ME 457 Checkpoint 1

1. Attitude Estimation
   1. Conceptual Roll Estimation

Accelerometer:

The accelerometer on the auto pHAT mounted on the Raspberry Pi provides accelerometer data for three axes, in a north, east, down configuration. Switch or reverse axes so that they are oriented in the right direction. The angle between the y and z accelerometer vectors is the pitch angle. to find the pitch angle use this equation:

This estimate will be in radians so it must be multiplied by the conversion factor, , to get in degrees.

gyroscope:

The gyroscope on the auto pHAT mounted on the Raspberry Pi provides gyroscope data for three axes, in a north, east, down configuration. Switch or reverse axes so that they are oriented in the right direction. The gyroscope provides angular velocity so to find angular position, roll angle, a numerical integration method must be used on the x-axis gyroscope data. Trapezoidal rule integration method:

The roll angle found from the gyroscope data needs to be divided by a scalar value to be accurate. To do this roll the Pi to a known angle then divide the estimate by the known angle. this will provide the scalar value.

* 1. Practical Roll Estimation

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Text

Description automatically generated



Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application

Description automatically generated

The above python script takes in data from the gyroscope and accelerometer and converts it to pitch and roll angles.

Lines 1-20: Initiates this script and imports all the functions and sensor data that are needed. Then assigns the sensor data array to variable imu\_data and assign the pitch, roll, and yaw angle array to variable message which will later be assigned filtered estimates.

Lines 23-30: Initiates a node, creating a call back function to make imu\_data a global variable, and creating a pubisher so the angle array can be published.

Lines 33-37: Setting the sampling frequency to 100 hertz. The myData array is populated with header labels.

Lines 43-59: This section creates an excel file named pitch\_roll and populates it with the myData.

Lines 61-63: First initializing the time variable, zero\_time, then the main function that processes the sensor data and outputs pitch and yaw estimates is defined.

Lines 66-95: Establishes the data rate according to what it was set to in line 33. initializing all the iterative variables outside of the loop. Start while loop that will calculate pitch and roll angle then filter the signal to remove noise.

Lines 97-110: First creating a subscriber so IMU data from the sensors can used. A Now variable is initialized and the time step between values is the sampling rate set in line 33. A conversion factor is created. This conversion factor changes radian to degrees and must be applied to the accelerometer pitch and roll data if output in degrees is desired. Last is a set of variables that the raw sensor data are assigned to. The gyro y and z were in the wrong direction, so a negative sign was appliedo to reverse the axes

Lines 128 -131: Pitch and roll estimates are calculated from the accelerometer.

Lines 149-161: Pitch and roll estimates are calculated from the gyroscope.

Lines 188-195: myData is populated with current iteration values from the IMU and calculated pitch and roll values. Then current myData is put in the file pitch\_roll.

Lines 198-203: rate.sleep() waits the remaining amount of time, set by the sampling frequency to begin the next iteration. Filtered pitch and roll estimates are assigned to message variable. Message variable is published

Lines 206-210: Runs the main function.

The roll angle from the accelerometer is not useful because there is too much noise. The roll angle from the gyroscope is not useful at low speeds because it is subject to drift. Filters will solve the issues.

1. Digital Filtering
   1. Conceptual Digital Filtering

Digital filters remove certain frequencies from a signal. There are two types, Infinite impulse response (IIR) and Finite impulse response (FIR). IIR filters are for applications where linear phase is not very important and memory is limited. FIR filters are for applications where linear phase is important and good memory and computational power are available. Second order Butterworth high pass and low pass filters IIR are going to be used. Low pass filters attenuate frequencies above the cut off frequency. High pass filters attenuate frequencies below the cut off frequency. Looking at a FFT of the roll data from the gyroscope and accelerometer will identify where an acceptable cut off frequency should be. To execute a second order butterworth low and high pass filter data needs to be processed through this set of equations. The input being the data x, the sampling frequency , and the cut off frequency . The output being filtered data y. The set of equations:

For low pass: For high pass:

The filter:

* 1. Implementing a digital filter in Python

After the script has calculated pitch and roll estimates from the gyroscope and accelerometer in the while loop the following code will filter the data set.

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Lines 109-120: Define the variables according to the sampling frequency and cut off frequency.

Lines 130-142: Here the pitch and roll from the accelerometer runs through the low pass filter equation and reassigns values to make room for the current index in the next iteration.

Lines 159-171: Here the pitch and roll from the gyroscope runs through the high pass filter equation and reassigns values to make room for the current index in the next iteration.

Using a cut off frequency of 5 hertz and a sampling frequency of 100 hertz:

Roll from the accelerometer is clear. The lowpass filter attenuated most of the frequencies higher than the cut off frequency. The Roll from the gyroscope no longer has drift. The high pass filter attenuated most of the frequencies lower than the cut off frequency.

1. Sensor Fusion
   1. Complimentary Filter Theory

The accelerometer works well at low frequencies and the gyroscope works well at high frequencies. Using a low pass filter on the accelerometer and a high pass filter on the gyroscope both using the same cut off frequency that is located between the accelerometer and gyroscope spikes on the FFT. The sum of both filters gives filtered data across the whole frequencies range. The data above was at low frequency. That is why the low pass filtered pitch from the accelerometer is picking it up. If the frequency was above 5 hertz the accelerometer would be about zero and the high pass filtered roll from the gyroscope would be picking it up. The equation is:

* 1. Implementing a Complimentary Filter

Text

Description automatically generated

Lines 174-177: Now that there is a filtered roll data from the gyroscope and the accelerometer they can be summed to give the complementary filter.

Here the complimentary filter gives accurate roll estimates and takes advantage of both sensors. The accelerometer for the lower frequencies and the gyroscope for the higher frequencies.