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

Group 3: Jesper Bogh Poder, Petr Batek and Bjarki Sigurdsson

Module 5: Aircraft attitude sensing.

1 Exercises

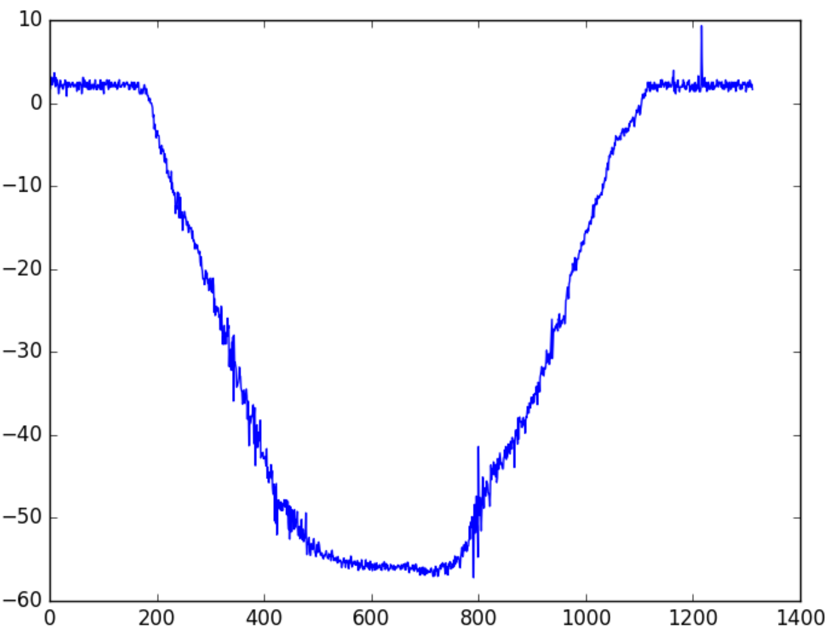
1.1 Attitude sensing using accelerometers

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 **??**).

1.1.1 Calculate pitch angle

Using the datafile `imu_razor_data_pitch_45deg.txt` and the script `imu_exercise.py`, which was provided, we have plotted the data using equation 1.

