Homework 3: Serial Communication

Nathan Phipps

Important notes about codes:
Due to this assignment **taking over 60 hours to complete**, I could not neither optimize my comments section in the code nor could I adapt codes to change from dynamic input specifically from nyquist frequencies. My codes do take into account frequency, but I have 3 separate codes for 3 separate nyquist rates that I chose, each of which needs to be commented in and out respectively when desired by the user. Since it is **not required** to have code that dynamically adapts to my nyquist frequency, my python code has 3 sections, each of which are at separate nyquist frequencies in python. I tried adapting this section, but could not adapt it, and it has taken too much time; therefore since it is not required, I will not continue to try to implement such a feature. Therefore, my main python code is at nyq = 100 with the other two nyq rate codes commented out in python. If you change your nyq rate in python, you will have to do it in arduino as well, but again the rubric didn't ask for the user to input nyq, they asked for the programmer to test separate frequencies, so unless altered, the codes are locked in at one specific nyquist rate; while I tested separate trials using **manual** input where nyquist frequency dependent code blocks were needed in order to obtain plots for the report section.

Report
In this report I examined a low pass filter at 3 different nyquist sampling rates. In examining these sampling rates, the end goal seems to be to produce a signal closest to our original signal by filtering a noisy version of that signal using a low pass filter. To determine the error between my signals I intended to use root mean square and percent error calculations between the root mean square of a filtered signal and a clean signal to visualize how close the filter could get to returning to the original reference signal. I also intended to do this same comparison between the clean sine and noise and the noise and filtered signals. However, my code would not implement properly to determine these errors. Attached at the end of the document is the calculations and equations I tried but failed to implement, a simpler form of these equations can be found in my python code, but is non functional

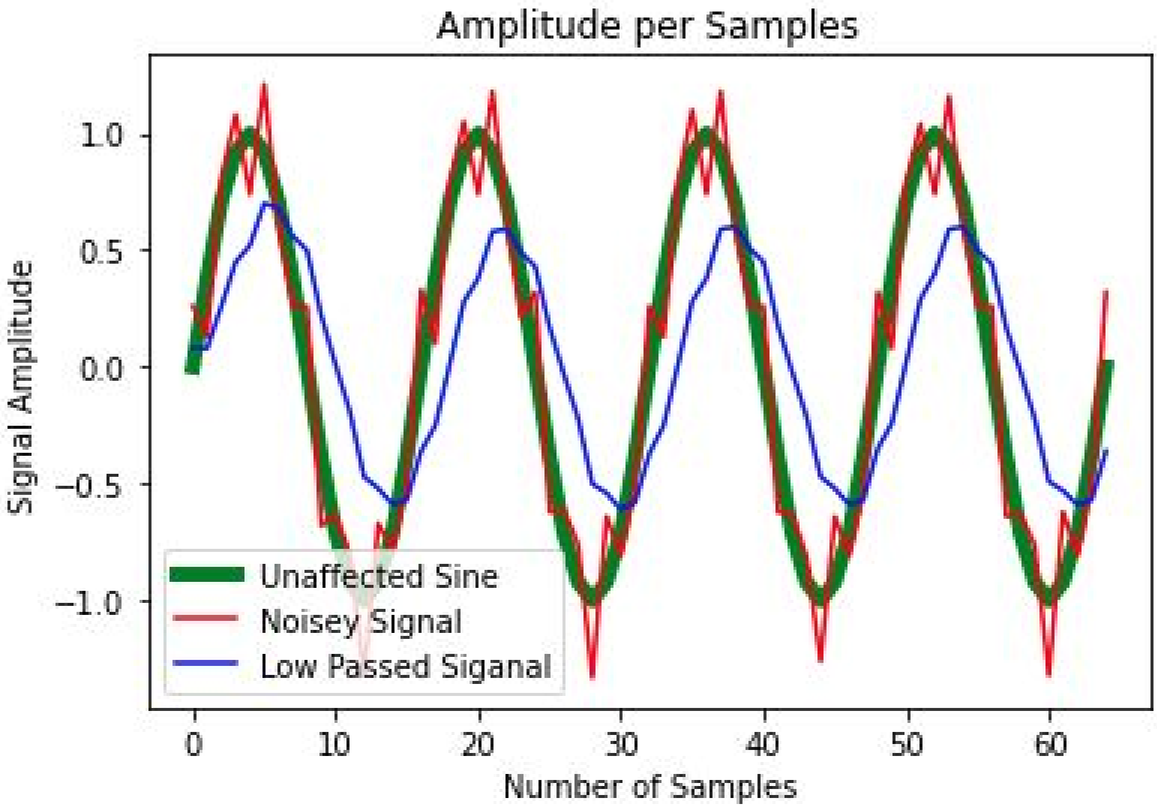
For alpha the higher it's cutoff frequency, the larger the alpha value, while the higher the nyquist sampling factor and the higher the signal frequency, the lower the alpha. More simply, alpha itself is the smoothing factor in the lowpass filter; soit smoothes out noise as depending on the cutoff. Additionally, if your cutoff is too low you could end up filtering out most of the signal altogether (think 1 hz cutoff on 100 hz signal, the result is low or no signal). It's important to note that if there appears to be no noise in the cutoff frequency range, the cutoff acts and smooths out this noise. So if one wants to smooth noise, a good starting point, if there is a visible and significant trend visible (such as a square or sine wave but with noise), figure out hertz or frequency and try to avoid cutting off at frequencies below the trend because the overall amplitude and signal may experience losses beyond noise but the losses instead may be in the realm of the overall trend or reference signal. In short, at lower cutoffs you get a lower alpha and start to lose signal if you go too low, while you may experience less smoothing depending on the original signal at higher frequencies, while the best way to clear noise is to assess any geometric curves or trends in your signal and attempt to lp filter out just above its measure frequency.

