Virtual Reality Coursework

Gvch48

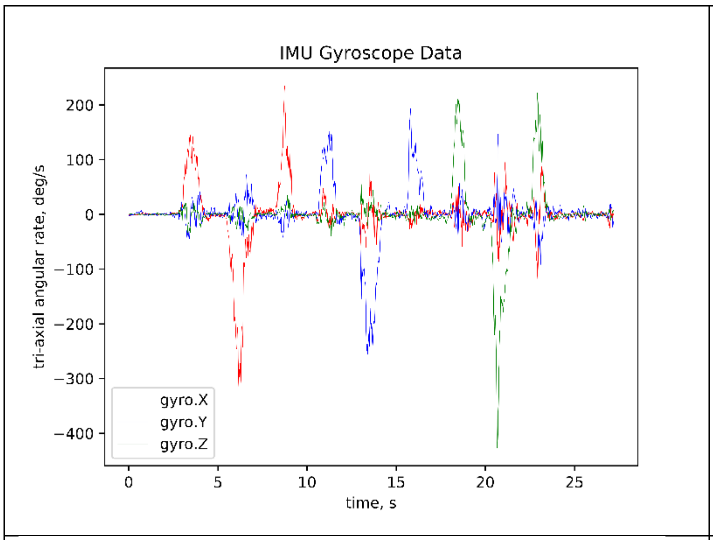
Initial comments on orientation

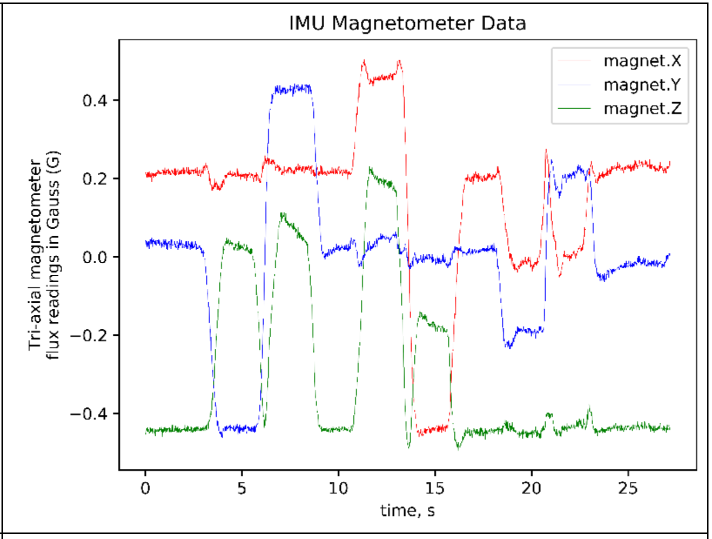
The default way to represent 3D coordinates in a list is [x,y,z], which corresponds to [pitch, yaw,roll] in LaValle's paper. From The accelerometer readings, it can be deduced that The Z axis of our dataset is "up", and as such, to fit with Lavalle's definition of pitch, yaw and roll, 3D coordinate lists in all my functions are in the order [y,z,x]

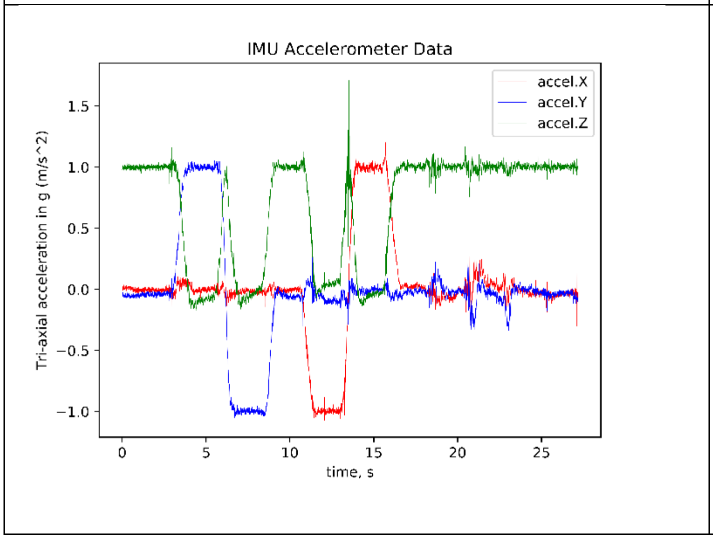
Problem 1:

The interfaces of each method are explained in comments in the code.

