Unmanned Aerial Systems Techology (RMUAST) Spring 2017
University of Southern Denmark

Created 2017-03-09 by Kjeld Jensen, kjen@mmmi.sdu.dk

Module 5: Aircraft attitude sensing.

1 Practical information

1.1 Materials

- PC with internet access.
- Python
- Optional: SparkFun Razor and/or VectorNav VN-100 IMU.

1.2 Agenda

1. Practical information.
   - New lab equipment.
   - The mission route plan part will be postponed to a later module.
   - Presentations for next module.
2. Discussion of results from last week's assignment
3. Theory for this module
4. Laboratory assignments for this module

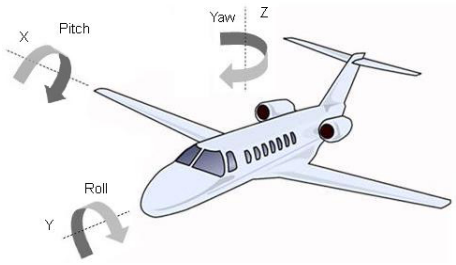
2 Preparation for next module

- Finish laboratory exercises from this module. Remember to submit your lab report no later than Saturday this week. If you submit via email please remember to include BOTH your team number and module number in the subject.
- Team 1 and 2 please prepare a presentation of your laboratory results for Monday next week.
- Nikolaj Bächler: Please prepare a presentation of *dronelov_2016.pdf*
- Stefan J. M-Møller: Please prepare a presentation of *bekendtg_droner_by.pdf*
- Petr Batek: Please prepare a presentation of *UAS Prototype Regulation final.pdf*

3 Theory covered in this module

3.1 Accelerometers

In aviation the three accelerometers contained in an Inertial Measurement Unit (IMU) are used for estimating the absolute pitch and roll (figure 1) orientation of the aircraft with respect to the Gravity vector.



Figur 1: *Pitch, roll og yaw axis of an aircraft.*

3.2 Gyros

The three gyro's in the IMU are used for measuring angular velocities about the pitch, roll and yaw axis of an aircraft (figure 1).

3.3 Kalman filter

The aircraft attitude may be estimated with a higher accuracy using a Kalman filter taking input from the accelerometers and gyros.

4 Exercises

4.1 Attitude sensing using accelerometers

The purpose of this exercise is to learn how to calculate the aircraft orientation based on these linear accelerations. The exercise is based on data sampled from a SparkFun Razor Inertial Measurement Unit (IMU) (figure 2).

4.1.1 Calculate pitch angle

The file `imu_razor_data_pitch_45deg.txt` contains a dataset that was sampled while the IMU was tilted forwards and back.

Use the Python script `imu_exercise.py` to perform a calculation of the pitch angle based on the accelerometer values. Use the equation 28 in the document *Tilt Sensing Using a Three-Axis Accelerometer.pdf*

Observe that the plot shows the expected output based on the description of the dataset.

4.1.2 Calculate roll angle

The file `imu_razor_data_roll_45deg.txt` contains a dataset that was sampled while the IMU was tilted to the side.

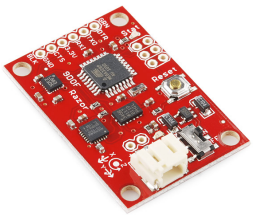
Use the Python script `imu_exercise.py` to perform a calculation of the roll angle based on the accelerometer values. Use the equation 29 in the document *Tilt Sensing Using a Three-Axis Accelerometer.pdf*

Observe that the plot shows the expected output based on the description of the dataset.

4.1.3 Accelerometer noise

The file `imu_razor_data_static.txt` contains a dataset that was sampled for approx 60 seconds while the IMU was held static on a reasonably level surface.

Using the same calculations as in 1. and 2. plot the pitch and roll angles based on this dataset.



Figur 2: *SparkFun Razor IMU.*

Figur 3: *VectorNav VN-100 IMU.*

Do the calculated angles show any significant noise or bias?

How could this be mitigated?

4.1.4 Low-pass filtering

Try adding a low-pass filter and see if you can reduce the noise.

How much delay do you then add to the estimation of the pitch and roll angle?

Is this acceptable given the update velocity of the stability controller?

4.1.5 Limitations of Euler angles

The equations 28 and 29 used above are not able to describe all states of orientation, this is well known as *Gimbal lock*. Which particular orientations may cause problems?

4.1.6 Extra: Quaternions

Implement a pitch and roll estimation ot suffering from the gimbal lock limitations using Quaternions.

4.2 Gyro measurements

The purpose of this exercise is to learn how to integrate the angular velocities measured by a gyro to obtain a relative measure of the angle about that axis.

The exercise is based on data sampled from a SparkFun Razor Inertial Measurement Unit (IMU) (figure 2) and a VectorNav VN-100 IMU (figure 3).

4.2.1 Calculating relative angle

The file `imu_razor_data_yaw_90deg.txt` contains a dataset that was sampled while the IMU was turned $90\,^\circ$ clockwise, then the IMU was turned $90\,^\circ$ counter-clockwise.

Use the Python script `imu_exercise.py` to perform a numerical integration of the angular velocity about the z-axis ω_z to obtain a measure of the relative angle. Use the actual time since the previous received angular velocity for the integration.

Observe that the plot shows the expected output based on the description of the dataset.

4.2.2 Static data

The file `imu_razor_data_static.txt` contains a dataset that was sampled for approx 60 seconds while the IMU was held static on a reasonably level surface.

Perform the same numerical integration of the angular velocity and observe the result.

4.2.3 Observing bias

The output from the integration shows that the relative angle is drifting. The drift is the visible effect of a bias on the angular velocity.

Try to estimate the bias and subtract it before integrating.

4.2.4 Bias sources

Consider the potential sources of noise and bias for a gyro?

4.2.5 Extra: Integration using average time

Perform a similar numerical integration but this time use a calculated average time between updates in the file.

Consider why there is a difference and what gives the most accurate result?

4.2.6 Extra: Record more datasets

Use the Python script `nmea_data_logger.py` and an IMU to record other datasets.

4.3 Kalman filter

The purpose of this exercise is to learn how a Kalman filter will improve the attitude estimation using the accelerometers and gyros as input.

4.3.1 Implementing a scalar Kalman filter

Implement a scalar Kalman filter estimating the Pitch angle in the Python script `imu_exercise_kalman.py`. To do this you must add the calculation of pitch and roll from the accelerometers and then the gyro relative angles from the previous exercises.

Please notice that within the script two sections are clearly marked *Insert initialize code below* and *Insert loop code below*. You do not need to change anything in the file outside these sections.