Discussion of Report Images:
It can be observed that at lower sampling rates you get sharper and more "pointed" trend lines, such as at the peaks of the sine at the nyquist rate of 16 (hence the inclusion of the clean sine graphs). This occurs because the sampling rate effectively pieces together small sections of signal, similar to integrating a signal but with different cross sectional bounds for the same signal. So if you integrate one cycle of the same signal at 3 different rates, the lowest rate effectively is more pointed because there's less samples to reconstruct the reference signal, while the higher rates will preserve the signal better,

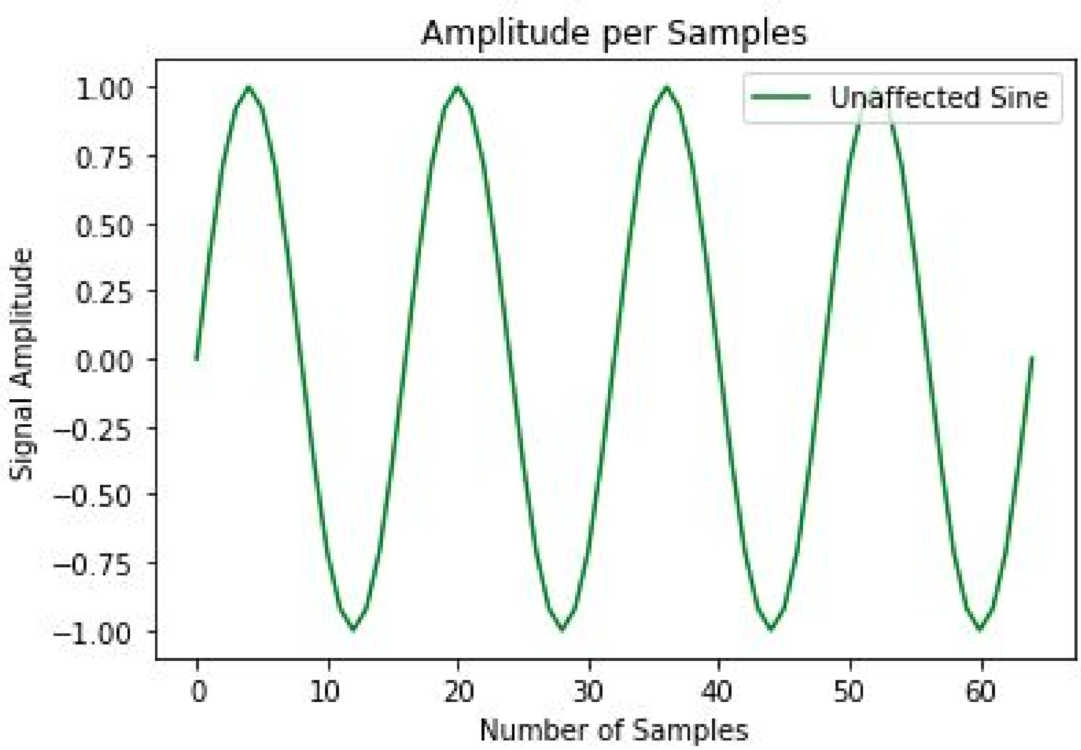
with practical and reasonable limits (i.e. to a certain extent a very, very high sampling rate, might be impractical because it's reconstruction of signal is only slightly better than a much lower and reasonable sampling rate, because the reconstructed signal approaches a limit, which is the original reference signal). At higher sampling rates the noise appears to show more jittery behavior as well. I'd imagine at higher sampling rates the signals would better display curves, but would eventually have little to gain at very high rates. The smaller nyquist frequencies distort the actual curves of the sine and fail to display the amount of actual noise that the higher frequencies show, which is something to note in the actual plots themselves.

