

POLITECNICO DI MILANO
Scuola industriale e dell'informazione
Corso di Laurea magistrale in Ingegneria Aeronautica



Drone fleet model identification and comparison

Relatore: Marco Lovera

Tesi di Laurea di:
Cristina Ghezzi, matricola 927943

Anno Accademico 2020-2021

*A tutte le persone che mi sono state vicine
e che hanno creduto in me.*

Ringraziamenti

Ringrazio i miei genitori per avermi dato la possibilità di inseguire i miei sogni: sono orgogliosa di essere vostra figlia.

Ringrazio i miei amici Marco, Petru, Riccardo e Diletta, per aver sempre creduto in me ed essermi stati vicino in questi anni.

Ringrazio Matteo, che mi ha aiutata a inseguire i miei sogni e a fare il primo passo verso la loro realizzazione.

Infine ringrazio Sergio, che in poco tempo mi ha fatto capire che la normalità può essere noiosa e che a volte un sorriso vale più di mille parole.

Contents

Ringraziamenti	5
1 Introduction	16
1.1 Goal of the thesis	17
1.2 State of the art	17
2 Drone dynamics: theoretical basis	19
2.1 Reference frames	19
2.2 Euler angles and Quaternions	21
2.3 Modelling UAV dynamics	24
3 Analysis of the ANT-X drone's data through Log Analyser	26
3.1 Description of Log analyser software	26
3.2 ANT-X drone specification	33
3.3 2DoF Drones tests	34
3.4 Test log analysis with Log analyser	34
4 Theory of model identification	44
4.1 State estimation problem	44
4.2 System identification	45
4.3 Model identification process	47
4.4 Input sequences for flight testing	47
4.5 Maximum likelihood estimation and Output-error method	49
4.5.1 Likelihood Principle	49
4.5.2 The output error method	50
4.6 Classification of model identification problems	51
4.7 Preliminary practical issues for model identification	51
4.7.1 Choice of sampling interval	52

4.7.2	Prefiltering	52
4.7.3	Stability	54
4.8	Validation of the model	54
5	Model identification applied to ANT-X drones	56
5.1	Grey-box identification	56
5.2	Closed loop identification for longitudinal dynamics: Position	57
5.3	Closed loop identification for longitudinal dynamics: Attitude	63
5.4	Open loop identification for longitudinal dynamics	67
6	Conclusions	73

List of Figures

2.1	Body frame	20
2.2	Ned frame	20
2.3	Frames x, y, z and b_1, b_2, b_3	21
2.4	Rotations	21
3.1	Position data without any plot customisation	27
3.2	Position data with Align starting time selected	28
3.3	Position data with Align starting time and trim starting and ending time selected	28
3.4	Log analyser first interface	29
3.5	Log analyser Log file loaded	30
3.6	Position flight log X0080_20201208_1252 with Matlab tool	31
3.7	Position flight log X0083_20201208_1339 with Matlab tool	31
3.8	Log analyser, position test: position vs set-point position	35
3.9	Log analyser, position test: position error	35
3.10	Log analyser, position test: velocity vs set-point velocity	36
3.11	Log analyser, position test: velocity (body frame) vs set-point velocity	36
3.12	Log analyser, position test: attitude angles vs set-point attitude angles	37
3.13	Log analyser, position test: attitude angles estimate vs mocap	37
3.14	Log analyser, position test: angular rates	38
3.15	Log analyser, position test: thrust	38
3.16	Log analyser, position test: moments	39
3.17	Log analyser, position test: battery	39
3.18	Log analyser, attitude test: attitude angles vs set-point	40
3.19	Log analyser, attitude test: attitude error	40
3.20	Log analyser, attitude test: angular rates vs set-point angular rates	41
3.21	Log analyser, attitude test: attitude estimate vs mocap	41

3.22	Log analyser, attitude test: moments	42
3.23	Log analyser, attitude test: thrust	42
4.1	Logical approach to the identification problem	46
4.2	Sine sweep	48
4.3	3211	48
4.4	Doublet	48
4.5	Gaussian probability	49
4.6	Block diagram of output-error method	50
4.7	error filtering	53
4.8	Prefiltering	53
5.1	Results from identification position in closed loop: estimated values	59
5.2	Results from identification position in closed loop: variance of the estimate	60
5.3	Results from identification position in closed loop: VAF	60
5.4	Results from identification position for the first drone tested: Position	61
5.5	Results from identification position for the first drone tested: Ac- celeration	62
5.6	Results from identification attitude in closed loop: estimated values	64
5.7	Results from identification attitude in closed loop: variance of the estimate	65
5.8	Results from identification attitude in closed loop: VAF	65
5.9	Results from identification angular position for the first drone tested: Attitude	66
5.10	Results from identification angular velocity for the first drone tested: Attitude	66
5.11	Block diagram	68
5.12	Results from identification position in open loop: X_u and X_q	69
5.13	Results from identification position in open loop: M_u and M_q	70
5.14	Results from identification position in open loop: X_d and M_d	70
5.15	Results from identification position in open loop: VAF	71
5.16	Results from identification position in open loop: position (Results from the last drone of the fleet)	71

List of Tables

3.1	ANT-X 2DoF specifications	33
5.1	Results from identification position in closed loop	59
5.2	Results from identification attitude in closed loop	64
5.3	Block diagram of closed loop model	68
5.4	Results from identification position in open loop	69

Abstract

In the last decades, the interest for unmanned air vehicles (UAVs) is increasing, due to their adaptability and capability to work in different conditions. They are employed from the civil to the military field where high-level performance are required.

The purpose of the thesis is to study the mathematical model of a drone fleet and to understand how the model differ from one to the other, considering a SIMO (single input, multiple output) problem. The drones are the ANTx 2DoF drone, and they were tested in laboratory in order to create a safe and controlled environment and to run experiments in a repeatable way. The model identification implemented is the grey-box modelling approach and it is performed in time domain. Both close and open loop are discussed. Experimental data are validated in order to guarantee accurate results.

Sommario

Negli ultimi decenni, l'interesse per gli aeromobili a pilotaggio remoto (APR) é in aumento, a causa delle loro capacità di adattarsi e di lavorare in diverse condizioni. I droni vengono usati sia in ambito civile che militare dove sono richieste prestazioni di alto livello.

L'obiettivo della tesi é di studiare il modello matematico di una flotta di droni e di vedere come differiscono uno dall'altro, considerando un problema di tipo SIMO (single input, multiple output). I droni sono droni ANTx 2Dof drone e sono stati testati in laboratorio dove le condizioni erano sicure e controllate, così da garantire la ripetibilità degli esperimenti. L'identificazione applicata é la grey-box nel dominio del tempo. Sia l'approccio in anello chiuso che in anello aperto sono stati studiati. I dati sono poi stati validati, così da dimostrare l'accuratezza dei risultati.

Chapter 1

Introduction

Chi ha provato il volo camminerá guardando il cielo, perché lá é stato e lá vuole tornare.

Leonardo da Vinci

Unmanned Aerial Vehicles (UAVs), commonly known as drones, are flying systems that can be remotely controlled or fly autonomously through software embedded in their systems. Louis Breguet built the first rotary wing aircraft "Gyroplane n: 01" in 1907. It had very limited flight capabilities and was controlled by people on the ground for stabilization. For its time, the design of this drone was visionary since the structure is very similar to that of modern UAVs.

Unmanned aerial vehicles have become increasingly prominent in a variety of aerospace applications, from the military field to the civilian one. The most common configuration, which is the subject of this thesis, is the X-configuration with four motors. Despite the advantages of weight, size and costs, multi-rotor UAVs are affected by instability. In the first experiments stabilisation was controlled by people, whereas today this task is accomplished by software.

The first step in understanding the drone behaviour is to study the real responses of the UAV subjected to different inputs. The second step is to build a mathematical model that, under the same inputs, gives a good approximation of the outputs of the real system. Obviously, the mathematical model will never perfectly fit the real system and this will be shown in depth in the next chapters.

1.1 Goal of the thesis

The goal of the thesis is to study and compare the behaviour of fifteen identical drones, tested with same input under the same conditions and to analyse how the mathematical model (one for each drone) differ one from the other.

The first step was developing a program that shows the data collected in the laboratory, where the drones were subjected to different tests. The main advantage that the developed program has, is that the user can overlap the data collected from different tests for UAVs and can easily understand the differences between the behaviour of the different systems under study.

Finally the system identification is performed, which is a methodology for building mathematical models of dynamic systems using measurements of the input and output signals of the system. This process is repeated for each of the fifteen UAV, thanks to grey box model approach (which will be explained later) .

1.2 State of the art

The first step towards model identification is done by studying the behaviour of the drone subjected to different tests. One of the most useful tools is Flight Review[1]. This online tool allows to load and analyse UAV autopilot flight data, as well as create a customized series of plots. This software is open source and developed by PX4 Development Team and Community.

Another software that works similarly is Flight log analyser[2], which is an application released for the R2021a version of MATLAB. These two tools do not allow overlapping data from different flights or different drones, making comparison between multiple tests difficult. An overview of several different software can be seen in: [3].

Once the flight data have been collected, the goal is to find the mathematical model that best fits them. Model identification is a really discussed topic in literature, an overview of the subject can be found in “Practical grey-box process identification: theory and applications” by Bohlin, Torsten P. [4]. This topic was also studied by Ljung in: [5] where the white-box, black-box and all the palette of grey shades-box modelling are used to build different mathematical models. A less general approach about model identification is presented in: [6] where the discussion is focused on flight vehicles model identification. In 2014, Qianying Li investigated the problem of “Grey-Box System Identification of a Quadrotor Unmanned Aerial Vehicle” [7]. This thesis focuses on the modelling and identification of a quadrotor

system through the implementation of a grey-box identification method for control purpose. A theoretical approach to the problem of identification can be seen in [8]: “When the Grey-box model is constructed and parametrised, the next step is to estimate the optimal values of parameters using experimental data. There are numerous estimation methods that can be used. However, probably the most common approaches are Prediction Error Method, Least Squares Estimation, or Maximum Likelihood Estimation.” Then the article proceeds to analyse the Maximum likelihood Estimation method in detail. This topic is really discussed in literature, in particular in [9], Identification of Linear Time-Invariant Systems is studied.

Model identification can be both in frequency domain and in time domain, the latter is discussed in [10], where longitudinal dynamics of a UAV is presented. Some software have been created to assist system identification, the most famous is the MATLAB System Identification Toolbox [11], which can be used in time domain as well as in the frequency one. This tool uses output-error method to estimate the unknown parameters in an Linear Time Invariant model. It maximize the likelihood function, that is equivalent to minimizes the cost function shown in [8], by using suitable initial parameter values. The unknown parameters are iteratively determined.

Finally a detailed paper about UAV identification is: [12], where identification of the attitude dynamics of a quadrotor is analysed.

Chapter 2

Drone dynamics: theoretical basis

In this chapter a detailed description about reference frames and the dynamics of the drone will be shown. Both of them will be deeply used in the thesis, so it is fundamental to give univocal definitions about how they are defined and implemented.

2.1 Reference frames

A generic rigid body has 6 Dof: three degrees described by coordinate x , y , z and other three degrees of freedom describing the orientation of the body: pitch, roll and yaw. These 6 Dof are usually defined with respect to two types of coordinate systems: Body frame and NED Frame which will be used to analyse flight data in the next chapters, thus they will be described in detail here.

Body frame

This type of reference frame consists of an origin that is typically placed at the body's center of gravity and three orthogonal axes that comprise a right-handed system (Figure 2.1). These axes are usually configured to the UAV in such a way that the x -axis is pointing forward, the y -axis is pointing to the right, and the z -axis is pointing down.

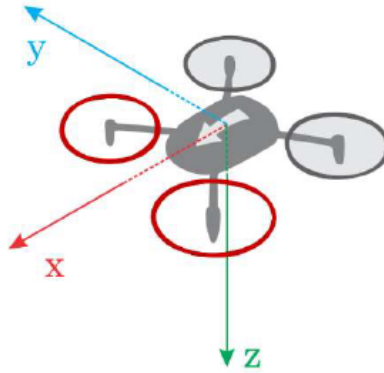


Figure 2.1: Body frame

Inertial frame: Nord East Down

The NED frame is defined such that the North and East axes form a plane tangent to the earth's surface at its present position, assuming a WGS84 ellipsoid model of the earth.

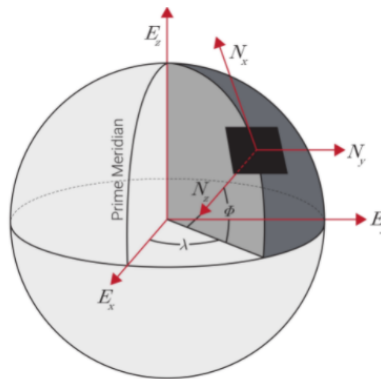


Figure 2.2: Ned frame

As shown in the Figure 2.2, the NED frame contains three orthogonal axes in which the N_x axis points to true North, the N_z axis points towards the centre of the Earth and the N_y axis completes the right-handed system pointing East.

2.2 Euler angles and Quaternions

In order to switch from this two reference systems, rotation matrices, Euler angles and Quaternions must be introduced.

The Euler angles are three angles which describe the orientation of a rigid body with respect to a fixed coordinate system.

Given two reference frames (x, y, z) and (b_1, b_2, b_3) , the Euler's theorem states that: "The most general displacement of a rigid body with a fixed point is a rotation about some axis through some angle" (Figure 2.3).

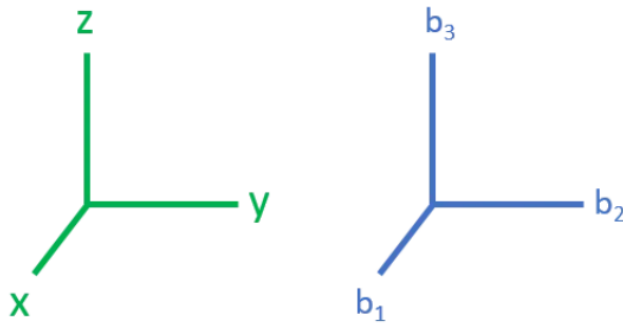


Figure 2.3: Frames x, y, z and b_1, b_2, b_3

Given the rotation about the x, y, z is possible to go from one reference frame to the other thanks to a sequence of three rotations (Figure 2.4).

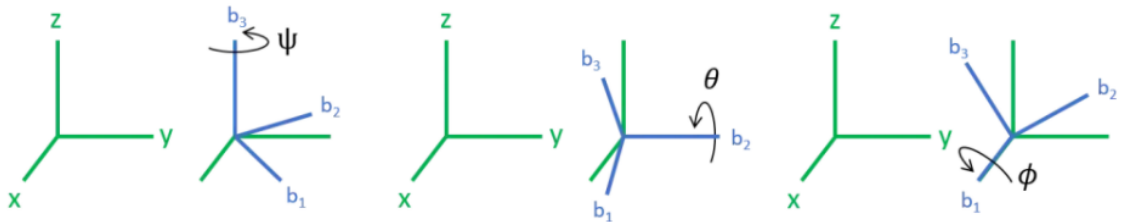


Figure 2.4: Rotations

Each rotation can be represented by a matrix A , called attitude matrix or Direction Cosine Matrix (DCM).

$$A_3 = \begin{bmatrix} \cos \psi & \sin \psi & 0 \\ -\sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.1)$$

$$A_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & \sin \theta \\ 0 & -\sin \theta & \cos \theta \end{bmatrix} \quad (2.2)$$

$$A_1 = \begin{bmatrix} \cos \phi & 0 & -\sin \phi \\ 0 & 1 & 0 \\ \sin \phi & 0 & \cos \phi \end{bmatrix} \quad (2.3)$$

A_3 is the rotation about the third axis of an angle ψ , A_1 is the rotation about the first axis of an angle θ , and A_2 is the rotation about the second axis of an angle ϕ . The main advantage of the Euler angles is the clear physical interpretation, but the con is that there is a singularity at $\sin \psi = 0$, $\sin \theta = 0$ and $\sin \phi = 0$. This issue is called: gimbal lock. It is necessary to do a step further and define Quaternions. This new parametrisation is less intuitive and has one redondant parameter, but there aren't singularities. Another advantage of the quaternion presentation is its computational efficiency. While the Euler angle formulation contains nonlinear trigonometric functions, the Quaternion expression results in much simpler. Given for example a rotation about the first axis:

$$\cos \phi = \frac{1}{2} \text{trace}(A) - 1 \quad (2.4)$$

$$e_1 = \frac{(A_{23} - A_{32})}{2 \sin \phi} \quad (2.5)$$

$$e_2 = \frac{(A_{31} - A_{13})}{2 \sin \phi} \quad (2.6)$$

$$e_3 = \frac{(A_{12} - A_{21})}{2 \sin \phi} \quad (2.7)$$

it is possible to evaluate:

$$q_1 = e_1 \sin \frac{\phi}{2} \quad (2.8)$$

$$q_2 = e_2 \sin \frac{\phi}{2} \quad (2.9)$$

$$q_3 = e_3 \sin \frac{\phi}{2} \quad (2.10)$$

$$q_4 = \cos \frac{\phi}{2} \quad (2.11)$$

The corresponding vector of Quaternions can be written as follow:

$$q = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} \quad (2.12)$$

where q_1, q_2, q_3, q_4 are real numbers such that $\|q\| = 1$. Is now possible to define the Rotation matrix as follows:

$$A(q) = \begin{bmatrix} q_1^2 - q_2^2 - q_3^2 + q_4^2 & 2(q_1q_2 + q_3q_4) & 2(q_1q_3 - q_2q_4) \\ 2(q_1q_2 - q_3q_4) & -q_1^2 + q_2^2 - q_3^2 + q_4^2 & 2(q_3q_2 + q_1q_4) \\ 2(q_1q_3 + q_2q_4) & 2(q_3q_2 - q_1q_4) & -q_1^2 - q_2^2 + q_3^2 + q_4^2 \end{bmatrix} \quad (2.13)$$

2.3 Modelling UAV dynamics

Starting from the Newton's second law:

$$F = ma \quad (2.14)$$

where F is the force, m the mass and a the acceleration, crossing both sides of the equation with a position vector (r) from some origin O , the rotational law of motion about the origin O is obtained:

$$\vec{r} \times F = \vec{r} \times ma \quad (2.15)$$

The equation to moment M can be simplified in this way for systems with constant mass distribution:

$$M = I \frac{d\omega}{dt} = I\Omega \quad (2.16)$$

where I is the moment of inertia and Ω the angular velocity. The moment of inertia is defined in physics for scalars as:

$$I = mr^2 \quad (2.17)$$

A more general representation where the mass is integrated over the entire body of the object is:

$$I = \int_{body} r^2 dm \quad (2.18)$$

Assume r made from coordinate with components $[x, y, z]$:

$$\mathbf{I} = \int_{body} \mathbf{r}^2 dm = \int_{body} [\mathbf{r} \times \mathbf{r} \times dm] \quad (2.19)$$

$$\mathbf{I} = \int_{body} \begin{bmatrix} 0 & -z & y \\ z & 0 & -x \\ -y & x & 0 \end{bmatrix} \begin{bmatrix} 0 & -z & y \\ z & 0 & -x \\ -y & x & 0 \end{bmatrix} dm \quad (2.20)$$

$$\mathbf{I} = \begin{bmatrix} \int (y^2 + z^2) dm & -\int (xy) dm & -\int (xz) dm \\ -\int (xy) dm & \int (x^2 + z^2) dm & -\int (yz) dm \\ -\int (xz) dm & -\int (yz) dm & \int (x^2 + y^2) dm \end{bmatrix} \quad (2.21)$$

$$\mathbf{I} = \begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{xy} & I_{yy} & -I_{yz} \\ -I_{xz} & -I_{yz} & I_{zz} \end{bmatrix} \quad (2.22)$$

With the hypothesis of small perturbations about the trim condition, the equations of motion expressed in the quadrotor body-fixed frame for the longitudinal dynamics are the following:

$$\dot{u} = X_u u + X_q q - g \sin(\theta) + X_\delta \delta_{long} \quad (2.23)$$

$$\dot{q} = M_u u + M_q q + M_\delta \delta_{long} \quad (2.24)$$

where u is the longitudinal body velocity and q the pitch rate. The first equation is about the position dynamics, instead the second one describes the attitude. These equation will be implemented in chapter number six, for the development of the model to identify.

