Project 2 – Sensor Analysis and Filtering

Course: Unmanned Aerial Vehicles (UAVs)

M.Sc. in Aerospace Engineering

Professor: Bruno Guerreiro

April 2024

Group 3

José Duarte Dias Corvo, student number 67118

Rodrigo Miguel Santos Silva Pardela Veríssimo, student number 67133

Vasco Rafael da Ponte Luís Nunes, student number 67304

NOVA SCHOOL OF
SCIENCE & TECHNOLOGY

Abstract

“This project aims at characterizing the sensor data obtained from the Crazyflie drone, and design Kalman filtering solutions using this data to obtain meaningful state estimates. In a first part, several experimental data sets will be used to characterize the sensors, both at rest and during flight. In a second part, Kalman filter designs will be tested for partial state estimation. Finally, a third part will focus on the design of an extended Kalman filter for the full state of the drone. Together with the first project, the techniques addressed in this project are necessary to achieve state feedback control as it will be addressed in the last project.” – Bruno Guerreiro, 23rd April 2024

NOVA SCHOOL OF SCIENCE & TECHNOLOGY

Index

NOVA SCHOOL OF SCIENCE & TECHNOLOGY

List of Figures

1. Introduction

As cited by Captain Brian Tice from United States Air Force [1], Unmanned Aerial Vehicles (or UAVs) are powered vehicles that do not carry human operators. These vehicles started to be developed on the twentieth century for military use, but have evolved into other uses such as agriculture, entertainment, or product deliveries.

Nowadays it's still a technology under intense development, where the safety and security regarding the usage of these vehicles in our society is a big concern. Operating an unmanned aerial vehicle is common when a human is in charge of the commands, the use of fully autonomous drones in complex and challenging environments is still underdeveloped and under intense research by universities and the relevant industries that could benefit from this.

In this second project (from a series of three projects), the group was designated to consider the use of filtering techniques to obtain state estimates from the noise sensors on board of a micro aerial vehicle called Crazyflie 2.1.

Figure 1 – Crazyflie 2.1 by Bitcraze Store

A Micro Aerial Vehicle (or MAV) is a man-portable aerial vehicle which, by EASA's (European Union Aviation Safety Agency) definition, weighs less than 250 grams [2].

The coding resolution of this project is in the format of a GitHub repository after working on the software MATLAB R2021a.

2. Project 2 Subjects and Goals

In the first part of this project, the group will need to characterize the sensors on board of the drone Crazyflie 2.1. In order to achieve this, several experimental datasets were provided by Professor Bruno Guerreiro:

- L2Data1 – Standing still with motors off;
- L2Data2 – Take-off, hover, and land;
- L2Data3 – Take-off, commute in axis x, and land;
- L2Data4 – Take-off, commute in axis y, and land;
- L2Data5 – Take-off, commute in axis z, and land;
- L2Data6 – Take-off, lemniscate trajectory ("∞"), and land.

Considering an ENU (East-North-Up) local tangent plane centred at the drone area of the CybAer laboratory in FCT-UNL (Faculdade de Ciências e Tecnologias da Universidade Nova de Lisboa), with the respective coordinates (Latitude 38.660319ºN, Longitude -9.204972ºW) as the inertial frame. It will be assumed that the drone is a rigid body in 3-D space with the relevant external forces acting on the centre of mass of the drone. In order to simplify, it will also be assumed that the body frame is also located at the centre of mass of the drone, considering the constant of mass ("$\boldsymbol{m} = 0.032\ kg$", as mentioned by Professor Bruno Guerreiro during the class on the 21st of march 2024 of "Unmanned Autonomous Vehicles") and moment of inertia's matrix ("$\boldsymbol{J} = diag(J_x, J_y, J_z)$").

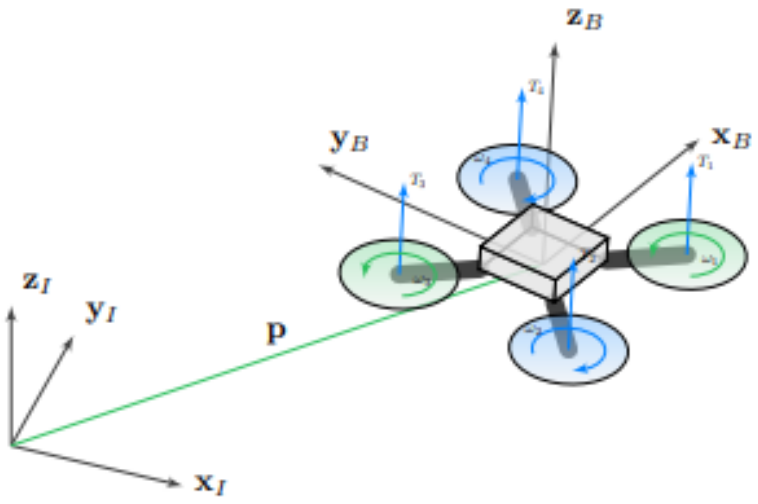

Figure 2 – Crazyflie 2.1 reference frames and configuration

In order to develop the project, the group will consider the following state-space variables of the drone:

- The drone's 3-D position (from the centre of mass "B" and origin of the body frame) relative to the inertial frame "I" ("$\boldsymbol{p} = {}^{I}p_B \in \mathbb{R}^3$");
- The velocity of the drone described on the body frame ("$\dot{\boldsymbol{p}} \in \mathbb{R}^3$");
- The attitude of the body frame regarding the inertial frame ("$\boldsymbol{R} \in SO(3)$"), parametrized by the $Z-Y-X$ Euler angles vector ("$\boldsymbol{\lambda} = [\phi \quad \theta \quad \psi]^T$");

- The relative angular velocity acting on the body frame ("$\boldsymbol{\omega} \in \mathbb{R}^3$").

Regarding the forces acting on the drone, it was considered the Earth Gravity Acceleration ("$\boldsymbol{g_{earth}} = 9.82\ [^m/_s]$") and Atmospheric Density ("$\boldsymbol{\rho_{earth}} = 1.217\ \left[^{kg}/_{m^3}\right]$") [3], the sum of the four forces of propulsion generated by the rotors ("$\boldsymbol{F_p} = F_{p_1} + F_{p_2} + F_{p_3} + F_{p_4}$") and the sum of the four moment vectors of the propulsion generated by the rotors thrust ("$\boldsymbol{n_p} = n_{p_1} + n_{p_2} + n_{p_3} + n_{p_4}$"). The maximum thrust needed by the sum of the forces of propulsion is obtained by the formula "$T_{max} = (m_{drone} + m_{maximum\ load}) \times g_{earth} \Leftrightarrow$ $\Leftrightarrow T_{max} = (0.032\ kg + 0.015\ kg) \times 9.82\ ^m/_s = 0.46154\ N$".

The group, in order to obtain the most accurate simulation of the drone, verified the external dimensions of the Crazyflie 2.1.

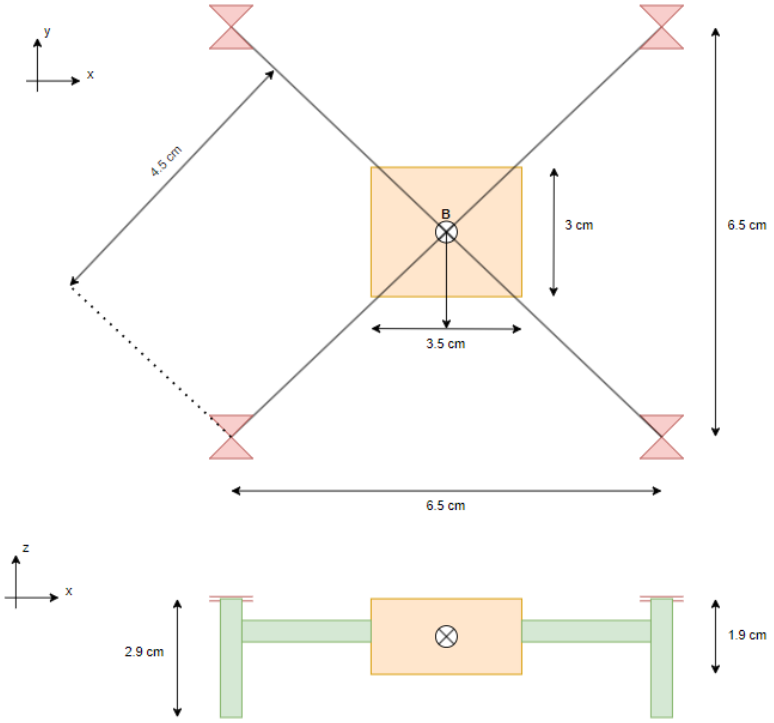

Figure 3 – Sketch displaying the dimensions of the Crazyflie 2.1 drone

Regarding the rotors, it can be assured that they have a vertical propulsion system (90º or $\frac{\pi}{2}\ rad$ regarding the axis angle xz).

The goals imposed for this part of the project are the following:

- **Goal number 1.1:** For the L2Data1 dataset given, plot the accelerometer and rate gyro data in all three axes, and obtain the statistical characterization of this data.
- **Goal number 1.2:** For the L2Data2 dataset given, focusing on the stable hover part, plot accelerometer and rate gyro in all three axes, and obtain the statistical characterization of this data.
- **Goal number 1.3:** Considering the accelerometer measurement equations, find the relation with the roll and pitch angle measurements (inclinometer).

- **Goal number 1.4:** Using the datasets L2Data3 and L2Data4 given, compare the result of the previous inclinometer measurements with the Crazyflie angle measurements available in the datasets.

Building on the previous sensor analysis, in the second set of goals the group will design, analyse and implement a linear Kalman filter for partial attitude estimation, with the following goals:

- **Goal number 2.1:** Design a linear Kalman filter to estimate both pitch and roll based on the previous measurements, defining clearly the system equations and the output/measurements equations.
- **Goal number 2.2:** Analyse the filter observability and closed-loop stability, discussing the inclusion of measurement biases.
- **Goal number 2.3:** Implement the filter and compare its performance with the attitude data from the Crazyflie, using two or more datasets (including L2Data6). For the implementation of the filter, the group will have the opportunity to implement in MATLAB R2021a.
- **Goal number 2.4:** Discuss the results when considering (or not) measurement biases in the state vector, as well as the choice of the process and measurement noise covariance matrices.

Regarding the section of extended Kalman filters, the previous analysis will be repeated, regarding now the nonlinear relations between accelerometers and the attitude.

For this section, the group will need to conclude another set of goals:

- **Goal number 3.1:** Design an extended Kalman filter to estimate both pitch and roll based on the original accelerometers and rate gyro measurements, defining clearly the (possibly nonlinear) system equations and the output/measurements equations.
- **Goal number 3.2:** Analyse the filter observability by linearizing the system and observation equations around a relevant point.
- **Goal number 3.3:** Implement the filter and compare its performance with the linear filter and with the attitude data from the Crazyflie, using two or more datasets (including L2Data6).
- **Goal number 3.4:** Discuss the results when considering (or not) measurements biases in the state vector, as well as the choice of the process and measurement noise covariance matrices.

3. Development of the goals imposed

In order to obtain the goals proposed by coding in MATLAB R2021a, the group based its research on the examples provided by Professor Bruno Guerreiro on the 2023/2024 version of FCT/UNL Moodle.

3.1. Goal number 1.1

For the L2Data1 dataset given, plot the accelerometer and rate gyro data in all three axes and obtain the statistical characterization of this data.

Using the dataset L2Data1 provided by Professor Bruno Guerreiro to plot the accelerometer and rate gyro data in all axes (x, y and z), the following figures are obtained.