$$\phi = atan2(\frac{G_{py}}{\sqrt{G_{px}^2 + G_{pz}^2}}) \tag{1}$$

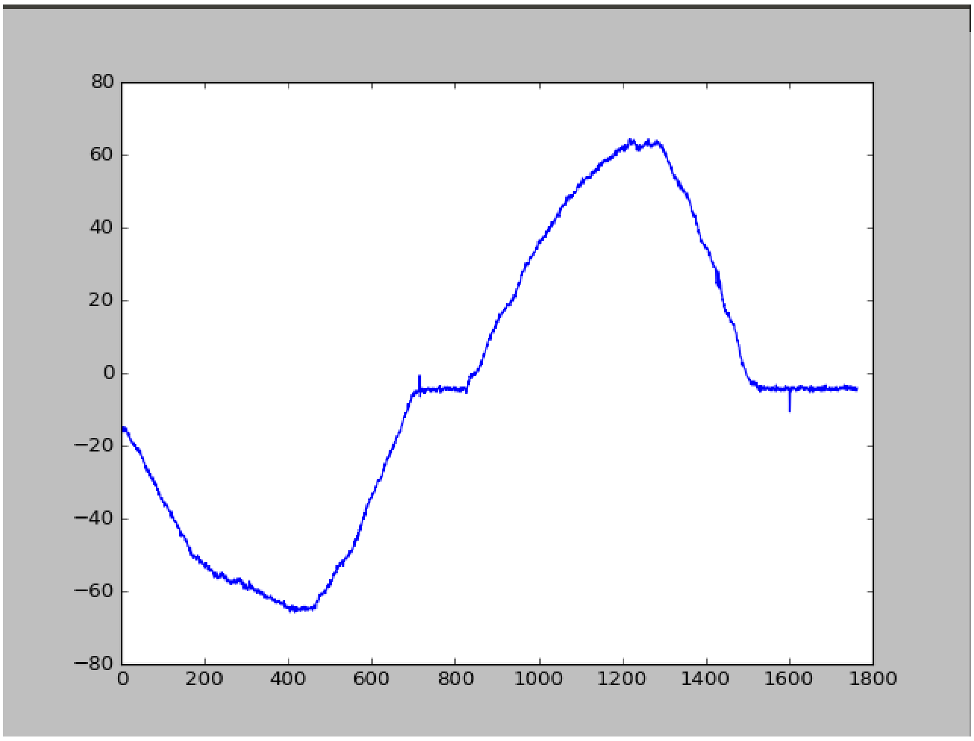


Figur 1: Plot showing the pitch angle versus the time

1.1.2 Calculate roll angle

Using the datafile `imu_razor_data_roll_45deg.txt` and the script `imu_exercise.py`, which was provided, we have plotted the data using equation 2. There is a bit of noise on the measurements but otherwise they seem reasonable.

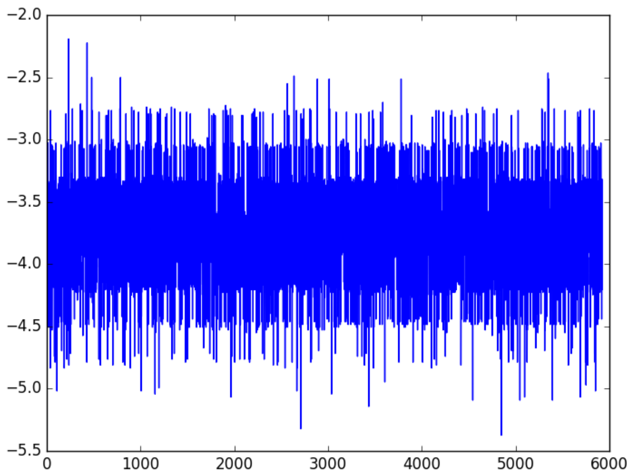
$$\theta = atan(\frac{-G_{px}}{G_{pz}}) \tag{2}$$



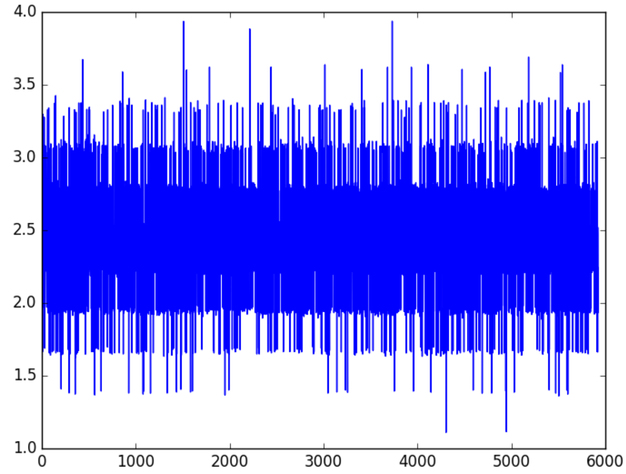
Figur 2: Plot showing the roll angle versus the time

1.1.3 Accelerometer noise

We used the data from `imu_razor_data_static.txt` to plot the sensor readings from a static accelerometer. Figures 3 and 4 show the pitch and roll angles, respectively. We see that both are quite noisy as well as having a bias of a few degrees. This could be mitigated via filtering.



Figur 3: Pitch noise from a static IMU as a function of time.