Below this page are the plots

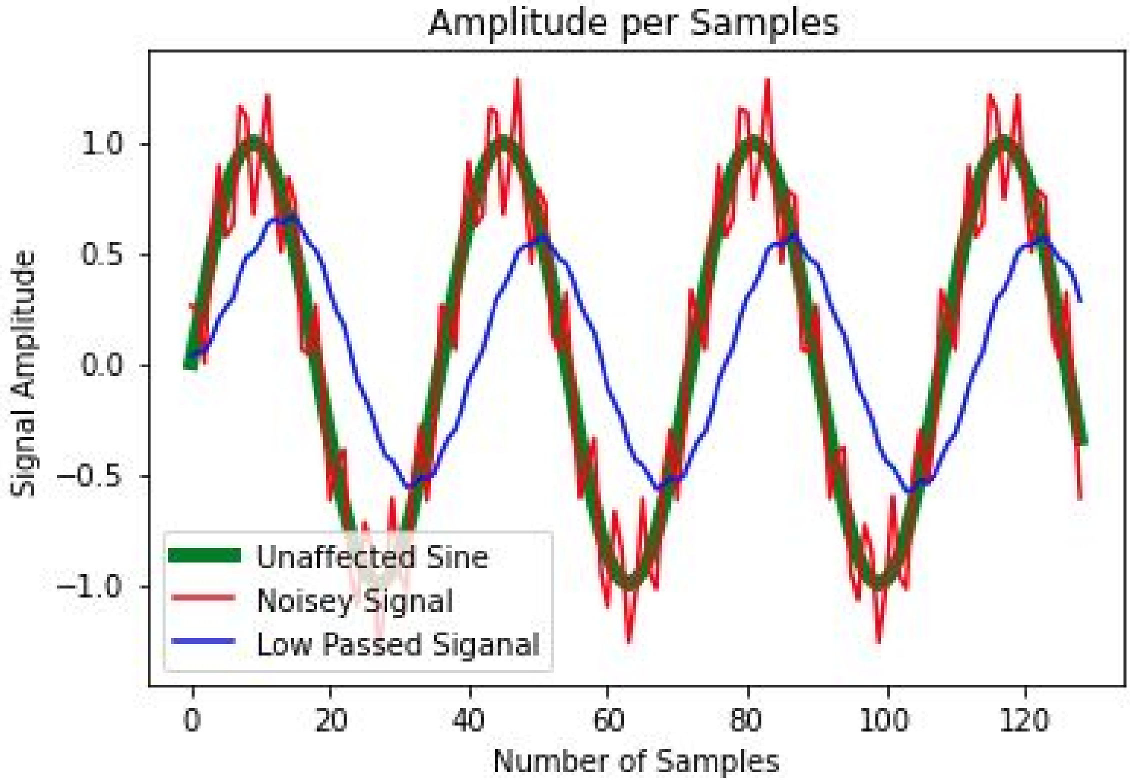
Nyquist = 16: At 100Hz, Nyquist * 4 = 64 samples