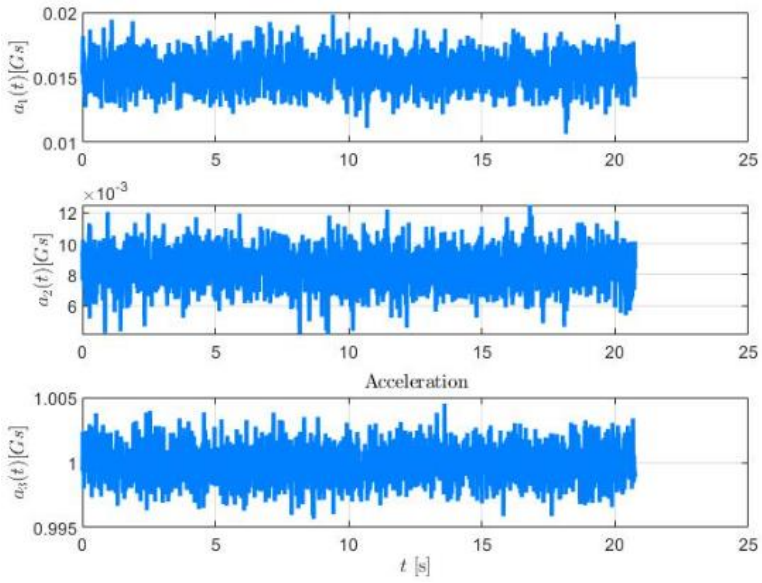

Figure 4 – Plot of the dataset L2Data1 for the Acceleration ("a $[Gs]$") in all axes over Time ("t $[s]$")

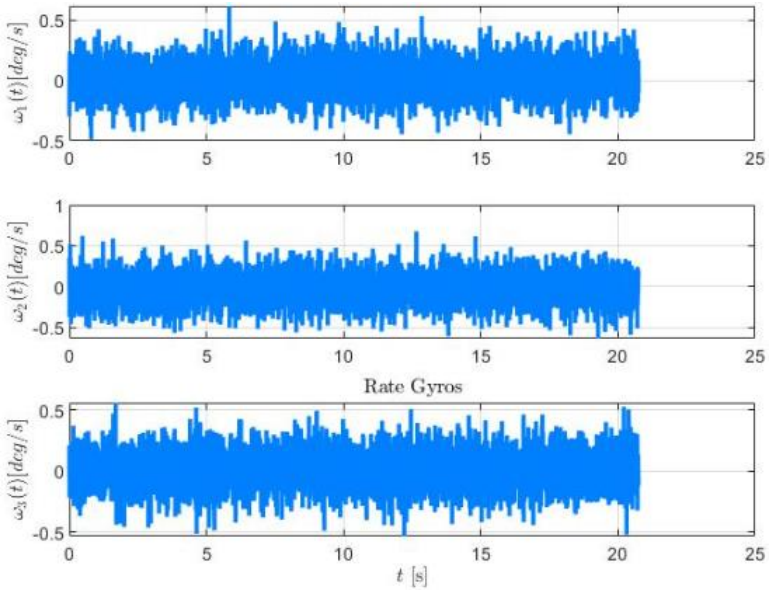

Figure 5 – Plot of the dataset L2Data1 for the Rate Gyro ("ω [deg/s]") in all axes over Time ("t [s]")

In order to characterize the state, it was obtained the functions for the mean and the covariance of the operating states of the drone.

For the dataset L2Data1, the following values were obtained:

- $Mean\ roll\ angle = \bar{x}_{roll} = 0.4630\ rad$;
- $Mean\ pitch\ angle = \bar{x}_{pitch} = 0.8492\ rad$;
- $Covariance\ roll\ angle = P_{roll} = 0.0151\ rad^2$;
- $Covariance\ pitch\ angle = P_{pitch} = 0.0256\ rad^2$.

3.2. Goal number 1.2

For the L2Data2 dataset given, focusing on the stable hover part, plot accelerometer and rate gyro in all three axes, and obtain the statistical characterization of this data.

In order to focus on the stable hover part, the use of the dataset L2Data2 given to plot the accelerometer and rate gyro data in all axes results in the following figures.

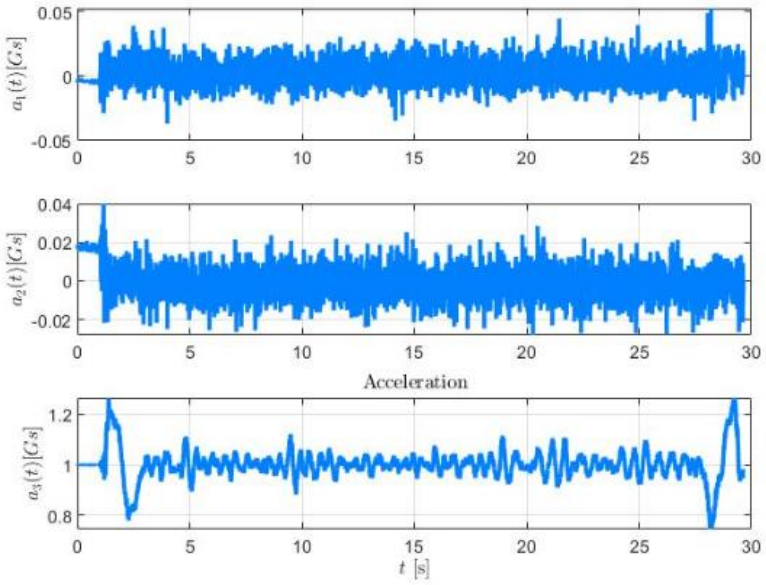

Figure 6 – Plot of the dataset L2Data2 for the Acceleration ("a $[Gs]$") in all axes over Time ("t $[s]$")

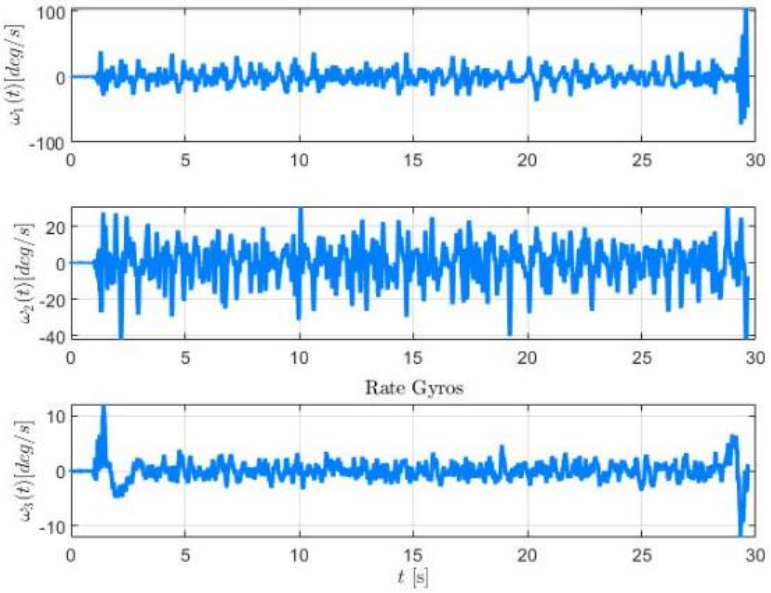

Figure 7 – Plot of the dataset L2Data2 for the Rate Gyro ("ω $[deg/s]$") in all axes over Time ("t $[s]$")

Similarly to what was developed in goal number 1.1, functions for the mean and for the covariance of the operating state of the drone with the dataset L2Data2 were obtained, with the following results:

- $Mean\ roll\ angle = \bar{x}_{roll} = 0.0729\ rad$;
- $Mean\ pitch\ angle = \bar{x}_{pitch} = 0.0181\ rad$;
- $Covariance\ roll\ angle = P_{roll} = 0.7679\ rad^2$;
- $Covariance\ pitch\ angle = P_{pitch} = 0.5938\ rad^2$.

3.3. Goal number 1.3

Considering the accelerometer measurements equations, find the relation with the roll and pitch angle measurements (inclinometer).

In most aerial vehicles, such as airplanes, helicopters, or rockets, it's possible to find a system named AHRS (Attitude and Heading Reference System) [4]. This system gives measurements as roll (ϕ) and pitch (θ) angles for the attitude, while the heading is given regarding magnetic measurements or a true North. This system usually consists in a three-axis accelerometer, a three-axis rate gyro and a three-axis magnetometer.

Assuming that the accelerometer remains stable regarding the vehicle, we can assure that the device is only measuring the acceleration due to gravity. This accelerometer can only measure the roll and pitch angles, being irrelevant to the measurement of heading.

The measurement of the accelerometer will be the following: "$\vec{a} = [a_x \quad a_y \quad a_z]$". Ignoring heading measurements/yaw angles (ψ), we obtain the following equation:

- $$\begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} = \begin{bmatrix} \cos(\phi) & 0 & -\sin(\theta) \\ \sin(\theta)\sin(\phi) & \cos(\phi) & \cos(\theta)\sin(\phi) \\ \sin(\theta)\cos(\phi) & -\sin(\phi) & \cos(\theta)\cos(\phi) \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix} \Leftrightarrow$$

$$\Leftrightarrow \begin{matrix} a_x = g * \sin(\theta) \\ a_y = -g * \sin(\phi)\cos(\theta). \\ a_z = -g * \cos(\phi)\cos(\theta) \end{matrix}$$

Resolving the equations for the roll and pitch angles:

- $\theta = \arcsin\left(\frac{a_x}{g}\right)$;
- $\phi = \arctan\left(\frac{a_y}{a_z}\right)$.

Knowing that, for a stationary sensor, "$g = \sqrt{a_x^2 + a_y^2 + a_z^2}$", we obtain the following expressions:

- $\theta = \arcsin\left(\frac{a_x}{g}\right) = \arcsin\left(\frac{a_x}{\sqrt{a_x^2+a_y^2+a_z^2}}\right)$;
- $\phi = \arctan\left(\frac{a_y}{a_z}\right)$.

3.4. Goal number 1.4

Using the datasets L2Data3 and L2Data4 given, compare the result of the previous inclinometer measurements with the Crazyflie angle measurements available in the datasets.

As it can be observed in the graphics of the measurements of the acceleration and the gyro rate provided by the drone Crazyflie 2.1, there are big discrepancies. These discrepancies are the result of the effects of noise and bias on the sensors used to obtain the roll and pitch, which in order can be used to obtain the acceleration of the drone.

The relative acceleration measured by the accelerometer can be obtained by "$a_m = a + b_a + n_a$", with "a_m" being the relative acceleration measured, "a" being the real acceleration of the drone, "b_a" being the effect of bias on the sensor and "n_a" being the effect of noise on the sensor of the vehicle.

The effects of bias and noise consists on the offset of the outputs of them, being approximately constant and derivative of different factors, such as imprecisions on the fabrication of the product or active and constant influences of other devices in the measurements of this sensor. The bias can always vary with time (degradation and aging of devices) or due temperature. This phenomenon can be described as "drift", such as there are calibration algorithms that have in mind these parameters.

The noise can be originated from different sources such as alternated current power supplies (AC), more accurately from the magnetic fields generated and their interference with the varying parameters of these fields with the variation of current from the power supplies, creating a residual tension (electromotive force) on the intrinsic circuits of the sensors. Another source of noise can be related to the temperature, implying changes in size of the components due to their thermal expansion coefficients, therefore implying inaccurate measurements seen as noise. The increase of the temperature can also mean that there's a transfer of energy into the electrons (responsible for the electrical currents), increasing the vibrations (kinetic energy increases), increasing the noise as well. If the resolution of the measured values of the sensors and the microcontroller (in particular their ADC's) are not ideal, these resolutions are responsible for another source of uncertainty in the measurements, increasing the noise.

It can be observed in the graphics of the angles measured over the acceleration that, in the case of the roll angle, the measurements have an elevated density of noise and a less smooth progress regarding the values from the Crazyflie 2.1 drone, meaning that the calculations present a high frequency around the axis x, which would be the ideal measurement and approximately constant in all progress along the time axis. These high frequency oscillations and non-constant amplitude consist in measurement noise, which will add up to the noise from the processing of values that also provokes oscillations around the reference value which can be derived from the actions of the drone in order to follow the reference value and counteract the noise that wasn't expected in the calculations of the drone dynamics (it's assumed that the drone already has implemented a control system).

The noise from the processing of values can be derived from phenomena such as variable drag forces or variable air density around the vehicle (the phenomena can be justified by changes of conditions unpredictable like the wind direction and strength).

Adding up to these conditions, possible variations in the real average roll angle (which it's assumed to be $0\,rad$) can be defined as part of the values for bias, being a constant offset on the graphic (assuming that the bias calculated before is also constant around certain points).

Regarding the pitch angles, it's noticeable the difference between both graphics, knowing that the values calculated don't represent values of the speeds that the drone can achieve. Therefore, it can be assumed that the drone is static in the air for the calculations of the pitch angles. Other than that, it is susceptible to the same phenomena as the roll angle, namely noise at high frequencies and biases, which although is difficult to observe, the noise appears to have approximately the same amplitude and bias.

