TÉCNICO LISBOA

Unmanned Aerial Vehicles

MEAer

Laboratory 1 - Modelling and Identification of the Parrot AR.Drone

Group 3:

1. João Peixoto, No. 95807 Signature:

2. José Bento, No. 95815 Signature: Jose Bento

3. Thomas Childs, No. 95847 Signature:

Instructor: Rita Cunha

2022/2023 – First Semester

Contents

1 Introduction and Goals

This laboratory work concerns the modeling and identification of a quadrotor, a Parrot AR.Drone. As such, there are five main goals to achieve:

- Kinematic and dynamic modelling of a quadrotor;
- Linearisation of the quadrotor dynamics around the hover condition;
- Analysis of the inner and outer loops for the height and vertical speed control;
- Identification of the height closed-loop control system;
- Identification of the pitch angle closed-loop control system.

Initially, a theoretical analysis of the model given for the drone will be carried out by studying its transfer functions and the different attitude control strategies used. Next, an experimental analysis: first, by simulating the model for the altitude and pitch control using as a basis the provided MATLAB and SIMULINK files; and finally, by carrying out the same tests as simulated in the laboratory.

2 Modelling

The first section of this laboratory work refers to the modelling of the quadrotor.

Theoretical notes

The two relevant frames of reference for the study of quadrotor and dynamics are {I}, a local inertial frame of reference aligned with the North-East-Down (NED) orientation, and {B}, the body-fixed frame with origin in the vehicle's centre of mass. given these two reference frames, the kinematics can, then, be expressed as

$$\begin{cases}\dot{\mathbf{p}} = R(\boldsymbol{\lambda})\mathbf{v} \\ \dot{\boldsymbol{\lambda}} = Q(\boldsymbol{\lambda})\boldsymbol{\omega}\end{cases} \tag{1}$$

where $\mathbf{p} = [x \quad y \quad z]^T \in \mathbb{R}^3$ denotes the position of the quadrotor {B} with respect to {I}; $\mathbf{v} = [u \quad v \quad w]^T \in \mathbb{R}^3$ is the linear velocity of the vehicle with respect to {I}, expressed in {B}; $\boldsymbol{\lambda} = [\phi \quad \theta \quad \psi]^T$ is the vector of Z-Y-X Euler angles, respectively the roll, pitch and yaw angles, where $\phi, \psi \in \mathbb{R}$ and $\theta \in]-\pi/2, \pi/2[$; and $\boldsymbol{\omega} = [p \quad q \quad r]^T \in \mathbb{R}^3$ is the angular velocity with regards to {B} expressed in {I}.

The matrix $R(\boldsymbol{\lambda})$ is the rotation matrix from {B} to {I}, given by:

$$\mathrm{R}(\boldsymbol{\lambda}) = \begin{bmatrix} \cos\theta\cos\psi & \sin\phi\sin\theta\cos\psi - \cos\theta\sin\psi & \cos\phi\sin\theta\cos\psi + \sin\theta\sin\psi \\ \cos\theta\sin\psi & \sin\phi\sin\theta\sin\psi + \cos\theta\cos\psi & \cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi \\ -\sin\theta & \sin\phi\cos\theta & \cos\phi\cos\theta \end{bmatrix} \tag{2}$$

The matrix $Q(\boldsymbol{\lambda})$ is the angular kinematics Jacobian given by

$$Q(\boldsymbol{\lambda}) = \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \frac{\sin\phi}{\cos\theta} & \frac{\cos\phi}{\cos\theta} \end{bmatrix} \tag{3}$$

As for the quadrotor dynamics, they can be described by the following system of differential equations:

$$\begin{cases} m\dot{\mathbf{v}} = -mS(\boldsymbol{\omega})\mathbf{v} + \mathbf{f} \\ J\dot{\boldsymbol{\omega}} = -S(\boldsymbol{\omega})J\boldsymbol{\omega} + \mathbf{n} \end{cases} \tag{4}$$

Note that the operator $S : \mathbb{R}^3 \mapsto \mathbb{R}^{3\times3}$ is such that $S(\boldsymbol{a}) = \boldsymbol{a} \times \boldsymbol{b}$; $m \in \mathbb{R}^+$ is the mass and $J \in \mathbb{R}^{3\times3}$ is the inertia tensor matrix given by:

$$J = \begin{bmatrix} J_{xx} & J_{xy} & J_{xz} \\ J_{xy} & J_{yy} & J_{yz} \\ J_{xz} & J_{yz} & J_{zz} \end{bmatrix} \tag{5}$$

$$\mathbf{f} = [f_x \quad f_y \quad f_z]^T \in \mathbb{R}^3 \quad \text{and} \quad \mathbf{n} = [n_x \quad n_y \quad n_z]^T \in \mathbb{R}^3$$

are the external forces and torques expressed in $\{B\}$, respectively.

2.1 Forces and moments

The Parrot AR.Drone is an example of the classic quadrotor design, presenting four rotors mounted symmetrically along two orthogonal axes, as depicted in figure 1, where the individual thrusts are denoted by $f_i \in \mathbb{R}, i \in \{1, \ldots, 4\}$, and the corresponding individual torques are given by $n_i = cf_i \in \mathbb{R}, i \in \{1, \ldots, 4\}$, where $c \in \mathbb{R}$ is a scalar constant. Two diametrally opposed pairs of rotors are counter-rotating, as shown; this is the only way to ensure the quadrotor can hover horizontally while maintaining a constant yaw.

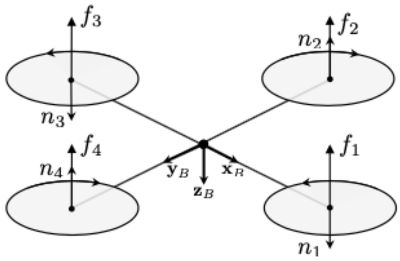


Figure 1: Model of the forces and moments that act on the quadrotor. Note that the reference frame depicted is $\{B\}$. [1]

From this representation, we may attempt to obtain the resultant forces and moments acting upon the drone, $\mho$ and $\ltimes$. If we ignore the aerodynamic resistance acting upon the drone (which can be a valid aproximation for a stationary hover), then the total force acting upon the quadcopter is given by the sum of the thrust produced by each rotor with the resulting gravitic force. Note that, using the reference frame $\{B\}$, each thrust component (and thus the total thrust, $\mathbf{f}_{thrust}$, as well) have a negative value: $\mathbf{f}_{thrust} = N\mathbf{f}_T$, with $\mathbf{f}_T := \begin{bmatrix} f_1 & f_2 & f_3 & f_4 \end{bmatrix}^T \in \mathbb{R}^4$ and the matrix N given by;

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

To calculate the gravitic force acting on the quadrotor body, also in the body reference $\{B\}$, we must transform the gravitic force expressed in $\{I\}$, for which we can resort to the inverse transformation matrix given in 2, $R(\boldsymbol{\lambda})^{-1} = R(\boldsymbol{\lambda})^T$ (given this is a rotation matrix), as described below:

$${}^{B}\mathbf{f}_{gravitic} = R(\boldsymbol{\lambda})^T . {}^{I}\mathbf{f}_{gravitic},$$

Note that this force has the direction of z_B. We are now able to sum $\mathbf{f}_{gravitic}$ and $\mathbf{f}_{thrust}$:

$$\mathbf{f} = {}^{B}\mathbf{f}_{gravitic} + \mathbf{f}_{thrust} = R(\boldsymbol{\lambda})^T \mathbf{f}_{gravitic} + N\mathbf{f}_T = M\mathbf{f}_{gravitic} + N\mathbf{f}_T, \quad M = R(\lambda)^T \tag{6}$$

To calculate the resulting moments, considering a hover flight, the moment along the z_B axis must be null, a condition which is assured by the counter-rotation of the motors 1 and 3, described above. The magnitude of the moment created by each propulsive force, in relation to the centre of mass of the drone, is given by bf_i (with b the distance of the centre of thrust of each rotor to the centre of mass of the quadrotor) and, according to the right-hand reference frame. The resulting moment due to the rotation of each rotor is given by cf_i and has direction contrary to the direction of rotation of each rotor. This is better described by the following equation:

$$\mathrm{n} = P\mathrm{f}_T \quad \text{with} \quad P = \begin{bmatrix} 0 & b & 0 & -b \\ b & 0 & -b & 0 \\ c & -c & c & -c \end{bmatrix} \tag{7}$$

