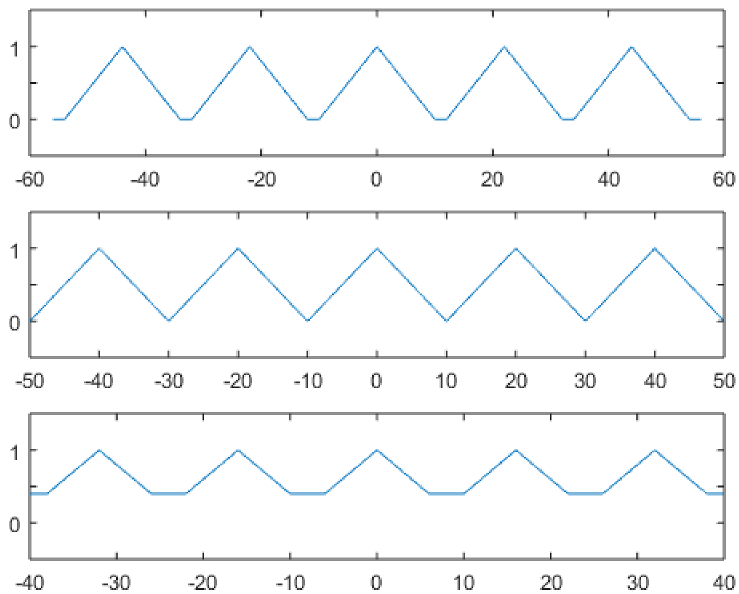
Frequency Spectrum of 1 kHz Sinc Wave Sampled Above, Below, and At Nyquist Rate:



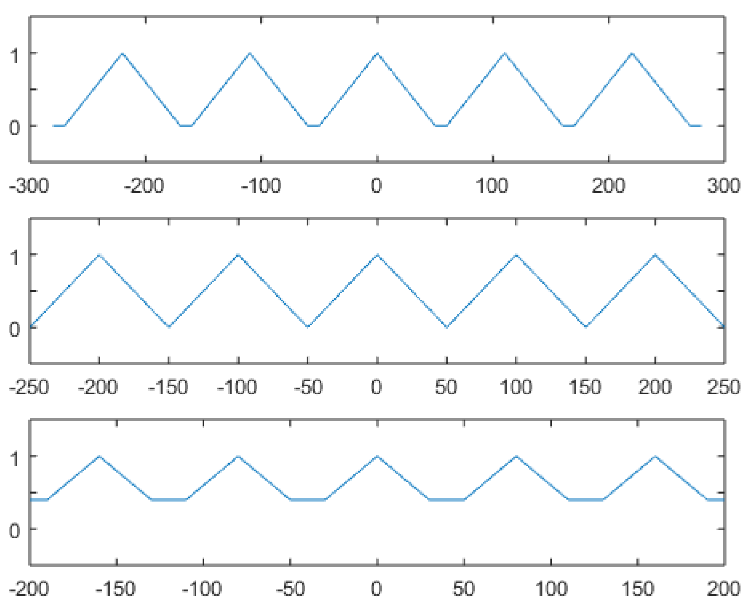
Frequency Spectrum of 5 kHz Sinc Wave Sampled Above, Below, and At Nyquist Rate:



Frequency Spectrum of 10 kHz Sinc Wave Sampled Above, Below, and At Nyquist Rate:



Frequency Spectrum of 50 kHz Sinc Wave Sampled Above, Below, and At Nyquist Rate:



The MATLAB script for the function that demonstrates the Nyquist sampling theorem is shown below.

```
function [] = nyquist_demo(function_type, bandwidth)
    % function_type (string): the type of time-domain function for which the
    % Nyquist rate is demonstrated; can be 'sine' or 'sinc'

    % bandwidth (integer): the bandwidth of the time-domain function in kHz

    nyquist_rate = 2*bandwidth;

    if strcmp(function_type, 'sine')
        f_range = 5*bandwidth + 5;
        f = -1*f_range:0.001:f_range;

        square = ones(1, 1000*(nyquist_rate + 1));
        oversampled = [zeros(1, 1000), square, zeros(1, 1000), square,...
            zeros(1, 1000), ones(1, 1000*bandwidth), 1,...
            ones(1, 1000*bandwidth), zeros(1, 1000), square,...
            zeros(1, 1000), square, zeros(1, 1000)];
        critsampled = ones(1, length(f));

        undersampled = [ones(1, 1000*(2*bandwidth*(1 - 0.2))), 2*ones(1, 1000*0.2*bandwidth),
ones(1, 1000*(2*bandwidth*(1 - 0.2))), 2*ones(1, 1000*0.2*bandwidth), ones(1, 1 +
1000*(2*bandwidth*(1 - 0.2))), 2*ones(1, 1000*0.2*bandwidth), ones(1, 1000*(2*bandwidth*(1 -
0.2))), 2*ones(1, 1000*0.2*bandwidth), ones(1, 1000*(2*bandwidth*(1 - 0.2)))];
        f_undersampled = -1000*bandwidth*(5 - 3*0.2) : 1000*bandwidth*(5 - 3*0.2);

        figure
        subplot(3, 1, 1)
        plot(f, oversampled)
        set(gca, 'YLim', [-1 3])
        subplot(3, 1, 2)
        plot(f, critsampled)
        set(gca, 'YLim', [-1 3])
        subplot(3, 1, 3)
        plot(f_undersampled, undersampled)
        set(gca, 'YLim', [-1 3])

    elseif strcmp(function_type, 'sinc')
        down_ramp = 1 : -1/1000 : 0;
        up_ramp = 0 : 1/1000 : 1;
        triangle = [up_ramp(1:1000), down_ramp(1:1000)];

        x_oversampled = -1*bandwidth*(5 + 3*0.2) : bandwidth/1000 : bandwidth*(5 + 3*0.2);
        x_critsampled = -1*bandwidth*(5) : bandwidth/1000 : bandwidth*(5);
        x_undersampled = -1*bandwidth*(5 - 5*0.2) : bandwidth/1000 : bandwidth*(5 - 5*0.2);

        oversampled = [zeros(1, 200), triangle, zeros(1, 200), triangle, zeros(1, 200),
up_ramp(1:1000), 1, down_ramp(1:1000), zeros(1, 200), triangle, zeros(1, 200), triangle, zeros(1,
```

```
200)];
        critsampled = [triangle, triangle, up_ramp(1:1000), 1, down_ramp(1:1000), triangle,
triangle];
        undersampled = [0.4*ones(1, 200), triangle(401:1600), 0.4*ones(1, 400),
triangle(401:1600), 0.4*ones(1, 400), up_ramp(401:1000), 1, down_ramp(1:600), 0.4*ones(1, 400),
triangle(401:1600), 0.4*ones(1, 400), triangle(401:1600), 0.4*ones(1, 200)];

        figure
        subplot(3, 1, 1);
        plot(x_oversampled, oversampled);
        set(gca, 'YLim', [-0.5 1.5])
        subplot(3, 1, 2);
        plot(x_critsampled, critsampled);
        set(gca, 'YLim', [-0.5 1.5])
        subplot(3, 1, 3);
        plot(x_undersampled, undersampled);
        set(gca, 'YLim', [-0.5 1.5])
    end
end
```

The script for performing the described experiments and generating the figures is shown below.

```
nyquist_demo('sine', 1)
nyquist_demo('sine', 5)
nyquist_demo('sine', 10)
nyquist_demo('sine', 50)
nyquist_demo('sinc', 1)
nyquist_demo('sinc', 5)
nyquist_demo('sinc', 10)
nyquist_demo('sinc', 50)
```

3. Summary

In this project, the Nyquist sampling theorem was demonstrated for continuous-time sine waves and sinc waves of unity amplitude. The MATLAB function that performed the demonstration was tested using waves at 1 kHz, 5 kHz, 10 kHz, and 50 kHz. The frequency spectra of the sampled signals were plotted as the sampling frequency was varied to be above, equal to, or below the Nyquist sampling frequency of each input signal.

4. Conclusions

The frequency spectrum of a sampled signal is the spectrum of that signal repeated at intervals equal to the sampling rate. For this reason, the minimum rate at which a signal must be sampled is twice that of its fundamental frequency in order to avoid the overlap of repeated instances of the spectrum. The overlap represents aliasing, or the loss of information encoded by signal due to the inability of a reconstruction filter to interpolate the original values of the signal between the samples. When sampled at the Nyquist frequency, the samples of a continuous-time signal can be used to recover the original continuous-time signal without loss of information by interpolating between samples in the time-domain or, equivalently, extracting a single instance of the signal’s frequency spectrum by applying a lowpass filter in the frequency domain. It is clear through the figures plotted in these experiments that the lowpass filter would be unable to extract the spectrum if the sampling rate falls below the Nyquist rate, at which point the repeated instances overlap and the filter would fail to isolate a single instance.