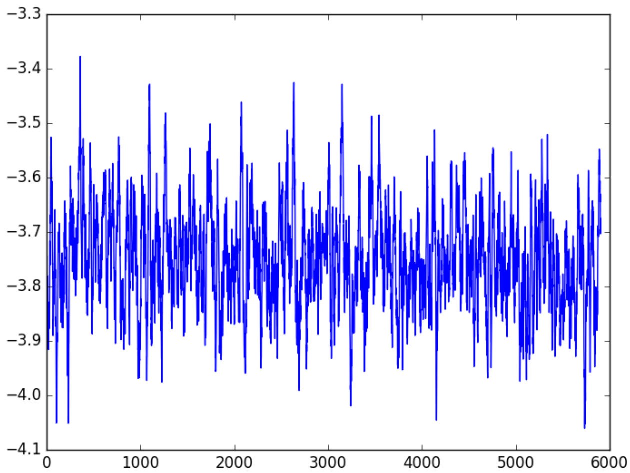


Figur 4: Roll noise from a static IMU as a function of time.

1.1.4 Low-pass filtering

Figure 5 shows the pitch noise from figure 3 after low-pass filtering using a box-car filter with a length of 20 samples. The magnitude of the noise has been considerably reduced while the bias remains. The downside of this – and the trade-off of performing more aggressive smoothing – is that the response of the system is delayed according to the chosen length of the filter.

The maximum output data rate is 3200Hz for which this particular filter introduces a delay of up to 6.25ms. This is already rather large for the purpose of stability control. This can be reduced by lowering the filter size but doing so reintroduces some of the noise.

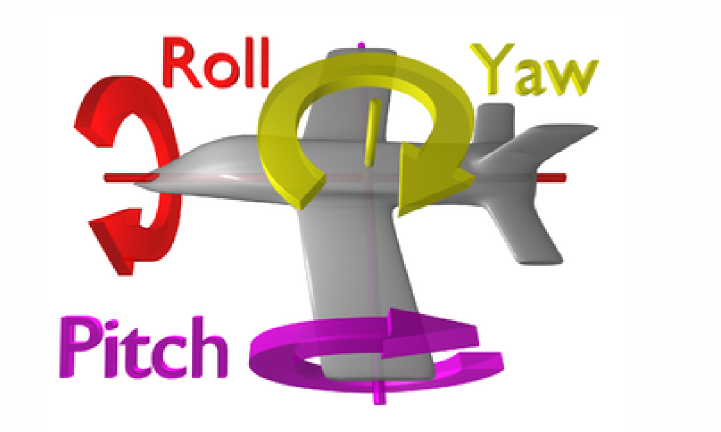


Figur 5: Pitch noise after low-pass filtering using a box-car filter with a length of 20 samples.

1.1.5 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?

When representing rotations using euler angles you can derive rotation matrices for each rotation and then multiply the three rotation matrices together in order to represent the full rotation. The problem with this representation can be illustrated using figure 6. If we were to have a 90 or -90 degrees pitch angle then we would align the y and z axis, which is called a gimbal lock.



Figur 6: Image showing the roll, pitch and yaw of an object.

1.1.6 Extra: Quaternions

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

1.2 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 **??**) and a VectorNav VN-100 IMU (figure **??**).

1.2.1 Calculating relative angle

Using the datafile `imu_razor_data_yaw_90deg.txt` and the script `imu_exercise.py`, which was provided, we have plotted the integrated angle signal. Integration was computed according to 3.

$$\psi[t] = \sum_{i=1}^{t} \texttt{gyro\_z}[i] * (t[i] - t[i-1]) \tag{3}$$

Figure 7 shows plot of integrated data. Results are as expected from assignment. First rotation in clockwise direction resulted in integration of -90° rotation and the second counter-clockwise rotation returned the sensor back to original orientation.

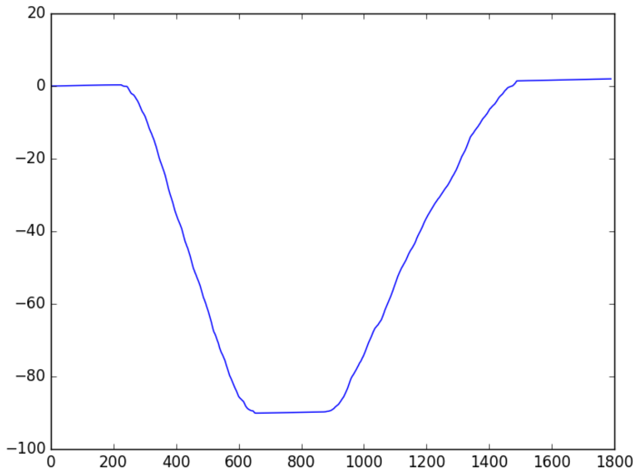
1.2.2 Static data

Plot of measurement integration while sensor was held static is shown in figure 8. One can see apparent drift in integrated orientation.