The parameters X_u , X_q , M_u , M_q are the dimensional stability derivatives with respect to velocities and rates, X terms have dependence from the mass, instead M terms have the dependence from inertia I seen before.

δ_{long} is the control input normalised between -1 and 1, X_δ and M_δ are the dimensional control derivatives with respect to the control input and finally g is the gravitational acceleration.

Chapter 3

Analysis of the ANT-X drone's data through Log Analyser

Here will be given a description of the developed tool Log Analyser. It will be shown how it was created and how it can be used. The software was entirely developed with the goal to be user friendly, but also with some useful features for further analysis.

3.1 Description of Log analyser software

It was developed in Python, with the support of PySimpleGUI package. For the following section, Figure 3.4 and 3.5 have to be taken as references. In this software, the user can load both ulog and CSV format. When the file is loaded, the software places it in the column of Normal or Custom mode, based on the topics contained in the file. Notice that not all the plots are available for both the modes: the grey labels (Figure 3.5) are available only in Custom mode.

On the right side of the Log analyser interface, there is a column with the list of all the plots available. They are divided in the following categories: Attitude, Position, Control and Debug. If the user flags a plot which is not available due to some missing data in the file loaded, the software will skip the plot and will go on with the other required.

At the bottom right there is the section for the plot customisation. If none of the plot customisation features are selected, the result shown by the software will be

the one in Figure 3.1. In this section there are two possible choices:

1. Align starting time to 0: if selected, this command will change the timestamps of all the files loaded and flagged in the left part of the window. The timestamps will go from 0 to the time the test ended. This command is very useful, since it aligns and makes comparable plots of different tests that start at different times (time is the x-axis of the most of the plots), as can be seen in Figure 3.2.
2. Starting and Ending time: this command will trim the plots at a given time (Figure 3.3). The plots can be exported as CSV files. They will maintain the customisation options selected and can be reloaded in the program from the “Load a CSV folder”

The main advantage of this software is the ability to overlap the data collected in different flights, so the comparison can be a lot more easier. The customisation options help the user to trim the log’s data and increase comparability.

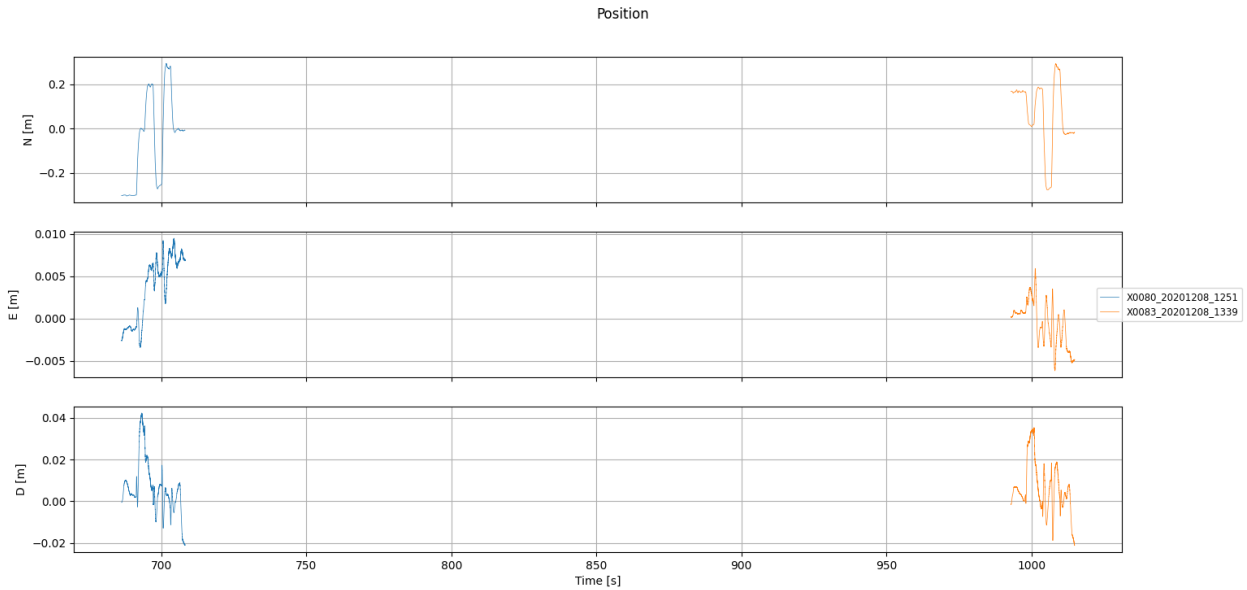


Figure 3.1: Position data without any plot customisation

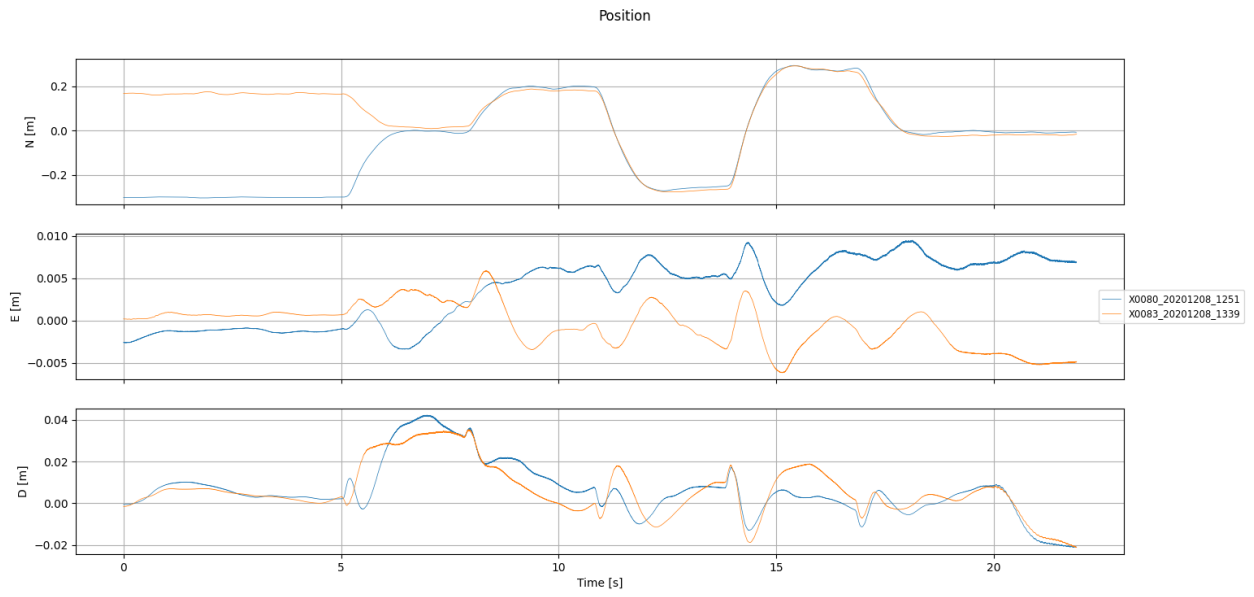


Figure 3.2: Position data with Align starting time selected

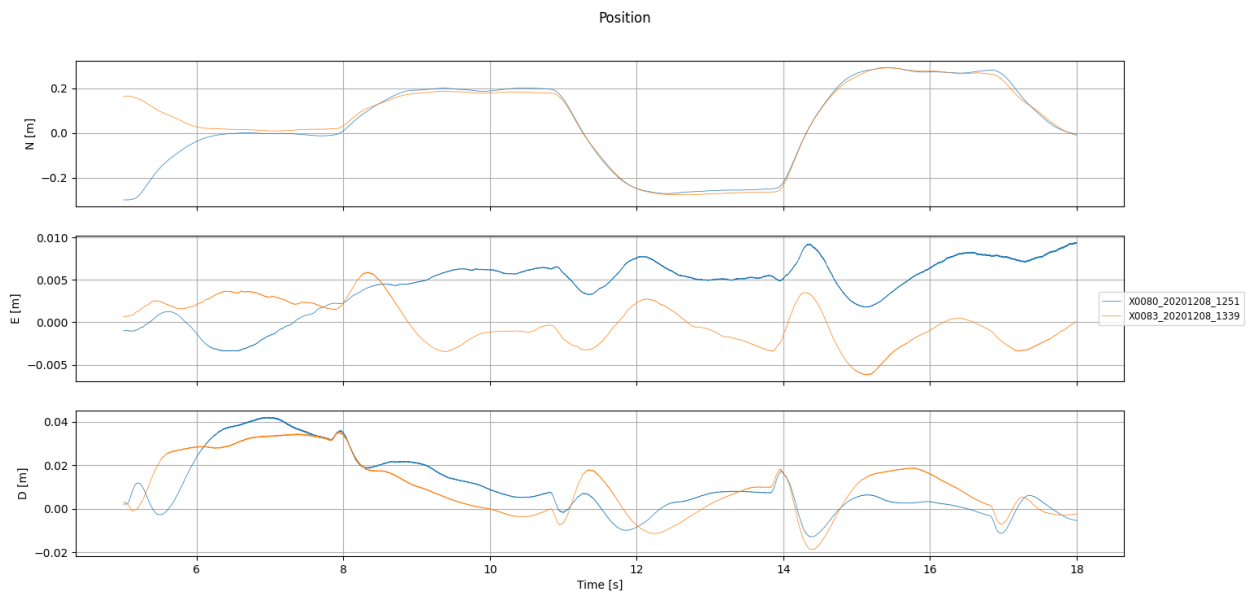


Figure 3.3: Position data with Align starting time and trim starting and ending time selected

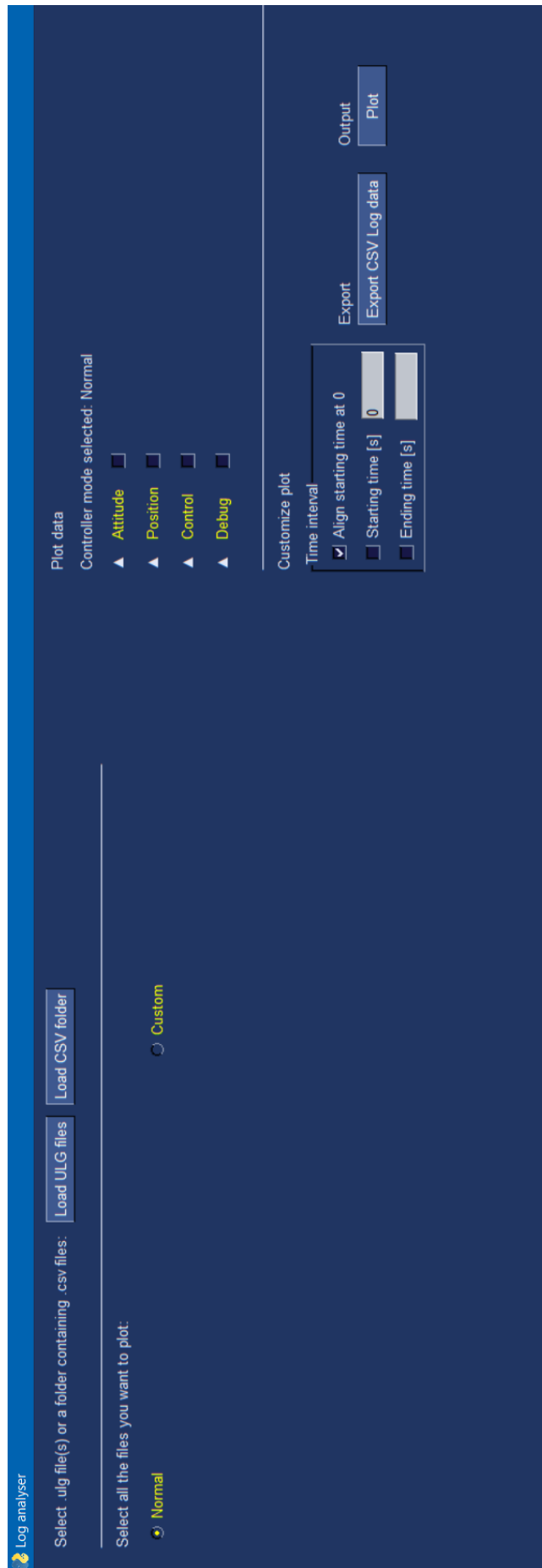


Figure 3.4: Log analyser first interface

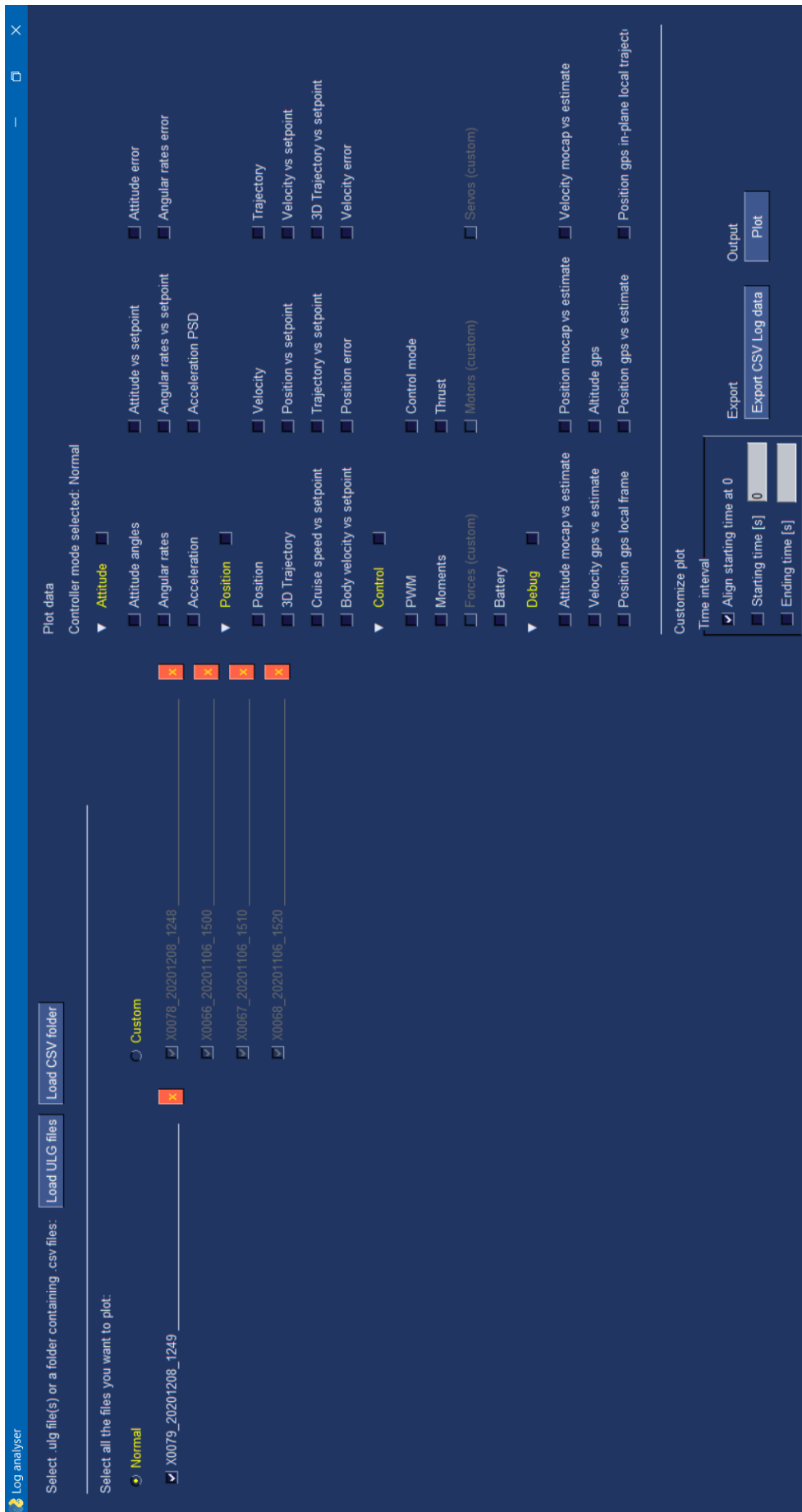


Figure 3.5: Log analyser Log file loaded

Now it will be shown a comparison between the software developed by Matlab and Log Analyser. For the plots with the latter software, use as reference Figure 3.3.

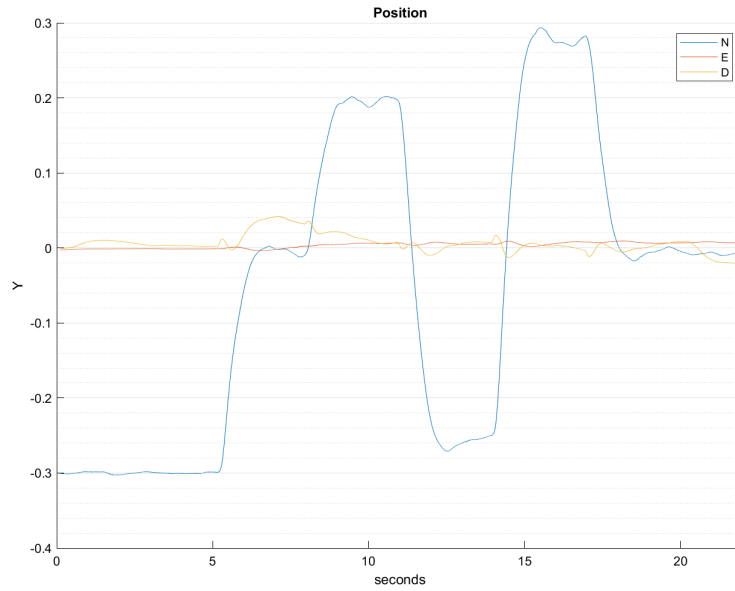


Figure 3.6: Position flight log X0080_20201208_1252 with Matlab tool

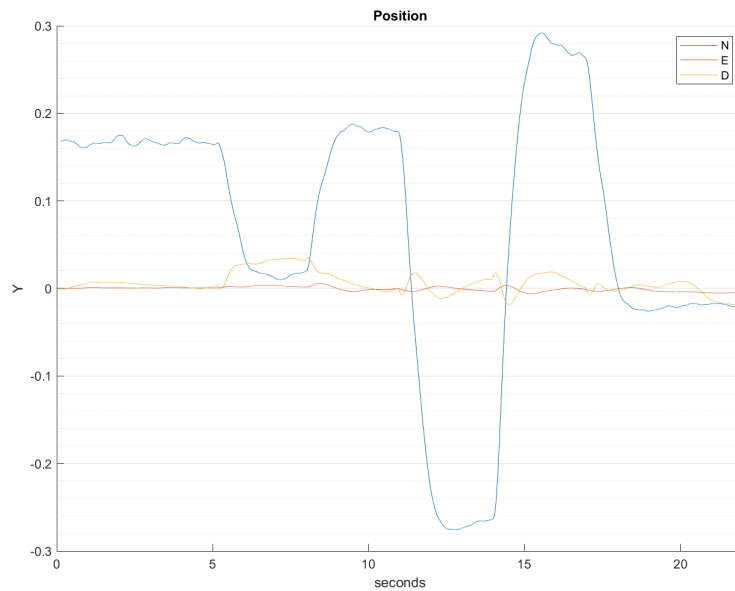


Figure 3.7: Position flight log X0083_20201208_1339 with Matlab tool

Flight Log Analyser (the tool developed by Matlab) it is not developed to support different flight logs, so a plot for each flight log is required (Figure 3.6 and 3.7). The comparison between flight logs can be really difficult, instead with Log Analyser it is pretty simple. Futhermore, the plot customisation shown before are not available with the Flight Log Analyser.

