Programming assignment 1

CM2022, Sports and Health Instrumentation

Aim

The assignment includes two tasks: 1A (Motion) and 1B (Rotation). To pass, both tasks must be successfully completed. Additionally, you are required to write a laboratory report based on your results. For higher grades, you must also complete one of the two in-depth tasks and provide an explanation of how this can be applied in a sports-related context. Please refer to the description in the instruction for guidance on report writing.

Purpose

These tasks will serve as an introduction to utilizing the IMU (Inertial Measurement Unit) found in today's smartphones, among other devices. The primary focus is on two key sensors: the accelerometer and the gyro. These sensors enable the determination of the unit's movement and orientation within a space. By combining data from these sensors with appropriate filters and algorithms, it becomes feasible to acquire information for classifying various types of movements. Please note that the first step is to download and extract the imu.zip file from Canvas to your computer. Afterward, access/open HTML5_PolarVeitySense_HT25.html using a Chrome browser It's essential to use Chrome since it currently on of few browsers that provides built-in Web Bluetooth support.

In the 'Show' menu, navigate to 'Developer' and select 'JavaScript Console.' Do Included in the imu.zip file is a file filter.js where you can input your code.

Task 1A, Motion

First write a code that combines the acceleration from all axes on the Polar Verity and display the result using the equation provided below. In the equation 'x', 'y' and 'z' represent raw acceleration values, while 'g' represents gravitational acceleration.

$$F = \sqrt{x^2 + y^2 + x^2} - g$$

Secondly, utilize an Exponential Weighted Moving Average Filter (EWMA) to filter the value and present the result. In the equation below, 'y(n)' represents the filtered value, 'y(n-1)' denotes the preceding filtered value, and 'x(n)' corresponds to the sampled value, such as 'F' in the equation mentioned earlier. Lastly, 'α' represents the filter parameter and should fall within the range of 0 to 1. Try different 'α'-values and reflect on how it effects the filter.

$$y(n) = \alpha * y(n-1) + (1-\alpha) * x(n)$$

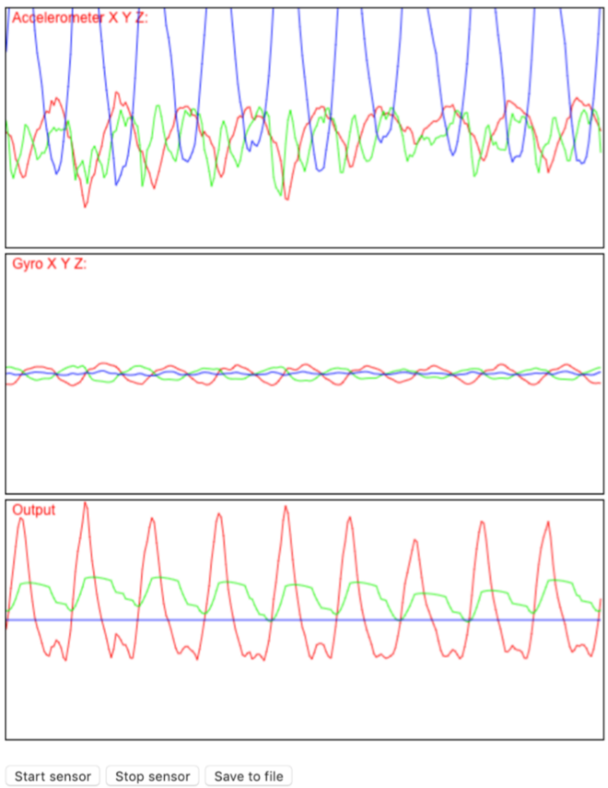
Introduce an additional Exponential Weighted Moving Average (EWMA) filter with distinct values for 'α' and 'β.' These values are employed depending on whether the last value of 'x()' is below or above the preceding value. The resulting graph should intersect with the initial graph, as demonstrated in the lower-left section of Figure 1. Consequently, while 'α' may remain relatively consistent, 'β' should be set at a higher value.

$$x(n) > x(n-1) \quad \rightarrow \; y(n) = \; \alpha * y(n-1) + (1-\alpha) * \; x(n)$$
$$x(n) < x(n-1) \quad \rightarrow \; y(n) = \; \beta * y(n-1) + (1-\beta) * \; x(n)$$

Finally, calculate the cadence. This can be achieved by detecting the intersection of the two graphs 'y1()' and 'y2()' generated as demonstrated in the equation below. Subsequently, record a timestamp when this intersection occurs. Upon its recurrence, compare the two timestamps to derive the cadence. To ensure a more stable cadence, the obtained result can be filtered using an Exponential Weighted Moving Average (EWMA) filter.

$$y1(n) > y2(n) \,\&\&\, y1(n-1) < y2(n-2)$$

The figure below displays the running HTML5 application. In the top-right corner, you can observe the raw acceleration data from the sensor. The blue color represents the X-axis, red represents the Y-axis, and green represents the Z-axis. The middle graph on the right represents the gyroscope data, and the bottom-right graph is available for displaying the results. On the left-hand side, there is a console window open for monitoring the running program.



Continue and write a code that will detect calculate the load where *i-1* represent the previous sample.

$$L = \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2 + (y_i - y_{i-1})^2}$$

Lastly write a code that detect when the Polar Verity Sense is continually being shaken for more than 1 sec.

Task 1B, Rotation

Use trigonometry to calculate the rotation of the Polar Verity Sense, starting with the roll and display the result.

$$accRoll(n) = \operatorname{atan}\left(\frac{x(n)}{\sqrt{y(n)^2 + z(n)^2}}\right) * {}^{180}\!/_{\pi}$$

Calculate the rotation, starting with the roll, using the gyro and display the result.

$$gyroRoll(n) = gyroRoll(n-1) * dT * gyroY(n)$$

Finally, apply a complementary filter for data fusion of the rotation, starting with the pitch, based on both data from the accelerometer and the gyro. Display the result. Try different 'α'-values and reflect on how it effects the filter.

$$comRoll(n) = \alpha * \big(comRoll(n-1) + dT * gyroRoll(n)\big) + (1 - \alpha) * accRoll(n)$$

In-depth assignments

There is two possible in-depth assignment, either write a code that estimate the displacement using a Kalman filter. Alternatively, follow the tutorial for the OBS system to stream IMU data and add in on top av a video stream. Reflect on the impact of the system and the obtained results.