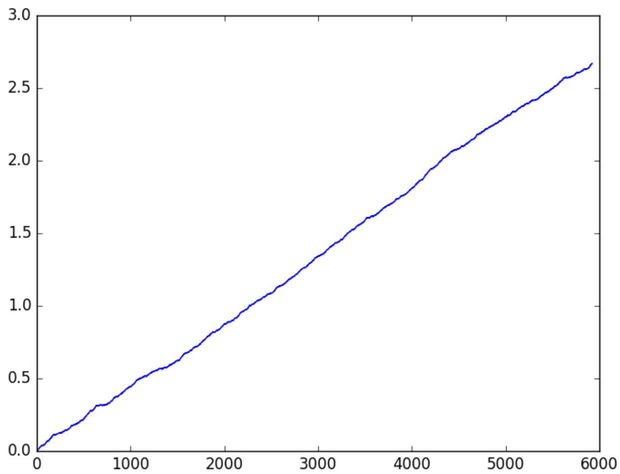
1.2.3 Observing bias

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.



Figur 7: Plot showing the integrated yaw angle from gyro



Figur 8: Plot showing the integrated yaw angle while gyro was held static

After inspection of figure 8 we can conclude that the gyro drifted approximately $2.6°$ in 60 seconds. Bias of the gyro is its average output, when the sensor stands still. Unit of bias are dps(degrees per second). It can be estimated from previous dataset according to 4.

$$\text{Bias}[dps] = \frac{\text{drift}[°]}{\text{time}[s]} = \frac{2.6}{60} = 0.043[dps] \tag{4}$$

1.2.4 Bias sources

Potential sources of bias can be following:

- calibration errors
- switch-on to switch-on - this error sources is coupled with temperature. After the sensor is turned on, it starts to warm up.
- temperature - changes in environment temperature can have effect on bias

1.2.5 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?

1.2.6 Extra: Record more datasets

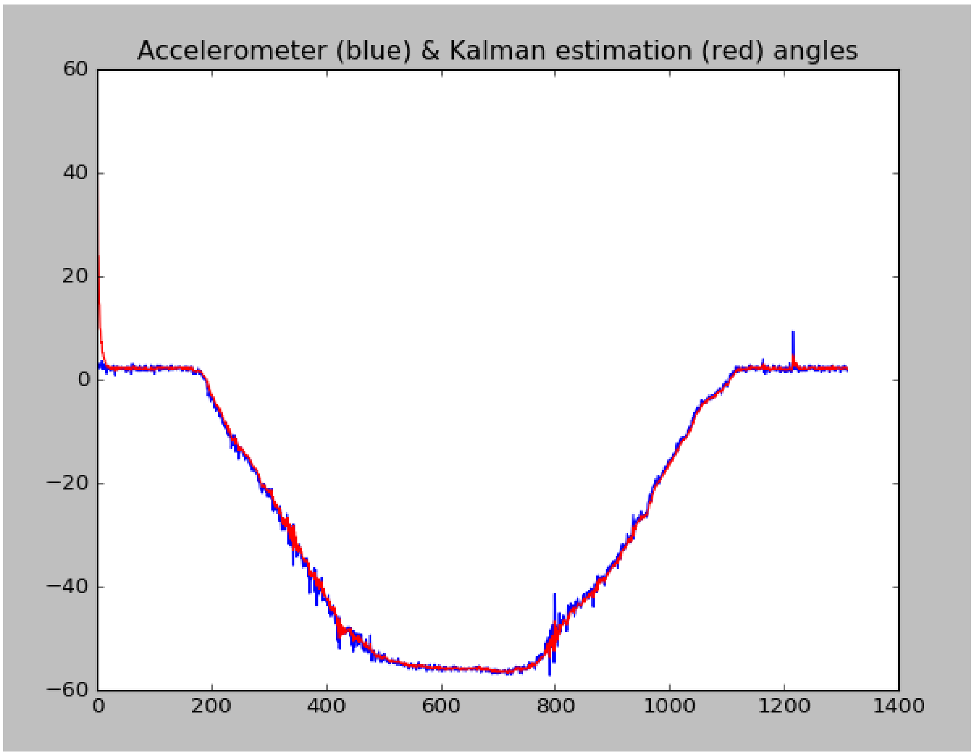
Use the Python script `nmea_data_logger.py` and an IMU to record other datasets.