The Normalisation methods check if the magnitude is 0, and If it is sets it to 1. Since a magnitude of zero means all the values are zero anyway, changing the magnitude prevents divide by zero errors, and since 0/mag = 0 it doesn't matter that I changed the magnitude to 1







Problem 2:

Graphs of Results included in Section for Problem 5

Problem 3:

Instead of using raw accelerometer values, I took the average over 30 readings.

Alpha = 0.001: No unusual spikes but also does not correct Tilt.

Alpha = 0.01: Spike in Z value when x-axis is rotated to +90 degree. Final orientation is no closer to 0 than with no Tilt Correction.

Alpha = 0.05: Same as for alpha = 0.01

Alpha = 0.1: Spikes in Y and Z when X is at +90 and -90 degrees. No improvement to tilt

Alpha = 0.15: Spikes in Y and Z like for alpha=0.1, but the spikes are wider (last longer). No improvement to tilt

I tested several more values for alpha and none of them made the final pitch and roll angles any closer to zero degrees (as it should be). I checked my code but could not spot the source of the issue. The graphs for Problem 5 are for Alpha=0.1

Problem 4:

Instead of using raw magnetometer values, I took the average over 20 readings.

Alpha = 0.001: Spikes in Y and Z values for X = +/-90 degrees. No yaw correction.

Alpha = 0.01: same as alpha=0.001

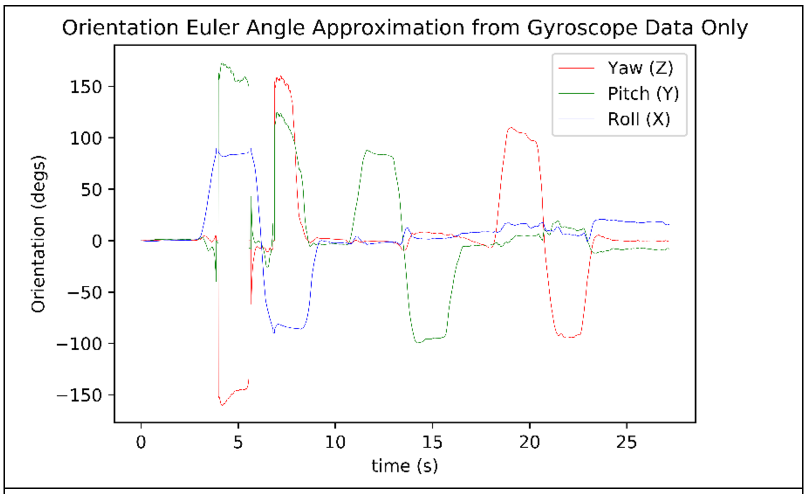
Alpha = 0.1: Too large, Z values change too much

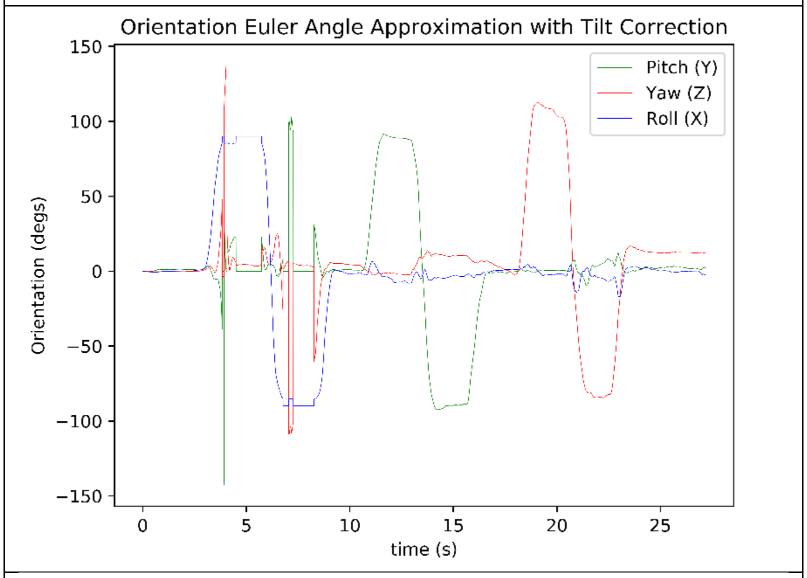
Alpha = 0.03: Largest value that doesn’t create erratic changes in X value, but also no yaw correction.