2.2 Alternative body frame

As indicated in the experimental guide [1], the orientation of the body frame, which will be designated $\{B'\}$, of the Parrot AR.Drone is different from the one so far considered (and represented in figure 1): it corresponds to the rotation of this frame by 45° in the horizontal plane. Thus, it is possible to derive the expressions for ${}^{B'}\mathbf{f}$ and ${}^{B'}\mathbf{n}$ by multiplying the previously obtained expressions by the corresponding 3×3 rotation matrix (note that the vertical component of the resulting force and moment is unaffected by this rotation). Thus the following expression follow, considering $R_z(45°)$ given by:

$$R_z\left(45°\right) = \begin{bmatrix} \cos\left(45°\right) & -\sin\left(45°\right) & 0 \\ \sin\left(45°\right) & \cos\left(45°\right) & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} & 0 \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{8}$$

$${}^{B'}\mathbf{f} = R_z\left(45°\right)\mathbf{f} = R_z(45°)M\mathbf{f}_{gravitic} + Nf_T. \tag{9}$$

$$\mathbf{n}' = R_z\left(45°\right)P\mathbf{f}_T \tag{10}$$

2.3 Control input

Using a new control input, as suggested $\mathbf{u} = \begin{bmatrix} T \\ \mathbf{n} \end{bmatrix} \in \mathbb{R}^4$, where the scalar T is given by $T = \sum_{i=1}^{4} f_i$ and corresponds to the total thrust ($\mathbf{f}_{thrust} = [1111]\mathbf{f}_T = -T\mathbf{e}_{z_B}$), and $\mathbf{n} \in \mathbf{R}^3$ is given by 7. The use of this control input is quite useful, as it means that the forces and moments are now being considered as dynamic inputs:

$$\mathbf{u} = L\mathbf{f}_T = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & b & 0 & -b/c \\ b & 0 & -b & 0 \\ c & -c & c & -c \end{bmatrix} \mathbf{f}_T$$

During the lab sessions these dynamic inputs will be translated into variables upon which to actuate (such as the differential thrust produced by each rotor, given by the forces $\mathbf{f}_T$). In this manner, it is even more useful to find the inverse transformation to obtain $\mathbf{f}_T$ given a control input $\mathbf{u}$, which is simply given by:

$$\mathbf{f}_T = L^{-1}\mathbf{u} = \frac{1}{4}\begin{bmatrix} 1 & 0 & 2/b & 1/c \\ 1 & 2/b & 0 & -1/c \\ 1 & 0 & -2/b & 1/c \\ 1 & -2/b & 0 & -1/c \end{bmatrix} \mathbf{u}.$$

2.4 Nonlinear state models

Defining a new position vector, ${}^B\mathbf{p} = R(\boldsymbol{\lambda})^T\mathbf{p}$ (which corresponds to the position vector of $\{B\}$ in the reference frame $\{I\}$, expressed in the reference frame $\{B\}$) and letting $\mathbf{x} = \begin{bmatrix} {}^B\mathbf{p}^T & \mathbf{v}^T & \boldsymbol{\lambda}^T & \boldsymbol{\omega}^T \end{bmatrix}^T \in \mathbb{R}^{12}$ and $\mathbf{x}_1 = \begin{bmatrix} \mathbf{p}^T & \dot{\mathbf{p}}^T & \boldsymbol{\lambda}^T & \boldsymbol{\omega}^T \end{bmatrix}^T \in \mathbb{R}^{12}$ denote two alternative state vectors for the system that describes the quadrotor, it is possible to derive the nonlinear dynamics as $\dot{\mathbf{x}} = \mathbf{g}(\mathbf{x}, \mathbf{u})$ and $\dot{\mathbf{x}}_1 = \mathbf{g}_1\left(\mathbf{x}_1, \mathbf{u}\right)$, respectively.

Remembering the kinematics of a 3D body, expressed in 1, remembering the expression for $Q(\boldsymbol{\lambda})$ (3) and the dynamics equations (4), and assuming the inertia tensor J to be approximately diagonal, one can substitute the known values for $\mathbf{f}$ and $\mathbf{n}$ in 4, obtaining

$$\begin{cases} \dot{\mathbf{v}} = -S(\boldsymbol{\omega})\mathbf{v} + gR(\boldsymbol{\lambda})^T\mathbf{e}_3 - \frac{1}{m}T\mathbf{e}_3 \\ \dot{\omega} = -J^{-1}S(\omega)J\boldsymbol{\omega} + J^{-1}\mathbf{n} \end{cases} \tag{11}$$

Next, one may rewrite the linear dynamics equation (the first equation of the system 1) using this information, obtaining

$$\ddot{\mathbf{p}} = g\mathbf{e}_{z_I} - \frac{1}{m}TR(\boldsymbol{\lambda})\mathbf{e}_{z_I} \tag{12}$$

As one must also define the nonlinear dynamics of the drone for the both alternative states x and x_1, it needs to be found the expression for

$${}^B\dot{\mathbf{p}} = \frac{\partial}{\partial t}(R(\boldsymbol{\lambda})^T\mathbf{p}) = \dot{R(\boldsymbol{\lambda})}^T\mathbf{p} + R(\boldsymbol{\lambda})^T\dot{\mathbf{p}} \tag{13}$$

Using the property $S(.)^T = -S(.)$ and $R(\mathbf{a} \times \mathbf{b}) = (R\mathbf{a}) \times (R\mathbf{b})$, it is known that $\dot{R(\boldsymbol{\lambda})} = R(\boldsymbol{\lambda})S(\boldsymbol{\omega})$ and $S(\boldsymbol{\omega}) = -S(\boldsymbol{\omega})^T$ and thus $\dot{R(\boldsymbol{\lambda})}^T = -S(\boldsymbol{\omega})^TR(\boldsymbol{\lambda})^T$, so:

$${}^B\dot{\mathbf{p}} = -S(\boldsymbol{\omega})^B\mathbf{p} + \mathbf{v} \tag{14}$$

The nonlinear dynamics $g(x,u)$ and $g_1\left(x_1,u\right)$ can then be written as

$$g(\mathbf{x},\mathbf{u}) = \begin{bmatrix} -S(\boldsymbol{\omega})^B\mathbf{p}+\mathbf{v} \\ -S(\boldsymbol{\omega})\mathbf{v}+gR(\boldsymbol{\lambda})^T\mathbf{e}_{z_I}-\frac{1}{m}T\mathbf{e}_{z_I} \\ Q(\boldsymbol{\lambda})\omega \\ -J^{-1}S(\boldsymbol{\omega})J\boldsymbol{\omega}+J^{-1}\mathbf{n} \end{bmatrix}, \qquad g_1\left(\mathbf{x}_1,\mathbf{u}\right) = \begin{bmatrix} \dot{\mathbf{p}} \\ -J^{-1}S(\boldsymbol{\omega})J\boldsymbol{\omega}+J^{-1}\mathbf{n} \end{bmatrix} \tag{15}$$

2.5 State and input vectors at equilibrium point

At equilibrium, the quadrotor is hovering about a given position $\mathbf{p_0}$ with an arbitrary yaw angle ψ_0. Let $\mathbf{x_0}$ and $\mathbf{u_0}$ denote the equilibrium state and input, respectively. $\mathbf{p_0}$ is constant, so the time derivatives will be 0. To determine the equilibrium values for T_0, ϕ_0 and θ_0, we will take the translation dynamics of the quadrotor, and write them as explicit functions of the linear accelerations $\ddot{x}$, $\ddot{y}$ and $\ddot{z}$, decomposing the rotation matrix $R(\boldsymbol{\lambda}) = R_z(\psi)R_y(\theta)R_x(\phi)$. These dynamics then become $mR_z(\psi)^T\left(g\mathbf{e}_{z_I}-\ddot{\mathrm{p}}\right) = TR_y(\theta)R_x(\phi)\mathbf{e}_{z_I}$, which can be written as:

$$\begin{cases} m(-\ddot{x}\cos\psi-\ddot{y}\sin\psi) = T\sin\theta\cos\phi, \\ m(\ddot{x}\sin\psi-\ddot{y}\cos\psi) = -T\sin\phi, \\ m(g-\ddot{z}) = T\cos\theta\cos\phi. \end{cases} \tag{16}$$

