ESE650 Project 2: Orientation Estimation

Due Date: **2/13/2018 at 1:20pm** on Canvas, and in class

In this project, you will implement a Kalman filter to track three dimensional orientation. Given IMU sensor readings from gyroscopes and accelerometers, you will estimate the underlying 3D orientation by learning the appropriate model parameters from ground truth data given by a Vicon motion capture system. You should then be able to generate a real-time panoramic image from camera images using the3D orientation filter.

Training Data Download: Now available at https://upenn.box.com/v/ese650Proj2-train

Test Data Release

Also available after 2/13/2018 1:20pm at https://upenn.box.com/v/ese650Proj2-test

Upload: on Canvas

(1) Code (due 2/13/2018 1:20pm, <pennkeyID>_project2.zip)

: Do not include data.

(2) Write-up (due 2/13/2018 11:59pm, <pennkeyID>_project2.pdf)

: Write the summary of your approach, result, and discussion.

: Make sure your result includes proper visualization of your orientation estimates and panoramic image.

Grading:Rubrics can be found on the Canvas assignment page.

*Clearly presenting your approach in the form of report and presentation, and having good algorithm performance are equally important.

Instructions and Tips

1. [CHECK DATA] You will find a set of IMU data, another set of data that gives the corresponding tracking information from the Vicon motion capture system, and also some files containing RGB images from a camera. Download these files and be sure you can load and interpret the file formats.

   * The files are given as '.mat' files. In order to use these files in Python:

   ```
   >>from scipy import io
   >> x = io.loadmat("filename.mat")
   ```

   This will return a dictionary form. Please disregard the following keys and corresponding values: '__version__', '__header__', '__global__'. The keys, 'cams', 'vals', 'rots', and 'ts' are the main data you need to use.

2. [SENSOR CALIBRATION] Note that the biases and scale factors of the IMU sensors are unknown, as well as the registration between the IMU coordinate system and the Vicon global coordinate system. You will have to figure them out.

3. [BEFORE IMPLEMENTING KF] You will write a function that computes orientation only based on gyro data, and another function that computes orientation only based on accelerometer data. You should check that each function works well before you try to integrate them into a single filter. This is important!

4. [IMPLEMENT KF] You will write a filter to process this data and track the orientation of the platform. You can try to use a Kalman filter, EKF or UKF to accomplish this. You will have to optimize over model parameters. You can compare your resulting orientation estimate with the “ground truth” estimate from the Vicon. A simple plotting program called “rotplot.py” is also provided for basic visualization.

5. [FOR TESTING] Make sure that your program can process new datasets containing only IMU readings without Vicon estimates. Your program should then filter this data, and then generate and display the estimated orientation.

6. [VISUALIZATION] You should then construct and display a panoramic image by stitching the associated RGB camera images in a proper way based on the estimated orientation.

7. [CLASS PRESENTATION] During the presentation in class, you are expected to bring your own laptop or use the classroom computer you will voluntarily present your algorithm and run your code on a set of test data. The projector has a HDMI port and you may need a HDMI adaptor for your laptop. You will be asked to run your code on the test set which will be released online.

8. Your write-up upload should include properly visualized results.
   The suggested (minimal) format is like the following:
   [Orientation plot] Euler angles of (1) your KF estimate, (2) estimate based on gyro only, (3) estimate based on accelerometer only
   [Panoramic image] A single image is fine. If you want to include a video, we recommend to give us some shared link to download (e.g., via google drive).