UDACITY

‹ Return to "Flying Car" in the classroom

DISCUSS ON STUDENT HUB

Building an Estimator

REVIEW

CODE REVIEW

HISTORY

Meets Specifications

Excellent Work Ern!

Congratulations on successfully 'Building an Estimator' and completing the the Flying Car and Autonomous Flight Engineer Nanodegree Program!

You have shown that you are mastering the skill to autonomous flight software engineering! I urge you take a look at the (Optional) Fixed-Wing Aircraft Project as you continue to build your skills code a fixed-wing aircraft, and then implement solutions to a significantly more challenging control problem. Keep up the hard work in all that you do; you can do whatever you set your mind too. Brilliant!

If you're interested in some additional reading material for the course, please refer to the following:

Indoor Navigation Strategies for Aerial Autonomous Systems: by Laura Elena Munoz Hernandez, Pedro Castillo-Garcia, and Pedro Garcia Gil

Planning Algorithms: by Steven M. LaValle. The following link is for the free e-book version: http://planning.cs.uiuc.edu/

Autonomous Flying Robots: Unmanned Aerial Vehicles and Micro Aerial Vehicles: by Daisuke Nakazawa, Farid Kendoul, Kenzo Nonami, Satoshi Suzuki, and Wang Wei

Probabilistic Robotics: by Dieter Fox, Sebastian Thrun, and Wolfram Burgard: Probabilistic Robotics: Early Draft

Writeup

The writeup / README should include a statement and supporting figures / images that explain how each rubric item was addressed, and specifically where in the code each step was handled.

Great work! Your writeup is thorough and you have included images for the flight results and charts for estimation and prediction for all of the sensor values.

Implement Estimator

The calculated standard deviation should correctly capture ~68% of the sensor measurements. Your writeup should describe the method used for determining the standard deviation given the simulated sensor measurements.

Excellent! You have exceeded the requirement for the standard deviation.

The improved integration scheme should result in an attitude estimator of < 0.1 rad for each of the Euler angles for a duration of at least 3 seconds during the simulation. The integration scheme should use quaternions to improve performance over the current simple integration scheme.

Perfect! You have implemented quaternions in the derived integration. You have exceeded the requirement for maintaining the required accuracy for more than three seconds.

The prediction step should include the state update element (`PredictState()` function), a correct calculation of the Rgb prime matrix, and a proper update of the state covariance. The acceleration should be accounted for as a command in the calculation of gPrime. The covariance update should follow the classic EKF update equation.

Excellent work! The RGBPrime matrix is correctly derived and the state covariance is updated. The acceleration components are properly accounted for and the algorithm follows a standard EKF implementation.

The update should properly include the magnetometer data into the state. Note that the solution should make sure to correctly measure the angle error between the current state and the magnetometer value (error should be the short way around, not the long way).

Wonderful! You have properly bounded the magnetometer and integrated the sensor's results into the state.

The estimator should correctly incorporate the GPS information to update the current state estimate.

YES!
The GPS information is correctly fed into the current state estimate.

Flight Evaluation

For each step of the project, the final estimator should be able to successfully meet the performance criteria with the controller provided. The estimator's parameters should be properly adjusted to satisfy each of the performance criteria elements.

Excellent! All performance criteria have been successfully passed.

The controller developed in the previous project should be de-tuned to successfully meet the performance criteria of the final scenario (<1m error for entire box flight).

DOWNLOAD PROJECT

RETURN TO PATH

Rate this review