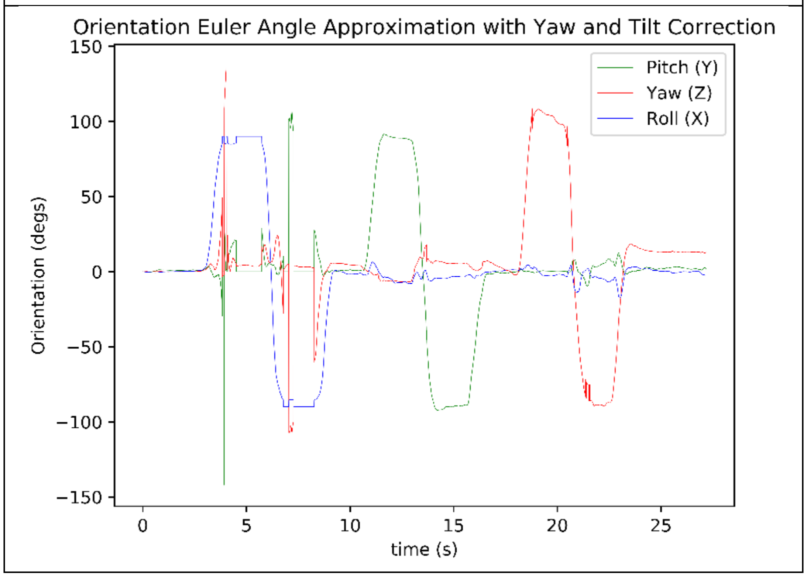
The graphs for problem 5 are for Alpha = 0.03

Clearly, similar to problem 3, there is a flaw in my code that is preventing drift correction from occurring. It is interesting that this has occurred, since varying alpha does cause the spikes to change which shows that the orientation is being adjusted, yet there is no notable improvement to the final orientation.

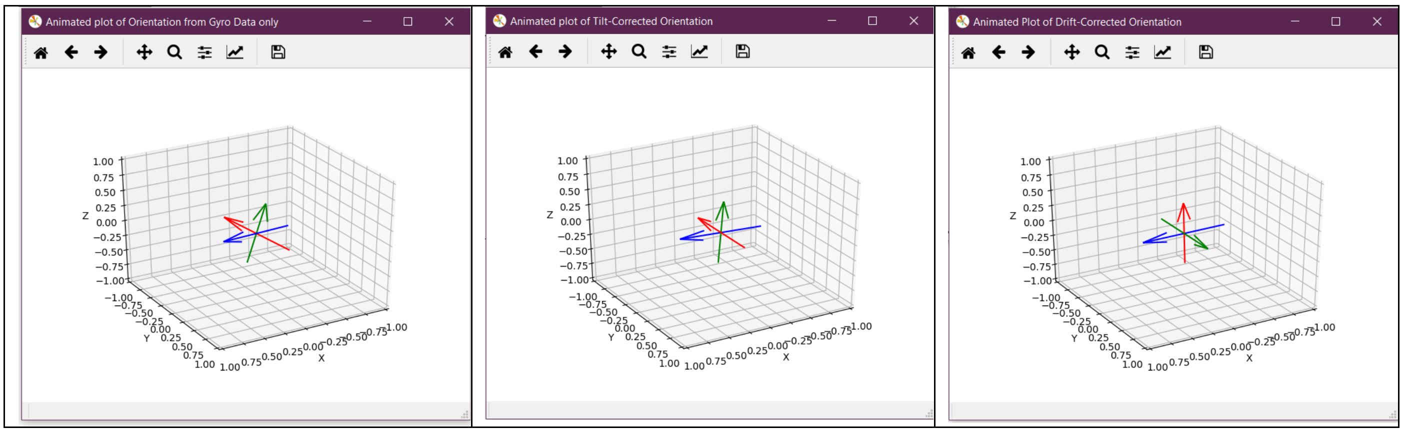
Problem 5:







2D graphs are also created and saved by running the code and following the console prompts


Animated plot of Orientation from Gyro Data only
Animated plot of Tilt-Corrected Orientation
Animated Plot of Drift-Corrected Orientation
X
Y
Z

3D plots can be viewed in full-speed and half-speed by running the code and following the console prompts

Problem 6

I set torso length to be 0.5m and neck length to be 0.2m, so the IMU starts 0.7m up in the Z axis. The animated plot shows that the IMU moves about 0.5m in each dimension over the course of 30 seconds. While this prediction is likely much worse than the reality that the IMU probably only moved a couple of centimetres, the results are very similar to those in LaValle's paper, where they measured movement of about 1 meter over 25 seconds using this Kinematically constrained method, which shows that my implementation of positional tracking is working as intended and is much better than using just double integration, although, as LaValle states, is "insufficient as a standalone technique".