From these, solving for T, ϕ and θ yields equations 17 through 19:

$$\tan\theta = \frac{\cos\psi\ddot{x}+\sin\psi\ddot{y}}{\ddot{z}-g} \tag{17}$$

$$\tan\phi = \frac{\sin\psi\ddot{x}-\cos\psi\ddot{y}}{\ddot{z}-g}\cos\theta \tag{18}$$

$$T = -\frac{m(\ddot{z}-g)}{\cos\theta\cos\phi} \tag{19}$$

Imposing $\ddot{x}=\ddot{y}=\ddot{z}=0$ gives $\theta_0=\psi_0=0$, thus $\boldsymbol{\lambda_0} = [0 \quad 0 \quad \psi_0]^T$, also constant. As for ${}^B\mathbf{p_0}$, applying the rotation matrix from {I} to {B} with $\boldsymbol{\lambda_0}$ results in a rotation about a single axis ${}^B\mathbf{p_0} = R_z\left(\psi_0\right)\mathbf{p}_0$. Both $\mathbf{v_0}$ and $\boldsymbol{\omega}_0$ are 3×1 null matrices. The state vector is, then:

$$\mathbf{x_0} = \begin{bmatrix} {}^B\mathbf{p}^T & \mathbf{v}^T & \boldsymbol{\lambda}^T & \boldsymbol{\omega}^T \end{bmatrix}^T = \begin{bmatrix} \left(R_z\left(\psi_0\right)\mathbf{p_0}\right)^T & 0_{3\times1}^T & \begin{bmatrix} 0 & 0 & \psi_0 \end{bmatrix}^T & 0_{3\times1}^T \end{bmatrix}^T \tag{20}$$

Noting that $\mathbf{p_0}$ is constant and $\theta_0=\psi_0=0$ results in $T_0=mg$, while $\mathbf{n_0}=0_{3\times1}$. The input vector is, thus:

$$\mathbf{u_0} = \begin{bmatrix} T_0 & \mathbf{n}_0^T \end{bmatrix}^T = \begin{bmatrix} mg & 0_{3\times1}^T \end{bmatrix}^T \tag{21}$$

The expressions for $\mathbf{x_0}$ and $\mathbf{u_0}$ are, essentially, showing that, in equilibrium, the system state is given by a constant position ${}^B\mathbf{p}$, with null linear and angular velocities and a constant attitude (in particular, no roll, no pitch and a constant yaw ϕ_0). The control input, in equilibrium, will also be constant, given by requiring from the actuators a thrust equal to the quadrotor's weight and a null moment $\mathbf{n_0}$.

2.6 Linearised model of the system

The incremental (or error, as referred to in the laboratory guide) variables considered are, respectively,

$$\begin{aligned} \tilde{\mathbf{x}} &= \begin{bmatrix} \tilde{\mathbf{p}}^T & \tilde{\mathbf{v}}^T & \tilde{\boldsymbol{\lambda}}^T & \tilde{\boldsymbol{\omega}}^T \end{bmatrix}^T \\ \tilde{\mathbf{u}} &= \begin{bmatrix} \tilde{\mathbf{T}} & \tilde{\mathbf{n}}^T \end{bmatrix}^T \end{aligned} \tag{22}$$

where

$$\begin{cases} \tilde{\mathbf{p}} = {}^{B}\mathbf{p} - R^T\mathbf{p}_0 = R^T\left(\mathbf{p}-\mathbf{p}_0\right) \\ \tilde{\mathbf{v}} = \mathbf{v} - \mathbf{v}_0 \\ \tilde{\boldsymbol{\lambda}} = \boldsymbol{\lambda} - \boldsymbol{\lambda}_0 \\ \tilde{\boldsymbol{\omega}} = \boldsymbol{\omega} - \boldsymbol{\omega}_0 \\ \tilde{T} = T - T_0 \\ \tilde{\mathbf{n}} = \mathbf{n} - \mathbf{n}_0 \end{cases} \tag{23}$$

Linearising:

$$\begin{gathered} \dot{\tilde{\mathbf{p}}} = \frac{\partial}{\partial t}\left(R(\boldsymbol{\lambda})^T\left(\mathbf{p}-\mathbf{p}_0\right)\right) = R(\dot{\lambda})^T\left(\mathbf{p}-\mathbf{p}_0\right) + R(\boldsymbol{\lambda})^T\dot{\mathbf{p}}, \\ \dot{\tilde{\mathbf{p}}} = -S(\tilde{\omega})\tilde{\mathbf{p}} + \tilde{\mathbf{v}}. \end{gathered} \tag{24}$$

Looking at the formulae presented in 23, it is easy to conclude that the time derivatives of the incremental variables are simply the time derivatives of the state and input variables. The new nonlinear dynamics, $g_2 : \mathbb{R}^{16} \to \mathbb{R}^{12}$, can be determined adapting the previous dynamics:

$$\dot{\tilde{\mathbf{x}}} = g_2(\tilde{\mathbf{x}}, \tilde{\mathbf{u}}) = \begin{bmatrix} -S(\tilde{\boldsymbol{\omega}})\tilde{\mathbf{p}} + \tilde{\mathbf{v}} \\ -S(\tilde{\boldsymbol{\omega}})\tilde{\mathbf{v}} - \frac{1}{m}\tilde{T}\mathbf{e}_z - g\mathbf{e}_z + gR\left(\tilde{\boldsymbol{\lambda}} + \boldsymbol{\lambda}_0\right)^T\mathbf{e}_z \\ Q\left(\tilde{\boldsymbol{\lambda}} + \boldsymbol{\lambda}_0\right)\tilde{\boldsymbol{\omega}} \\ -J^{-1}S(\tilde{\boldsymbol{\omega}})J\tilde{\boldsymbol{\omega}} + J^{-1}\tilde{\mathbf{n}} \end{bmatrix} \tag{25}$$

The Taylor series expansion for a multivariable function states that, ignoring the higher order terms and taking into account that $g_2 = 0$ when evaluated at equilibrium, the dynamics can be expressed as

$$g_2(\tilde{\mathbf{x}}, \tilde{\mathbf{u}}) = \frac{\partial g_2(\tilde{\mathbf{x}}_0, \tilde{\mathbf{u}}_0)}{\partial \tilde{\boldsymbol{x}}}\tilde{\mathbf{x}} + \frac{\partial g_2(\tilde{\mathbf{x}}_0, \tilde{\mathbf{u}}_0)}{\partial \tilde{\boldsymbol{u}}}\tilde{\mathbf{u}} \equiv \mathbf{A}\tilde{\mathbf{x}} + \mathbf{B}\tilde{\mathbf{u}} \tag{26}$$

It is necessary, then, to determine the partial derivatives of g_2 in order to compute the matrices $\mathbf{A}$ and $\mathbf{B}$. Remembering 25, $\frac{\partial\dot{\tilde{\mathbf{p}}}}{\partial\tilde{\mathbf{x}}}$ is immediate, and remembering 3 so is $\frac{\partial\dot{\tilde{\boldsymbol{\lambda}}}}{\partial\tilde{\mathbf{x}}}$:

$$\left.\frac{\partial\dot{\tilde{\mathbf{p}}}}{\partial\tilde{\mathbf{x}}}\right|_{\tilde{\mathbf{x}}_0,\tilde{\mathbf{u}}_0} = \begin{bmatrix} 0_{3\times3} & I_{3\times3} & 0_{3\times3} & 0_{3\times3} \end{bmatrix} \tag{27}$$

$$\left.\frac{\partial\dot{\tilde{\boldsymbol{\lambda}}}}{\partial\tilde{\mathbf{x}}}\right|_{\tilde{\mathbf{x}}_0,\tilde{\mathbf{u}}_0} = \begin{bmatrix} 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & I_{3\times3} \end{bmatrix} \tag{28}$$

