Assignment 2 - State Estimation

Dr. Joseph Conroy (conroyj@umd.edu)
Dr. Krishna Kidambi (kidambi@umd.edu)

10 points
Due: 04 November 2021

1 Overview

This assignment is designed to support development of skills for basic state estimation given onboard sensing. You may discuss approaches with classmates, but each student must turn in their own assignment and plots.

2 Implement Orientation State Estimation

Using the programming language of your choice, including MATLAB, implement orientation state estimators using the **Kalman Filter**, **Extended Kalman Filter**, and **Complementary Filter** approaches using downloaded real data bags as sources of data to work with. For this assignment, do not download complete estimation code off the internet, please implement the actual core of the fusion code from scratch, so you get practice doing this. You may download supporting libraries for math (example: Eigen) or for display of the data. You may work using quaternions or Euler angles as your orientation states, though quaternions are encouraged. The quaternion primer downloadable from the ELMS website in the 'files' section may be useful. The Kalman Filter is, naturally, a linear estimator: for this one, a portion of the "Position Control" bag is probably going to be best to use to demonstrate fusion. You don't need to demonstrate each algorithm working with all data sets. A reasonable subset that shows the algorithm working is fine.

2.1 Download Data Bags

Bags are a few GB in size each and can be downloaded from here:

https://drive.google.com/drive/folders/1vQ4_oUIN72g6hcUrbxHs_GptT2pmk2_z?usp=sharing

2.2 Coordinate Systems

Note that the onboard vehicle topics utilize the standard ROS coordinate system of FLU (forward, left, up) for the body-fixed axes and ENU (east, north, up) for the local inertial frame. The vicon coordinate systems were set such that the coordinate system in the body frame is (X=right, Y=forward, Z=up) and the inertial frame was (X=towards the wall mounted TV, Y towards the window outside in the Brin lab, and Z up). The vicon body coordinate system was co-aligned with the vicon local inertial system at the start of each data bag collection. You will find that the MAVROS on-board data and vicon data are 90 degrees off from each other in yaw.

2.3 Relevant Topics

The bags contain many topics that will not be needed, but are included to enhance your familiarity with what you have access to through MAVROS. Also included is the "hires/image_raw" topic so you can see what the drone was actually doing during the manuevers. 'rqt_image_view' works well to display the imagery while you play the bag. Principally, you will be interested in the following topics to complete this assignment: '/mavros/imu/data_raw' (containing the calibrated accelerometer and rate gyro data), '/imu0' and '/imu1' (calibrated imu data from both IMUs on the VOXL board), and '/mavros/imu/mag' (magnetometer data). Here's a little bit on the VOXL-specific IMUs:

https://docs.modalai.com/voxl-flight-datasheet-imus/#voxl-flight—voxl-imu0.

The '/vicon/m500_joec/m500_joec' topic is the ground truth Vicon data to be used for comparison (not as part of the estimator).

For reference, a description of the published topics from MAVROS is available here:

http://wiki.ros.org/mavros#Published_Topics

3 Plot Results

1.) Professionally document and provide plots of each of the three estimator fused outputs alongside ground truth. Ground truth in this case can be either the onboard PX4 estimated data or the Vicon ground truth appropriately rotated into the same frame of reference. The MAVROS topics with the onboard fused data can be '/mavros/imu/data' OR '/mavros/local_position/pose' for example. Reminder that the Vicon ground truth is '/vicon/m500_joec/m500_joec'. You are NOT expected to "beat" the onboard PX4 filtering in terms of accuracy as this filtering method has been an open-source effort years in the making, however the nature of your implementation should suggest that the fusion of sensors is actually working. For reference, the onboard estimation algorithm for PX4 is an extended Kalman filtering process that is not easy to follow in the actual firmware code, but is reasonably well documented here: https://docs.px4.io/master/en/advanced_config/tuning_the_ecl_ekf.html

2.) Plot and discuss how the onboard estimate (orientation extracted from '/mavros/imu/data' OR '/mavros/local_position/pose') compares to the Vicon estimated orientation.