When the user has selected the graphs to plot, it is possible to export the file in CSV format. The software will not save the plots selected, but it will export the flight data in CSV format, with the customisation options selected. In this way if, for example, the button starting time is selected, the data will be trimmed at that time and then saved in a folder selected by the user. With the button Load CSV folder, it is possible to load the folder in the software and plot all the graphs needed.

3.2 ANT-X drone specification

The drone fleet under analysis is made up of fifteen ANT-X 2DoF drones.

“The ANT-X 2DoF Drone is a laboratory test-bed designed to be employed for aerospace control research and educational purposes. It is a quadrotor drone constrained to a structure in such a way that the two degrees of freedom of translation and rotation are retained. [...] The 2DoF Drone is designed to run experiments in a safe and controlled environment, and in a repeatable way”. [13]

Main specifications:

Characteristic	Value
Weight[Kg]	0.27
Dimensions[cm]	20 x 20 4
Hovering[min]	7'30”
Battery	LiPO 950mAh 3S
Propellers	3-bladed

Table 3.1: ANT-X 2DoF specifications

The software components on these drones is PX4 firmware, ROS(Robot Operating System) and MATLAB. ROS enables communication between nodes in the network. In the case of the 2DoF Drone, connects the ground control station and the FCC. PX4 is an open source flight control software and its flight logs can be analysed by Flight Review. This online tool allows users to upload flight data, and analyze them through the browser. It will be shown and compared with Log Analyser later in this chapter.

3.3 2DoF Drones tests

For each drone, the following sequence of tests was executed in this order:

1. Hovering test: the drone is armed, then a level attitude set-points is sent to the drone, simulating the hovering condition. After 20s, the drone goes to idle mode and then is disarmed.
2. Attitude steps: the drone is armed, and a sequence of step attitude set-points of increasing magnitude and alternated sign are sent to the drone, then, the drone is disarmed.
3. Position steps: the drone is armed and a sequence of step position set-points of increasing magnitude and alternated sign are sent to the drone, then, the drone is disarmed.

The first and second tests were carried out in 1DoF mode, instead the third was carried out in 2DoF mode. To guarantee the degrees of freedom and to block the other movements, the drone was connected to some bars through some bearings: in the 2DoF mode the drone was sliding, in the other was fixed to them. An example of the test can be seen at: [14]. The same position and attitude controller was employed in all the tests [15]. In order to achieve maximum repeatability, the battery was replaced with a fully charged one at the beginning of each sequence of 3 tests and the tests scripts were automated. Level attitude calibration was carried out before each sequence of three tests.

3.4 Test log analysis with Log analyser

In this section will be showed some of the plots which can be obtained with the developed software and the log data from the Position and Attitude test. Only three flights will be taken into account and overlapped.

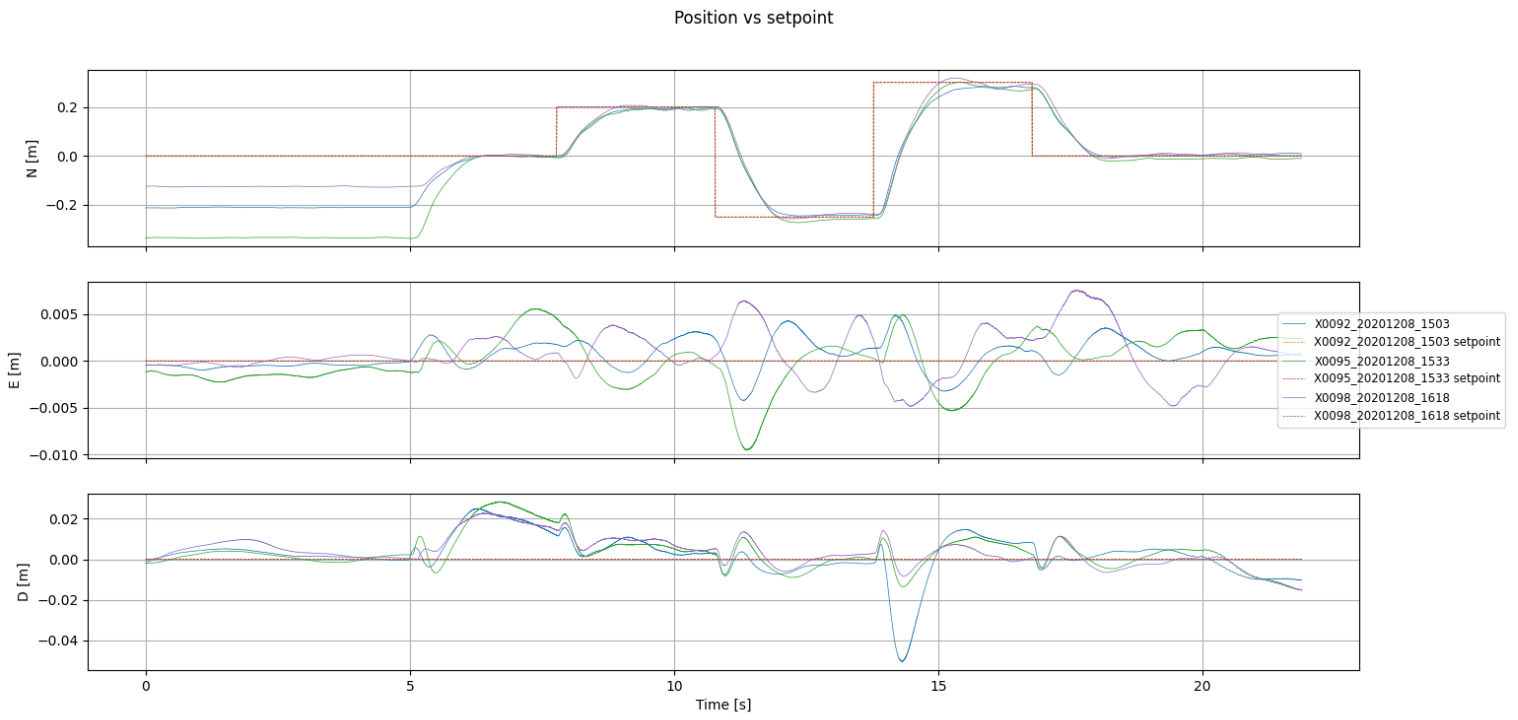


Figure 3.8: Log analyser, position test: position vs set-point position

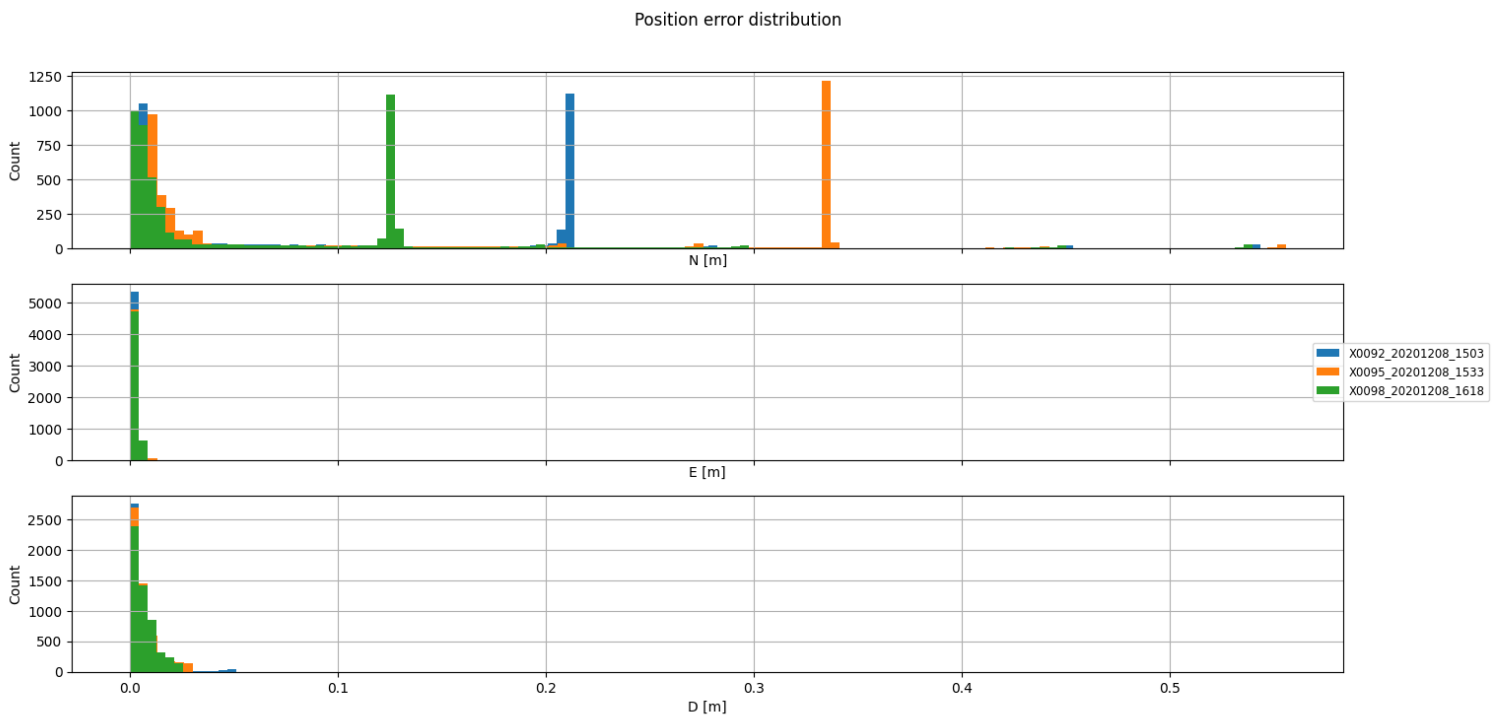


Figure 3.9: Log analyser, position test: position error

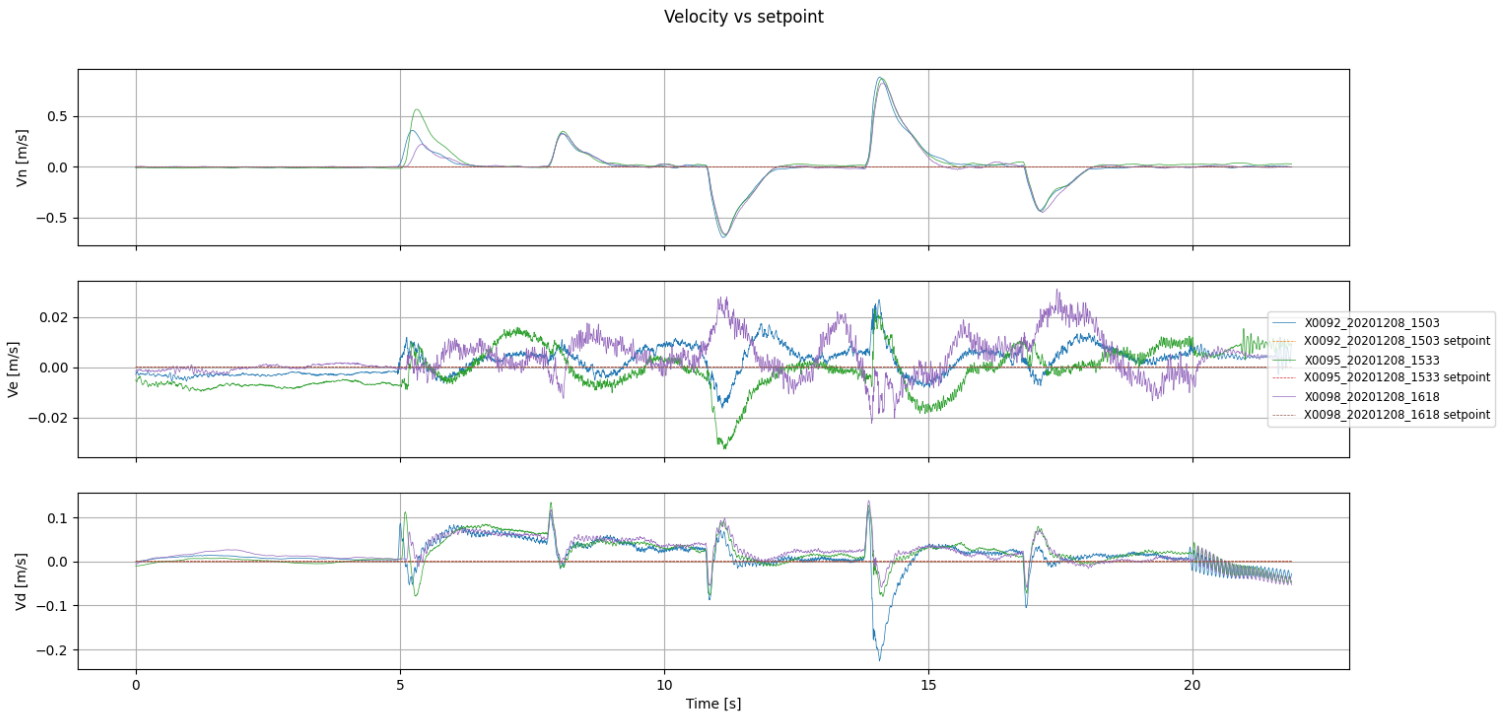


Figure 3.10: Log analyser, position test: velocity vs set-point velocity

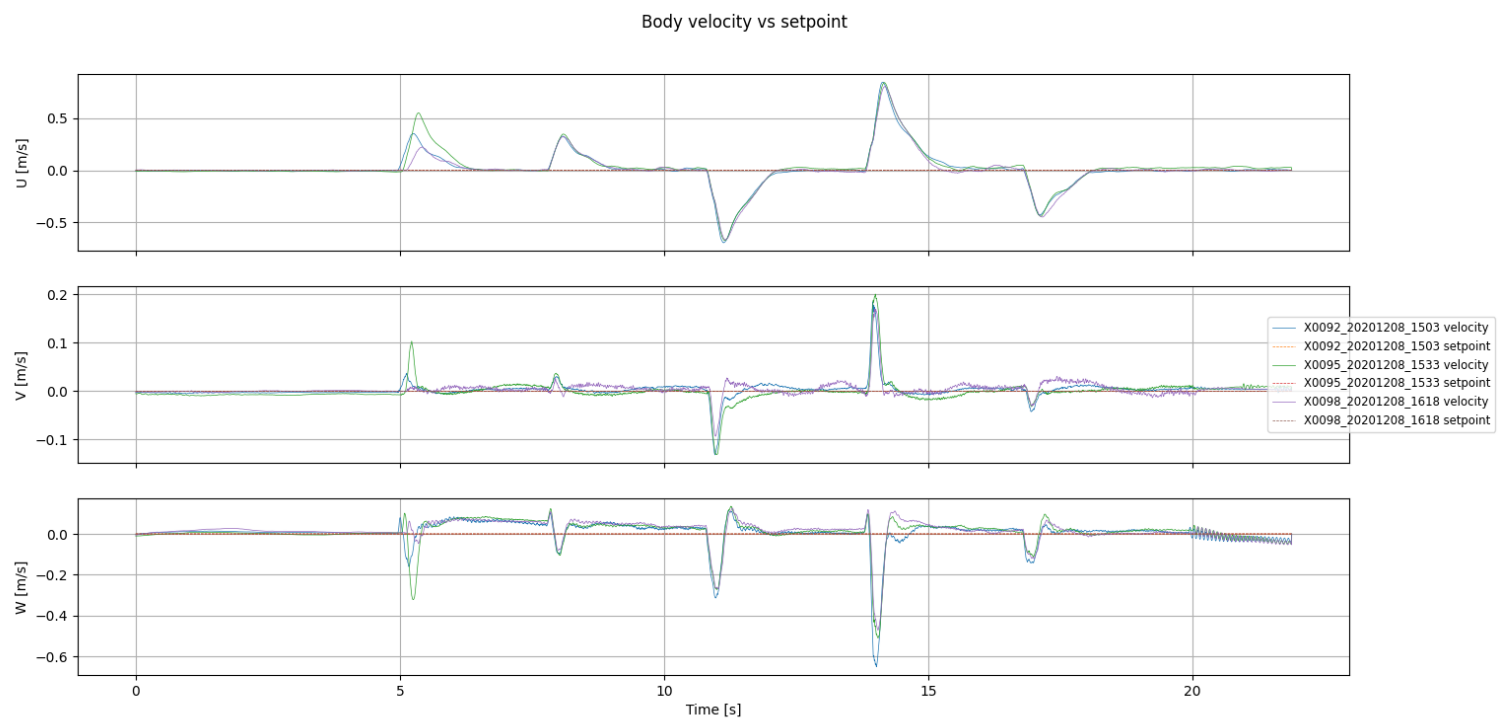


Figure 3.11: Log analyser, position test: velocity (body frame) vs set-point velocity

Attitude angles vs setpoint

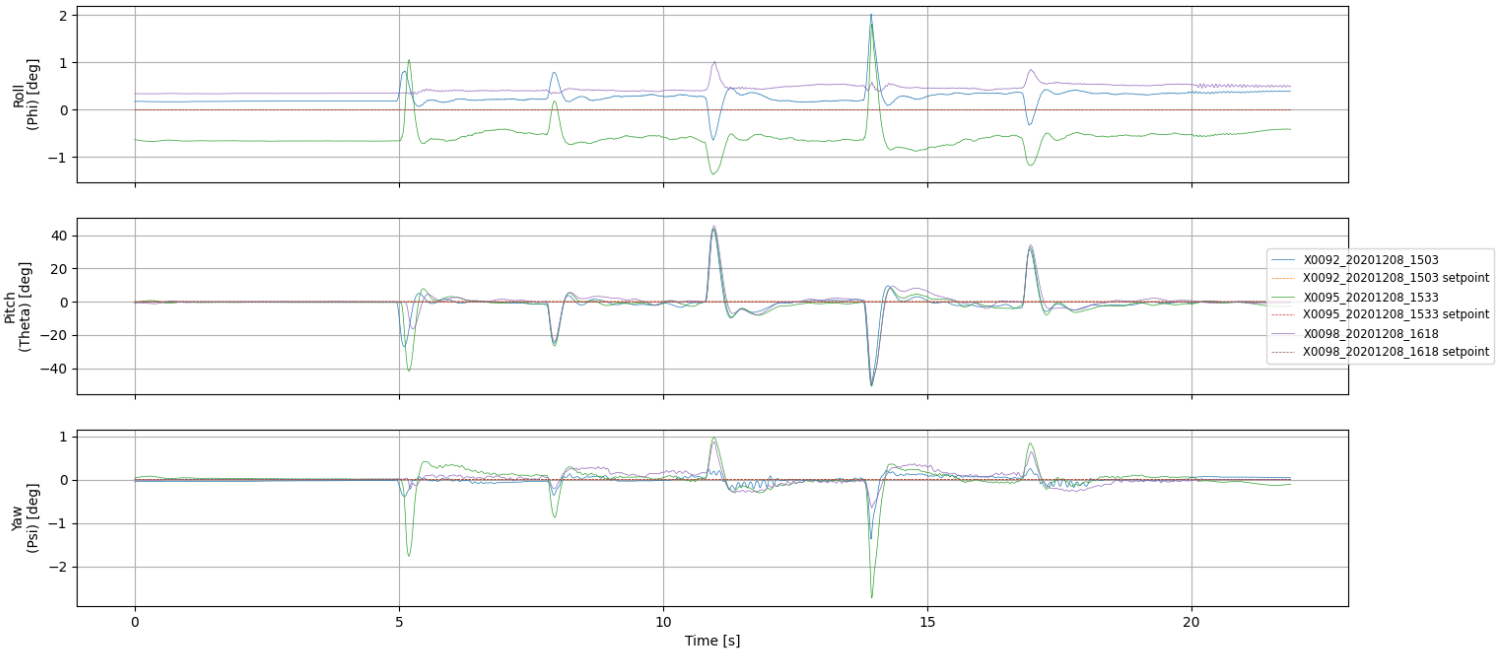


Figure 3.12: Log analyser, position test: attitude angles vs set-point attitude angles