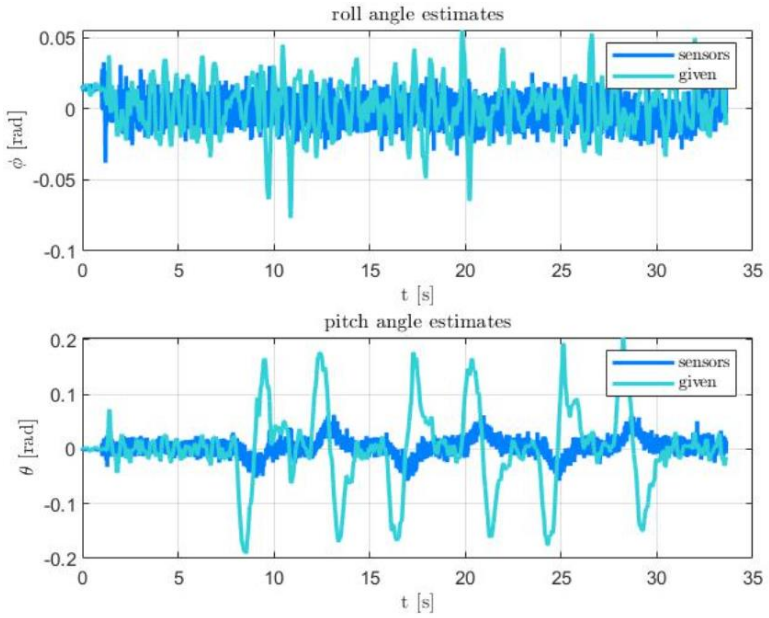

Figure 8 – Roll ("$\phi\ [rad]$") and Pitch ("$\theta\ [rad]$") Angles over Time ("$t\ [s]$") for the dataset L2Data3

Regarding the movement with the dataset L2Data4 with movement in the axis y (the analysis of the roll angle is similar as the analysis for the pitch angle), the graphic analysis is similar to the one above, even though it's possible to observe that there's a bigger influence of the sensor measured values in the roll angles and a lower influence of the sensor measured values in the pitch angles.

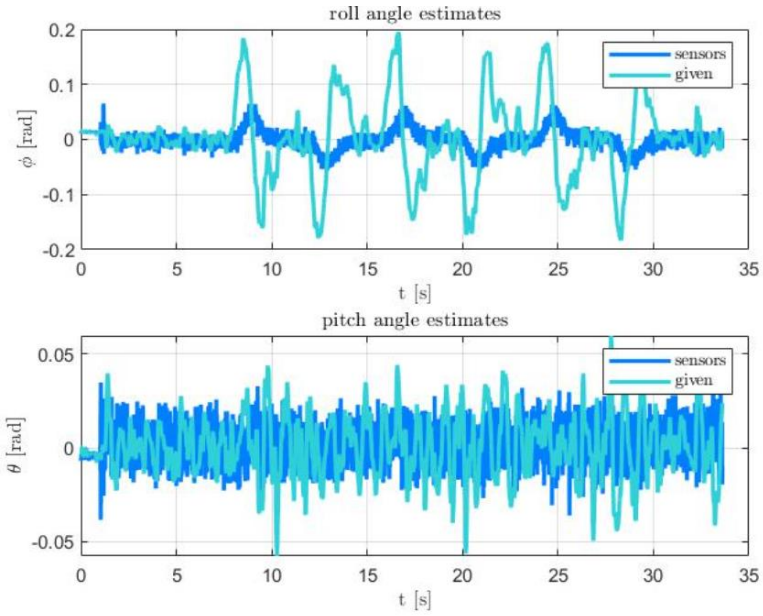

Figure 9 – Roll ("ϕ $[rad]$") and Pitch ("θ $[rad]$") Angles over Time ("t $[s]$") for the dataset L2Data4

3.5. Goal number 2.1

Design a linear Kalman filter to estimate both pitch and roll based on the previous measurements, defining clearly the system equations and the output/measurements equations.

The Kalman filter is a mathematical method that uses the measurements of variables along time (influenced by noise and other uncertainties) and generates results that tend to be similar over time to the real values of those variables.

In this goal it will be implemented a linear Kalman filter in order to obtain an approximation to the real values of the parameters developed between the goals 1.1 to 1.4.

As showed before, the accelerometer is an important device due to its use to measure roll and pitch angles. The outputs (which are the roll and pitch angles measured by the accelerometer) allow to estimate the states and the real values of the wanted parameters.

In order to estimate the roll and pitch angles, these two will be part of the state variables.

Assuming that "$h(x,u) = [\phi_m \quad \theta_m]^T$", the measurements equations are the following:

- $\phi_m = arctan\left(\frac{a_{y_m}}{a_{z_m}}\right)$;
- $\theta_m = arcsin\left(\frac{a_{x_m}}{g}\right)$.

In order to obtain the model of a linear sensor ("$y(t) = Cx(t) + v(t)$") and having in account that the drone won't show high angles on the equilibrium points chosen, it's possible to linearize the respective equations assuming that:

- "$\phi_m = arctan\left(\frac{a_{y_m}}{a_{z_m}}\right) = \frac{a_{y_m}}{a_{z_m}} = \frac{a_y + b_y + n_y}{a_{z_m}} = \phi + b_\phi + n_\phi$" due to the assumption that "a_{z_m}" is constant in all points of equilibrium:
- $\theta_m = arcsin\left(\frac{a_{x_m}}{g}\right) = \frac{a_{x_m}}{g} = \frac{a_x + b_x + n_x}{g} = \theta + b_\theta + n_\theta$;
- Therefore, "$h_{lin}(x,u) = \left[\left(\phi + b_\phi + n_\phi\right) \quad (\theta + b_\theta + n_\theta)\right]^T$".

Having in account all the outputs and functions described before and with the goal to estimate a value close to the real one of the roll and pitch angles, these parameters will be a part of the state. The dynamics of these angles are given by "$\dot{\lambda} = Q(\lambda) * \omega$".

In order to obtain a model for the linear installation ("$\dot{x}(t) = Ax(t) + Bu(t) + w(t)$"), and considering that the drone won't have high angles of movements in the equilibrium points:

- "$\dot{\lambda} = Q(\lambda) * \omega = \omega$", which will lead to "$\dot{\phi} = \omega_x$" and "$\dot{\theta} = \omega_y$".

Once these dynamics are obtained and after acknowledging that these dynamics are depending solely on "ω", they'll be considered to be part of the state. Both roll and pitch angles have biases, therefore the roll bias ("b_ϕ") and the pitch bias ("b_θ") will enter into the variables of the state vector.

Without knowing the value for the inputs, it's considered an equilibrium input equal to the real input, therefore the inputs won't impact the variation of states:

- "$\dot{x} = A\delta x + B\delta u + w, \delta u = 0$", assuming that "$\dot{\omega}_x = 0$", "$\dot{\omega}_y = 0$" and "$\dot{x} = Ax + w$".

The measurements of the roll and pitch angles contain biases, meaning that these biases initially enter the state vector:

- $\begin{bmatrix} \phi \\ \omega_x \\ \theta \\ \omega_y \\ b_\phi \\ b_\theta \end{bmatrix}$.

The biases vary due to certain factors ("drift") as it was described before, however, considering the equilibrium points considered they are assumed to be close to null:

- $\dot{b}_\phi = 0$;
- $\dot{b}_\theta = 0$.

After neglecting the biases, the linearized dynamics are the following:

- $f_{lin}(x,u) = \begin{bmatrix} \omega_x \\ 0 \\ \omega_y \\ 0 \\ 0 \\ 0 \end{bmatrix}$.

Simplifying the matrixial notation:

- "$\dot{x}(t) = Ax(t) + Bu(t) + w(t)$", which can be simplified into "$\dot{x} = Ax + w$" since "$B = 0_{4\times6}$" with bias ("$B = 0_{4\times4}$" without bias);
- "$y(t) = Cx(t) + v(t)$" which can be simplified into "$y = Cx + v$".

Since the linearized functions for both "$f(x,u)$" and "$h(x,u)$" are defined for their respective equilibrium points (knowing that these points don't vary much between them), via approximations (described before) it's known that it isn't needed to calculate the matrices "A", "B", "C" and "D" through derivations of the respective non-linearized functions for the states and inputs of the equilibrium points:

- $A = \frac{df_d(x_{eq})}{dx^T}$;
- $B = \frac{df_d(u_{eq})}{du^T}$;
- $C = \frac{dh_i(x_{eq})}{dx^T}$;
- $D = \frac{dh_i(v_{eq})}{dv^T}$.

In order to obtain the matrices A and C:

- "$\begin{bmatrix}\dot{\phi}_m\\ \dot{\omega}_x\\ \dot{\theta}_m\\ \dot{\omega}_y\\ \dot{b}_\phi\\ \dot{b}_\theta\end{bmatrix} = \begin{bmatrix}\omega_x\\ 0\\ \omega_y\\ 0\\ 0\\ 0\end{bmatrix} = A * \begin{bmatrix}\phi\\ \omega_x\\ \theta\\ \omega_y\\ b_\phi\\ b_\theta\end{bmatrix} + \mathrm{B} * \mathrm{u} + \mathrm{w} \Leftrightarrow \mathrm{A} = \begin{bmatrix}0&1&0&0&0&0\\0&0&0&0&0&0\\0&0&0&1&0&0\\0&0&0&0&0&0\\0&0&0&0&0&0\\0&0&0&0&0&0\end{bmatrix}$" with the biases included;
- "$\begin{bmatrix}\dot{\phi}_m\\ \dot{\omega}_x\\ \dot{\theta}_m\\ \dot{\omega}_y\end{bmatrix} = \begin{bmatrix}\omega_x\\ 0\\ \omega_y\\ 0\end{bmatrix} = A * \begin{bmatrix}\phi\\ \omega_x\\ \theta\\ \omega_y\end{bmatrix} + B * u + w \Leftrightarrow \mathrm{A} = \begin{bmatrix}0&1&0&0\\0&0&0&0\\0&0&1&0\\0&0&0&0\end{bmatrix}$" without the biases, resulting in the state vector "$x = \begin{bmatrix}\phi_m\\ \omega_x\\ \theta_m\\ \omega_y\end{bmatrix}$";
- "$y = \begin{bmatrix}\phi_m\\ \theta_m\end{bmatrix} = C * \begin{bmatrix}\phi\\ \omega_x\\ \theta\\ \omega_y\\ b_\phi\\ b_\theta\end{bmatrix} + \begin{bmatrix}n_\phi\\ n_\theta\end{bmatrix} \Leftrightarrow \mathrm{C} = \begin{bmatrix}1&0&0&0&1&0\\0&0&1&0&0&1\end{bmatrix}$" with the biases included;
- "$y = \begin{bmatrix}\phi_m\\ \theta_m\end{bmatrix} = C * \begin{bmatrix}\phi_m\\ \omega_x\\ \theta_m\\ \omega_y\end{bmatrix} + \begin{bmatrix}n_\phi\\ n_\theta\end{bmatrix} \Leftrightarrow \mathrm{C} = \begin{bmatrix}1&0&0&0\\0&0&1&0\end{bmatrix}$" without the biases.

It's worth taking notice that matrix "D" won't be used since the inputs don't influence the outputs.

It's also important to calculate the matrix "Q", knowing that this matrix can't be null due to the fact that would imply that the filter is not controllable (since the controlability matrix would become null). In order to not influence the dependency of the matrix lines for controlability, the matrix "$Q = I_{6\times6}$" with biases ("$Q = I_{4\times4}$" without the biases).

3.6. Goal number 2.2

Analyse the filter observability and closed-loop stability, discussing the inclusion of measurement biases.

With the matrices "A", "C" and "Q" defined, it's possible to analyse the observability, the controlability and the closed-loop stability of the Kalman filter.

To acknowledge the observability of the filter, the following equations need to be resolved:

- Including the bias in the filter, "$O_{(A,C)} = \begin{bmatrix} C \\ CA \\ CA^2 \\ CA^3 \\ CA^4 \\ CA^5 \end{bmatrix} = \left[\begin{array}{cccccc} 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ \hline 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ & & \underline{0_{2\times6}} & & & \\ & & \underline{0_{2\times6}} & & & \\ & & \underline{0_{2\times6}} & & & \\ & & 0_{2\times6} & & & \end{array}\right]$", with "$rank(O_{(A,C)}) = 4 \neq n = 6$", meaning that the Kalman filter with bias is not observable;
- Removing the biases from the filter, we obtain "$O_{(A,C)} = \begin{bmatrix} C \\ CA \\ CA^2 \\ CA^3 \end{bmatrix} = \left[\begin{array}{cccc} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \hline 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ & \underline{0_{2\times4}} & & \\ & 0_{2\times4} & & \end{array}\right]$", with "$rank\left(O_{(A,C)}\right) = 4 = n$", meaning that the Kalman filter without bias is observable.

