UDACITY

PROJECT

Building an Estimator

A part of the Flying Car Nanodegree Program

PROJECT REVIEW | CODE REVIEW | NOTES

Meets Specifications

You've done a **fantastic job** on this project!

Keep up the great work 👍

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.**
	🚀 Nicely done writing a comprehensive write-up including statements to explain how each rubric item was addressed and figures to support each statement!

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.**

🚀 The calculated standard deviation does correctly captures ~67% of the sensor measurements and the writeup does include how the standard deviations were calculated from previously recorded data.

✓ **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.**

🚀 The integration scheme does result in an attitude estimator of < 0.1 rad for each Euler angles at least 3 seconds during the simulation. Well done using quaternions to improve the performance!

✓ **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.**

🚀 The Rgb prime matrix is being correctly calculated and the state covariance has been properly updated by following the classic EKF update equation.

✓ **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).**

🚀 Magnetometer data has been properly included in the state and the error is on the short way around.

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

🚀 Good job incorporating the GPS information to update 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.**
	🎉 Awesome job meeting all of the performance criteria with the provided controller! The estimator parameters have been properly adjusted to satisfy all of the performance criteria.

✓	**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).**