Attitude angles: estimate vs mocap

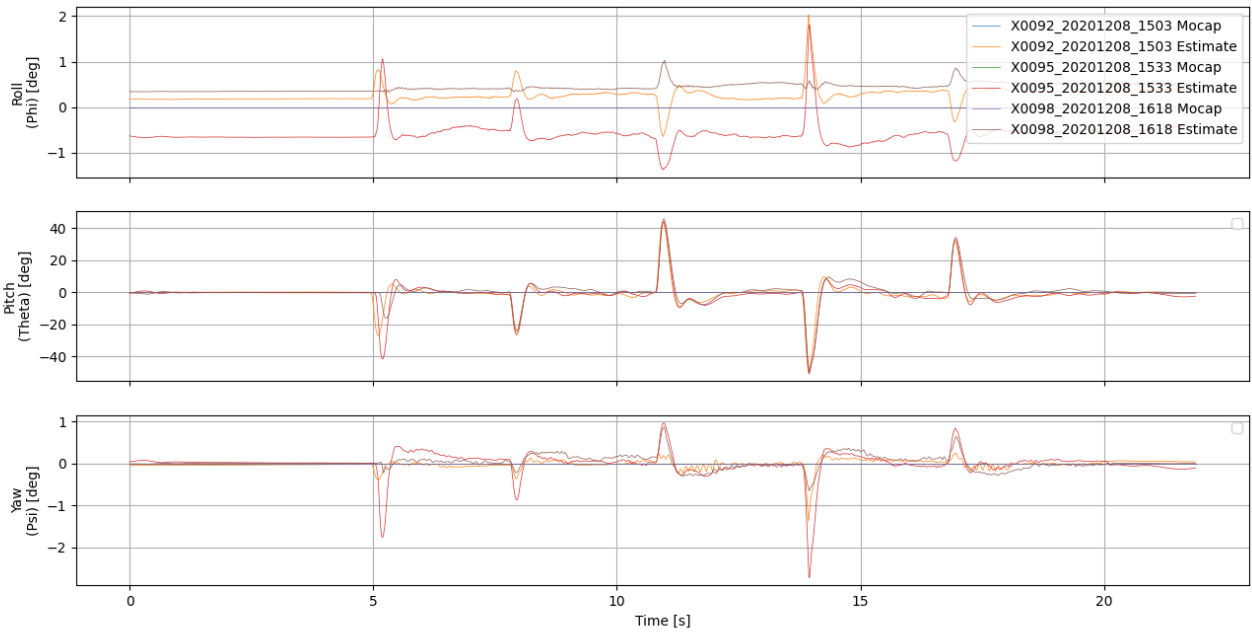


Figure 3.13: Log analyser, position test: attitude angles estimate vs mocap

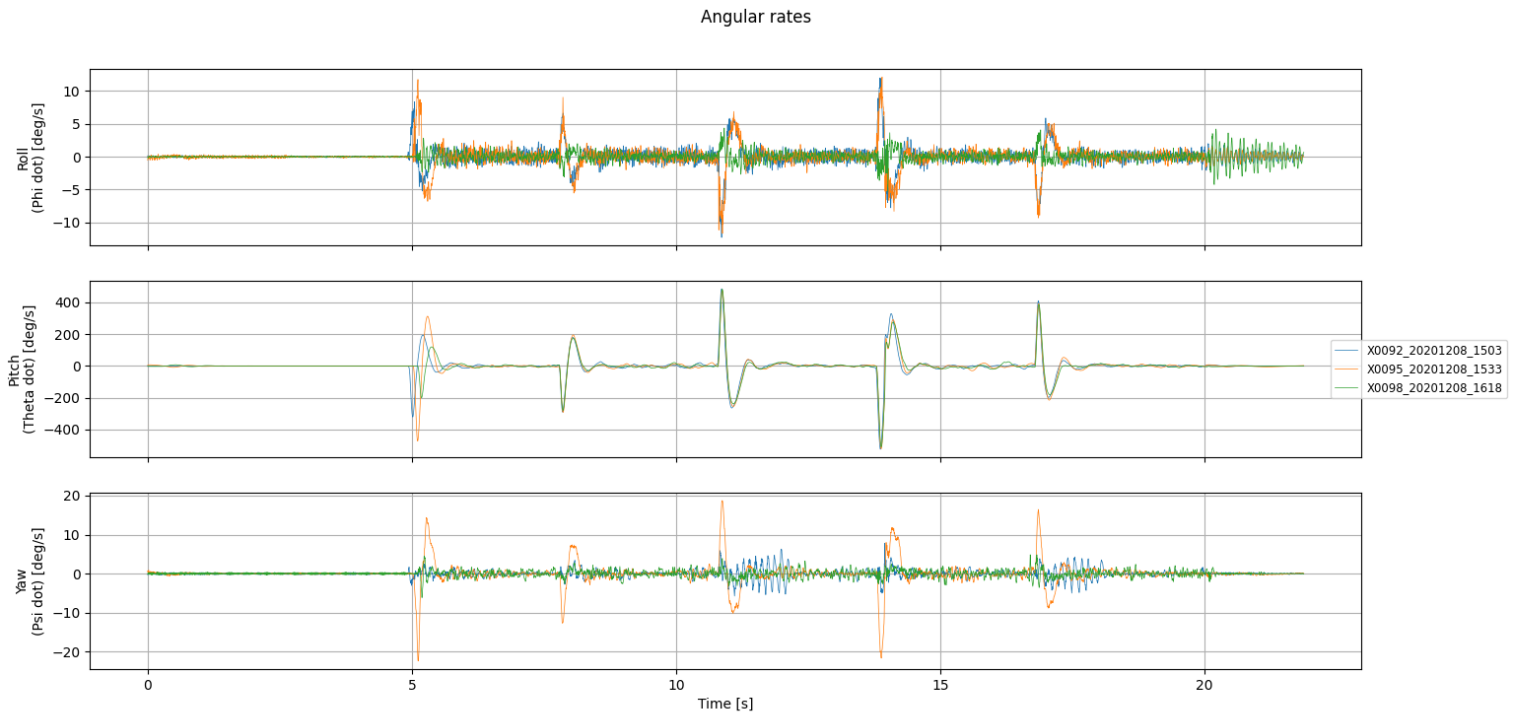


Figure 3.14: Log analyser, position test: angular rates

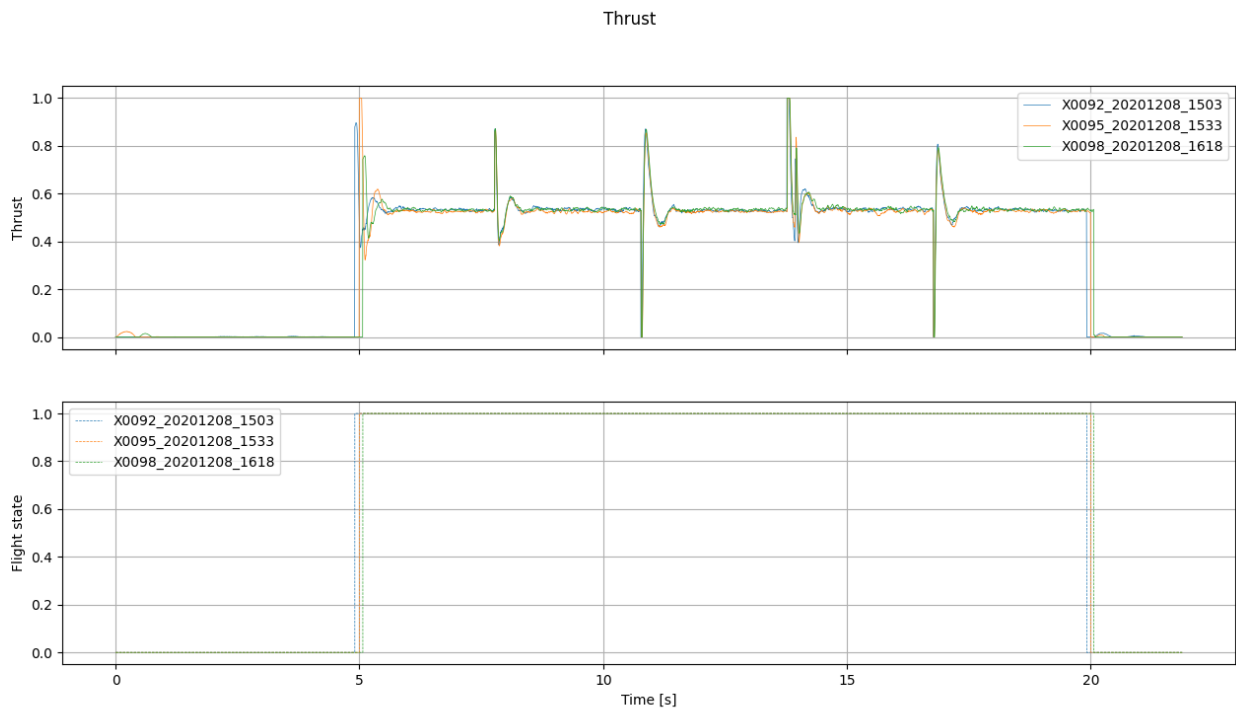


Figure 3.15: Log analyser, position test: thrust

Moments (control actions)