The other partial derivatives are a bit more troublesome; however, noting that

$$\begin{gathered} G \equiv \left.\frac{\partial\dot{\tilde{\mathbf{v}}}}{\partial\tilde{\boldsymbol{\lambda}}}\right|_{\tilde{\mathbf{x}}_0,\tilde{\mathbf{u}}_0} = g\frac{\partial}{\partial\tilde{\boldsymbol{\lambda}}}\left(R\left(\tilde{\boldsymbol{\lambda}} + \boldsymbol{\lambda}_0\right)^T\mathbf{e}_3\right) = \\ = g\frac{\partial}{\partial\tilde{\boldsymbol{\lambda}}}\left.\begin{bmatrix} -\sin\tilde{\theta} \\ \sin\tilde{\phi}\cos\tilde{\theta} \\ \cos\tilde{\phi}\cos\tilde{\theta} \end{bmatrix}\right|_{\tilde{\mathbf{x}}_0,\tilde{\mathbf{u}}_0} = g\begin{bmatrix} 0 & -\cos\tilde{\theta}_0 & 0 \\ \cos\tilde{\phi}_0\cos\tilde{\theta}_0 & -\sin\tilde{\phi}_0\sin\tilde{\theta}_0 & 0 \\ -\sin\tilde{\phi}_0\cos\tilde{\theta}_0 & -\cos\tilde{\phi}_0\sin\tilde{\theta}_0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & -g & 0 \\ g & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \end{gathered} \tag{29}$$

it is possible to determine $\frac{\partial\dot{\tilde{\mathbf{v}}}}{\partial\tilde{\mathbf{x}}}$:

$$\left.\frac{\partial\dot{\tilde{\mathbf{v}}}}{\partial\tilde{\mathbf{x}}}\right|_{\tilde{\mathbf{x}}_0,\tilde{\mathbf{u}}_0} = \begin{bmatrix} 0_{3\times3} & 0_{3\times3} & G & 0_{3\times3} \end{bmatrix} \tag{30}$$

Through some mathematical calculations, it is also possible to conclude that

$$\left.\frac{\partial\dot{\tilde{\boldsymbol{\omega}}}}{\partial\tilde{\mathbf{x}}}\right|_{\tilde{\mathbf{x}}_0,\tilde{\mathbf{u}}_0} = \begin{bmatrix} 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \end{bmatrix} \tag{31}$$

Thus, the complete matrix becomes:

$$\mathbf{A} = \begin{bmatrix} 0_{3\times3} & I_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & G & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & I_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \end{bmatrix} \tag{32}$$

Likewise, for the inputs, it can be deduced that

$$\begin{aligned} \left.\frac{\partial \dot{\mathbf{p}}}{\partial \tilde{\mathbf{u}}}\right|_{\tilde{\mathbf{x}_0},\tilde{\mathbf{u}_0}} &= \begin{bmatrix} 0_{3\times1} & 0_{3\times3} \end{bmatrix}, & \left.\frac{\partial \dot{\mathbf{v}}}{\partial \tilde{\mathbf{u}}}\right|_{\tilde{\mathbf{x}_0},\tilde{\mathbf{u}_0}} &= \begin{bmatrix} -\frac{1}{m}\mathbf{e}_3 & 0_{3\times3} \end{bmatrix}, \\ \left.\frac{\partial \dot{\boldsymbol{\lambda}}}{\partial \tilde{\mathbf{u}}}\right|_{\tilde{\mathbf{x}_0},\tilde{\mathbf{u}_0}} &= \begin{bmatrix} 0_{3\times1} & 0_{3\times3} \end{bmatrix}, & \left.\frac{\partial \dot{\boldsymbol{\omega}}}{\partial \tilde{\mathbf{u}}}\right|_{\tilde{\mathbf{x}_0},\tilde{\mathbf{u}_0}} &= \begin{bmatrix} 0_{3\times1} & J^{-1} \end{bmatrix} \end{aligned} \tag{33}$$

which translate into

$$\mathbf{B} = \begin{bmatrix} 0_{3\times1} & 0_{3\times3} \\ -\frac{1}{m}\mathbf{e}_3 & 0_{3\times3} \\ 0_{3\times1} & 0_{3\times3} \\ 0_{3\times1} & J^{-1} \end{bmatrix} \tag{34}$$

2.7 Obtaining the transfer functions

For this section, the individual elements of $\tilde{\mathbf{x}}$ and $\tilde{\mathbf{u}}$ will be considered, such that

$$\begin{aligned} \tilde{\mathbf{x}} &= \begin{bmatrix} \tilde{x} & \tilde{y} & \tilde{z} & \tilde{\phi} & \tilde{\theta} & \tilde{\psi} & \tilde{u} & \tilde{v} & \tilde{w} & \tilde{p} & \tilde{q} & \tilde{r} \end{bmatrix}^T \\ \tilde{\mathbf{u}} &= \begin{bmatrix} \tilde{T} & \tilde{n}_x & \tilde{n}_y & \tilde{n}_z \end{bmatrix}^T \end{aligned} \tag{35}$$

and $\mathbf{X}(\mathbf{s})$, $\mathbf{U}(\mathbf{s})$ their respective Laplace transforms:

$$\begin{gathered} \mathbf{X}(\mathbf{s}) = \begin{bmatrix} X(s) & Y(s) & Z(s) & \Phi(s) & \Theta(s) & \Psi(s) & U(s) & V(s) & W(s) & P(s) & Q(s) & R(s) \end{bmatrix} \\ \mathbf{U}(\mathbf{s}) = \begin{bmatrix} (s) & N_x(s) & N_y(s) & N_z(s) \end{bmatrix} \end{gathered} \tag{36}$$

Applying the Laplace transform to the equations contained in the state model, it is possible to compute the following transfer functions:

$$\begin{aligned} G_\phi(s) &= \frac{\Phi(s)}{N_x(s)} = \frac{\Phi(s)}{P(s)}\frac{P(s)}{N_x(s)} = \frac{1}{J_{xx}}\frac{1}{s^2}, \\ G_\theta(s) &= \frac{\Theta(s)}{N_y(s)} = \frac{\Theta(s)}{Q(s)}\frac{Q(s)}{N_y(s)} = \frac{1}{J_{yy}}\frac{1}{s^2}, \\ G_\psi(s) &= \frac{\Psi(s)}{N_z(s)} = \frac{\Psi(s)}{R(s)}\frac{R(s)}{N_z(s)} = \frac{1}{J_{zz}}\frac{1}{s^2}, \\ G_x(s) &= \frac{X(s)}{N_y(s)} = \frac{X(s)}{U(s)}\frac{U(s)}{\Theta(s)}\frac{\Theta(s)}{N_y(s)} = -\frac{g}{J_{yy}}\frac{1}{s^4}, \\ G_y(s) &= \frac{Y(s)}{N_x(s)} = \frac{Y(s)}{V(s)}\frac{V(s)}{\Phi(s)}\frac{\Phi(s)}{N_x(s)} = \frac{g}{J_{xx}}\frac{1}{s^4}, \\ G_z(s) &= \frac{Z(s)}{T(s)} = \frac{Z(s)}{W(s)}\frac{W(s)}{T(s)} = -\frac{1}{m}\frac{1}{s^2}. \end{aligned} \tag{37}$$

2.8 Physical interpretation of the transfer functions

From the expressions obtained above in 37, the relation between $X(s)$ and $\Theta(s)$ can be determined:

$$\frac{X(s)}{\Theta(s)} = \frac{X(s)}{N_y(s)}\frac{N_y(s)}{\Theta(s)} = \frac{G_x(s)}{G_\theta(s)} = -\frac{g}{s^2} \tag{38}$$

Similarly,

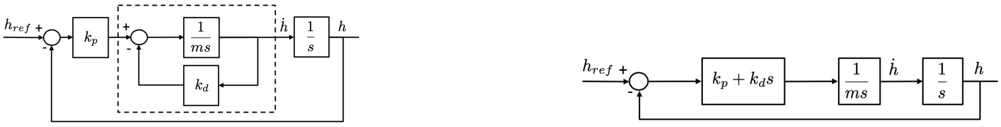
$$\frac{Y(s)}{\Phi(s)} = \frac{Y(s)}{N_x(s)}\frac{N_x(s)}{\Phi(s)} = \frac{G_y(s)}{G_\phi(s)} = \frac{g}{s^2} \tag{39}$$

Some conclusions can be drawn. First, $X(s)$ and $N_y(s)$ are not independent and the dependency can be made explicit by writing their relationships with $\Theta(s)$, and a similar argument can be made for $Y(s)$ and $N_x(s)$. In addition, both $\tilde{x}$ and $\tilde{y}$ are double integrators of θ and ϕ, respectively. For $\tilde{x}$, for movement along the x-axis there must be a non-zero pitch angle in order for a net force to occur. A similar argument about the y-axis can be made for the case of $\tilde{y}$. Finally, both expressions have the expected signs: a negative pitch angle will generate a movement along the positive x-semiaxis, meanwhile movement in the positive y-semiaxis requires a positive roll angle.

An interesting note is that a different value of ϕ_0 should not alter the relationships between these variables, as it was not used when defining $\tilde{\mathbf{x}}$ since it was written in the {I} frame and not {B}, therefore it changing is irrelevant for the system.

