Unmanned Aircraft Design, Modeling and Control
Extended Kalman Filter

For questions: slynen@ethz.ch

Abstract

This exercise is derives and implements an attitude estimator using the indirect extended Kalman filter formulation. Sensor inputs are a gyroscope and an orientation estimate from a camera..

Note: The tasks are given as TODOs in the file `attitude.m`. See below for hints about the different steps of the exercise.

1 Getting familiar with the simulation environment

a) Download and unzip the archive containing the exercise materials from the lecture homepage. Start Matlab. **Add the subfolder `helper_files` to your path**.

b) In the subfolder EKF there is the simulation environment. It consists of:

- `attitude.m` Defines the simulation parameters and contains your tasks. This file needs to be executed to run the simulation.
- `plot_trajectory.m` Plots the trajectory we are simulating.

c) In the subfolder `helper_files` you can find methods for quaternion math (the ones you wrote in your last exercise) and some other generic helpers:

- `find_closest.m` Finds the elements in another sequence that has minimal distance to a given element.
- `mulquat.m` Computes the resulting quaternion from the operation $q \otimes p$.
- `invertquat.m` Computes the quaternion inverse (complex conjugate).
- `quat2rotvel.m` Converts a rotation within a time-span dt to respective rotational velocity.
- `quat2rot.m` Converts quaternion to respective rotation matrix as in $C_W B = [f(q_W B)]_{3x3}$.

- `quat2rpy.m` Converts quaternion to roll-pitch-yaw for plotting.
- `skew.m` Computes $\lfloor v_X \rfloor$ for the 3x1-vector v.
- `quat_skew.m` Computes $\lfloor q_X \rfloor$ for the 3x1-imaginary part of the quaternion q.

In the data file `realdata.mat` you find:

- `q_CW_m`: the measured orientation of the world frame of reference w.r.t. the camera frame of reference q_{CW_m} over time
- `t_q_CW_m`: the time stamps corresponding to q_{CW_m}
- `omega_m`: the measured rotational velocity of the IMU over time
- `t_omega_m`: the time stamps corresponding to ω_m
- `q_WI_gt`: ground truth attitude $q_{WI_{gt}}$ over time
- `t_q_WI_gt`: the time stamps corresponding to $q_{WI_{gt}}$.

d) Visualize the trajectory by running: `plot_trajectory.m`

2 Hints for solving the tasks:

a) Filter initialization:

- The IMU and camera are fixed together as a rigid body.
- The rotation between them is $q_{IC} = [0.0020, -0.0330, -0.0050, 0.9994]^T$ with: $\vec{v}_C = C(q_{IC})^T \vec{v}_I$
- We would like to estimate the IMU attitude w.r.t. the world frame of reference q_{WI}.
- We know nothing about the IMU bias so the best initial estimate is that there is none.

b) State and covariance propagation:

- The manufacturer states an angular velocity noise characteristic of $4.5 rad/\sqrt{h}$.
- Assume the bias noise to be 100 times smaller than the angular velocity noise.
- ${}^dF$ and ${}^dQ$ has been already calculated for you in the framework. Details about the calculations can be found in the suggested quaternion tutorial.
- Implement the zeroth order quaternion integration presented in the lecture.

c) The update step:

- Assume the measurement noise to have a standard deviation of 0.1 rad.

- Derive the measurement matrix H.
- Calculate the error of your estimate vs. your measurement.
- Correct your state.