ROB-6213 Project 1

Due: Tuesday, March 22, 2022, 11:59 pm

1 Overview

It is time to put together everything that you've learned in this course until today! In this phase, you'll use vicon and IMU for estimation. For this phase of the project, you will implement a Extended Kalman Filter (EKF) for state estimation. You will use the body frame acceleration and angular velocity from the onboard IMU as your control inputs. The measurement will be given by the pose or velocity from the Vicon. The body frame of the robot is coincident with the IMU frame.

2 Sensor Data

The data for each trial is provided in a mat file. The `mat` file also contains Vicon data. The Vicon data is stored in two matrix variables, `time` and `vicon`. The `time` variable contains the timestamp while the `vicon` variable contains the Vicon data in the following format:

$$\begin{bmatrix} x & y & z & \text{roll} & \text{pitch} & \text{yaw} & v_x & v_y & v_z & \omega_x & \omega_y & \omega_z \end{bmatrix}^T$$

The on board processor of the robot collects synchronized camera and IMU data and sends them to the desktop computer. At this stage, the camera data should not be used. The sensor data is decoded into standard MATLAB format. Note that since the sensor data is transmitted via wireless network, there may or may not be a sensor packet available during a specific iteration of the control loop. A sensor packet is a *struct* that contains following fields:

```
sensor.is_ready    % True if a sensor packet is available, false otherwise
sensor.t           % Time stamp for the sensor packet, different from the Vicon time
sensor.omg         % Body frame angular velocity from the gyroscope
sensor.acc         % Body frame linear acceleration from the accelerometer
sensor.img         % Undistorted image.
sensor.K           % Calibration matrix of the undistorted image
sensor.id          % IDs of all AprilTags detected, empty if no tag is detected in the image
sensor.p0          % Corners of the detect AprilTags in the image,
sensor.p1          % the ordering of the corners, and the distribution of the tags
sensor.p2
sensor.p3
sensor.p4
```

3 Kalman Filter

In this project, you will use the Extended Kalman Filter (EKF) as presented during lecture 7 to estimated the position, velocity, and orientation, and sensor biases of an Micro Aerial Vehicle. To simplify the overall process, we have already made a skeleton code that is able to synchronize the data that you need to use. The Vicon velocity is given in the world frame, whereas the angular rate in the body frame of the robot. You will use the body frame acceleration and angular velocity from the on board IMU as your inputs. You will have to present two filter versions. For this reason the project has been dived in two parts in separate folders. In the first one, the measurement update will be given by the position and orientation from vicon, whereas in the second one you will use only the velocity from the Vicon. In the released code, two separate folders have been created to accommodate each part. The function init.m is called within the KalmanFilt_Part1.m and KalmanFilt_Part2.m functions to initialize the process and select the specific dataset. The data is already parsed. You are authorized to modify the for loops within KalmanFilt_Part1.m and KalmanFilt_Part2.m as well as the prediction and update functions defined within each folder for your filter. Moreover, you are not allowed modify the definition of the prediction and update functions. Any other type of modification is not allowed and will result in 0 score. In this way, the sensor data has already been parsed to facilitate the process. The Euler angles convention to use is ZYX. You can re-use the function eul2rotm if you need. You should have two measurement models, one for each part of the project. The first one for the position and orientation (part 1) and the second one for the velocity (part 2). In both parts, the process model is the same. The project should be completed using Matlab 2021b. We will only debug solutions made with that version. The use of another maltab version is at your own risk and no debugging will be performed if the code does not work.

4 Report

You need to summarize your results in a report submitted in pdf format and generated with latex or word. Please add on top of your manuscript your name and NYU ID. The report should not be more than 8 pages including plots. You will have to use the plot function that has been provided in the code to generate your results for the 3 given datasets. In addition to the results, please include your approach in terms of process and measurement models and any explanation you think is appropriate. Do not just write equation, but try to add your logic process and explain why and how you used the equations or models you have in your code. Moreover, briefly comment your plots and compare your results to the vicon data. The plot function already overlaps your filter estimates with the ground truth provided by the vicon.

5 Grade Policy and Submission

The overall score will be 100 and will be subdivided in the following way, part 1 (60 points), part 2 (30 points), and report quality and readability (10 points). Your code in each part should not take more than 1 minute to run. We will not consider submitted codes that go over this running time. Do not modify any part of the code except the for loop with the prediction and update step functions and the dataset number to test the 3 dataset provided. Any other type of modification will result in 0 points. All the files, including code and report, should be submitted in an unique zip file.