2.9 Inner-outer loop and alternative control schemes for height

Letting $h = -z$ denote the height of the quadcopter above the ground and h_{ref} denote the reference height, it is quite straightforward to determine the transfer functions that correspond to the inner-outer loop control scheme and the alternative control scheme, both depicted in figure 2.



((a)) Inner-outer loop height control scheme. ((b)) Alternative height control scheme.

Figure 2: Height control schemes.

Inner-outer loop

The internal feedback loop of the first control system is the one highlighted in the outlined square. Given there is negative feedback at the summation point, the closed loop system for the inner loop is given by:

$$G_{in}(s) = \frac{\frac{1}{ms}}{1 + k_d\frac{1}{ms}} \tag{40}$$

Substituting the highlighted square by this transfer function, it is easy to compute the closed loop transfer function for the global system (note that the feedback is, once again, negative). $H(s)$ and $H_{ref}(s)$ are, respectively, the Laplace transforms of h and h_{ref}.

$$G_{out}(s) = \frac{H(s)}{H_{ref}(s)} = \frac{k_p G_{in}(s)\frac{1}{s}}{1 + k_p G_{in}(s)\frac{1}{s}} = \frac{k_p \frac{\frac{1}{m}}{s+\frac{k_d}{m}}\frac{1}{s}}{1 + k_p\frac{\frac{1}{m}}{s+\frac{k_d}{m}} s\frac{1}{s}} = \frac{\frac{k_p}{m}}{s^2 + \frac{k_d}{m}s + \frac{k_p}{m}} \tag{41}$$

Alternative control scheme

Proceeding in an analogous way, but with the ease of only having one control loop to consider, follows directly:

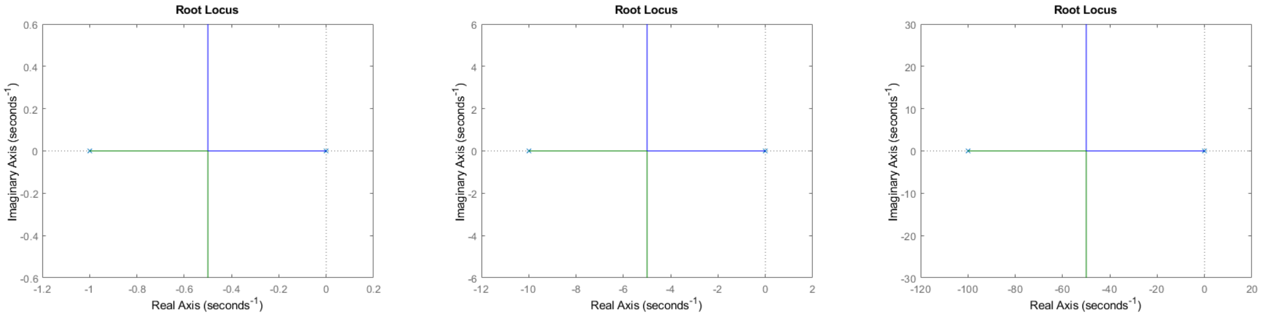
$$\frac{H(s)}{H_{ref}(s)} = \frac{(k_p + k_d s)\,\frac{1}{ms^2}}{1 + (k_p + k_d s)\,\frac{1}{ms^2}} = \frac{\frac{k_d}{m}s + \frac{k_p}{m}}{s^2 + \frac{k_d}{m}s + \frac{k_p}{m}} \tag{42}$$

Comparison of the two height control possibilities

Both control systems implement a closed-loop feedback with the aim of controlling the height tracking error. Both transfer functions (41 and 42) present the same characteristic polynomial, $s^2 + \frac{k_d}{m}s + \frac{k_p}{m}$, which means that both systems have the same pair of poles. The main differences are that, in the inner-outer loop control scheme, $\dot{h}$ is also in a feedback loop, and in the gain k_d is part of the inner loop; in the alternative control scheme, there is a proportional and derivative controller without an inner loop. The alternative control scheme presents a disadvantage: it requires a pole far from the origin for the system to be causal, due to the use of a PD controller. Also, the derivative part of the controller means that the system is more susceptible to high-frequency noise;

however, the fact that this system has a zero (while the inner-outer loop configuration has no zeros) means that the ensuing poles will be more negative, and thus the system response will be likely faster. The fact that the first control system proposed does not include a derivator is an advantage, as it avoids situations where the input derivative may be infinite (such as for a step). In addition, any perturbations in the inner loop will be attenuated in the inner loop, and thus will affect the outer loop minimally. The only disadvantage is that this is a more complex implementation, and requires the acquisition of more sensor data with a high enough data rate. If these sensors are available (as they are for the Parrot AR.Drone), this is not a big issue.

2.10 Root-locus and effect of controller gains



((a)) $k_d/m = 1$ ((b)) $k_d/m = 10$ ((c)) $k_d/m = 100$

Figure 3: Root-locus for different values of $k_d > 0$, considering the drone mass $m = 0.420$kg.

Figure 3 shows the root-locus for 3 different values of the positive but arbitrary gain k_d in the control scheme depicted in figure 2(b).

The poles of the system are located in the left complex semiplane for $k_p > 0$, so the system is always stable. The system becomes faster as k_p increases, since the pole closer to the origin moves away from it, however, for large enough values of k_p, the poles, which were located along the real axis, become complex conjugates. Hence, oscillations appear.

As for k_d, a given increase in the k_d/m quotient translates into the same increase in the absolute value of the pole farthest from the origin, as well as in the gains (i.e. the value for which the two poles "meet", $-k_d/2m$, and become complex conjugates is increased by the same factor). As such, an increase in k_d/m makes the system faster.

3 Identification of the closed-loop height dynamics

3.1 The Simulink model

The `ARDroneHoverHeight.slx` model was modified, since this is the one used for the control of the drone itself. Modifications were also required for the simulink dedicated to the simulation, but these were relatively similar to the ones presented in figure 5.

