PID:17/Shreya Rastogi May 2020

GY-87 and Filter Design

1 Aim

To understand GY-87 sensor, collect data using it and then filter that data for noise using the complimentary filter. To also understand about the function of Kalman Filter and compare it with complimentary filter.

2 Questions

1. What is GY-87 ? What are some of its features? What is it's benefit over MPU6050?
2. What are filters and why do we use them in our systems?
3. What kind of filter will be used for a balance bot system? Why do we use a high pass filter for Gyroscope and a low pass filter for Accelerometer?
4. What is a Kalman Filter and why should we use it instead of a complimentary filter?

Once you have answered the above mentioned questions, you will perform the experiment.

3 Experiment

In this experiment you have been provided with 2 files, "**filter.m**" and "**data.csv**". The octave file "filter.m" consists of a skeleton code for the implementation of extraction of noisy data from GY-87 sensor, passing of that data through the necessary filters for noise removal and finally combining that data to get the tilt angle of the balance bot. The "data.csv" file contains readings from sensor module GY-87. These readings are obtained from 12 registers named as below:

Sr. No.	Register Name	Sr. No.	Register Name
1.	ACCEL_XOUT_H	7.	GYRO_XOUT_H
2.	ACCEL_XOUT_L	8.	GYRO_XOUT_L
3.	ACCEL_YOUT_H	9.	GYRO_YOUT_H
4.	ACCEL_YOUT_L	10.	GYRO_YOUT_L
5.	ACCEL_ZOUT_H	11.	GYRO_ZOUT_H
6.	ACCEL_ZOUT_L	12.	GYRO_ZOUT_L

Table: Register Names

These readings help us find out the tilt of the body on which the sensor is mounted.
Readings of registers from Sr. No. 1 to 6 in the Table 1 are from the Accelerometer present in the sensor module. Similarly, readings of registers from Sr. No. 7 to 12 in the Table 1 are from the Gyroscope.
Accelerometer and Gyroscope in the GY-87 sensor module gives us readings for all three axes (X, Y and Z). In order to get data for a particular axis from Accelerometer or Gyroscope, you have to combine readings from its respective HIGH and LOW registers.

The "filter.m" file consists of functions for implementation of the code. Go through the explanation offered for each function and then write the code.

Function description:

1. **accel_data and gyro_data:** Used for combining the High and Low values of the registers mentioned in Table 1 for respective axis. accel_data function will work on readings of registers from Sr. No. 1 to 6 in the Table. Similarly, gyro_data will work on readings from Sr. No. 7 to 12 in the Table. Then the output of accel_data and gyro_data are passed as the argument to lowpass and highpass functions respectively.

2. **lowpass and highpass:** These two functions are going to be used for filtering the data of the sensors as the data is highly noisy. One of the arguments in these two functions is f_cut i.e. cut off frequency. Keep f_cut as 5. You already understand that cut off frequency is the point which defines the point of blocking/passing of the signal based on their frequency.

3. **comp_filter_pitch and comp_filter_roll:** Use these two functions to compute Pitch and Roll by implementing the concept of Complementary Filter on the values filtered by low pass filter and high pass filter. Both of these functions take six values as argument.
   For calculating the Pitch: use the changes in values of accelerometer readings of Y-axis and Z-axis and gyroscope readings of X-axis.
   For calculating the Roll: use the changes in values of accelerometer readings of X-axis and Z-axis and gyroscope readings of Y-axis.

4. **run:** In this function, you will call all of the above mentioned function to calculate the Pitch and Roll. Once they are calculated, you will save them in an array named “Output”. This array will be of size 8000x2.

After executing the final function, the output of the experiment that is the pitch and roll will get saved to "output.csv" file.