By observing that the existence of biases on the roll and pitch angles generated imply the filter to be non-observable, the real value of these angles will always be aggregated to the respective biases, meaning that these angles are adding up with their respective biases, impeding a distinction from both when estimating. This means that these states are linearly dependent of each other.

The stability of a Kalman filter can be confirmed if three different conditions are confirmed:

- If the filter has uniform complete observability (related to "$(A, C)\ with\ R > 0$");
- If the filter has uniform complete controllability (related to "(A, Q)");
- If the initial state "$x(0) \sim \mathcal{N}(\overline{x_0}, P_0)$", with the covariance matrix "$P_0 = 0$".

In order to analyse the close-loop stability of the filter, the controlability needs to be analysed:

- Including the bias in the filter, "$C_{(A,Q)} = [Q \quad QA \quad QA^2 \quad QA^3 \quad QA^4 \quad QA^5] =$

$$= \left[\begin{array}{cccccc|cccccc|cccc} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & & & & \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & & & & \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0_{6\times6} & 0_{6\times6} & 0_{6\times6} & 0_{6\times6} \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & & & & \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & & & & \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & & & & \end{array}\right]$$

  obtaining "$rank\left(C_{(A,Q)}\right) = 6 = n$", meaning that the Kalman filter with bias is controllable;

- Disregarding the bias from the filter, "$C_{(A,Q)} = [Q \quad QA \quad QA^2 \quad QA^3] =$

$$= \left[\begin{array}{cccc|cccc|cc} 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & & \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0_{4\times4} & 0_{4\times4} \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & & \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & & \end{array}\right]$$

  ", obtaining "$rank\left(C_{(A,Q)}\right) = 4 = n$", meaning that the Kalman filter without bias is controllable.

Knowing that the filter is not observable with bias, the Kalman filter is not stable in its closed-loop, even though its controllable, which would mean that the values most likely won't converge to a value close to reality.

By removing the bias from the filter, the Kalman filter is both observable and controllable. Knowing that the covariance matrix "$P_0 = I_4 > 0$", it's possible to assure that this Kalman filter is stable in its closed-loop.

3.7. Goal number 2.3

Implement the filter and compare its performance with the attitude data from the Crazyflie, using two or more datasets (including L2Data6). For the implementation of the filter, the group will have the opportunity to implement in MATLAB R2021a.

In order to implement a Kalman filter, the biases for roll and pitch angles won't be considered.

The first step is to define the initial conditions to implement the filter, in particular the dynamics models ("A" and "B") and the sensors models ("C" and "D"). The models obtained initially do not possess biases in their states.

The uncertainty associated with the processing noise defined in the covariance matrix "Q" (defined as an identity matrix). This uncertainty is adjusted to the results and comparisons realized with data closer to the real values accessible (in this case these values are the ones returned from the drone).

The sensors covariance matrix "R" (more specifically the matrix from the outputs considered by the sensors model) refers to the square of the standard deviation of the sensors caused by noise in the measurements, meaning that the diagonal values of this matrix are obtained calculating the covariance of the outputs "ϕ_m" and "θ_m" (as it was initially executed) in a set of data where these outputs don't vary. This is done due to the fact that any noise can cause variation, therefore, in order to obtain these values, the dataset used was L2Data2.

The matrix "R" for the hovering stage is as follows:

- $R = \begin{bmatrix} P_\phi & 0 \\ 0 & P_\theta \end{bmatrix} = \begin{bmatrix} 9.7197 \times 10^{-10} & 0 \\ 0 & 1.2843 \times 10^{-11} \end{bmatrix}$.

Regarding the initial conditions of the estimator, "P_0" refers to the initial covariance associated with the prediction previously made and reflects on the initial trust rank of the predictions. Similarly, the matrix "Q" also needs some tweaking in order to improve the accuracy of the estimations. The initial state considered for the estimations and predictions is equal to the first state measured or deduced and, in case of a big uncertainty of predictions, it compensates using a lower number in "P_0".

In this case, the uncertainty is given by an identity matrix multiplying with the covariance obtained in the state to be evaluated. The noise is given by the covariance of the outputs multiplied by an identity matrix 2×2 (having the same number of lines as the matrix "C").

The initial values used to estimate the state are the ones from the models of "$\hat{x}$" and "P" for "$K = 0$". With the states and initial values defined, the Kalman filter can be implemented.

In order to predict step:

- $\hat{x}(k+1|k) = A(k)\hat{x}(k|k) + B(k)u(k)$;
- $P(k+1|k) = A(k)P(k|k)A(k)^T + Q(k)$.

Afterwards, in order to update step, it's needed to use the following equations:

- $\hat{x}(k+1|k+1) = \hat{x}(k+1|k) + K(k+1)(y(k+1) - C(k+1)\hat{x}(k+1|k)$;
- $P(k+1|k+1) = [I - K(k+1)C(k+1)]P(k+1|k)$;
- $K(k+1) = P(k+1|k)C(k+1)^T[C(k+1)P(k+1|k)C(k+1)^T + R(k+1)]^{-1}$.

In order to explain the concept of the steps in a Kalman filter, this filter imposes an estimation of the state variables in an initial cycle ("predict step"), allowing the system to act regarding the measurements, receiving the new estimated data in another cycle ("update step"). Afterwards, the cycle resets and keeps on having these consecutive transitions between steps [5].

The matrices "A" and "C" will be equal to the ones used before. The matrix "B" will remain null.

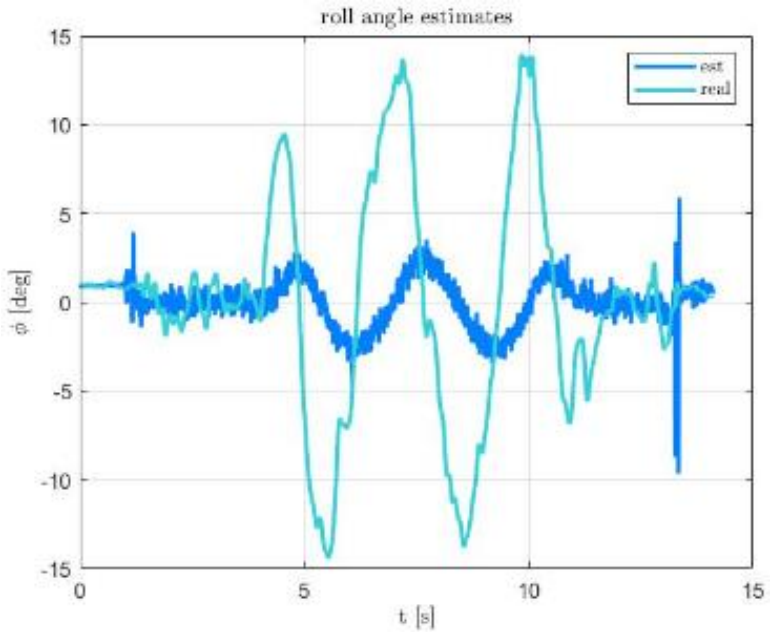

Figure 10 – Roll Angle ("ϕ $[deg]$") over Time ("t $[s]$") in the dataset L2Data6 with a Kalman filter without biases

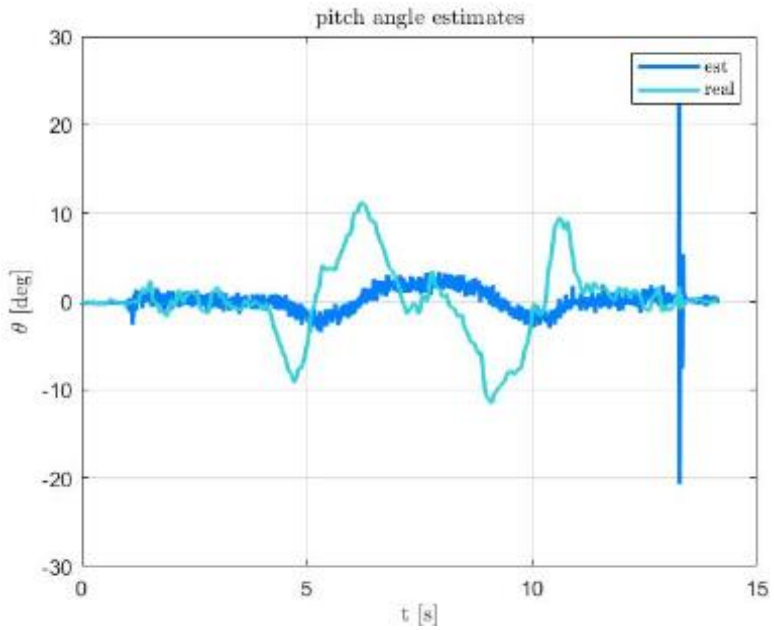

Figure 11 – Pitch Angle ("θ $[deg]$") over Time ("t $[s]$") in the dataset L2Data6 with a Kalman filter without biases

The second dataset chosen to obtain the data was L2Data3 (movement in axis x).

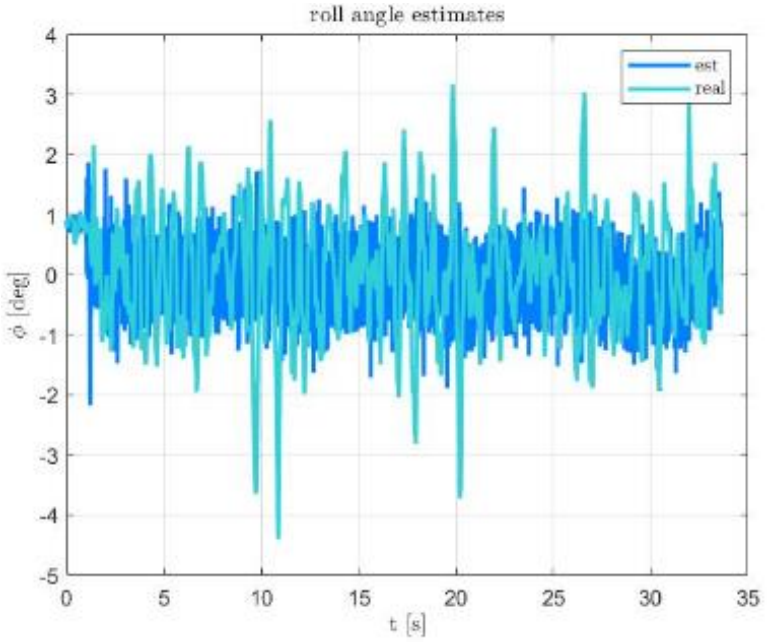

Figure 12 – Roll Angle ("ϕ $[deg]$") over Time ("t $[s]$") in the dataset L2Data3 with a Kalman filter without biases

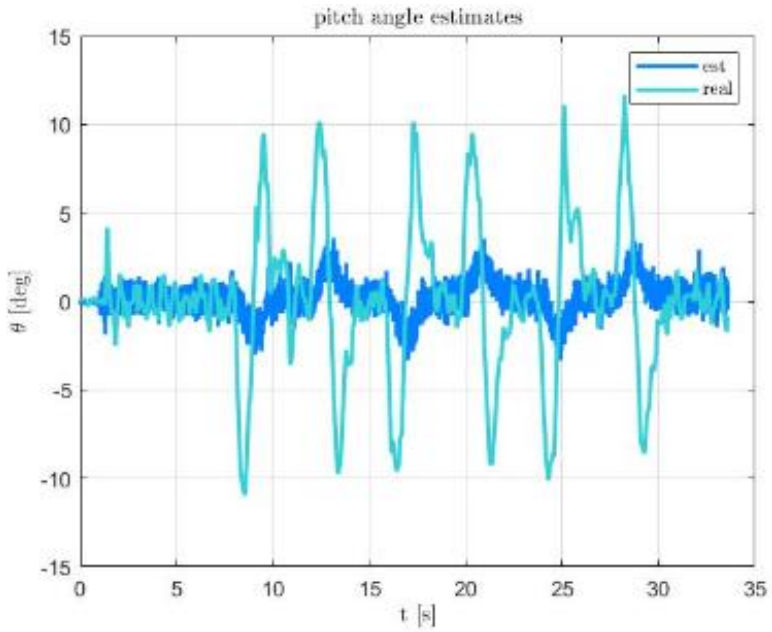

Figure 13 – Pitch Angle ("$\theta\ [deg]$") over Time ("$t\ [s]$") in the dataset L2Data3 with a Kalman filter without biases

3.8. Goal number 2.4

Discuss the results when considering (or not) measurement biases in the state vector, as well as the choice of the process and measurement noise covariance matrices.

In this goal, the filter will be implemented while including the biases in the state vector and afterwards the results will be evaluated.

