Code Challenge Guidelines

Intuitive Surgical, Inc.
July 18th, 2022

Summary: Create a command-line program that reads a file with a series of data from two sensors and compares them to determine if a sensor has failed.

Platform: The program must be in Python and should not require any imports. Any imports that are used should be from the Python standard library and a justification should be made for why they are being used.

Task:

You are given a motor and gearbox assembly with two sensors: an incremental quadrature encoder and a potentiometer. The encoder is on the motor shaft (i.e., the input to the gearbox), and the potentiometer is on the output shaft of the gearbox. There is a 30:1 gear reduction between the encoder and potentiometer.

The encoder generates 2048 counts per revolution of the motor. The reading from the encoder is presented to you as a signed number whose value is in encoder counts.

The reading from the potentiometer is a voltage from 0 to 5V, which cycles linearly over each revolution by a sawtooth wave as pictured in Figure 1. This voltage is read by an 8-bit A/D converter and presented to you as an unsigned number, where a value of 0 corresponds to 0V and a value of 255 corresponds to 5V.

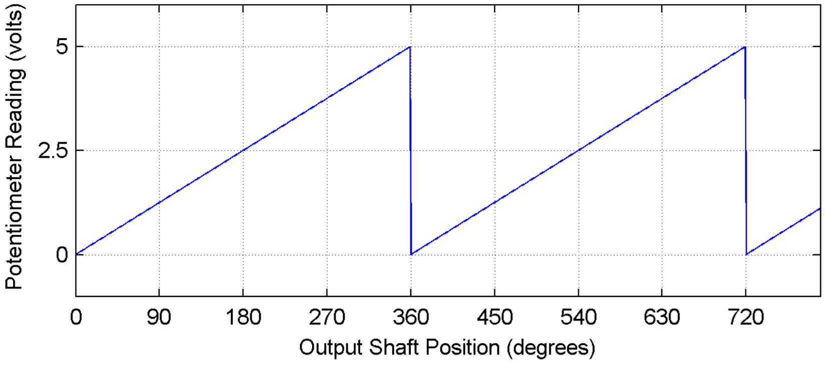

Figure 1. Potentiometer reading vs. output shaft position.

Note that the potentiometer is an absolute sensor – at boot time (the beginning of the data file), it has a non-zero value corresponding to the current position of the output shaft. However, the encoder always reads zero at boot time, so there is a fixed offset between the two sensors that must be computed every time you start up. You may assume that both sensors are functioning correctly during the first 0.5 seconds of each data file. This period may be used to initialize and/or calibrate your algorithms.

Write a command-line program that takes as its first argument a path to a text file with a series of readings from each sensor. The program should compare the two sensor data series and print to the console a single message indicating whether or not an error was detected. If an error is detected, the time at which an error was first detected should also be printed.

Input Data:

Each sensor data file is a text file with lines of the format

time encoder potentiometer

where *time* is in floating-point seconds, *encoder* is a signed integer and *potentiometer* is an unsigned integer. The sensor readings are for a motor moving through the trajectory pictured in Figure 2.

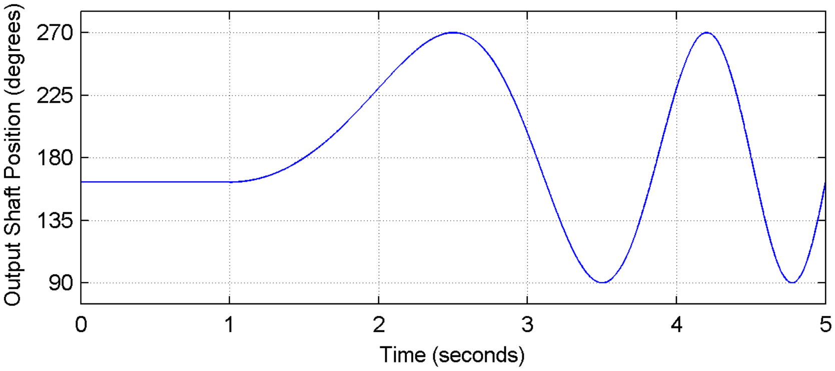

Figure 2. Motor trajectory for sample datasets.

The following sample data files are provided:

perfect.txt:	encoder and potentiometer readings match to the limit allowed by their resolution
normal.txt:	the potentiometer readings are noisy
error.txt:	the encoder fails during the trajectory

Your program will be tested against other data sets. The trajectories may vary from the sample data set, but the amount of potentiometer noise will be similar to that of the sample data set. You may assume that the potentiometer reading does not wrap during a dataset, i.e. that the position never crosses from 0° to 360° or vice-versa.