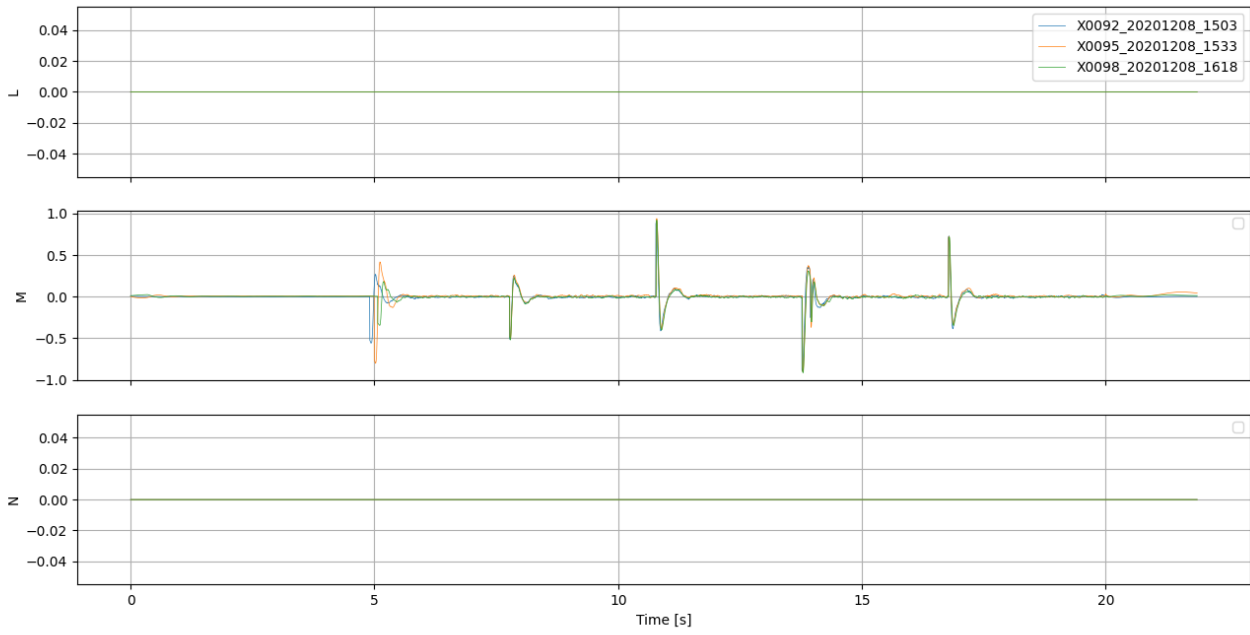


Figure 3.16: Log analyser, position test: moments

Battery

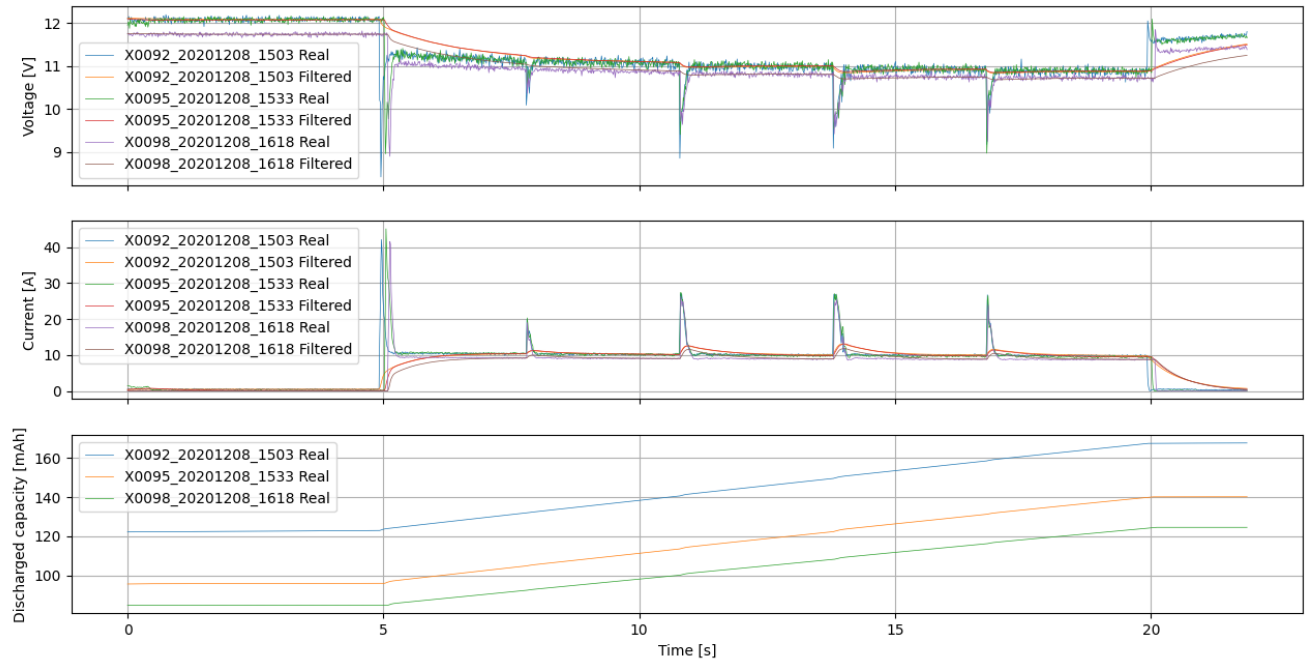


Figure 3.17: Log analyser, position test: battery

Attitude angles vs setpoint

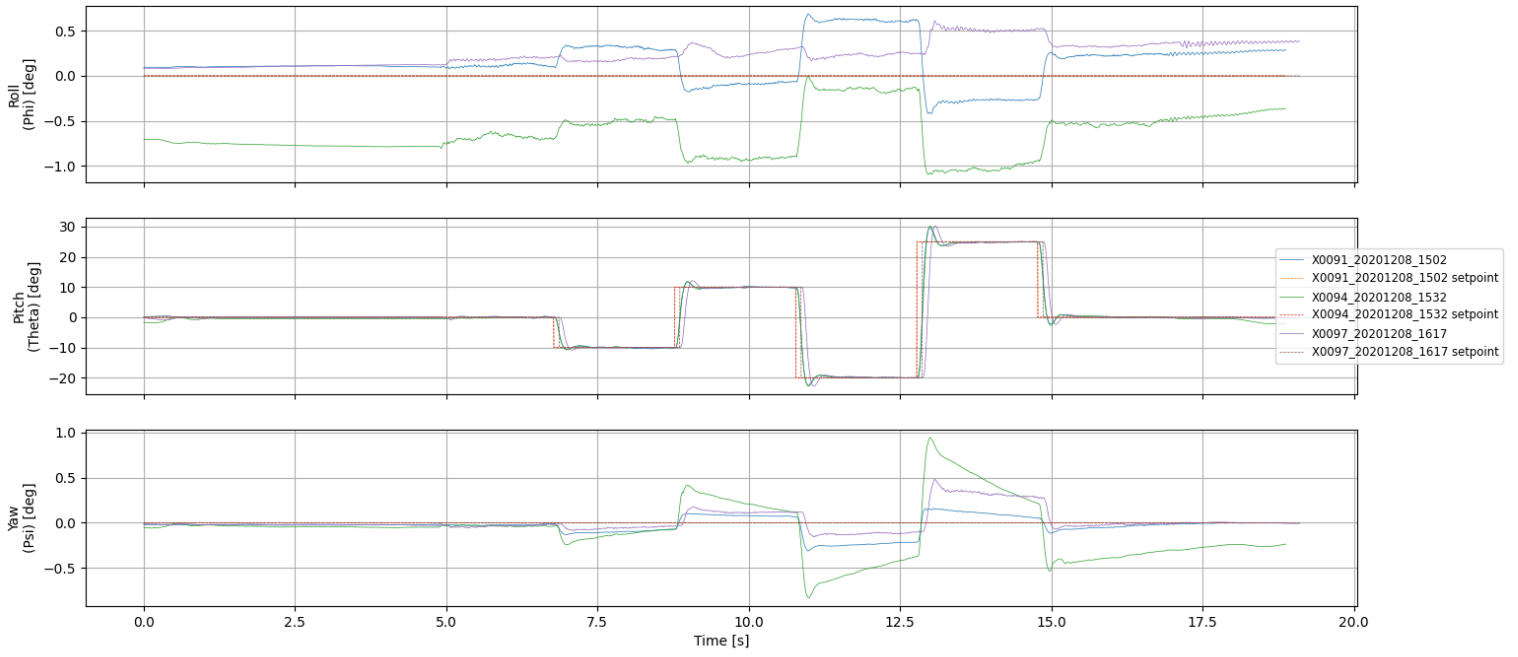


Figure 3.18: Log analyser, attitude test: attitude angles vs set-point

Attitude error distribution

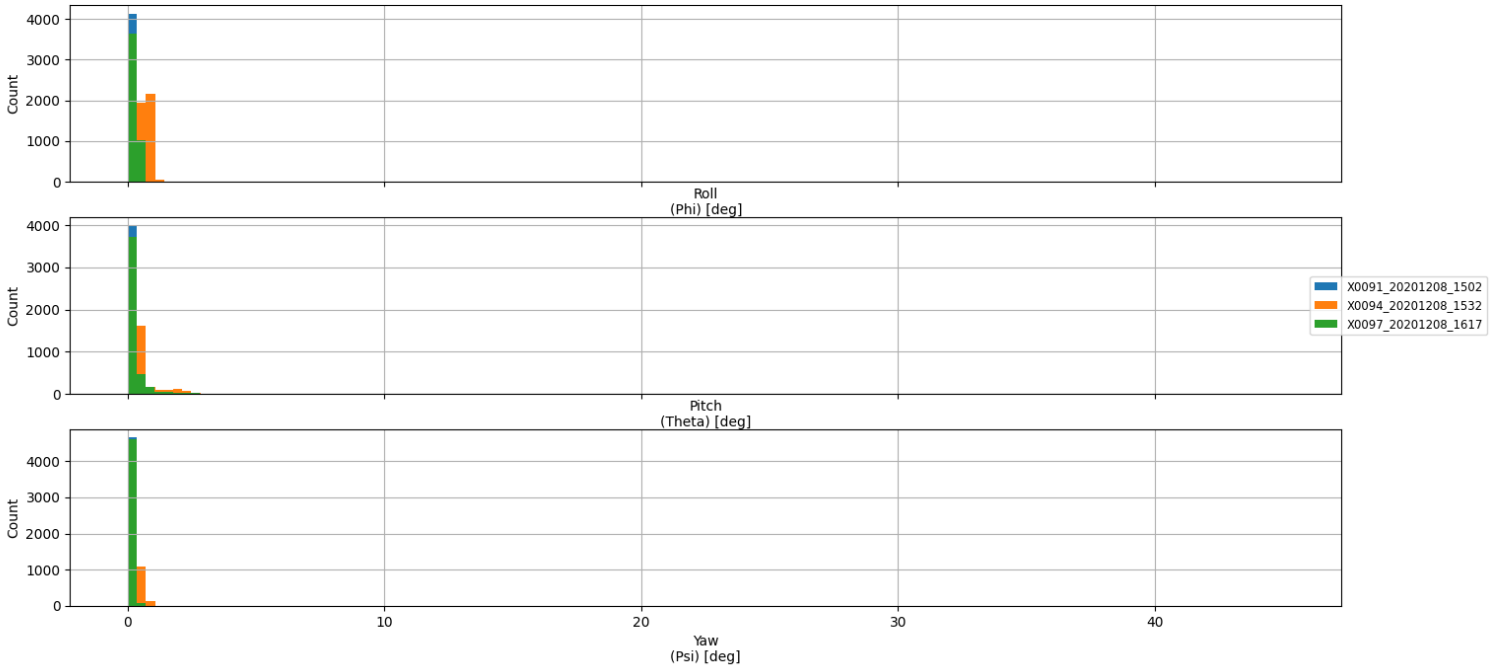


Figure 3.19: Log analyser, attitude test: attitude error

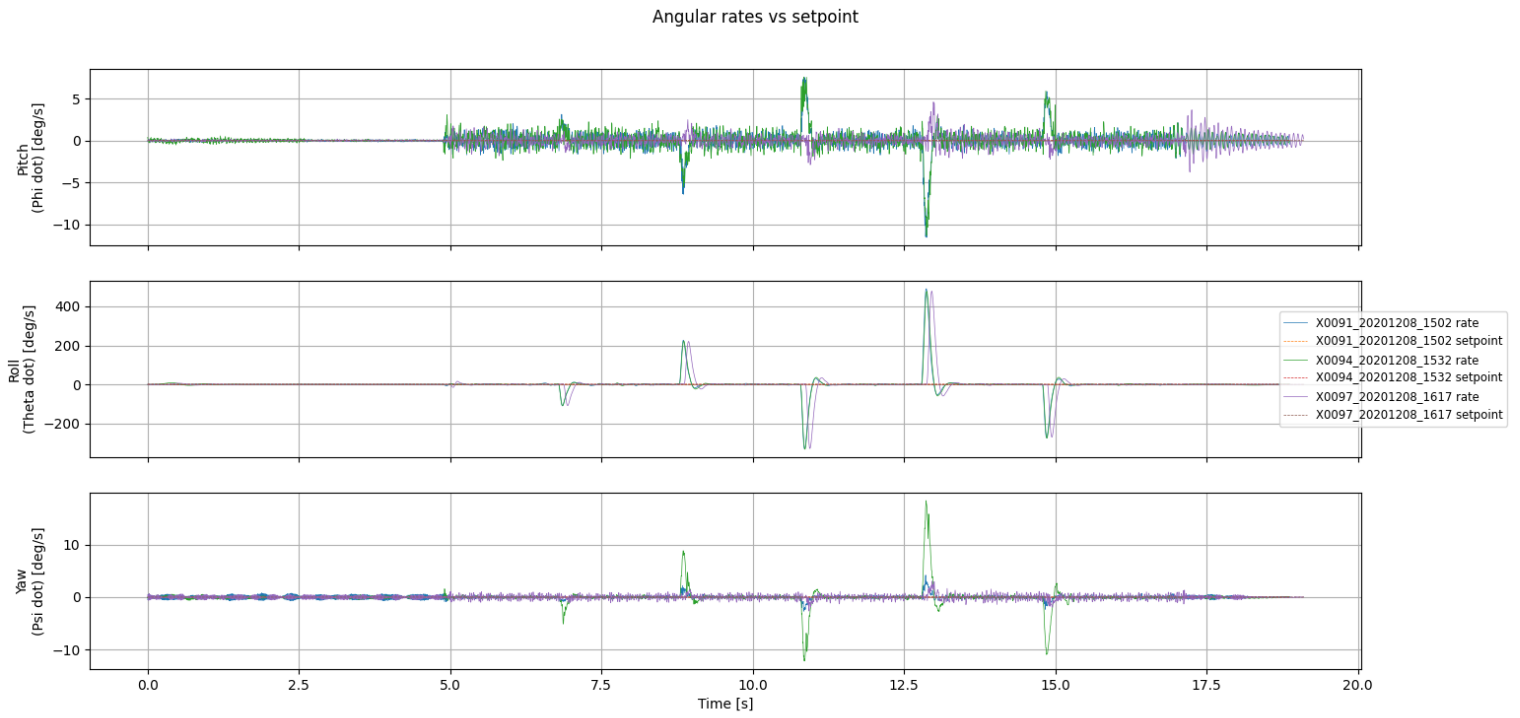


Figure 3.20: Log analyser, attitude test: angular rates vs set-point angular rates

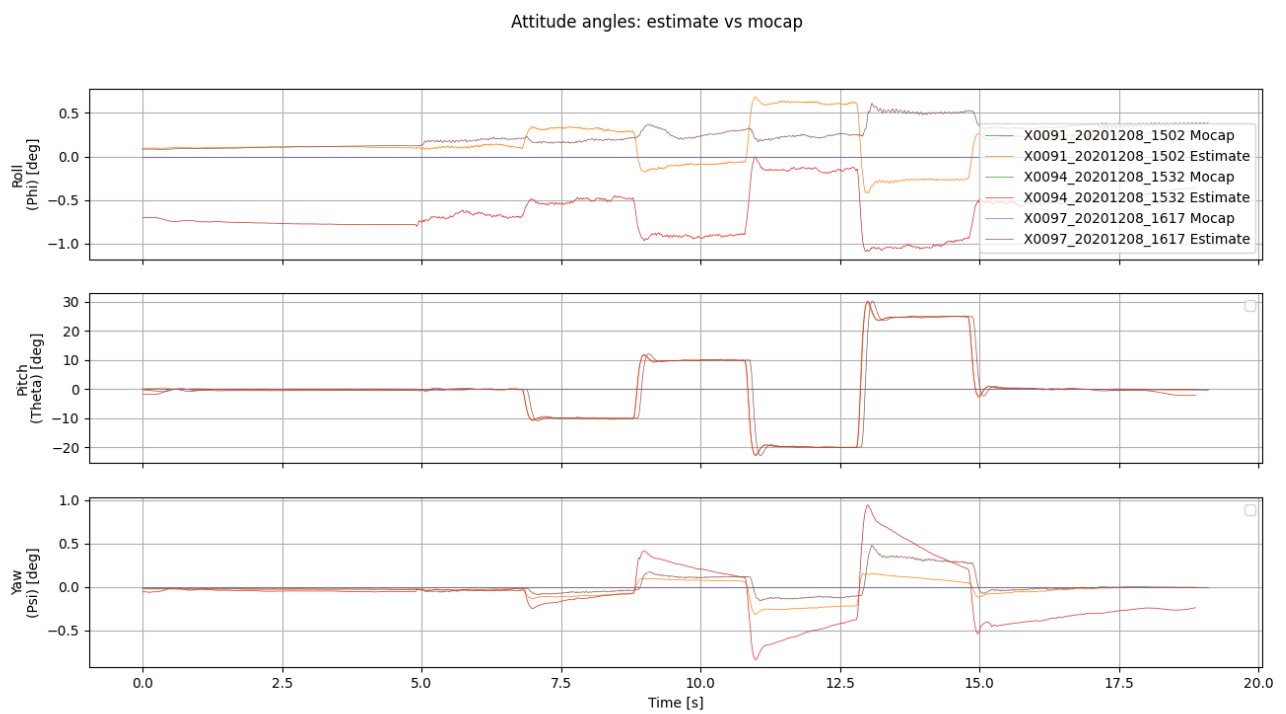


Figure 3.21: Log analyser, attitude test: attitude estimate vs mocap

Moments (control actions)

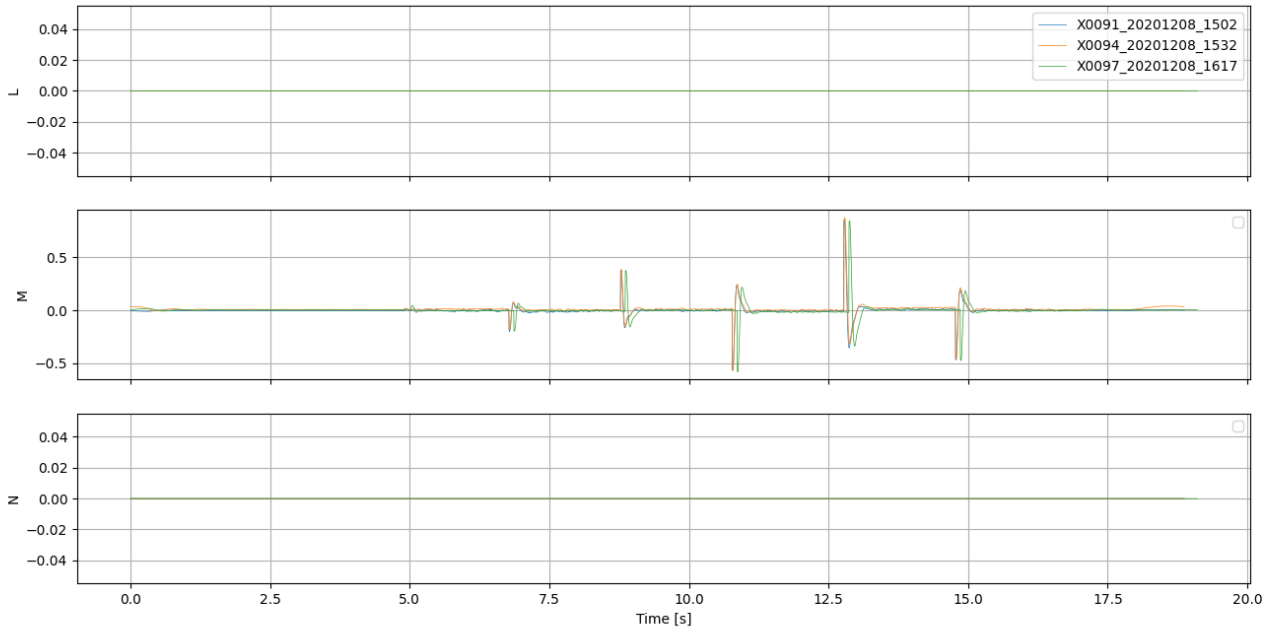


Figure 3.22: Log analyser, attitude test: moments

Thrust

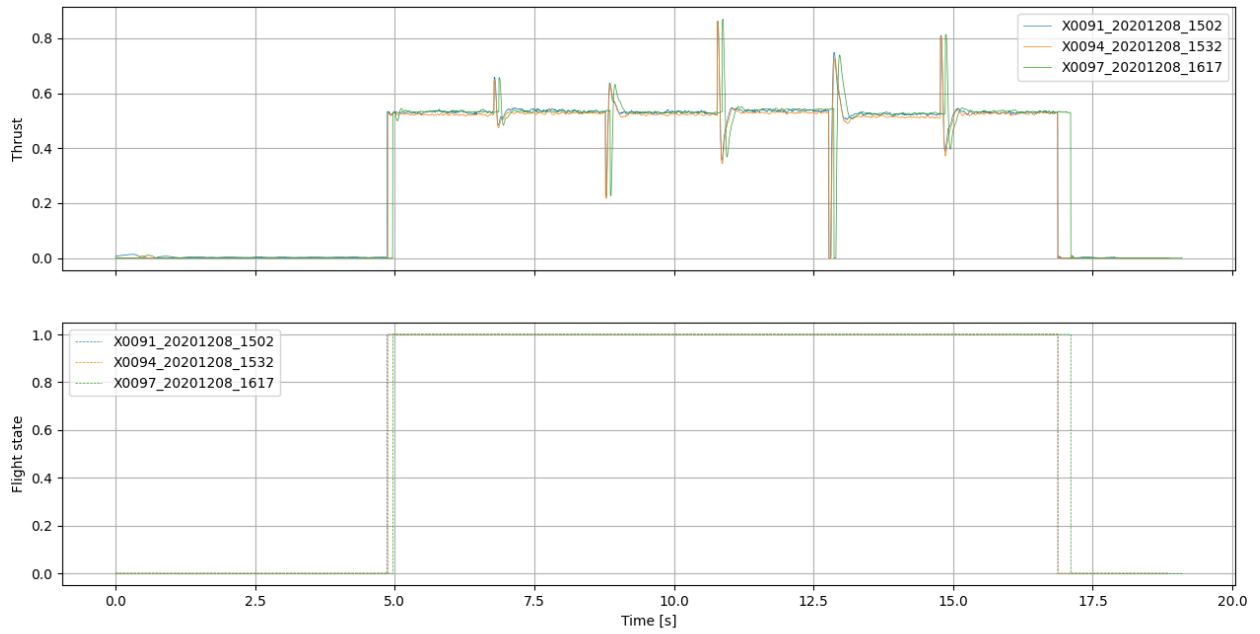


Figure 3.23: Log analyser, attitude test: thrust

In the Position set-point tests, in “position vs set-point” figure (Figure 3.8), is highlighted the step input given to the drones. The “position error” plot instead, gives a better understanding of which is the flight that best follows the set-point. In this case, the green one is the most accurate.

All the plot are in NED frame, except for the one where is specified to be in the body one, this is the case of “velocity vs set-point velocity” plot. The way to go from one reference frame to the other is showed in the second chapter.

Finally in the last plot of position set-point, the battery one (Figure 3.17), is easy to check that the battery was replaced with a fully charged one at the beginning of each sequence of tests, as we said before.

Similar consideration can be done for the attitude set-point plots.

After an qualitative analysis of the data available in the previous plots, it is evident that all the drones had a very similar behaviour, so a deeper analysis is needed. Model identification will be introduced in order to understand the mathematical model behind each drone. Both closed loop and open loop identification will be performed. For the first one, only mathematical model for position set-point will be described, for the latter both position and attitude model will be taken into account.

Chapter 4

Theory of model identification

In this Chapter the theory of model identification will be described: from the definition to state estimation problem, to the process of the model identification and the inputs used in laboratory for flight testing. The grey box modeling approach will be shown, as well as a way to validate the results of the obtained solutions. The entire process from flight testing to the results of the problem will be taken into account.

4.1 State estimation problem

There is an estimation problem every time it is necessary to gather information about an unknown or uncertain parameter by means of experimental observations. Experiments collect data about a quantity (parameter called θ) which want to be estimated.

This parameter can be either:

1. Constant (for example the moment of inertia of an aircraft)
2. Time-varying $\theta(t)$ (for example the attitude of an aircraft during flight).

Information are collected as data samples and are represented as a function of an index t observation instants: $d(t) \ t \in t_1, t_2, t_3, \dots, t_N$. This means that exists a function which correlates data to the parameter that needs to be estimated: $\hat{\theta} = f(d)$. Define $\hat{\theta}$ as the estimate of θ , the collected data plugged into the function f will result in the estimate of the parameter. Depending on the nature of the estimation

problem, the function f might be simple or a complex procedure/algorithm.

In the case θ is constant, the estimation problem can be called:

1. Parametric if θ is finite-dimensional ($\theta \in R^n$)
2. Non-parametric if θ is infinite-dimensional

In the case θ is time-varying, the estimation problem can be called:

1. Prediction problem if given d , $\theta(t)$ for $t > t_n$, needs to be estimated
2. Filtering problem if given d , $\theta(t)$ for $t = t_n$, needs to be estimated
3. Smoothing problem if given d , $\theta(t)$ for $t_1 < t < t_n$, needs to be estimated

In the prediction problem, past data in the window $[t_1, t_n]$ are used to estimate what the value of θ which will be at a future time instant; the further the parameter needs to be evaluated in the future, the harder it will be to obtain its expression.

4.2 System identification

System identification or model identification is the engineering name for the estimation problem. A system identification problem exists every time that, for a given system:

1. There is a mathematical problem which contains one or more uncertain or unknown parameters.
2. It is possible to carry out experiments on the system, to collect data through which uncertainty on the parameters can be reduced.

The difference with the estimation problem is that now the mathematical model of the system is taken into account. What will follow, will be a much more realistic and practical formulation of the general problem of estimation.

The Figure 4.1 shows how the system identification problem is implemented in real life experiments. The values of the parameters are modified by comparing measurements collected on the real system with the simulations of the model. Define:

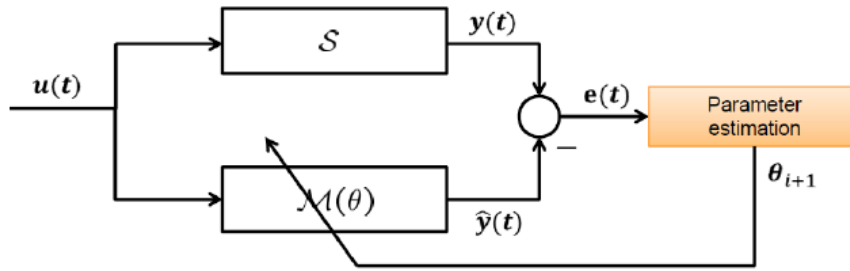


Figure 4.1: Logical approach to the identification problem

1. A system S generating the data
2. A model class $M(\theta)$

For example S might be the flight dynamics of a real aircraft and M its mathematical model representation. This one is not a specific model unless the value of θ is fixed, within the set of values that makes sense for the given problem.