As previously observed, the results show discrepancies when compared with reality, which is why the technique of "sensor fusing" is used in order to improve the results. This technique will improve the values of the estimations by increasing the number of measurements of the same magnitude, with more sensors. This system is aggregated with a gyroscope which measures the angular velocity, allowing to estimate the angle pretended. Theoretically, this sensor has bigger noise than the accelerometer, implying less accurate measurements to the noise. However, with these measurements and with a covariance value associated to the noise properly estimated, this system provides more information regarding the state that's wanted to be estimated.

The pitch angle measured by the gyroscope at a certain moment can be obtain by "$\phi_{m_{gyroscope}}(k) = (\omega_x * T_s) + \phi_{m_{gyroscope}}(k-1)$", "$T_s$" being the period of time between measurements.

Assuming that the accelerometer has a bigger bias than the gyroscope due the bigger deviation from its mean when comparing to the average values of reality, the outputs associated with the accelerometer are the following:

- $y_1 = \phi_{m_1} = \phi + b_{\phi_1} + n_{\phi_1}$;
- $y_2 = \theta_{m_1} = \theta + b_{\theta_1} + n_{\theta_1}$.

Therefore, the outputs associated with the gyroscope are the following:

- $y_3 = \phi_{m_2} = \phi + b_{\phi_2} + n_{\phi_2} \approx \phi + n_{\phi_2}$;
- $y_4 = \theta_{m_2} = \theta + b_{\theta_2} + n_{\theta_2} \approx \theta + n_{\theta_2}$.

With the addition of this new sensor and the approximation made, the linear dependency of the bias with the respective angles is removed, allowing the estimation of "b_{ϕ_1}" and "b_{θ_1}".

It's important to mention that the values of "ϕ" and "θ" aren't exempt of biases, since these values are considered to be equal to "ϕ_{m_2}" and "θ_{m_2}", respectively. However, realistically these values continue to be smaller than "ϕ_{m_1}" and "θ_{m_1}".

- $y_1 = \phi_{m_1} = \phi + b_{\phi_1} + n_{\phi_1} = \phi_{b_2} + b_{\phi_1} + n_{\phi_1}$;
- $y_2 = \theta_{m_1} = \theta + b_{\theta_1} + n_{\theta_1} = \theta_{b_2} + b_{\theta_1} + n_{\theta_1}$.

Since there was no addition of states, the matrix "A" remains the same. However, the matrix "C", due to the two new outputs (in particular the roll and pitch angles measured by the gyroscope), the matrix has the following configuration:

- $C = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$

Regarding the matrix "R", the matrix was obtained using the same data as before and calculating the covariance for both roll and pitch angles of the gyroscope:

- $R = \begin{bmatrix} P_{\phi_{accelerometer}} & 0 & 0_{2\times2} \\ 0 & P_{\theta_{accelerometer}} & \\ 0_{2\times2} & & P_{\phi_{gyroscope}} \quad 0 \\ & & 0 \quad P_{\theta_{gyroscope}} \end{bmatrix} =$

$$= \begin{bmatrix} 1.2509 \times 10^{-6} & 0 & 0_{2\times2} \\ 0 & 1.3762 \times 10^{-6} & \\ 0_{2\times2} & & 2.0212 \times 10^{-6} \quad 0 \\ & & 0 \quad 8.494 \times 10^{-6} \end{bmatrix}.$$

In order to improve the results obtained earlier, it has emerged a necessity to modify the matrices "Q" and "P_0" (initial covariance matrix and the initial trust rank of the previous predictions).

Regarding the state predictions of the roll and pitch angles, they were considered as relatively reliable. In regard to the predict steps made by the model relatively to the roll and pitch angles, the first and third diagonal elements of "Q" relatively to the respective elements in "P_0" are quite small, while the second and fourth elements of the same diagonal are relatively high regarding the angular velocities, meaning that there's a low level trust in the predict step of the linear model of the instalment. The biases and the last two elements of the diagonal of "Q" are very small when compared to "R", since the model predicts that these parameters won't vary over time, therefore the value chosen implies the prediction of a small time-consuming "drift", which is coincident with what happens in reality.

The modifications explained represent the following matrixes:

- "$Q = \begin{bmatrix} 1 & 0 & 0 & & & \\ 0 & 100 & 0 & & 0_{3\times3} & \\ 0 & 0 & 1 & & & \\ & & & 100 & 0 & 0 \\ & 0_{3\times3} & & 0 & 0.000001 & 0 \\ & & & 0 & 0 & 0.000001 \end{bmatrix}$";

- "$P_0 = \begin{bmatrix} 0.01 & 0 & 0 & & & \\ 0 & 10000 & 0 & & 0_{3\times3} & \\ 0 & 0 & 0.01 & & & \\ & & & 10000 & 0 & 0 \\ & 0_{3\times3} & & 0 & 0.00000001 & 0 \\ & & & 0 & 0 & 0.00000001 \end{bmatrix}$".

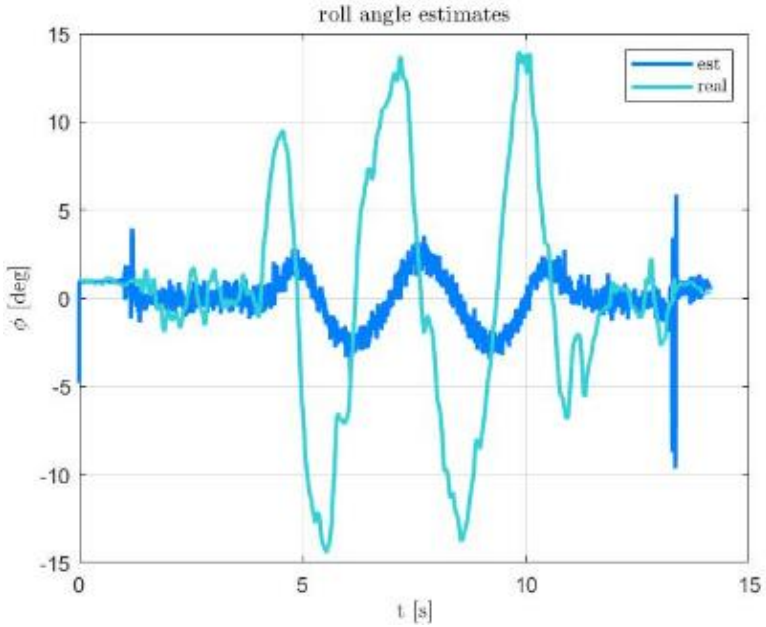

Figure 14 – Roll Angle ("ϕ $[deg]$") over Time ("t $[s]$") in the dataset L2Data6 with a Kalman filter with biases

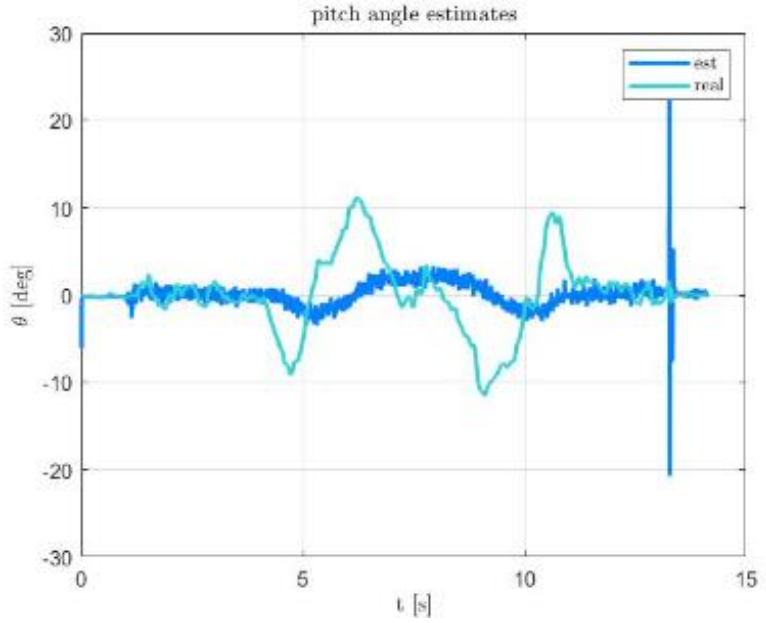

Figure 15 – Pitch Angle ("ϕ $[deg]$") over Time ("t $[s]$") in the dataset L2Data6 with a Kalman filter with biases

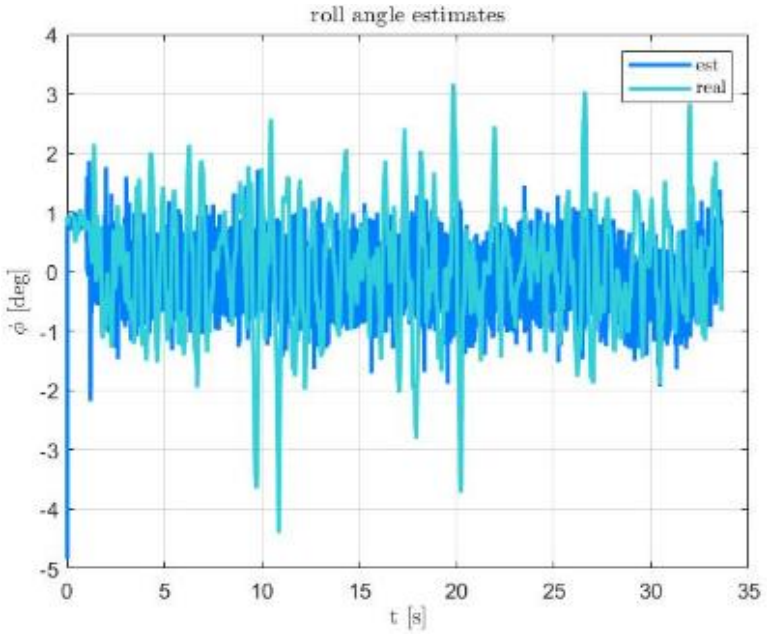

Figure 16 – Roll Angle ("ϕ $[deg]$") over Time ("t $[s]$") in the dataset L2Data3 with a Kalman filter with biases

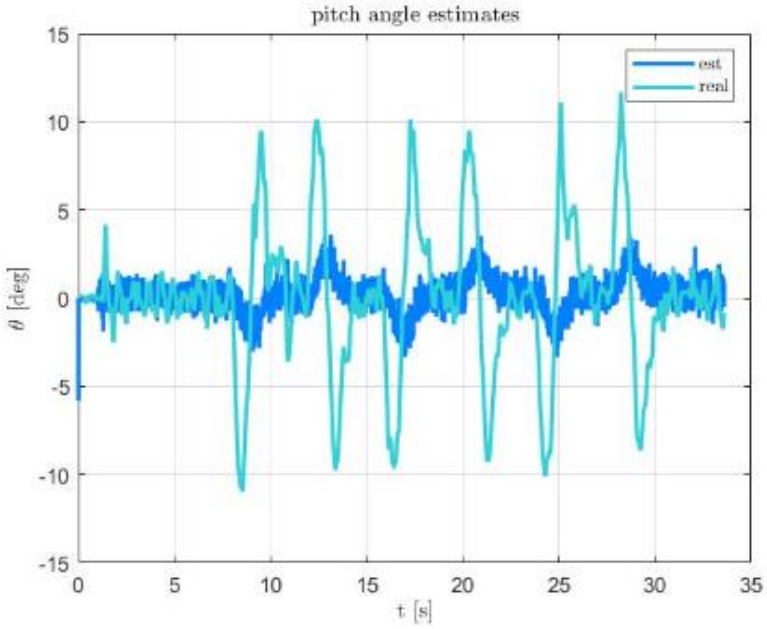

Figure 17 – Pitch Angle ("ϕ $[deg]$") over Time ("t $[s]$") in the dataset L2Data3 with a Kalman filter with biases

As it's possible to observe, the results obtained still vary from reality, with an amplitude smaller and with an excess of noise, which can be caused by the angles measured by the accelerometer (more specifically the covariance associated with the noise or the measurement uncertainty regarding the accelerometer), since the roll and pitch angles are obtained by considering an almost static movement, showing efficiency and an adequate covariance for situations which in fact it comes closer from that condition. However, when there's movement (which is the case for most datasets), the efficiency of the measurements of the accelerometers is small. Assuming in this case that the values obtained don't

represent the angles pretended (or they don't have an approximation angle acceptable), with angle variations less pronounced and limited and having in account factors like the similarity in results obtained through the gyroscope, inadequacy of the sensor for the movements pretended, and low measurements variations (with consequent low and non-representative uncertainty), the associated covariances will increase according to the accelerometer sensor, resulting in a smaller influence of the respective sensor. The associated covariances are passed for the previously made predictions and, essentially, the measurements by the oscilloscope reveal a relatively high degree approximation to the values of reality.