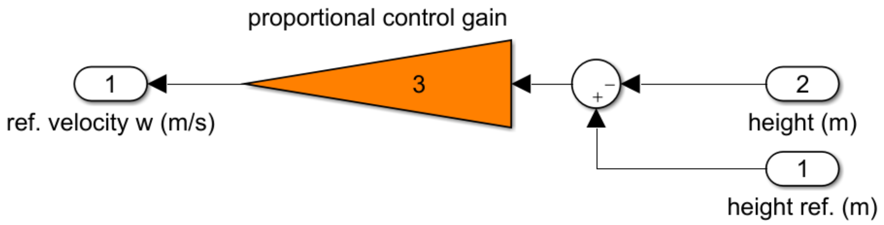


Figure 4: Gain block k_p

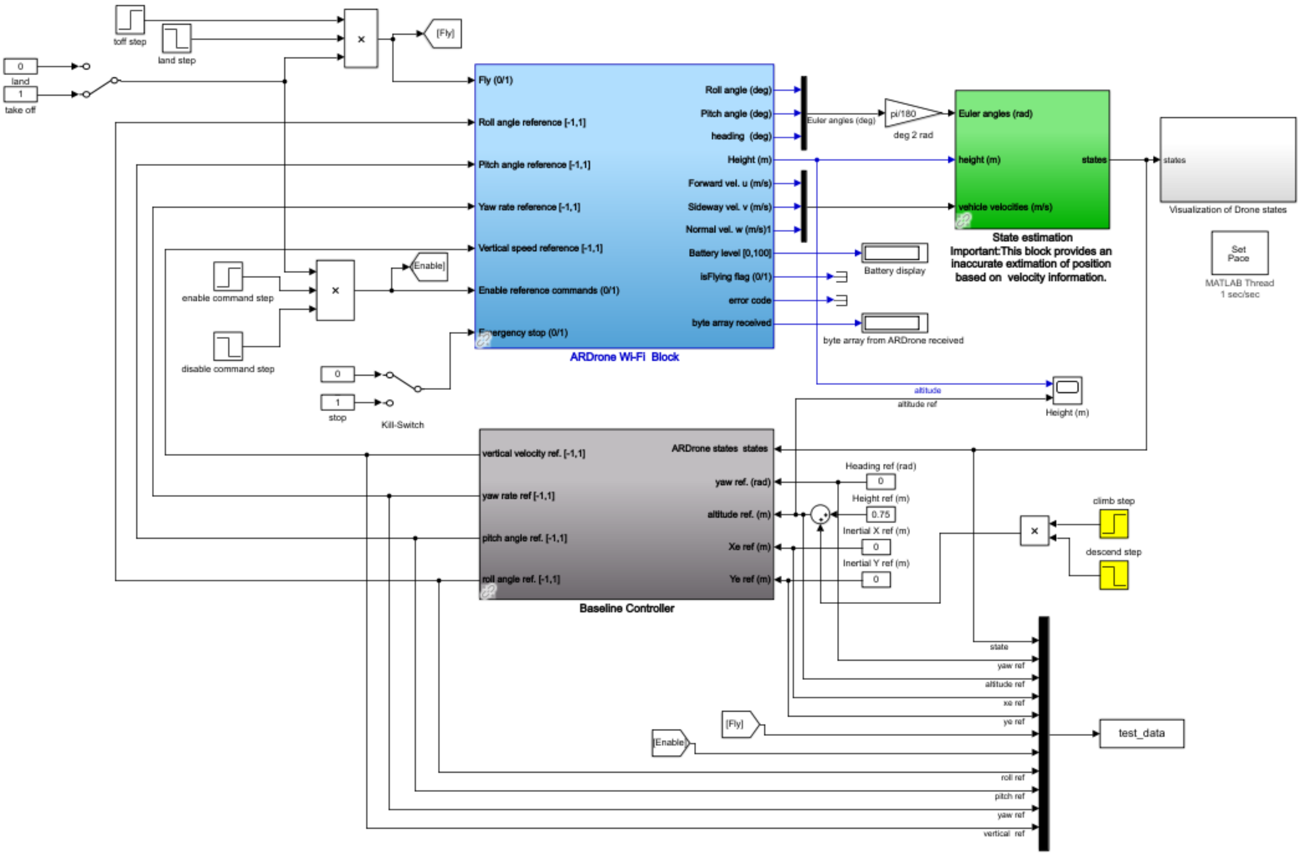


Figure 5: Simulink model adapted for height reference steps

The most important modification consisted in the introduction of two step blocks (highlighted in yellow). These produce an altitude reference step 20 seconds after the start of the simulation. To make sure that the drone followed this reference, steps were also placed on the ”Enable reference commands” entry that set this port to 1 before the altitude reference step and reset it to 0 after both steps had been applied. Another step then issues the land instruction, after which the test is concluded.

So as to modify the gain k_p, the controller block was located and the gain value was changes, as can be seen in figure 4.

In order to get a good system estimation, it is important that the system does not saturate, since this would introduce non-linearities in a system that is being approximated as linear. This means that the reference and gains must be chosen so that there are no saturations. Particularly in the case of the altitude control, the vertical speed is being used to control the altitude and it saturates at ± 1 m/s. The higher the controller gain k_p, the greater the reference for the same error. Therefore, if the gain is increased during the testing, the altitude reference step has to decrease so as to ensure the vertical speed requested to the controller does not saturate.

3.2 Step responses for different gains

Figure 6 shows the tests conducted for the height step response. Each of the plots corresponds to a controller gain k.

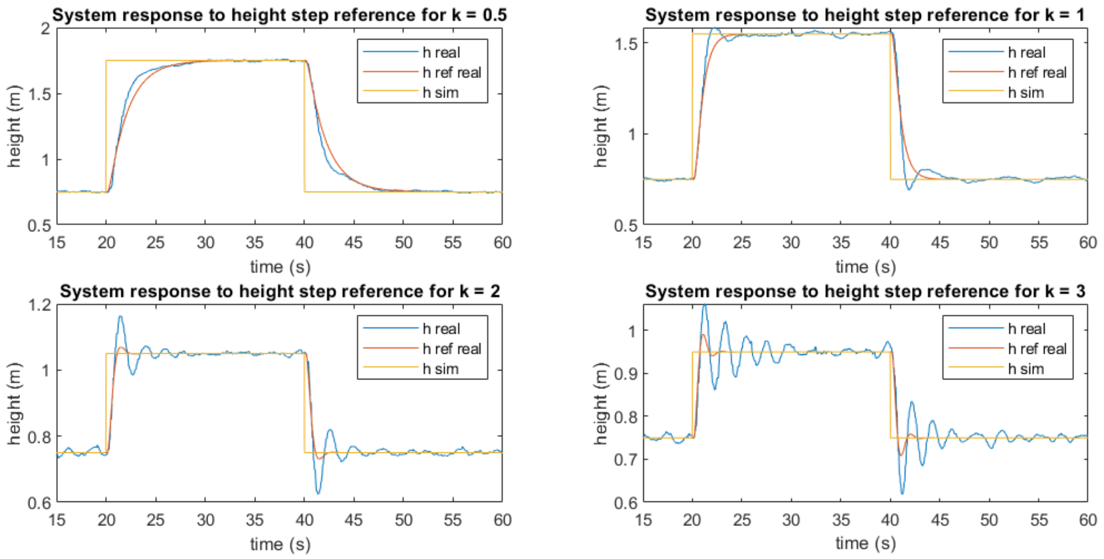


Figure 6: Time response of the altitude closed-loop system for different controller gains.

As was explained in the previous section, in order to obtain a good system identification, saturation must be avoided to ensure the linearity of the system. Analysing the Simulink model, the vertical speed reference is given by the following expression:

$$\dot{h}_{ref} = (h_{ref} - h)k_p = \pm 1 \; (m/s) \tag{43}$$

Table 1 presents the controller gains and the respective cumulative steps used in testing. Assuming the system is stable and the maximum error is achieved when the step is first applied, it is estimated that the maximum vertical speed reference will be given by the following formula: $\dot{h}_{ref} = \Delta h_{step} k_p$.

Gain (k)	Cumulative Step (m)	Max $\dot{h}_{ref}$ (m/s) (m/s)
0.5	1.0	0.5
1.0	0.8	0.8
2.0	0.3	0.6
3.0	0.2	0.6

Table 1: Controller gains, cumulative steps and estimated maximum vertical speed reference used in testing.

Since all the maximum vertical speed reference values are below 1 m/s, it is possible to conclude that the system did not saturate during these tests and, therefore, these results are plausible for system identification.

3.3 System identification for altitude steps

In this section, the MATLAB system identification toolbox was used to estimate the transfer functions and poles of the altitude closed-loop system. The condition that the system had to have 2 poles and no zeros was imposed, just like the transfer function found in the theoretical section for this system.

From every experiment, the moment the step was performed was extracted from the rest of the data to be used in the model identification. In other words, the takeoff and landing sections were removed, as these were not the response of the step input. The experimental data used for the model estimation for each value of the proportional gain k_p is shown in figure 8.

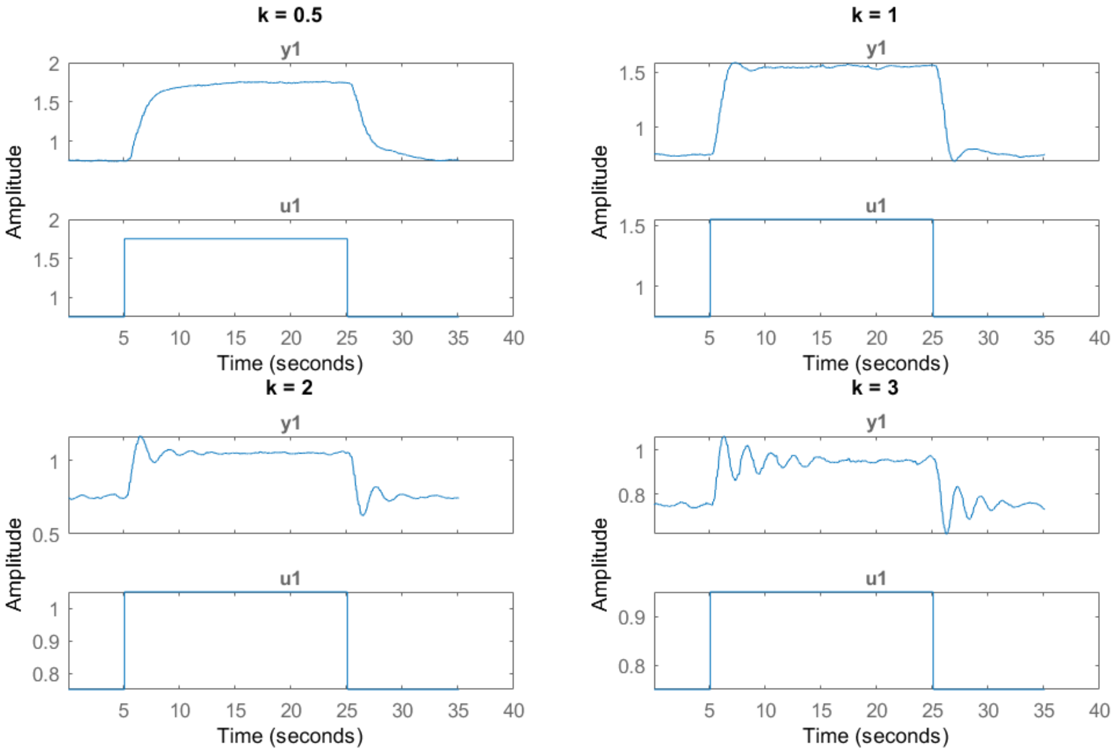


Figure 7: Data used for system identification

Using these data sets, the following transfer functions were identified:

$$\frac{H(s)}{H_{ref}(s)} = \frac{1.6012}{s^2 + 2.9668s + 1.6034}, \qquad k_p = 0.5 \tag{44}$$

$$\frac{H(s)}{H_{ref}(s)} = \frac{2.8742}{s^2 + 2.3263s + 2.8761}, \qquad k_p = 1 \tag{45}$$

$$\frac{H(s)}{H_{ref}(s)} = \frac{5.6800}{s^2 + 1.4185s + 5.6796}, \qquad k_p = 2 \tag{46}$$

$$\frac{H(s)}{H_{ref}(s)} = \frac{8.3516}{s^2 + 0.7844s + 8.3518}, \qquad k_p = 3 \tag{47}$$

From 44 through 47, information about the system can be inferred comparing these with the theoretical transfer functions. The system identification fit results are represented in table 2

Gain (k_p)	**Fit quality (%)**	$\frac{k_d}{m}$	$\frac{k_p}{m}$	**m (kg)**
0.5	95.02	2.9668	1.6012	0.3123
1.0	92.85	2.3263	2.8742	0.3479
2.0	90.38	1.4185	5.6800	0.3521
3.0	84.78	1.4185	8.3516	0.3592

Table 2: System identification fit results and transfer function information

Note that the fit quality decreases with the increase of k_p. This is due to the fact that for higher values of k_p there are more oscillations that the system identification struggles to capture. Nevertheless, the fit quality is acceptable for all values of k_p tested.

From the theoretical transfer function, it was established that the numerator was equal to $\frac{k_p}{m}$. Therefore, in order to estimate the mass of the quadcopter, for each test, the gain k_p was divided by the numerator of the transfer function identified by MATLAB. This resulted in an average mass of 0.3429 kg. From information available online, we would expect the mass of the drone to be be around 0.420 kg, which leads us to be believe that this method of calculating the mass is not very exact, and based on many simplifying assumptions.

Next, we re-ran the simulations, using the transfer functions obtained by fitting the experimental data to the second-order model as our system model. The comparison of the experimental results with these new simulated ones can be seen in figure 8.

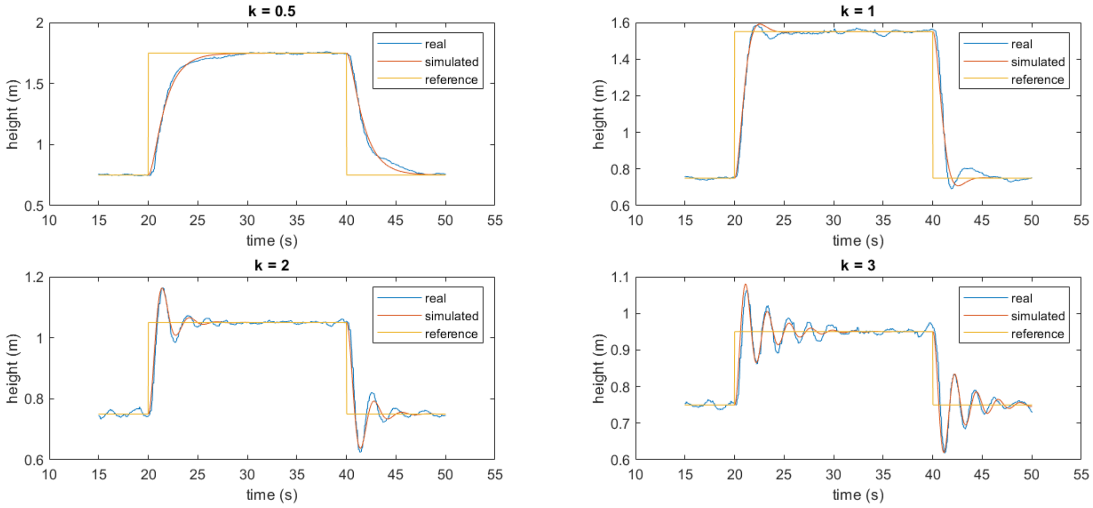
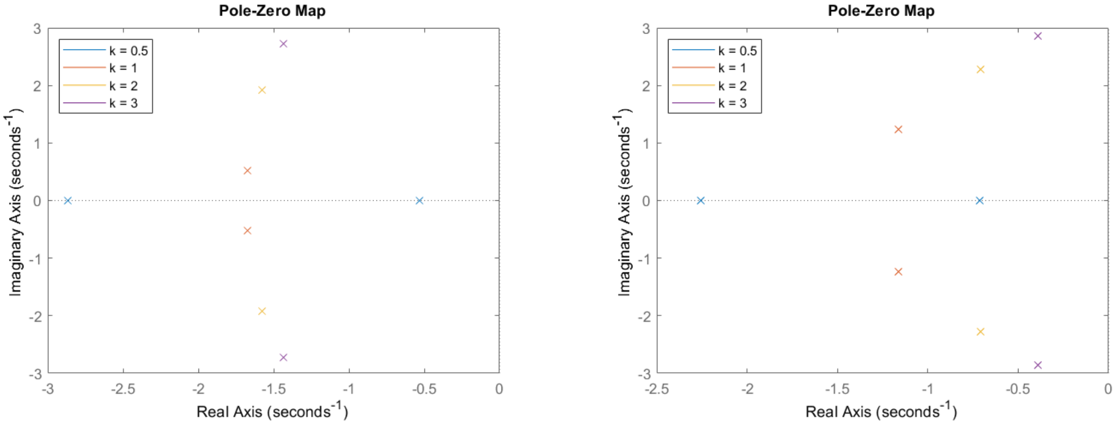


Figure 8: Test data compared with simulations generated by the estimated transfer function.

Looking at figure 8, the real and simulated responses are quite similar, following the same global trend with slightly larger oscillations for the real response. The most striking detail is the overshoot present for all values of k tested except 0.5, the only overdamped system. The response gets faster as k increases, as predicted theoretically.

3.4 Pole-zero map and comparison with the theoretical root locus



((a)) Simulation

((b)) Experimental results

Figure 9: Pole-zero map of the systems identified by MATLAB for each gain k_p for the simulated and real systems

Comparing figures 3 and 13, the simulation results of figure 9(a) are rather similar to the theoretical root locus presented. The simulation pole-zero map has a ”meeting point” for the poles at approximately -1.75, hinting that the ratio k_d is equal to 3.5. The experimental results, on the contrary, show a significant divergence from both the theoretical and simulation results, owing to some other factors that were not taken into account, such as drag forces or an imbalance in forces and moments due to the placement of the battery and of the protective cover.

In addition, the Simulink model already contemplates the existence of a delay in the feedback loop, something that the MATLAB root locus does not take into account. This results in a model that does not exactly replicate the real system. A possible solution comes in the form of a Padé approximation using the MATLAB function `pade` that returns the numerator and denominator of the so-called Padé approximant. The transfer function is then multiplied by the Padé approximation.

4 Identification of the closed-loop pitch dynamics

4.1 Obtaining experimental amplitude for bode diagram

Frequency (rad/s)	Reference amplitude (rad)	Measured amplitude (rad)	Gain (db)
1	0.2	0.101	-5.98
2	0.2	0.131	-3.66
3	0.3	0.225	-2.51
5	0.5	0.255	-5.84
10	0.5	0.118	-12.50
15	0.5	0.055	-19.17
20	0.5	0.028	-25.01

Table 3: Pitch frequency response for several sinusoidal reference signals. Data used for constructing Bode plot.