In a realistic situation the model class is not able to capture the dynamics of the real system, no matter the value of θ plugged in the model. This is true because of the simplified physics applied to the problem to avoid unnecessary complexity of the model. The true system is too complicated to be captured by the mathematical point of view. For analysis purposes, things are simpler if it is assumed that the true system does belong to the model class $S \in M(\theta)$, so there is an optimal θ that allows picking an element belonging to the model class which matches the true system. Thanks to this assumption a solution to the problem can be found. The typical approach to the solution of the problem is:

1. Define a metric $J(\theta)$, which is a function of the data and the model class. J must measure how well the model class matches the data for a given θ , so the smaller is the metric, the better the model approximates the measured data.
2. Solve the optimization problem:

$$\hat{\theta} = \operatorname{argmin} J(\theta) \quad (4.1)$$

4.3 Model identification process

No matter what specific application of the model identification problem, the process is characterized by the same steps:

1. Definition of the model class depending on the intended application of the identified model (simulation, prediction, control law design, ...) and on the specific requirements of the application (flight control, aeroelastic analysis, ...). Appropriate modelling choices and assumptions are made and the parameters to be estimated are defined, at this point structural identifiability must be verified.
2. Experiment design: the input sequence used to excite the system must ensure experimental identifiability, but on the other hand, the experiment must be safe and repeatable. This means that the response to the applied input must not lead the system into unsafe conditions and that the shape of the input sequence must be suitable for: either repeatable manual application or easy automatic implementation. Depending on the specific application, experiments are carried out: in open-loop disabling the feedback systems if there is any, or in closed-loop, to guarantee stability during tests. The control system tends to suppress the effect of perturbations so it is harder to carry out parameter estimation.

4.4 Input sequences for flight testing

Sine sweep is the most commonly used input, it is a sinusoid with a frequency increasing over time (Figure 4.2)

The law of increasing of the frequency is defined by the experiment. This input is an approximate way of implementing a sum of many sinusoids with frequencies ranging from the minimum to the maximum one. The sine sweep would be almost impossible to be applied by a human and non automatic test pilot, so there are some simpler sequences that can simulate that behavior (Figure 4.3).

The 3211 sequence is a sequence of steps applied to the pilot controls. Its name comes from the time duration of signals: the last two step have the duration of 10 seconds, the one before lasts 20 seconds and the first one lasts 30 seconds.

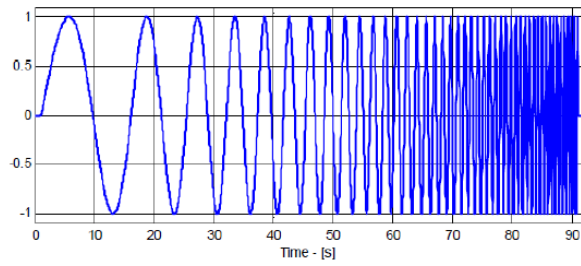


Figure 4.2: Sine sweep

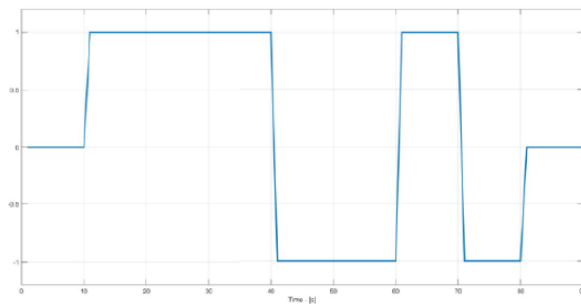


Figure 4.3: 3211

The simplest sequence of all is the Doublet (Figure 4.4), which consists of the application of a step input, then reverse it and then go back to trim. This sequence is not as informative as the 3211 sequence or the sweep but is very easy to use and usually contains enough information to clearly identify models to be sure that they actually capture the response of the true system in a satisfactory way.

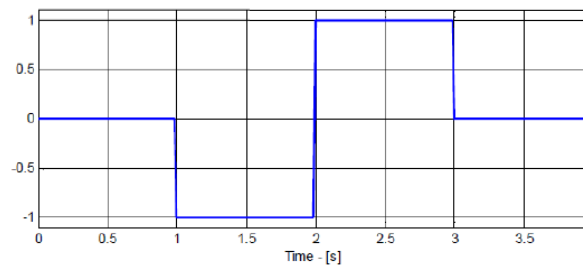


Figure 4.4: Doublet

4.5 Maximum likelihood estimation and Output-error method

Let's now see how to build actual estimators for given problems. There are many approaches to estimator design, but the most classical one is the maximum likelihood estimation.

4.5.1 Likelihood Principle

To formulate the principle an initial problem statement is needed: assume that N independent identically distributed observations x_i with $i = 1 \dots N$ are available. The probability density function f of the random variable is:

$$x_i \cong f(q_i|\theta), \quad i = 1, \dots \quad (4.2)$$

At this point the likelihood function L is defined as the joint probability function of the observed data-set:

$$L(x|\theta) = f(x_1|\theta)f(x_2|\theta), \dots, f(x_n|\theta) \quad (4.3)$$

The joint probability function is given by the product of the probability density function of each random variable evaluated at the corresponding measurement. Considering the probability density function $f(q_i|\theta)$ to be a Gaussian, and picking values of x near the expected value, the probability density function will be high, this means that x is a highly probable value (Figure 4.5).

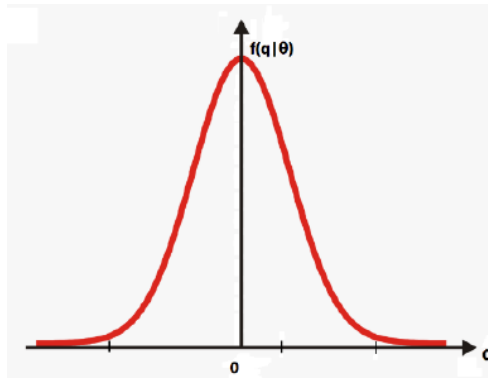


Figure 4.5: Gaussian probability

On the other hand by picking values of x that are far to the right or far to the left the probability density function will be low, this means that x is a low probable value. The maximum likelihood principle consists in choosing as estimate θ_n of the parameter the value that makes the likelihood as large as possible:

$$\hat{\theta}_n : L(x|\theta_n) \leq L(x|\theta) \quad (4.4)$$

4.5.2 The output error method

General theory of the maximum likelihood estimators will be now applied. The name output-error for this set up comes from the fact that measurement noise is the only noise considered, no disturbance is acting on the plant and the map from input u to output y is deterministic (Figure 4.6).

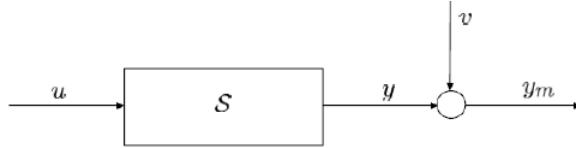


Figure 4.6: Block diagram of output-error method

v is the measurement error, assuming that systematic errors are not present, the expected value of the product of v at different time instants is zero. This means that the measurement error at a certain time i is uncorrelated from the corresponding measurement error at the instant j . v is gaussian, has zero mean and there is no correlation between pairs of samples at different time instants: if at time i the error is small or large, the value is uncorrelated to the one at time j .

The assumption of no disturbances on the plant is quite restrictive as in real experiments there are always some. If the developed theory is applied to some experiments, the requirement must be respected as much as possible.

The next step is to define the error e as the difference between the measured output (y_m) and the identified one (y):

$$e(k, \theta) = y_m(k) - y(k, \theta) \quad (4.5)$$

and to define the corresponding cost function. Now the cost function has to be minimised, an optimisation problem has to be solved, taking care to find the global minimum of the cost function.

4.6 Classification of model identification problems

There are several ways identify the mathematical model, but the most common ones are: black-box and grey-box model identification.

Black-box identification problem is the formulation with a model structure that does not have any physical consideration, it is just defined in an optimal way to extract as many information as possible from the data. Grey-box identification instead is the formulation with a model structure that comes from physics, therefore it is physically motivated. This approach is used for the purpose of the thesis coupled with output error method and maximum likelihood principle. The implementation of this procedure will be seen in the next chapter through the command `greyest` of Matlab.

Some further distinctions can be done:

1. Continuous-time vs discrete-time: the models may be formulated in continuous time, mainly used in grey-box modelling, or in discrete time, mainly used in black-box modelling, to reflect the discrete nature of sampled data.
2. Linear vs nonlinear models: Linear models is used to capture flight dynamics near a given trim condition. Nonlinear models instead are used if the dynamic of the problem involves large angles and/or fast maneuvering flight.

4.7 Preliminary practical issues for model identification

There are some important factors to be taken into account in the setup of the experiment:

1. Choice of sampling interval
2. Prefiltering
3. Stability

4.7.1 Choice of sampling interval

The choice of the sampling interval is a free parameter and needs to be chosen carefully, as this will affect the estimation process considerably. Therefore some guidelines must be followed.

Since a variable that has a certain frequency has to be sampled, the sampling needs to be sufficiently fast to capture its dynamics.

From this point of view the Shannon-Nyquist theorem gives a lower bound about the frequency at which the sample is needed: the sampling has to be sufficiently fast with respect to the relevant dynamics. This means that the sampling interval should be at least 5 to 10 shorter than the faster dynamics of interest.

Another aspect is related to the higher frequency at which the sensor can sample, but choosing an arbitrarily small sampling interval is not necessarily a good idea: it may increase the number of data-points to be processed with little or no gain in terms of information about the parameters.

Given the expressions of the cost function this means that the larger the data-set, the longer it will take to obtain the optimal solution.

4.7.2 Prefiltering

Prefiltering appears also in the nonlinear case, but, for sake of simplicity, it will be discussed in the case of a LTI system. It was assumed that the true system belongs to the model class, the dynamics of the problem was approximated to be totally described by the parameters in the vector θ . This assumption is hardly never true in real life: if the data-set contains contributions from modes or dynamics not included in the model class the approximation will not be as good as expected.

The best thing is to setup the experiment in a such a way that the system is excited only by frequencies of interest. Unfortunately this is not possible in real life: post processing of the data is needed. This method is called prefiltering.

The only annoyance associated with this idea is that the error filtering is needed at each iteration of the optimisation problem for the cost function. Therefore, it represents a significant increase in the cost of the procedure. However, for linear

systems it is possible to filter data before the optimisation process. To better understand data prefiltering, consider the block diagram in Figure 4.7.

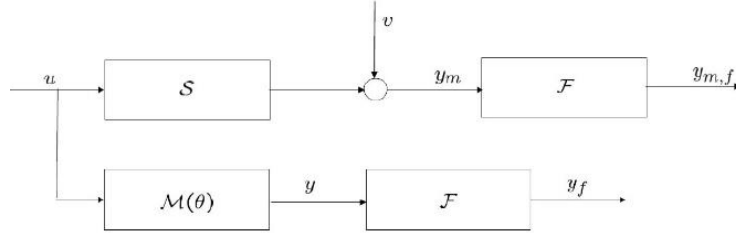


Figure 4.7: error filtering

The system S is tested using the input u , then the output is measured (y_m), containing the measurement error v , and finally it is processed with the filter F to obtain the measured filtered output ($y_{m,f}$). In the lower branch the same input is applied to the model class and the simulation of the model is generated. Then the simulation is filtered to obtain a filtered version (y_f). The filtered error (e_f) can be computed as:

$$e_f(k; \theta) = L(z)e(k; \theta) = L(z)y_m(k) - L(z)y(k; \theta) = y_{m,f}(k) - y_f(k; \theta) \quad (4.6)$$

$L(z)$ is a linear digital filter design to attenuate the contribution of undesired dynamics. Since all the blocks are linear, the order of the blocks can be changed in order to obtain a different solution for the branch of the model class (Figure 4.8):

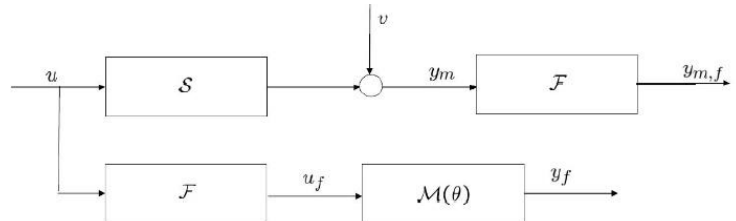


Figure 4.8: Prefiltering

Instead of filtering the output, the input of the model is filtered.

Thanks to the fact that the input is filtered, filtering at each iteration is not needed. This is what is called data prefiltering and it is possible only in the linear case. Using data prefiltering in nonlinear problem can lead to unpredictable results.

4.7.3 Stability

In the output-error method, at each iteration of the optimisation algorithm, the estimate of θ and the new cost function are evaluated. Assuming that the true system is asymptotically stable there is no guarantee that at each iteration the estimation will get closer to the true value. If a model that diverges, the cost will be very different at each iteration and the algorithm for the minimisation will fail. There are several ways to not let this happen:

1. Use a constrained optimisation algorithm to enforce bounds on the parameters. This is not easy because it depends on how the parameters relate to the eigenvalue of the dynamics.
2. Check stability at each iteration and modify the current estimate to lead to a stable system. This issue in nonlinear problems is a lot harder to handle.

4.8 Validation of the model

Once the identified model is found, it must be validated. This means that some parameters to evaluate how much the model behaves as the real system, are needed. The most useful validation parameters are the following:

$$VAF = \max\left(1 - \frac{\text{Var}(y_{\text{model}} - y_{\text{estimate}})}{\text{Var}(y)}, 0\right)100 \quad (4.7)$$

$$FIT = \max\left(1 - \frac{\|y_{\text{model}} - y_{\text{estimate}}\|^2}{\|y_{\text{model}} - E(y_{\text{model}})\|^2}, 0\right)100 \quad (4.8)$$

$$PEC = \frac{1}{\sqrt{N}} \|y_{model} - y_{estimate}\|^2 \quad (4.9)$$

VAF is the Variance accounted for, FIT is the percentage of fitting and PEC is the Prediction error cost. When $y_m = y_{estimate}$, VAF and FIT have a value of 100%, instead PEC would be 0, these are the best values possible, but they are reachable only from a theoretical point of view.

Chapter 5

Model identification applied to ANT-X drones

In this chapter grey box model identification is described both in a theoretical and practical point of view. Both closed and open loop model identification are applied for the dataset. The parameters identified are collected and shown in tables and the meaning of the results is deeply analysed.

5.1 Grey-box identification

The identification performed for the purpose of the thesis is the grey-box identification, through the Matlab function: Greyest. Both identification in the time and frequency domain are possible, but in this thesis only the time domain is taken into account. This function uses output error method and maximum likelihood estimation, described in the previous chapter. Define now the prediction error e as the difference between the measured output and the output of the model:

$$e(k, \theta) = y_m(k) - y_{model}(k, \theta) \quad (5.1)$$

maximize the likelihood function is equal to minimize the cost function:

$$J(\theta) = \frac{1}{N} \sum (e(k, \theta)^2) \quad (5.2)$$

This is the cost function implemented with Greyest. This command is an iterative algorithm that needs some starting values for the estimates. It is important to find the most reliable ones and ensure that the algorithm finds the global minimum of the cost function. The experimental data were collected in closed loop, and the

identification has been performed both in closed loop and in open loop. Closed loop will be analysed first: in this case position and attitude for longitudinal dynamics identification were performed. The drone is supposed to be Linear and Time Invariant (LTI) system with sampling time: 250Hz.

5.2 Closed loop identification for longitudinal dynamics: Position

The ANT-X drones were subjected to a set-point position input, as shown in Figure 3.1. The red line is the input given to the drone and the blue line instead is the real response of the system. The aim of model identification is to create a mathematical model that gives a similar response shown by the blue line. The entire system is approximated by a second order transfer function with position set-point as input and position and acceleration as outputs. The following transfer function is implemented:

$$\frac{y}{y_0} = \mu \frac{\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2} \quad (5.3)$$

with the gain $\mu = 1$. The parameters to estimate are: ω_n (natural frequencies) and ξ (damping). The initial parameters for estimation are: $\omega = 2.8$ and $\xi = 0.8$. Once the parameters are estimated, the response of the model will be compared with the real one seen in Figure 3.1 and the quality of the model will be analysed. u is the longitudinal velocity [m/s], instead x is the position [m] and x_0 the setpoint position. The corresponding state space model is:

$$\dot{x} = \begin{bmatrix} u \\ a_x \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\omega_n^2 & -2\xi\omega_n \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix} + \begin{bmatrix} 0 \\ \omega_n^2 \end{bmatrix} x_0 \quad (5.4)$$

$$y = \begin{bmatrix} x \\ a_x \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ -\omega_n^2 & -2\xi\omega_n \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix} + \begin{bmatrix} 0 \\ \omega_n^2 \end{bmatrix} x_0 \quad (5.5)$$

Recall the concept of prefiltering discussed in the previous section, the output of the real system (so the data collected in the laboratory: position and acceleration in NED frame) and the input of the model (position set-point in NED frame) are

filtered in such a way that the system is excited only by frequencies of interest.

The identification of the unknown parameters is performed for each drone of the fleet and the results are presented in Table 5.1. A more effective way of presenting the data will be done in Figure 5.1, 5.2, 5.3. The validation data are presented with the VAF parameter discussed in the previous chapter. The data of the response of the real system were compared to the data simulated with the identified model.