The increase of the respective values of the matrix "R" (in this case the first and second elements) were adjusting in order to converge the mean difference between the real values and the values obtained with the Kalman filter to null.

The matrix "R" is updated into the following:

- $R = \begin{bmatrix} 1.2509 \times 10^{-5} & 0 & 0_{2\times2} \\ 0 & 1.3762 \times 10^{-4} & \\ 0_{2\times2} & & \begin{matrix} 2.0212 \times 10^{-6} & 0 \\ 0 & 8.494 \times 10^{-6} \end{matrix} \end{bmatrix}$.

As the following figures show, the graphics present an acceptable accuracy.

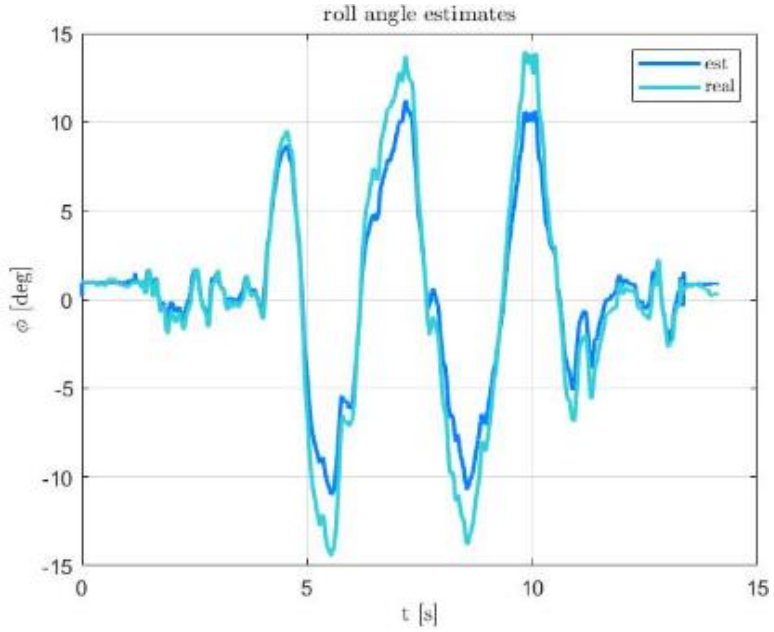

Figure 18 – Roll Angle ("ϕ $[deg]$") over Time ("t $[s]$") in the dataset L2Data6 with an updated Kalman filter with biases

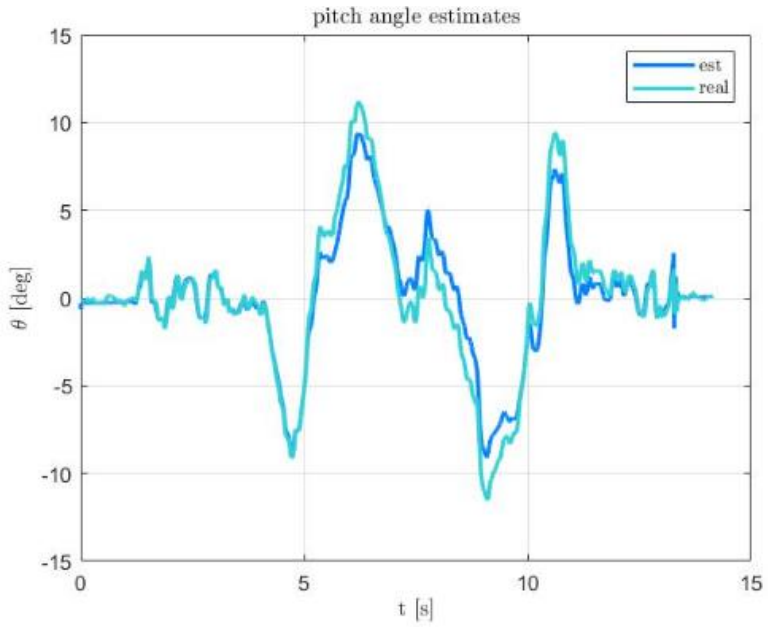

Figure 19 – Pitch Angle ("θ $[deg]$") over Time ("t $[s]$") in the dataset L2Data6 with an updated Kalman filter with biases

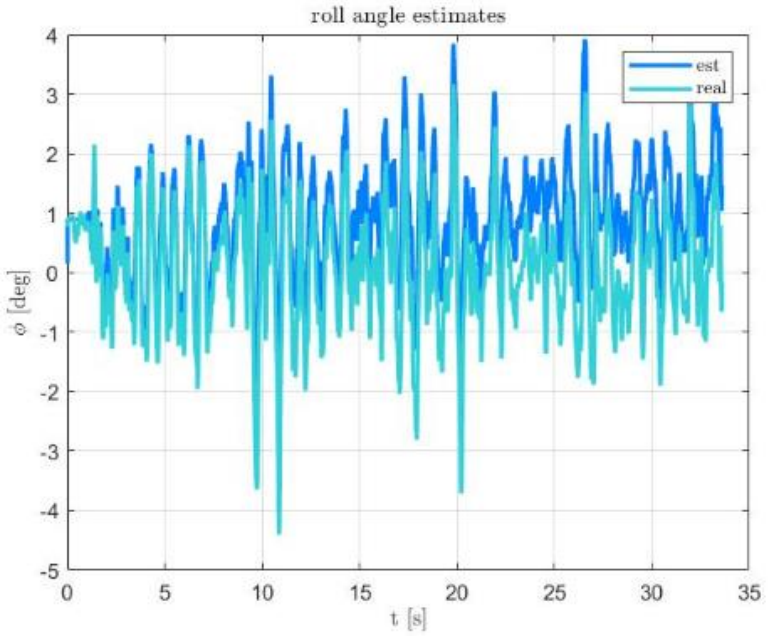

Figure 20 – Roll Angle ("ϕ $[deg]$") over Time ("t $[s]$") in the dataset L2Data3 with an updated Kalman filter with biases

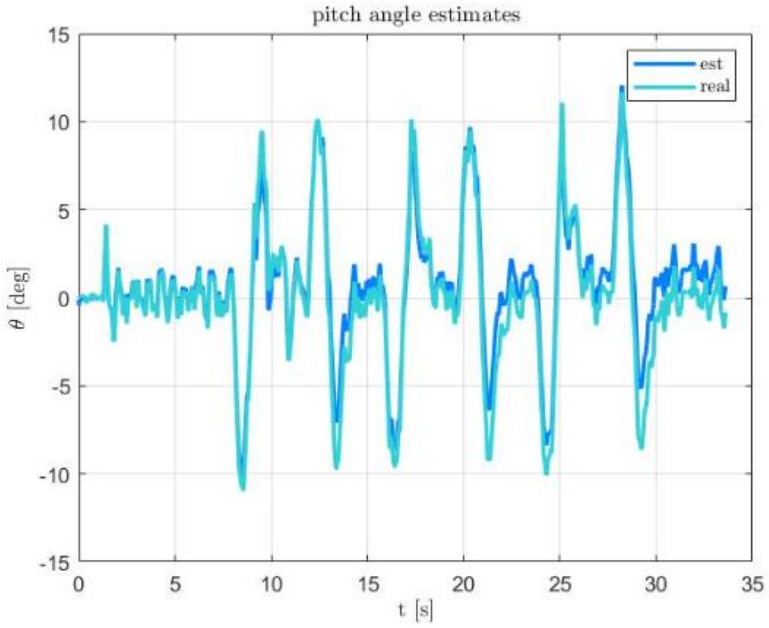

Figure 21 – Pitch Angle ("$\theta\ [deg]$") over Time ("$t\ [s]$") in the dataset L2Data3 with an updated Kalman filter with biases

3.9. Goal number 3.1

Design an extended Kalman filter to estimate both pitch and roll based on the original accelerometers and rate gyro measurements, defining clearly the (possible nonlinear) system equations and the output/measurements equations.

An extended Kalman filter is a non-linear version of a Kalman filter, being applied into non-linear systems of equations in order to linearize around a relevant point, allowing its application.

In order to apply an extended Kalman filter, it's necessary to define the initial conditions, the dynamics models, and the sensors of this new filter. Similarly to the goal number 2.1, the state vector and the measurements equations will be the same. This happens due to the fact that, in the non-linear functions obtained, both "ω" will be close to null, meaning that the results of this filter will be similar to the linearized Kalman filter:

- $\dot{\phi} = \omega_x$;
- $\dot{\omega}_x = 0$;
- $\dot{\theta} = \omega_y$;
- $\dot{\omega}_y = 0$;
- $\dot{b_\phi} = 0$;
- $\dot{b_\theta} = 0$;
- $\phi_m = \phi + b_\phi + n_\phi$;
- $\theta_m = \theta + b_\theta + n_\theta$.

The state dynamics and the non-linear measurements will be the following:

- $x(k+1) = A(kx(k) + B(k)u(k) + B_w(k)w(k)$;
- $y_i(k) = C_i(k)x(k) + D_i(k)u(k) + D_{vi}(k)v(k)$
- $A(k) = \frac{df_d(\hat{x})}{dx^T}$;
- $B_w(k) = \frac{df_d(\bar{w})}{dx^T}$;
- $C_i(k) = \frac{dh_i(\hat{x})}{dx^T}$;
- $D_{vi}(k) = \frac{dh_i(\bar{v})}{dv^T}$.

The matrices "A" and "C" will be similar to the same matrices from the goal number 2.2, with the matrices "B" and "D" continuing to be null. The matrices "A", "B", "C" and "D" will remain the same because the orientations aren't relevant, which is why the linearization of each point will be similar to the ones defined in other moments of time. However, it will be needed to linearize each instant of time around each estimated point.

For the same reasons as the previous goal, the matrices "P_0", "Q" e "R" will maintain the same format since the conditions relevant to them didn't change.

The matrices "B_w" and "D_v" (related to the processing noise and the noise regarding the measurements, respectively) are linear varying over time. In order to simplify and due to the

direct relation of the noise with the outputs and state variations, the matrices "B_w" and "D_v" will be assumed as identity matrices.

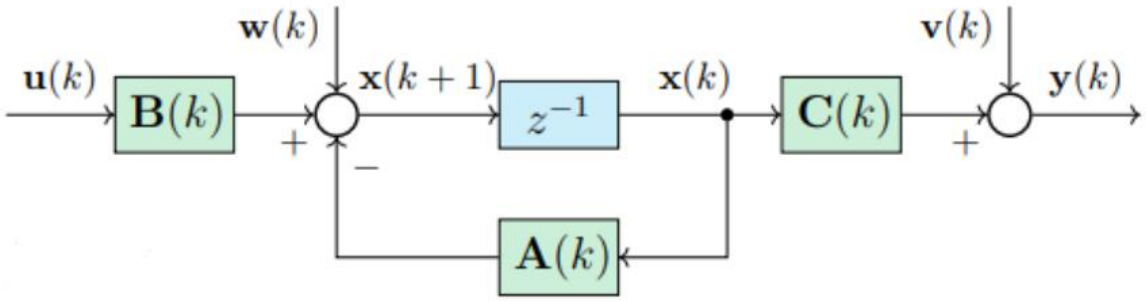

Figure 22 – Discrete Linear Model with noises associated

Contrary to the previous linear filter, in order to predict step, it was used a non-linear model provided by the following equations:

- $\dot{\phi} = \omega_z * \cos(\phi) * \tan(\theta) + \omega_y * \sin(\phi) * \tan(\theta)$;
- $\dot{\theta} = \omega_y * \cos(\phi) - \omega_z * \sin(\phi)$.

To use these equations, it was considered that:

- $\dot{\lambda} = [\dot{\phi} \quad \dot{\theta} \quad \dot{\psi}]$;
- $\dot{\lambda} = Q * \left([\phi \quad \theta \quad \psi] \quad * \quad \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \right)$.

Having in account that the dynamics of the Euler angles above depend on the angular velocity and "ω_z", the group decided to add the parameter measured as a state and have the angular velocity measured by the sensor ("$\omega_{z_m} = \omega_z + b_{\omega_z} + n_{\omega_z} = \omega_{z_{biased}} + n_{\omega_z}$") as the output, ignoring the bias to be able to estimate "$\omega_{z_{biased}}$".

The following matrix is the state vector:

- $x = \begin{bmatrix} \phi \\ \omega_x \\ \theta \\ \omega_y \\ b_\phi \\ b_\theta \\ \omega_z \end{bmatrix}$.