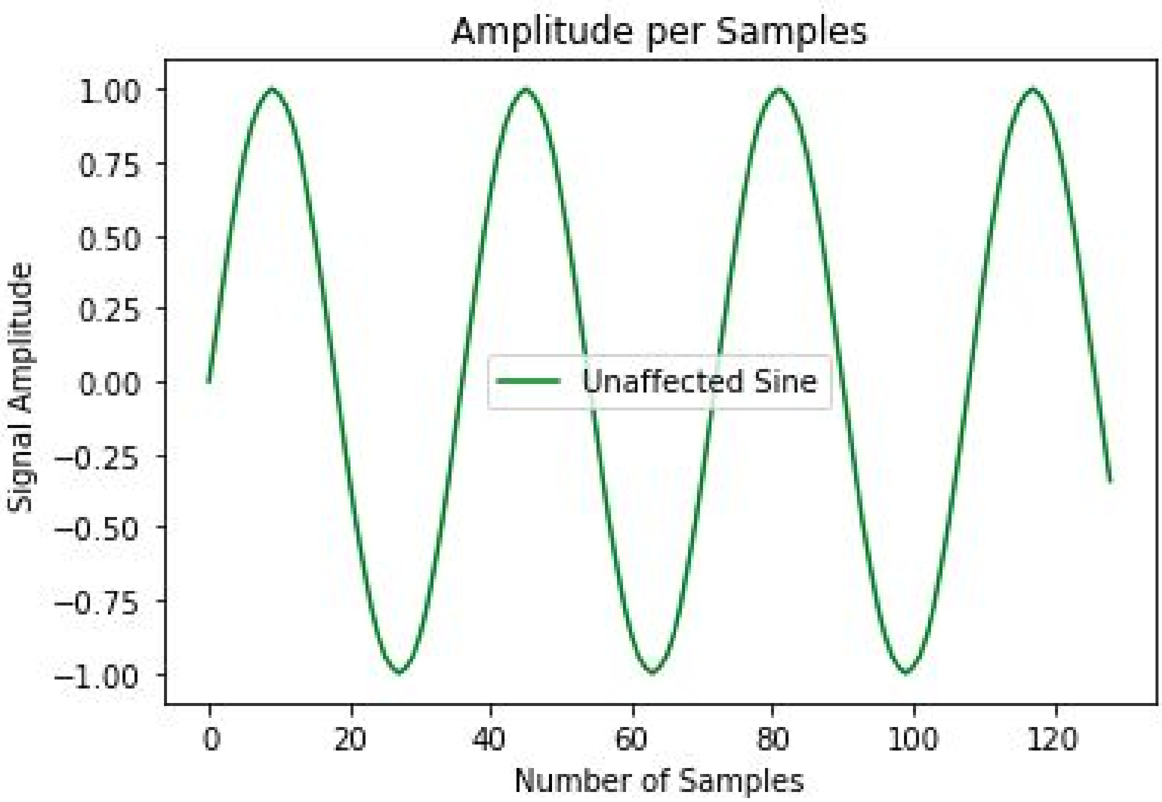
Nyquist = 16, Clean Sine



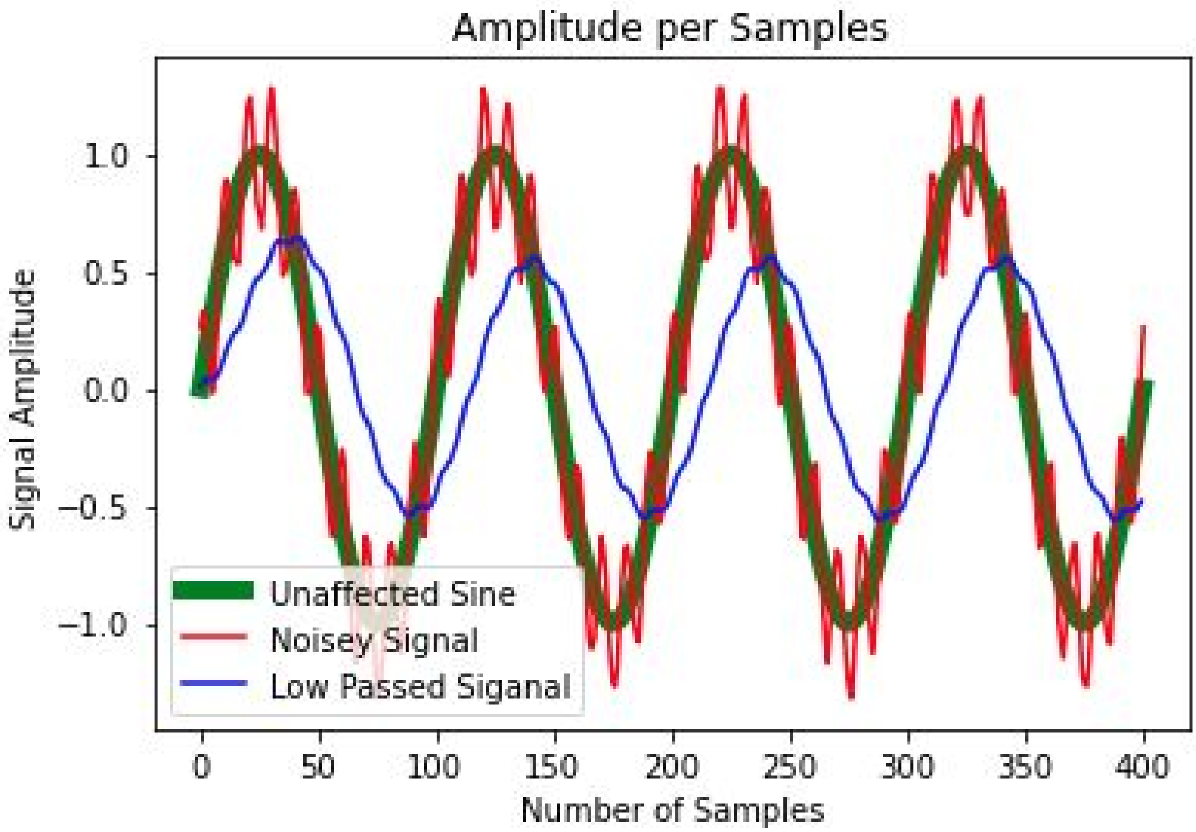
Nyquist = 32: At 100Hz, Nyquist * 4 = 128 Samples, All Signals