N	ω_n	ξ	variance ω_n	variance ξ	VAF pos	VAF acc	$3\sigma \omega_n$	$3\sigma \xi$
1	3,3103	0,8105	1,0669E-04	1,0131E-05	99,63	60,42	0,0309	0,0095
2	3,1224	0,7950	1,9535E-04	2,1468E-05	98,86	51,45	0,0419	0,0139
3	3,6604	0,7759	5,3509E-04	3,4565E-05	99,58	59,29	0,0693	0,0176
4	3,3311	0,8655	3,6416E-04	3,5270E-05	99,14	41,24	0,0572	0,0178
5	4,2393	1,0816	5,2083E-04	3,9616E-05	99,60	38,60	0,0685	0,0189
6	3,2659	0,8122	1,2951E-04	1,3316E-05	99,48	64,24	0,0341	0,0109
7	3,4646	0,7894	8,6194E-05	6,6542E-06	99,68	68,06	0,0279	0,0077
8	3,5398	0,9267	3,0774E-04	2,7980E-05	99,54	37,00	0,0526	0,0159
9	3,5272	0,8240	1,2294E-04	9,5965E-06	99,67	61,41	0,0333	0,0093
10	3,3026	0,8114	8,0392E-05	8,0647E-06	99,42	67,76	0,0269	0,0085
11	3,2867	0,7938	4,9400E-05	5,3831E-06	99,46	73,59	0,0211	0,0070
12	3,7659	0,8681	4,0345E-04	2,7788E-05	99,64	51,37	0,0603	0,0158
13	3,5118	0,8540	1,1644E-04	1,0149E-05	99,52	66,34	0,0324	0,0096
14	3,4077	0,7766	1,7796E-04	1,4722E-05	99,12	65,82	0,0400	0,0115
15	3,2462	0,8133	4,6347E-05	5,6791E-06	99,42	74,21	0,0204	0,0071
MEAN	3,4655	0,8399	2,1617E-04	1,8025E-05	99,45	58,72	0,0411	0,0121
MEDIAN	3,4078	0,8121	1,2951E-04	1,3315E-05	99,52	61,41	0,0341	0,0109

Table 5.1: Results from identification position in closed loop

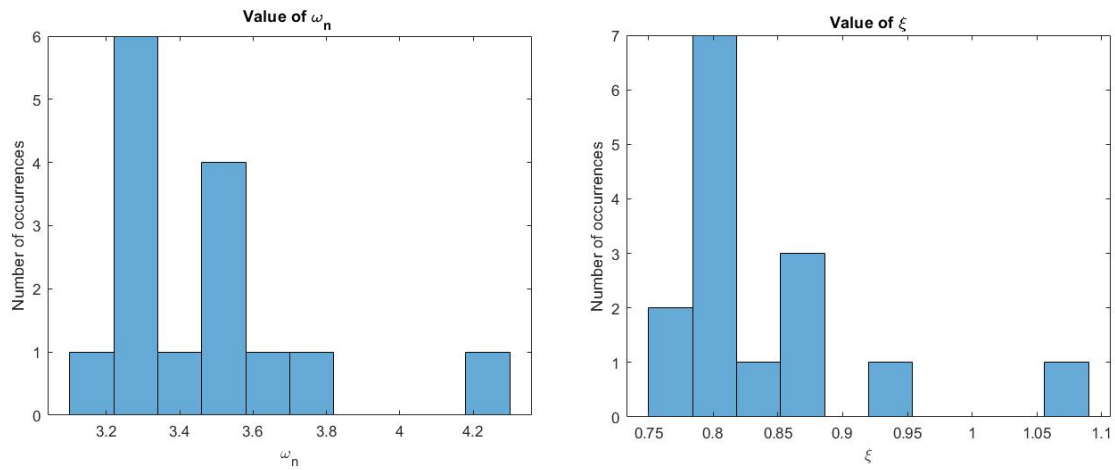


Figure 5.1: Results from identification position in closed loop: estimated values

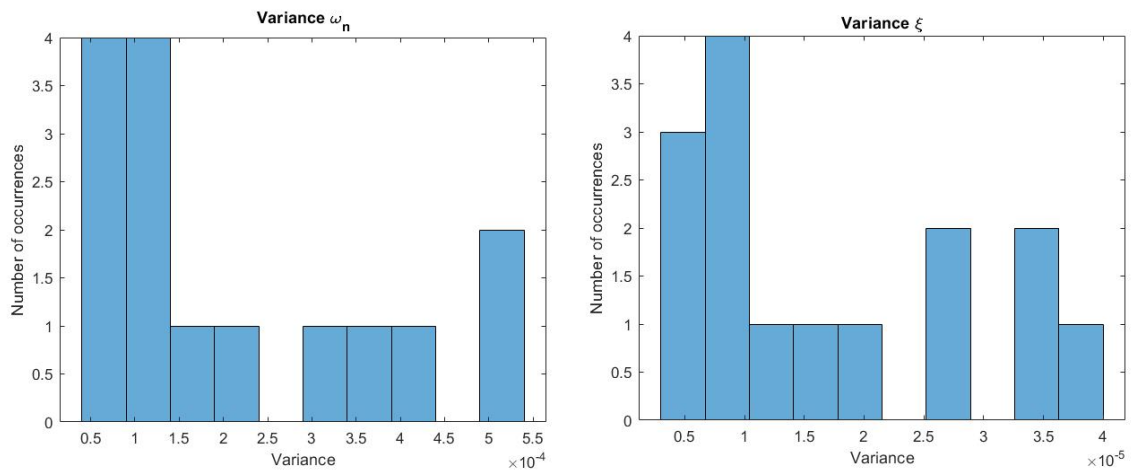


Figure 5.2: Results from identification position in closed loop: variance of the estimate

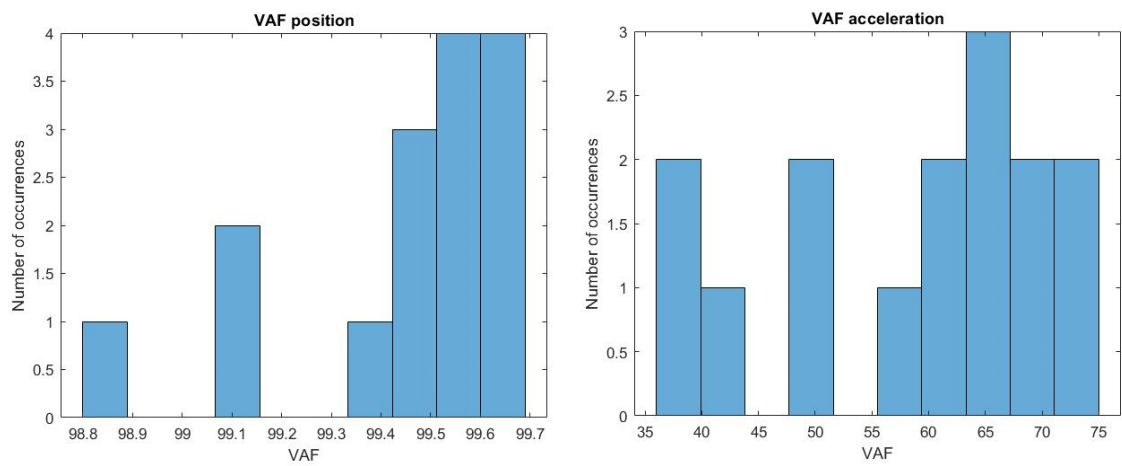


Figure 5.3: Results from identification position in closed loop: VAF

The accuracy of the position estimation is really high as is shown by Figure 5.1, instead the acceleration is much less accurate, with a pretty low VAF. In Figure 5.2 is highlighted the low variances of the estimates.

Before drawing some conclusions about the presented data, it is important to take a look at how the mathematical model reproduce the response of the drone to a certain input. The first drone tested of the entire fleet will be taken into account and it will be subjected to the same step input in position, that was used to excite the real system in the laboratory.

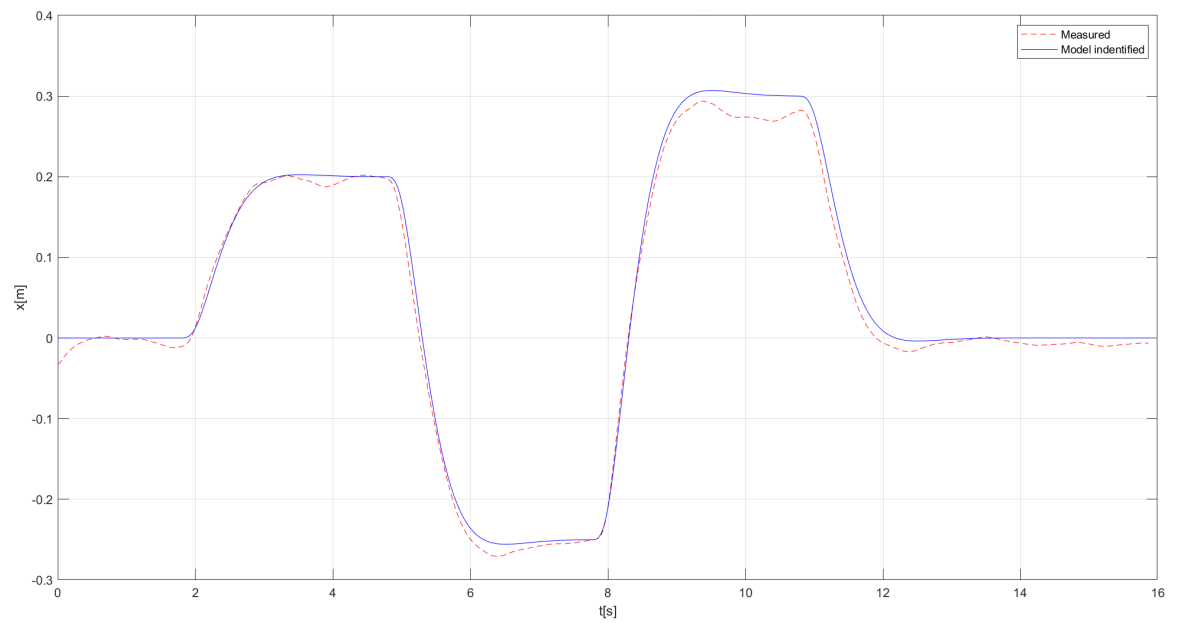


Figure 5.4: Results from identification position for the first drone tested: Position

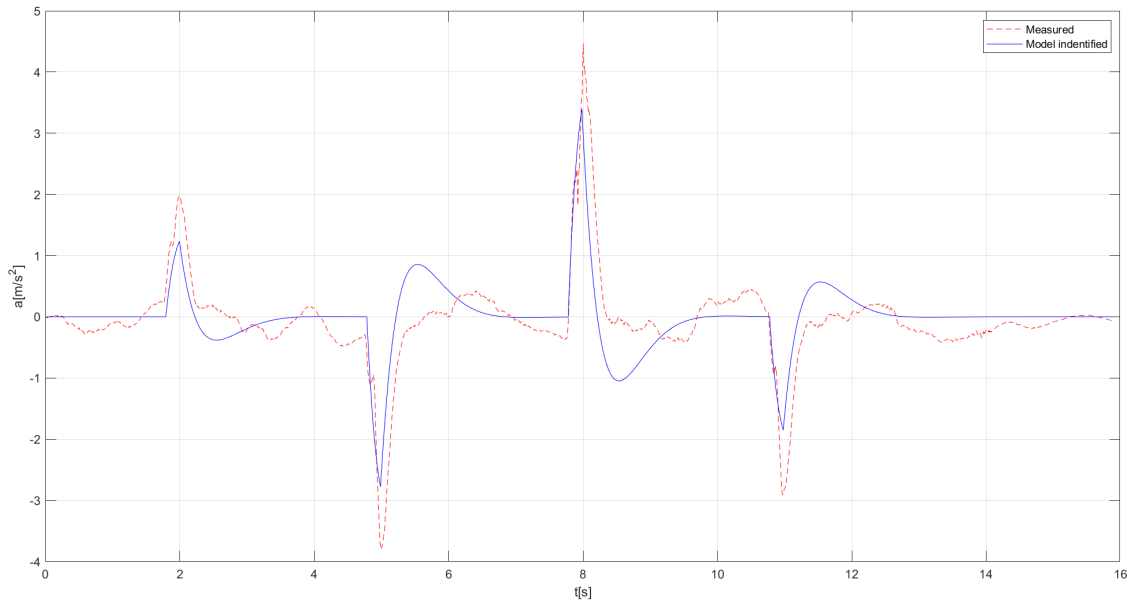


Figure 5.5: Results from identification position for the first drone tested: Acceleration

The histograms show that all the drones have a very similar behaviour. This means that the drone presented in the last two figures (Figure 5.4 and 5.5) can be taken as a good example of behaviour of the fleet. The Figure 5.5 shows that the response of the identified model does not reproduce in an accurate way the real response of the system.

The position approximation has a better fitting to the real data compared with the acceleration one. The main reason is that the latter is a second derivative of the position, so the approximation of the behaviour is more complex.

The reason why the identified model differ from one drone to the other is mainly given by the different lubrication of the bearings and calibration of the bars used to run the experiments.

5.3 Closed loop identification for longitudinal dynamics: Attitude

The ANT-X drones were subjected to a set-pitch angle position input, as shown in Figure 3.11. The red line is the input given to the drone and the blue line instead is the real response of the system. The following state space model is implemented:

$$\dot{x} = \begin{bmatrix} \theta \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\omega_n^2 & -2\xi\omega_n \end{bmatrix} \begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ \omega_n^2 \end{bmatrix} \theta_0 \quad (5.6)$$

$$y = \begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix} \quad (5.7)$$

Where θ is the pitch angle and θ_0 the setpoint position angle. The following data are obtained with identification, which was performed in the same way of position identification, the only difference now is that now the test done in laboratory was with 1DoF. The initial parameters for identification are: $\omega = 10$ and $\xi = 0.5$:

N	ω_n	ξ	variance ω_n	variance ξ	VAF θ	VAF $\dot{\theta}$	$3\sigma \omega_n$	$3\sigma \xi$
1	15,9712	0,5636	9,3363E-04	2,9683E-06	99,86	96,86	0,0917	0,0052
2	16,1709	0,5655	9,7379E-04	3,0840E-06	99,85	97,14	0,0937	0,0053
3	16,0971	0,5640	8,8610E-04	2,7936E-06	99,86	97,16	0,0893	0,0050
4	16,1611	0,5612	8,82710E-04	2,7774E-06	99,86	97,27	0,0891	0,0050
5	16,5547	0,5731	8,9318E-04	2,6729E-06	99,88	97,49	0,0897	0,0049
6	16,0264	0,5535	1,0043E-03	3,2672E-06	99,82	96,93	0,0951	0,0054
7	15,9807	0,5557	8,0436E-04	2,5816E-06	99,87	97,30	0,0851	0,0048
8	16,3708	0,5541	7,3686E-04	2,2420E-06	99,89	97,66	0,0814	0,0045
9	16,2787	0,5598	7,5203E-04	2,3114E-06	99,89	97,60	0,0823	0,0046
10	16,4102	0,5629	7,4439E-04	2,2552E-06	99,89	97,74	0,0819	0,0045
11	16,1706	0,5505	7,9394E-04	2,4995E-06	99,87	97,49	0,0845	0,0047
12	16,0658	0,5587	9,9415E-04	3,2026E-06	99,84	97,00	0,0946	0,0054
13	15,8809	0,5575	8,6971E-04	2,7955E-06	99,86	96,96	0,0885	0,0050
14	16,6049	0,5658	9,5544E-04	3,0165E-06	99,83	97,78	0,0927	0,0052
15	15,9781	0,5543	8,6837E-04	2,8116E-06	99,85	97,24	0,0884	0,0050
MEDIAN	16,1814	0,5601	9E-04	2,7519E-06	99,86	97,31	0,0885	0,0050
MEAN	16,1611	0,5598	9E-04	2,7936E-06	99,86	97,27	0,0891	0,0050

Table 5.2: Results from identification attitude in closed loop

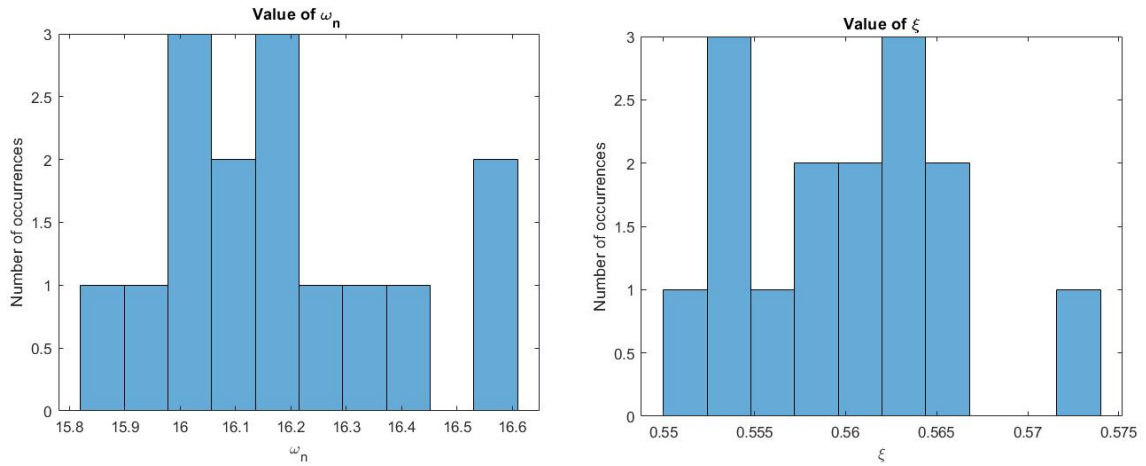


Figure 5.6: Results from identification attitude in closed loop: estimated values

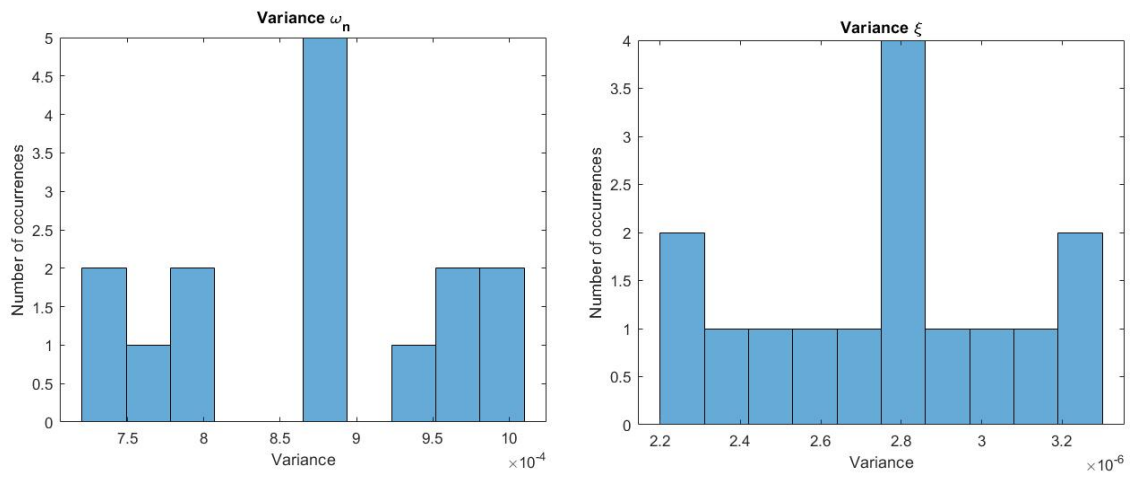


Figure 5.7: Results from identification attitude in closed loop: variance of the estimate

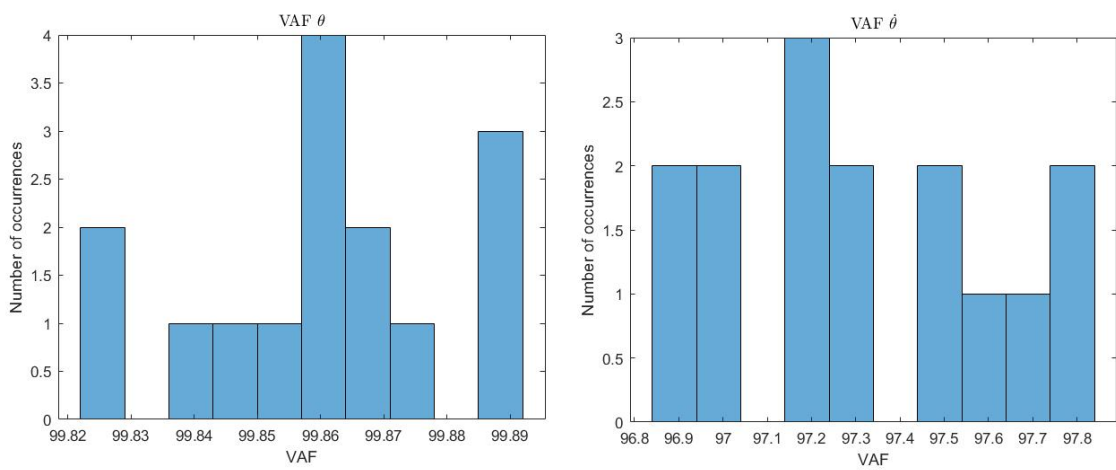


Figure 5.8: Results from identification attitude in closed loop: VAF

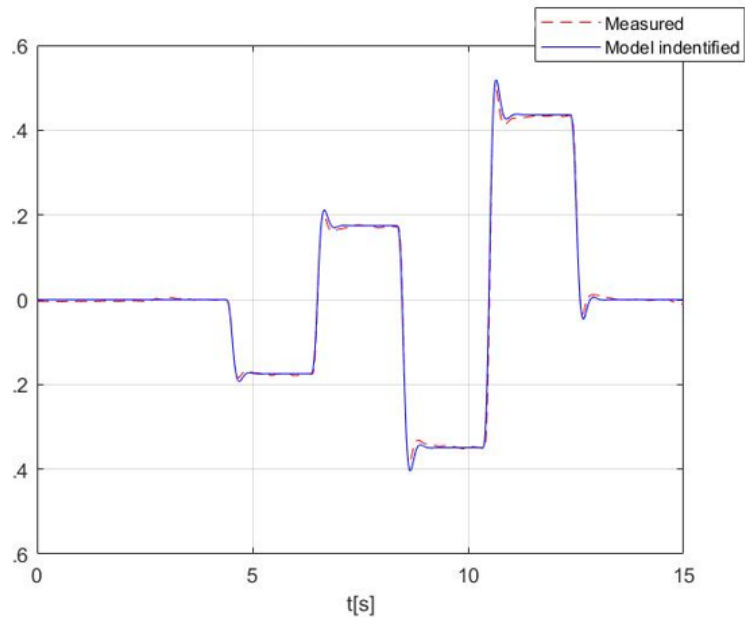


Figure 5.9: Results from identification angular position for the first drone tested: Attitude