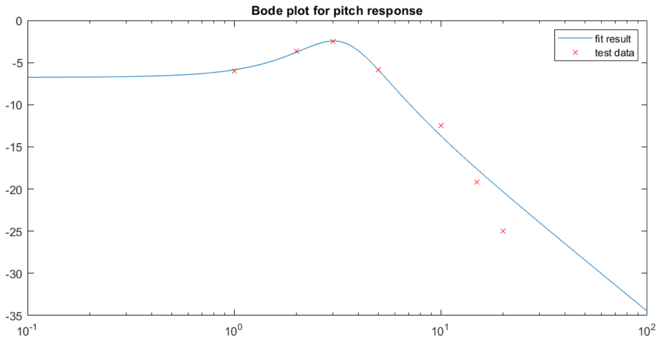


Figure 10: Experimental frequency response points and fit result.

Since the sample time is 0.03s (210 rad/s of sample frequency), it does not make sense to evaluate the frequency response of the system for frequencies higher than half of this value (105 rad/s), since the Nyquist theorem would not be met. It was noted that for frequencies higher than 10 rad/s, it was much harder to calculate the response's amplitude without more complicated methods (like fitting a sine wave to the data), since the $\frac{reference\,frequency}{sample\,frequency}$ ratio was higher than that for the lower reference frequencies.

4.2 Estimating the transfer function from the frequency response

From the data in table 3, using MATLAB `nlinfit`, the parameters ω_n, ζ, k and a of the following transfer function were estimated:

$$G(s) = \frac{k(s+a)}{s^2 + 2\zeta\omega_n s + \omega_n^2} \tag{48}$$

To fit the data, the model used was

$$|G(j\omega)| = \frac{k\sqrt{\omega^2 + a^2}}{\sqrt{(\omega_n^2 - \omega^2)^2 + (2\omega_n\zeta\omega)^2}} \tag{49}$$

The fit results were $\omega_n = 3.4057$ rad/s, $\zeta = 0.4872$, $k = 1.8838$, $a = 2.8285$. This system results in two complex conjugate poles at $-1.66 \pm 2.97i$, as well as a non-minimum phase zero at -2.83. Thus, the system is underdamped.

4.3 Response of the system to a pitch reference

$$\theta_{ref}(t) = \begin{cases} 0, & 0 \leq t < 20s \\ 0.2, & 20 \leq t < 23s \\ 0, & t \geq 23s \end{cases} \tag{50}$$

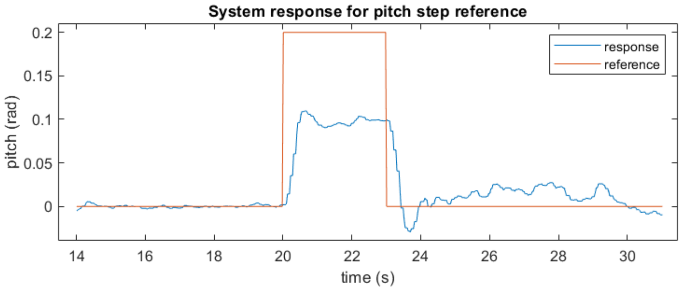


Figure 11: System response to a 3 second pitch reference step of 0.2 rad

While the reference is 0, there are slight oscillations in the pitch angle, mostly after the downwards step, in line with attempts to stabilise the drone through slight nudges performed in the laboratory. During the step section itself, it is possible to conclude that the system gain is not unitary, since the response did not reach 0.2 rad/s (instead, stabilising at around 0.1 rad/s). There appears to be a small delay in the response as well, to be expected because of the delay in communication with the drone. This delay is accounted for in the simulation.

4.4 System identification from concatenated responses

As well as estimating the model using MATLAB `nlinfit` with the frequency response data collected, another approach was also taken. Similarly to the analysis conducted for the height closed-loop system, the MATLAB system identification tool was used for the pitch closed-loop again.

In order to get good training data, the responses from most of the tests conducted for pitch were concatenated into one single response. To achieve this, the discontinuities were ignored, without any major issue appearing. This response was then used to identify a model with two poles and one zero. The results can be seen in figure 12. The fit achieved was 84% and the frequencies and damping of the poles can be found in figure 13(b). The values estimated by this model are similar to those obtained in section 4.2. As was expected, this estimated model also has two complex conjugate poles on the left complex semi-plane, was well as a non-minimum phase zero.

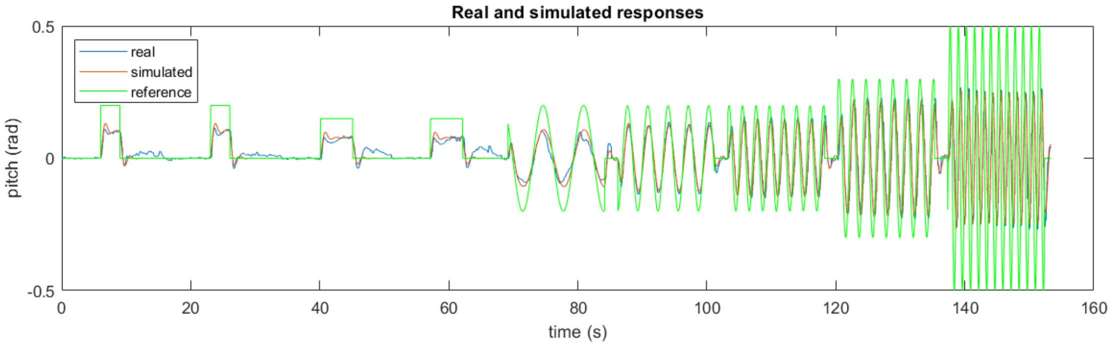
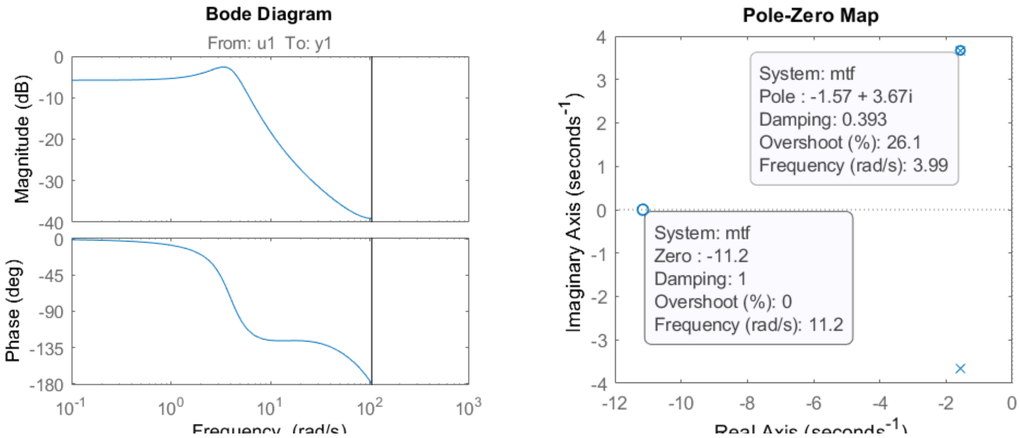


Figure 12: System responses concatenated for system identification



((a)) Simulation ((b)) Experimental results

Figure 13: Pole-zero map and Bode plot of the system identified from all the tests conducted for pitch

5 Conclusions

This laboratory activity allowed the analysis the quadrotor model considered in great detail, both initially, from a theoretical standpoint, and subsequently, as the model was tested against the experimental results obtained. The process, initially, went through all steps of the modelling process, namely identifying simplified forces and moments acting upon the quadcopter, deriving the adequate control inputs to use in the controller design, the nonlinear state models, finding and linearising the model around the equilibrium point, obtaining and reflecting about the physical meaning of ensuing transfer functions, choosing between two (different) possible altitude control approaches, and studying the impact of changing the gains on the chosen control strategy.

The experimental part was the study, separately, of the altitude control and pitch dynamics of the quadcopter. Both the simulated and the experimental responses to different types of inputs were studied, and, when required, varying the controller gains as well. In general, the experimental results were close to what was expected; however, a more detailed analysis enabled the quantification of the degree of adequacy of the theoretical model. In particular, fitting the experimental data to the second-order models derived generated numerically satisfying results. Obviously, the model has its limitations, and part of the work at this stage was reflecting on ways to improve it (for example, using corrections such as those given by the Padé approximation).

In summary, the experiment was successful and validated the theoretical model derived for the quadcopter.

References

[1] *Modeling and Identification of the Parrot AR.Drone.* 2022.