Regarding the dynamics of the angular velocities and the biases caused by the reasons presented earlier for the other filter, an approximation was made by linearizing these dynamics, assuming that they are null.

Due the dynamics previously presented of the angles, "ω_z" will become a state important to estimate, therefore, a new output will be introduced directly into the gyroscope and given by:

- $\omega_{z_{measured}} = \omega_z + b_{\omega_z} + n_{\omega_z} = \omega_{z_{biased}} + n_{\omega_z}$.

In order to simplify, the linear functions representative of the sensor's outputs "$h_{lin}(x, u)$" (where the vector "v" of measurement noise) has a direct relation with the respective outputs, meaning that:

- $D_v(k) = \frac{dh(\bar{\bar{v}})}{dv^T} = I_{5\times5}$.

The matrix "B_w" follows the same logic in order to simplify the linear function representative of the state dynamics ("$f_{lin}(x, u)$"), where "w" (containing the processing noises associated with the state dynamics) has a direct relation with the respective dynamics, meaning that:

- $B_w(k) = \frac{df_d(\bar{\bar{w}})}{dw^T} = I_{7\times7}$.

3.10. Goal number 3.2

Analyse the filter observability by linearizing the system and observation equations around a relevant point.

In order to analyse the filter observability, the matrices "A" and "C" for this new filter. The matrix "A" will be equivalent to every point. To linearize the system, the matrices will be the following:

- $C = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$;
- $A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$

With "A" and "C" defined, the study of the observability of the filter continues by calculating "$O_{(A,Q)} = \begin{bmatrix} C \\ CA \\ CA^2 \\ CA^3 \\ CA^4 \\ CA^5 \\ CA^6 \end{bmatrix}$":

- "$O_{(A,Q)} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ \hline 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline \multicolumn{7}{c}{0_{5\times 7}} \\ \hline \multicolumn{7}{c}{0_{5\times 7}} \\ \hline \multicolumn{7}{c}{0_{5\times 7}} \\ \hline \multicolumn{7}{c}{0_{5\times 7}} \\ \hline \multicolumn{7}{c}{0_{5\times 7}} \end{bmatrix}$", with "$rank\left(O_{(A,C)}\right) = 7 = n$",

meaning that this Kalman filter is observable.

3.11. Goals number 3.3 and 3.4

Implement the filter and compare its performance with the linear filter and with the attitude data from the Crazyflie, using two or more datasets (including L2Data6).

Discuss the results when considering (or not) measurements biases in the state vector, as well as the choice of the process and measurement noise covariance matrices.

The extended Kalman filter follows the same logic as the linear one. However, predictions are made based on the system's non-linear model, which will not present very varied results compared to the linear model, since the linear model proved to be similar throughout the simulations, largely due to the linearization points considered, namely the slight orientations and little variation in them, implying a not so considerable deviation from the equilibrium point and the non-linear model. In addition, in the covariance prediction, "P", "Q" is influenced by the new "B_w" matrix and in the Kalman gain update step, "R" is influenced by "D_v", allowing the filter to follow the non-linearity of the respective parameters.

In order to compare the extended Kalman filter with the linear filter, the initial values must be determined. The initial values used to estimate the state are the same ones from the models of "$\hat{x}$" and "P" for "$K = 0$".

In order to initialize the filter:

- $\hat{x}(0|0) = x_0$;
- $P(0|0) = P_0$.

The predict step is determined by:

- $\hat{x}(k+1|k) = f_d(\hat{x}(k|k), u(k), \overline{\overline{w}})$;
- $P(k+1|k) = A(k)P(k|k)A(k)^T + B_w(k)QB_w(k)^T$.

The update step (for each sensor i) is determined by:

- $\hat{x}(k+1|k+1) = \hat{x}(l+1|k) + K_i(k+1)\big(y_i(k+1) - C_i(k)\hat{x}(k+1|k)\big)$;
- $P(k+1|k+1) = [I - K_i(k+1)C_i(k)]P(k+1|k)$;
- $K(k+1) = P(k+1|k)C_i^T(k)\big[C_i(t)P(t)C_i^T(t) + D_{vi}(t)RD_{vi}^T(t)\big]^{-1}$.

Without adjusting the covariances, the Kalman filter obtained the following figures.

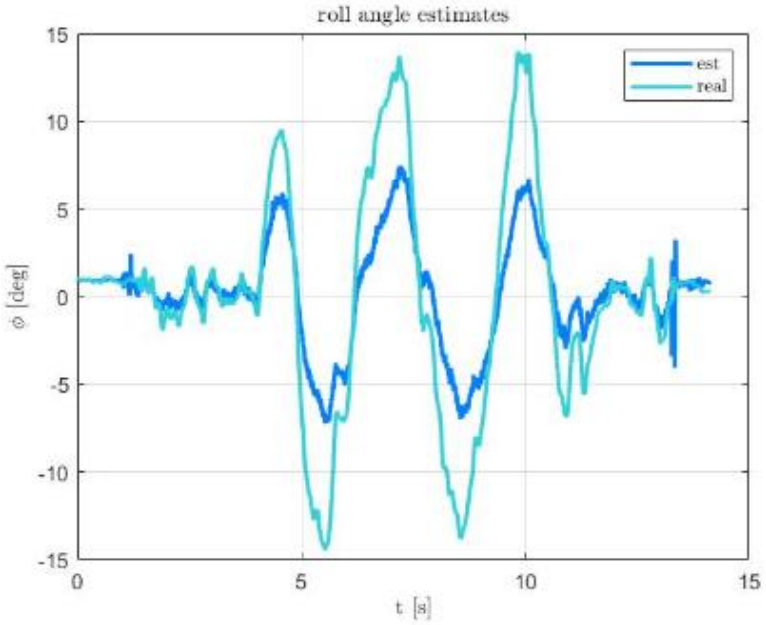

Figure 23 – Roll Angle ("ϕ $[deg]$") over Time ("t $[s]$") in the dataset L2Data6 with an extended Kalman filter before the adjustment of the covariances

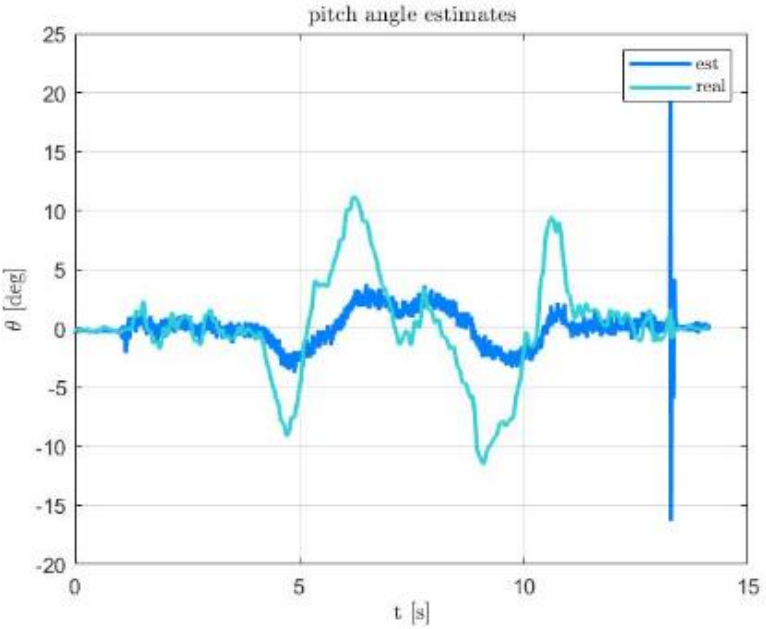

Figure 24 – Pitch Angle ("θ $[deg]$") over Time ("t $[s]$") in the dataset L2Data6 with an extended Kalman filter before the adjustment of the covariances

By the adjustment of the covariances, the extended Kalman filter develops the following figures.

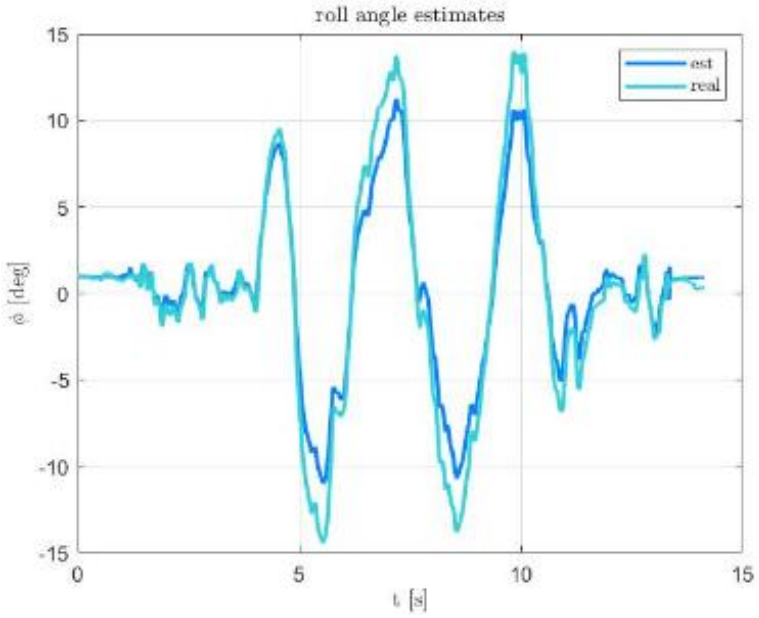

Figure 25 – Roll Angle ("ϕ $[deg]$") over Time ("t $[s]$") in the dataset L2Data6 with an extended Kalman filter after the adjustment of the covariances

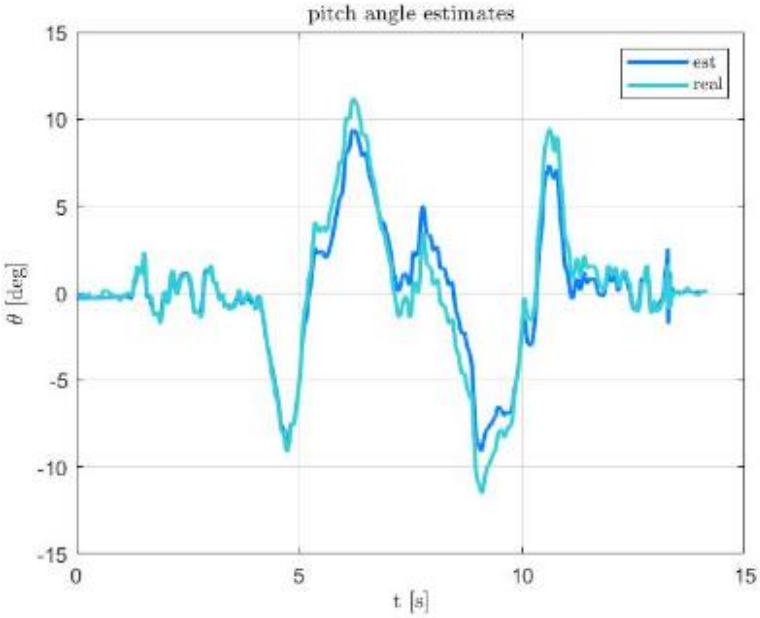

Figure 26 – Pitch Angle ("θ $[deg]$") over Time ("t $[s]$") in the dataset L2Data6 with an extended Kalman filter after the adjustment of the covariances

After obtaining more realistic results with the covariances adjusted, it's possible to compare the results with the linear Kalman filters obtained earlier.

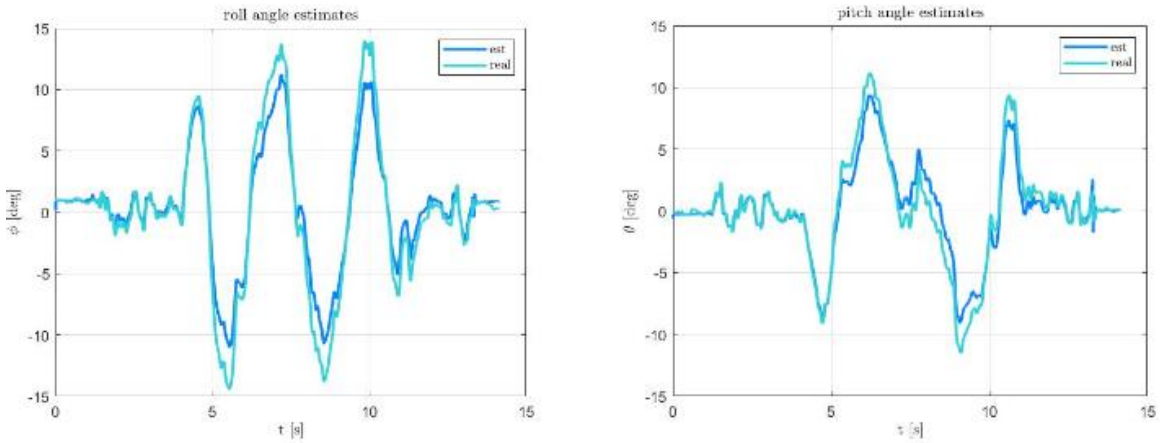

Figure 27 – Roll and Pitch Angles ("ϕ $[deg]$" and "θ $[deg]$" respectively) over Time ("t $[s]$") in the dataset L2Data6 with the linear Kalman filter obtained earlier

After a brief look into both filters, it's possible to consider both filters equal, with both of them following very closely the real values. A closer look from point to point reveals that the extended Kalman filter doesn't improve the values when comparing with the linear filter. In the case of the Crazyflie 2.1, the non-linear equations don't improve the values, even if they vary from the equations of the linearized filter.