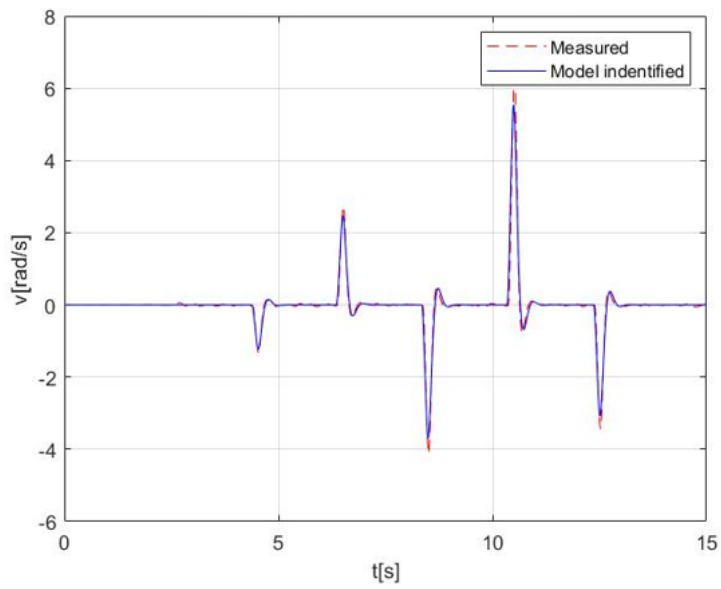


Figure 5.10: Results from identification angular velocity for the first drone tested: Attitude

Comparing the data obtained from Figures 5.4, 5.5 and Figures 5.9, 5.10, it is obvious that the approximation performed in Attitude gives a better approximation of the real system. This happens because in the attitude test, there were only one degree of freedom and the drone could have only a rotation of some angle, sliding was not possible: friction was lower and the data collected more accurate.

The histograms show that all the drones of the fleet have a very similar behaviour and that the model identified, as before, are similar one from the other.

5.4 Open loop identification for longitudinal dynamics

As was discussed before, the results obtained from the previous analysis are similar for all the drones considered, for this reason open loop model identification is studied. Data were collected in closed loop in the laboratory, so when the identification of the open loop system will be performed, the dynamics of the controllers will be taken into account for the validation part of the process. The data used for this study are the one from the 2 Dof position test, so position will be approximated. For this identification, the dynamics of the drone is required (Chapter 2.3).

The equation for the model that has to be identified:

$$\begin{bmatrix} \dot{u} \\ \dot{q} \\ \dot{\theta} \\ \dot{x} \end{bmatrix} = \begin{bmatrix} X_u & X_q & -g & 0 \\ M_u & M_q & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} u \\ q \\ \theta \\ x \end{bmatrix} + \begin{bmatrix} X_\delta \\ M_\delta \\ 0 \\ 0 \end{bmatrix} \delta_{longitudinal} \quad (5.8)$$

$$\begin{bmatrix} u \\ x \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ q \\ \theta \\ x \end{bmatrix} \quad (5.9)$$

After the identification is performed and the unknown parameters

$$X_u, X_q, M_u, M_q, X_\delta, M_\delta \quad (5.10)$$

are identified, the next step is to use the model identified in open loop and to close it, so that the data obtained can be validated through the comparison with the

data obtained in laboratory.

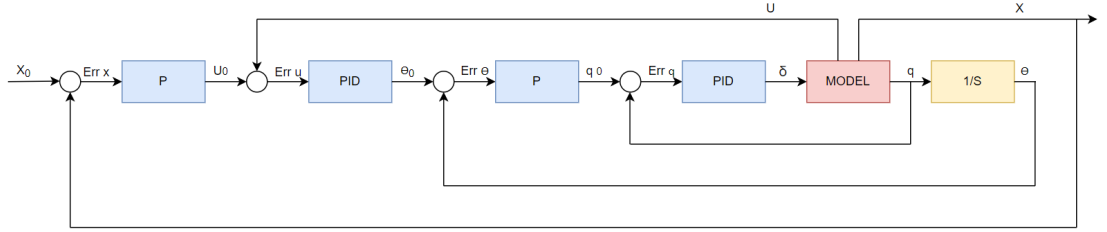


Figure 5.11: Block diagram

The block diagram (Figure 5.11) represents the closed loop system implemented. The model (the red block) is the identified one, P is the Proportional controller and PID is the Proportional, Integral, Derivative one. The inner block is the attitude one, instead the outer is the position one.

The gains used for the controllers are the following:

Position	P	$K_p = 2$		
	PID	$K_p = 0.5$	$K_i = 0.4$	$K_d = 0$
Attitude	P	$K_p = 12$		
	PID	$K_p = 0.9$	$K_i = 0.21$	$K_d = 0.0016$

Table 5.3: Block diagram of closed loop model

The transfer function of the controller implemented are the following:

$$f_p = K_p \quad (5.11)$$

$$f_{pid} = K_p + \frac{K_i}{s} + \frac{K_d s}{T_s s + 1} \quad (5.12)$$

Where T_s is the Time sampling: $T_s = 0.004s$

The data collected from the identification and validation are the following:

N	Xu	Xq	Mu	Mq	Xd	Md	VAF x
1	-0,00354	-18,9084	1,037E-07	-6,6228	98,7534	4,7850	95,94
2	-0,02476	0,7101	2,893E-09	-7,5262	97,7980	5,5599	95,74
3	-0,02420	-1,6767	3,916E-10	-4,8001	99,7335	4,1036	96,38
4	-0,04235	2,1828	4,409E-06	-8,4703	98,1318	5,9631	95,62
5	-0,03692	1,4420	7,246E-03	-13,6122	97,2853	10,0917	96,70
6	-0,04667	6,3011	5,625E-04	-12,4027	96,3890	8,7747	96,22
7	-0,02610	-10,329	4,677E-04	-11,3455	86,8309	6,9253	95,89
8	-0,02746	-1,1612	2,321E-05	-10,4661	97,6953	7,8523	95,82
9	0,00007	-2,5369	4,982E-04	-5,85315	98,3472	5,6470	96,22
10	-0,00086	3,1884	6,394E-03	-26,4081	89,1074	21,3173	96,19
11	-0,05836	0,6971	5,123E-08	-6,1304	98,8318	4,6231	96,38
12	-0,04582	1.0000	4,273E-09	-7,1812	98,9792	5,0558	96,27
13	-0,04043	-0,4418	1,213E-10	-5,2889	98,9168	5,1516	96,03
14	0,00107	-5,8023	1,860E-03	-7,9183	98,2862	5,3322	95,56
15	-0,20491	85,4117	1,334E-03	-0,9503	101,0434	0,1461	96,12
MEAN	-0,03874	4,0051	1,226E-03	-8,9984	97,0753	6,7552	96,07
MEDIAN	-0,02746	0,6971	2,321E-05	-7,5262	98,2861	5,5599	96,12

Table 5.4: Results from identification position in open loop

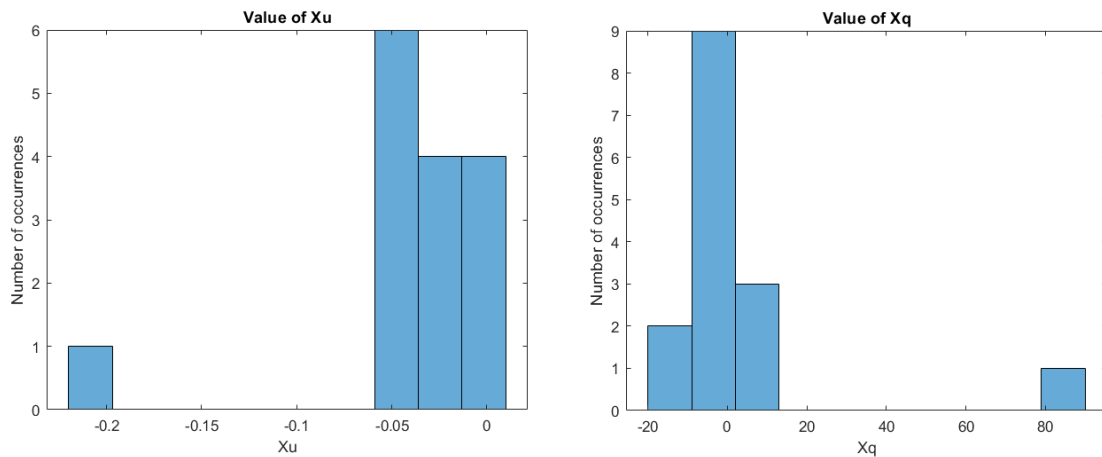


Figure 5.12: Results from identification position in open loop: Xu and Xq

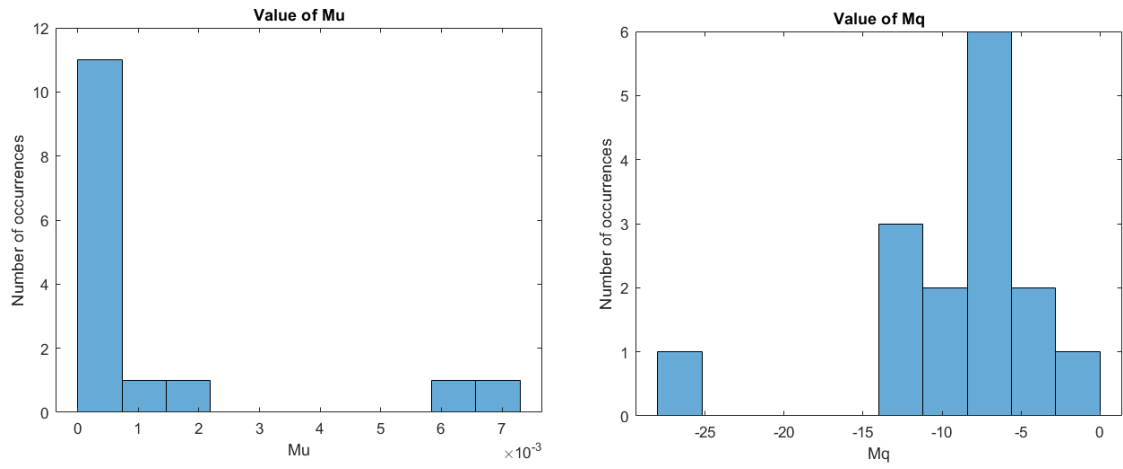


Figure 5.13: Results from identification position in open loop: Mu and Mq

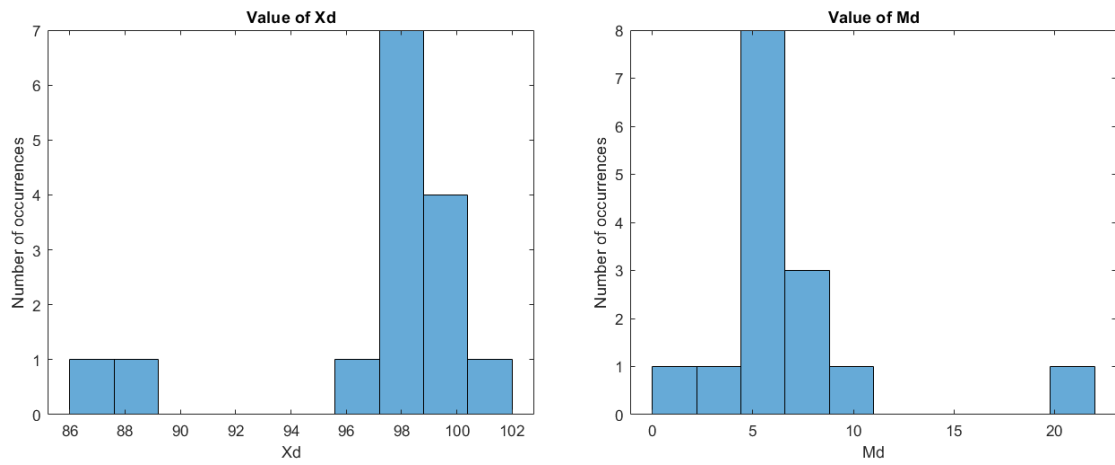


Figure 5.14: Results from identification position in open loop: Xd and Md

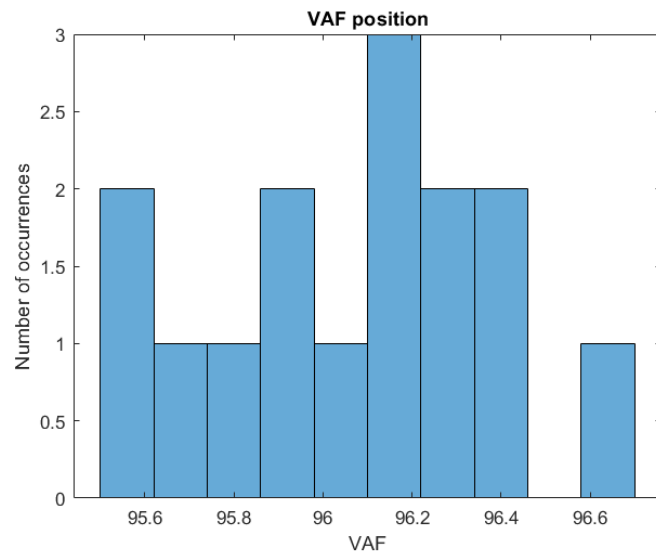


Figure 5.15: Results from identification position in open loop: VAF

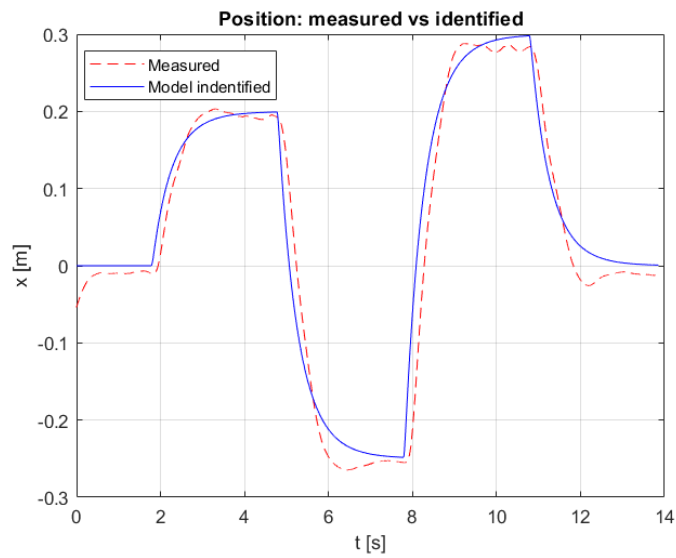


Figure 5.16: Results from identification position in open loop: position (Results from the last drone of the fleet)

The initial guess for the parameters are the following: $Xu = -0.1$, $Xq = 20$, $Mu = 0.1$, $Mq = -110$, $Xd = 100$, $Md = 10$.

Comparing the results obtained from the identification performed in closed loop, the identified variables obtained in open loop analysis differ much more one to another.

Chapter 6

Conclusions

The drone fleet was subjected to three types of tests (hovering, attitude steps, position steps) and their responses were collected. Through the developed software Log Analyser, a qualitative comparison was performed: all the drones had a really similar behaviour, as expected.

Then the problem of system identification has been considered. The fleet was subjected to closed loop identification, where position, acceleration, pitch angle and pitch rate were estimated. All the identified parameters had similar values from one drone to the other. The goal of the identification in closed loop is to identify the entire controlled system. Given that the drones have a similar behaviour, it is reasonable that the values obtained after the identification, are similar one to the other.

Finally the identification in open loop was performed. The aim was to identify the system in open loop through the UAV dynamics. The results obtained were really different one from the other. This result underlines that, even if the drones were built in the same way, subjected to the same inputs and in the same environment, they worked in different ways, with different variables in their dynamic.

It is obvious that the controllers (P and PID) have an important role in the development of a drone and for its correct behaviour. Notice that the gains of the controllers implemented in the fifteen drones are the same, even if the identified variables differ.

Bibliography

- [1] https://docs.px4.io/v1.9.0/en/log/flight_review.html.
- [2] <https://it.mathworks.com/help/uav/ref/flightlogalyzer-app.html>.
- [3] Emad Ebeid, Martin Skriver, Kristian Husum Terkildsen, Kjeld Jensen, and Ulrik Pagh Schultz. A survey of open-source uav flight controllers and flight simulators. *Microprocessors and Microsystems*, 61:11–20, 2018.
- [4] Torsten P Bohlin. *Practical grey-box process identification: theory and applications*. Springer Science & Business Media, 2006.
- [5] Lennart Ljung. Perspectives on system identification. *Annual Reviews in Control*, 34(1):1–12, 2010.
- [6] L Nugroho and R Akmeliawati. Comparison of black-grey-white box approach in system identification of a flight vehicle. In *Journal of Physics: Conference Series*, volume 1130, page 012024. IOP Publishing, 2018.
- [7] Qianying Li. Grey-box system identification of a quadrotor unmanned aerial vehicle. 2014.
- [8] Jiri Rehor and Vladimir Havlena. Maximum likelihood estimation of lti continuous-time grey-box models. *IFAC Proceedings Volumes*, 47(3):3739–3744, 2014. 19th IFAC World Congress.
- [9] Ivan Markovsky and Rik Pintelon. Identification of linear time-invariant systems from multiple experiments. *IEEE Transactions on Signal Processing*, 63(13):3549–3554, 2015.
- [10] M Anis Jamil, Mansoor Ahsan, M Jabran Ahsan, and M A Choudhry. Time domain system identification of longitudinal dynamics of a uav: A grey

- box approach. In *2015 International Conference on Emerging Technologies (ICET)*, pages 1–6, 2015.
- [11] Lennart Ljung. *System identification toolbox: User's guide*. Citeseer, 1995.
- [12] Riccardi F.; Panizza P.; Lovera M. Identification of the attitude dynamics for a variable-pitch quadrotor uav. 2014.
- [13] https://ant-x.gitlab.io/documentation_2dofdrone/_chapters/intro.html.
- [14] <https://www.youtube.com/watch?v=ttkqtivyccs>.
- [15] Slx-topx4 demos, v1.1: demo/drone2dof/2dofbaseline.