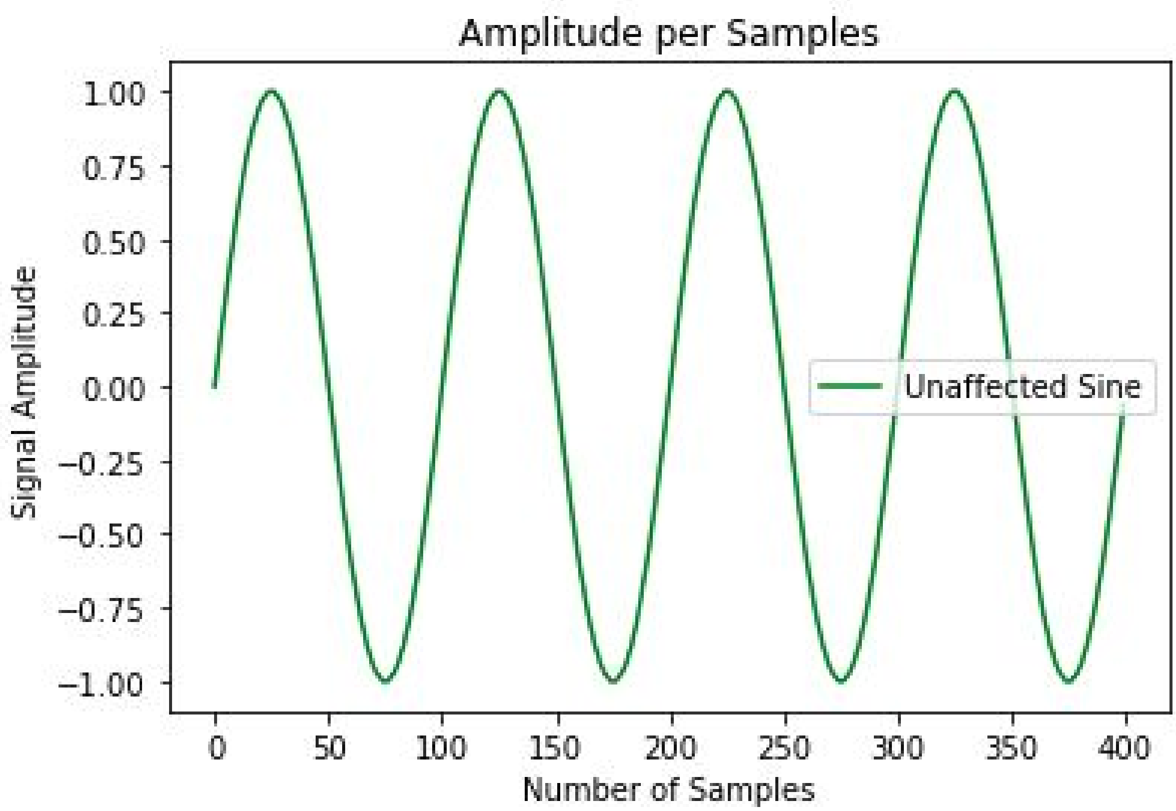
Nyquist 32: Sine Signal



Nyquist = 100: At 100 Hz, Nyquist *4 Cycles = 400 Samples



Nyquist = 100, Clean Sine



Equations I used when trying to solve error between signals:

$$RMS = \sqrt{\frac{x_1^2 + x_2^2 + x_3^2 + \dots + x_n^2}{n}} = \sqrt{\frac{1}{T}\int_0^T v(t)\,dt}$$

$$\text{Filtered RMS} = FR = \sqrt{\frac{x_1^2 + x^2 + x_3 + \dots + x_n^2}{n}}$$

$$\text{Reference signal (clean sine)} = CR = \sqrt{\frac{y_1^2 + y_2^2 + y_3^2 + \dots + y_n^2}{n}}$$

$$\text{Percent Error} = \left|\frac{V_A - V_E}{V_E}\right| * 100 = \left|\frac{FR - CR}{CR}\right| * 100\%$$