1.3 Kalman filter

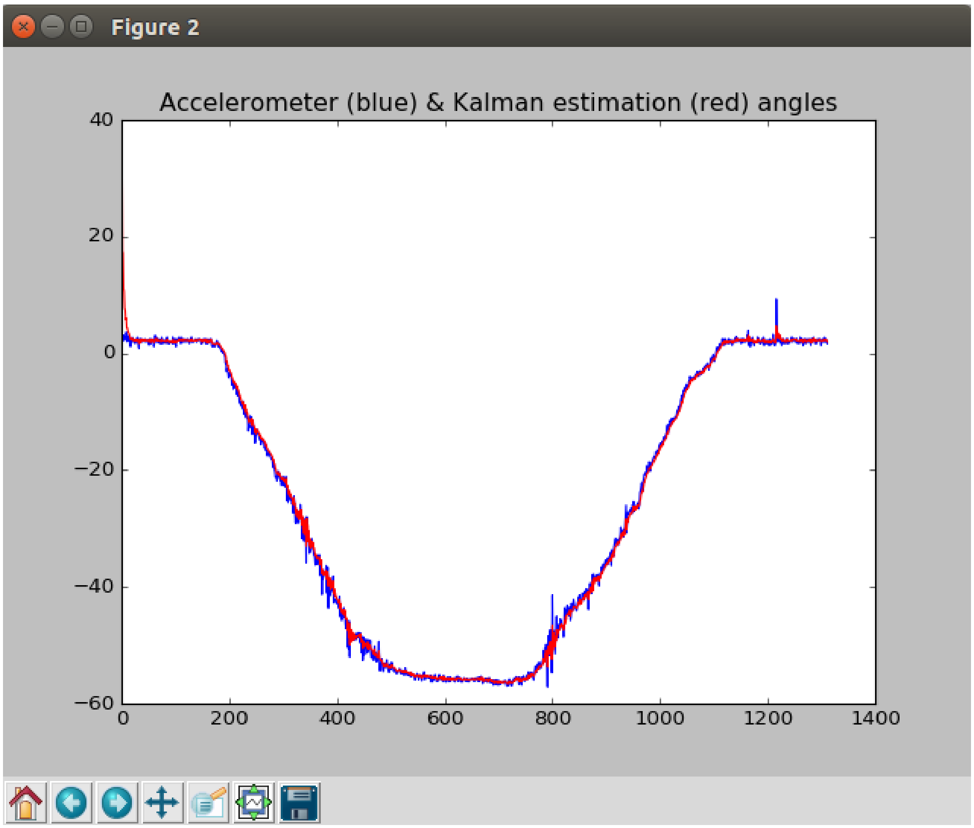
The purpose of this exercise is to learn how a Kalman filter will improve the attitude estimation using the accelerometers and gyros as input.

1.3.1 Implementing a scalar Kalman filter

Figure 9 and 10 shows our implemented Kalman filter, where we see that a larger initial variance makes the filter converge faster.



Figur 9: Plot showing the implemented kalman filter with $\rho_0 = 10$ *and* $x_0 = 5$



Figur 10: Plot showing the implemented kalman filter with $\rho_0 = 15$ *and* $x_0 = 5$