

Project 4: Estimation

Matthias Schmitt

February 6, 2021

1 Standard deviation of the measurement noise

Following <https://github.com/udacity/FCND-Estimation-CPP#step-1-sensor-noise> and plugging the resulting time series into Excel, I got $\sigma_{acc} = 0.491$ and $\sigma_{pos} = 0.7$, very close to the values in the config file.

2 Improved rate gyro attitude integration

In `UpdateFromIMU()`, we integrate body rates using `IntegrateBodyRate` (which already does the multiplication and updates the attitude variable in place).

Then we only have to normalize yaw. The fusion and update was provided (corresponding e.g. to equation (41) in the *Estimation for quadrotors* paper – henceforth just ‘the paper’).

3 Implementation of the prediction step for the estimator

First we implement the `PredictState()` function. It is totally confusing that *gyro* is given as a parameter to this function – that should be removed as one may keep looking for the ‘mistake’ of not needing it (like I did).

R'_{bg} is given by equation (52). g' is computed according to equation (51) in the paper. The covariance update is done in `Predict()` corresponding to the formula

$$\Sigma_{upd} = g' \times \Sigma \times g'^T + Q$$

which is part of the Kalman algorithm given in section 3 of the paper.

4 The magnetometer update

Section 7.3 of the paper provides all the formulas we need. The actual update is already provided, we just have to fill the estimated state `zFromX` and initialize `h'`. Last step before calling `Update` for the state re-estimation is to normalize the yaw estimate by adding or subtracting 2π depending on the current difference between the measurement and the current yaw estimate.

5 The GPS update

Section 7.3 of the paper provides all the formulas we need. The actual update is already provided, we just have to fill the estimated state `zFromX` and initialize `h'`.