The same procedures were done for the dataset L2Data3.

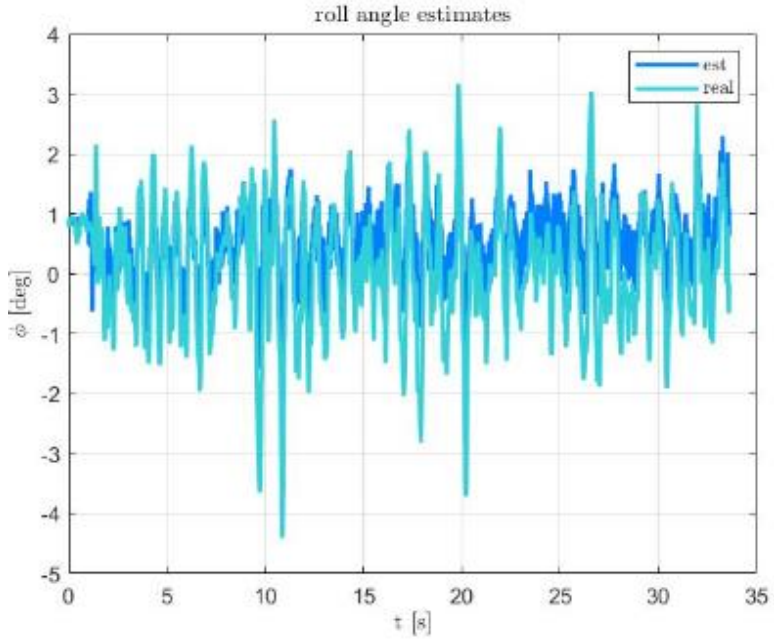

Figure 28 – Roll Angle ("ϕ $[deg]$") over Time ("t $[s]$") in the dataset L2Data3 with an extended Kalman filter before the adjustment of the covariances

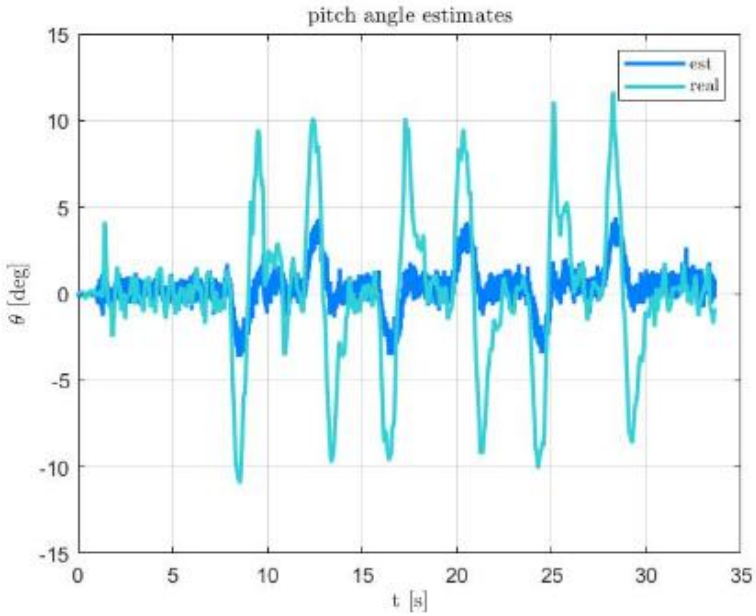

Figure 29 – Pitch Angle ("θ $[deg]$") over Time ("t $[s]$") in the dataset L2Data3 with an extended Kalman filter before the adjustment of the covariances

As was the case with the dataset L2Data6, there's a need to adjust the covariance in order to obtain more realistic results.

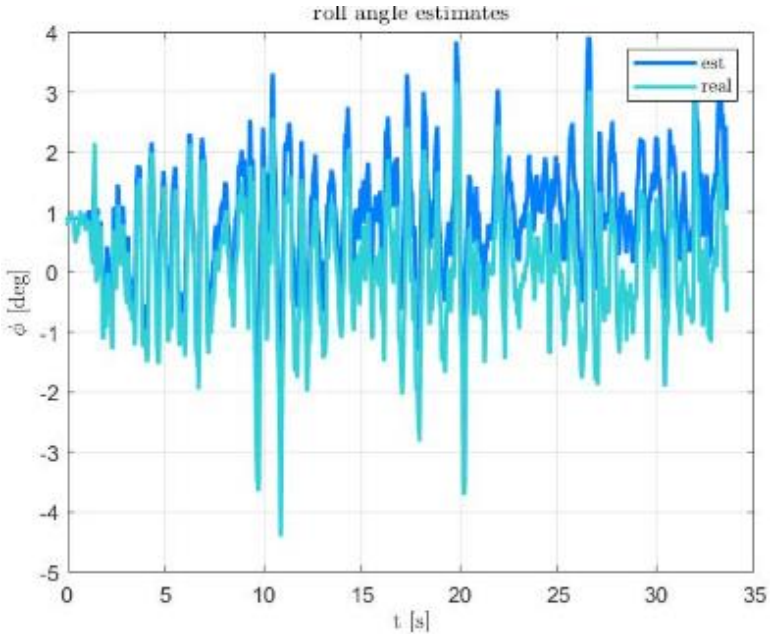

Figure 30 – Roll Angle ("ϕ $[deg]$") over Time ("t $[s]$") in the dataset L2Data3 with an extended Kalman filter after the adjustment of the covariances

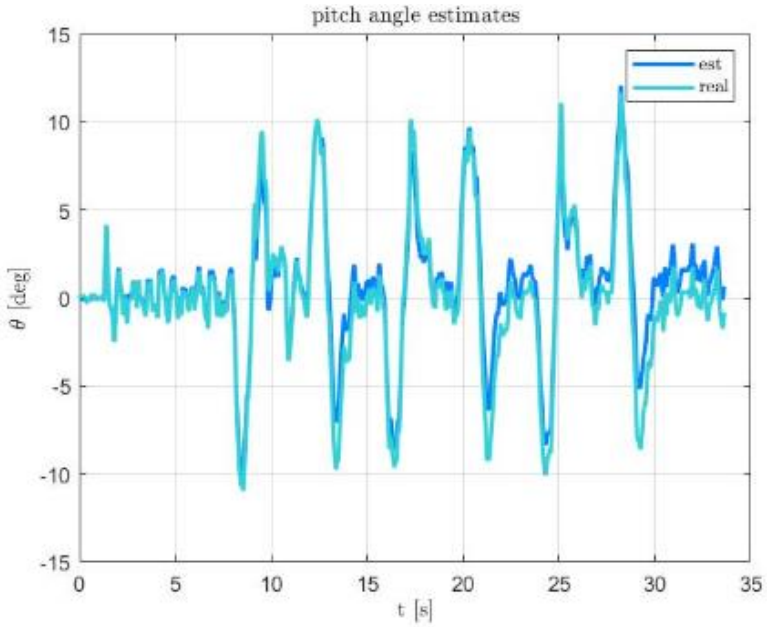

Figure 31 – Pitch Angle ("θ [deg]") over Time ("t [s]") in the dataset L2Data3 with an extended Kalman filter after the adjustment of the covariances

As was the case with the other dataset, there's going to be a comparison between the extended Kalman filter and the linear Kalman filter obtained before.

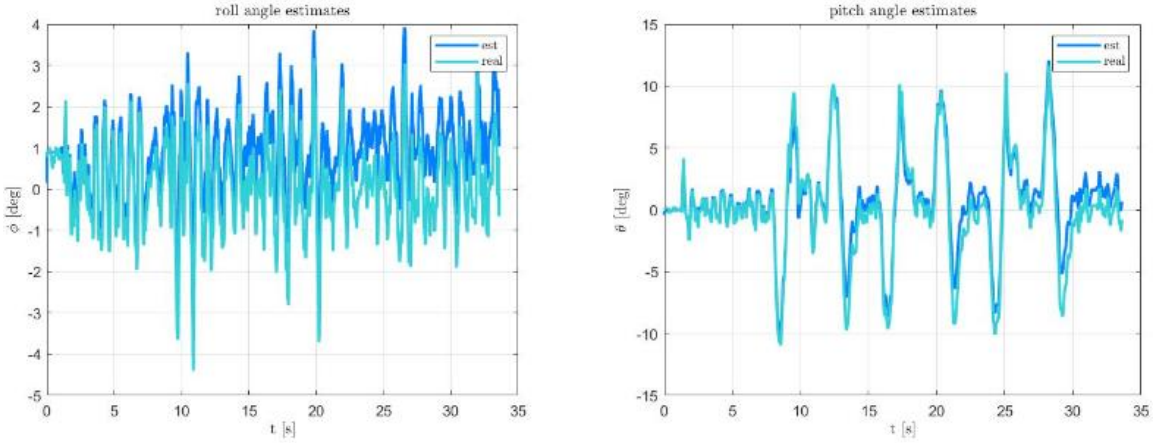

Figure 32 – Roll and Pitch Angles ("ϕ [deg]" and "θ [deg]" respectively) over Time ("t [s]") in the dataset L2Data3 with the linear Kalmen filter obtained earlier

Comparing the two filters visually, the same conclusions can be deduced for the dataset L2Data3 as it was deduced to the dataset L2Data6.

4. Conclusions and Commentaries

Throughout this project it was possible to study and implement the given Kalman filter theory in a complementary study project, analysing sensors that are normally used in everyday vehicles (aircraft, cars, etc), analysing the sensor results, taking a critical view of what was expected, and obtaining and comparing the results with the sensor readings.

It was a challenging job where the results obtained were not the most desirable, with several difficult steps to overcome.

Gyroscope measurements are already very precise in relation to the real values, and the degree of confidence given to the gyroscope is what allowed these results to be obtained, undermining the reliability of the predictions made, which by the way did not accurately describe the drone's movement.

The predictions from extended to linear vary little because the linear model used already differed a little from the drone's progress in reality due to the unreasonable angles. In addition, the fact that the "B_w" matrix is an identity implied a covariance prediction similar to the prediction step made using the linear filter, and because the "D_v" is an identity, the Kalman gain in the extended model doesn't differ from the linear one either. What helps is the small variation in the results from the extended Kalman filter to the non-linear Kalman filter.

5. Webgraphy

[1] – Captain Brian P. Tice, "Unmanned Aerial Vehicles" – https://web.archive.org/web/20090724015052/http://www.airpower.maxwell.af.mil/airchronicles/apj/apj91/spr91/4spr91.htm

[2] – EASA (European Union Aviation Safety Agency) website, "Open Category – Low Risk – Civil Drones" where we checked the classification of a Micro Aerial Vehicle – https://www.easa.europa.eu/en/domains/civil-drones-rpas/open-category-civil-drones

[3] – NASA's Earth Fact Sheet where we obtained the Bulk Parameter of Mean Surface Gravity ("$\boldsymbol{g_{earth}} = 9.82\ [^{m}/_{s}]$") and the Terrestrial Atmosphere Surface Density ("$\boldsymbol{\rho_{earth}} = 1.217\ \left[^{kg}/_{m^3}\right]$") – https://nssdc.gsfc.nasa.gov/planetary/factsheet/earthfact.html

[4] – Roll and Pitch Angles from Accelerometer Sensors to develop goal 1.3 – https://mwrona.com/posts/accel-roll-pitch/

[5] – Maria Isabel Ribeiro, "Kalman and Extended Kalman Filters: Concepts, Derivation and Properties" – http://users.isr.ist.utl.pt/~mir/pub/kalman.pdf

6. Attachments

Github repository by José Corvo:

- Link for the Main Folder – https://github.com/Jose-Corvo-Lv99/UAV-s_Grupo3_FCT_2023-24