University of Colorado Denver, Spring 2020
Special Topics Course: Autonomous Vehicle Design
Homework 2: Serial Communication, Plotting, and Filtering

General Description

The Arduino will generate a noisy periodic signal, filter it with a first-order low pass digital filter and serially transmit both the noisy signal and filtered signal values to a python program. The python program will write the signal data to files and plot the signal data.

Arduino State Behavior

Arduino serially transmits a ready signal to the python program, once per second. It continues transmitting the ready signal until an acknowledgement from the python program is received.

Arduino waits to receive signal frequency from python program. Sends acknowledgement back to python program after receiving signal frequency.

Arduino generates the signal at the specified sampling interval, transmitting the signal data to the python program. After transmitting all signal data, the Arduino program sends an end of data signal to python and then returns to a state of waiting to receive a signal frequency from the python program.

Python State behavior

Open serial port connection to Arduino

Wait to receive ready signal from Arduino, transmit acknowledgement to Arduino in response to receiving ready signal.

Python program transmits a signal frequency in units of Hz to Arduino 10 times per second until Arduino acknowledges receipt of signal frequency.

Python program enters state where it receives signal values from Arduino. It remains in this state until Arduino sends an end of signal value message or until there is a signal interrupt, SIGINT.

Python program saves writes the signal data to a file or files (may want to save noisy data in one file and filtered data in another file). Plot the noisy versus filtered data. The program may perform these actions while receiving the signal data, or after all data has been received. When all data has been plotted, the plots should be saved a .png or .jpg images.

SIGINT behavior – when the signal interrupt, SIGINT occurs, the python program should transition to a shutdown state. Any data received and not yet written to a file, should be saved. Files and serial connections should be closed before terminating the program.

Program Requirements

Periodic ideal signal – Use a periodic signal such as a sinusoidal wave or a square wave. The peak to peak amplitude should range from -1 to +1.

Noisy signal – Add noise to the periodic ideal signal. The random number generator may be helpful for random noise or you may choose to add scaled amplitude values of other periodic signals at other frequencies.

ideal = (signal freq * 2pi * t)

noise = 1.3*cos(some freq * 2pi * t) + 0.4 * sin(some other freq * 2pi * time)

data = ideal + noise

Timing – use a microsecond timer to control time intervals when signals are sampled.

Signal Frequency – The maximum signal frequency is 250 Hz. The python program should guard against sending a frequency larger than that.

Baud rate – consider the sampling rate and amount of data transferred from Arduino to python when determining baud rate.

Nyquist – don't forget the Nyquist sampling criteria. The minimum criteria is not the best choice. Oversampling is better than under-sampling. However, there are practical limits to the sample rate. While 2 times is too little, 1000 times is likely impractical. Experiment to see what rate performs well.

Arduino – write the program for an Arduino Nano or Uno.

Comments – programs should include comments regarding algorithms employed, assumptions made, and supporting detail/reasoning.

Implementation details are up to the programmer. You may want to add LEDs to the Arduino program to signify the state. When working with hardware, in a limited debug environment, LEDs are often useful indicators. For the python program, you may want to allow the user to control the input of signal frequency, or you may want a function that loops through a few test frequencies.

Undergraduate Student Requirements

Submit the Arduino sketch and python program. You may also submit a written analysis of test results and discussion of observations, topics for further study, etc., if so desired.

Graduate Student Requirements

Submit the Arduino sketch, python program, and written report. The written report is an analysis of filter performance at three different Nyquist sampling rates. You may want to sample at 2 times, 4 times, and 16 times the signal frequency while keeping the filter coefficients the constant across all

tests. You are not expected to find an optimal rate but should compute the error between ideal and filtered output at each sampling rate. Is there an “optimal” practical rate? At some point, more samples do not improve performance.

There is no length requirement for the report. It is expected to be a brief written report. You do not have to derive filter equations or discuss theory in detail. Mainly, I am interested in the filter’s performance. The report should address the alpha weighting factor calculation and its related cutoff frequency. The general report form is an introduction, methodology, analysis and conclusion.

Grading

Programs will be tested with python3 running in Ubuntu 18.04 and an Arduino Uno/Nano. Graduate student reports should be submitted in a Word document format. Use single-spacing in a paragraph, leaving a blank line between paragraphs. Any tables and figures should be numbered. No cover sheet nor table of contents is necessary, but do include your name, the report date, and a report title.

Total points possible, undergraduate: 75
Total points possible, graduate: 100