Design and Implementation of Flight Dynamics Control Strategies for a Smartphone-based Quadrotor

Thesis for obtaining the degree of

MASTER OF SCIENCE IN ENGINEERING
with emphasis in Automation

Alejandro Astudillo Vigoya
alejandro.astudillo@correounivalle.edu.co

School of Electrical and Electronic Engineering
UNIVERSIDAD DEL VALLE
Cali, COLOMBIA

November 29, 2017

Supervised by:

Dr.-Ing. Esteban Rosero
Industrial Control Research Group - GICI
School of Electrical and Electronic Engineering
Universidad del Valle

Bladimir Bacca Ph.D.
Perception and Intelligent Systems Research Group - PSI
School of Electrical and Electronic Engineering
Universidad del Valle

Abstract

The field of autonomous systems control is young, but operational experience is rapidly growing, making research on collaborative systems of great importance. Improving aerial robots in particular could be key in facing future environmental challenges..... In this work, two main problems are addressed: the cooperative source seeking problem and the cooperative level curve tracking problem by a group of agents under undirected constrained communications. .....

Resumen

The field of autonomous systems control is young, but operational experience is rapidly growing, making research on collaborative systems of great importance. Improving aerial robots in particular could be key in facing future environmental challenges..... In this work, two main problems are addressed: the cooperative source seeking problem and the cooperative level curve tracking problem by a group of agents under undirected constrained communications. .....

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Quadrotors

Figure 1.1: LJI 500-X4 carbon fiber frame[1]

1.2 Motivation

Quadrotor control is a difficult and interesting problem. A quadrotor has six degrees of freedom (three translational and three rotational) and four independent inputs (forces applied by the motors). As established by [1] and [2], quadrotor dynamics are affected by nonlinearity, parameters perturbations, uncertainties and disturbances: this include unknown and variable payloads, aerodynamical parameters of the system, wind changes, and sensors inaccuracies. Numerous studies have been developed in designing optimal and robust controllers that allow unmanned aircraft systems

[1]LJI 500-X4 frame image taken from https://goo.gl/hHfHQR

(UAS) to fly and accomplish missions rejecting disturbances and being robust to parameter uncertainties as seen in [3, 4, 5, 6].

Although there are embedded systems with high computational capacity that can serve as controllers of a quadrotor, smartphones are available, easily accessible for people and also have a large computational capacity, hence multiple instrumentation and communication elements integrated in the same device. The use of smartphones also facilitates distribution and installation of updates of the control application as it is a commonly known device and has application distribution platforms. Some attempts to joint smartphones and aerial robots have been made: in [7], a smartphone was used as a mission planner for a quadrotor and in [8], a smartphone was used as flight controller using its sensors and power to stabilize the quadrotor and control its altitude. A smartphone has been used as a flight controller and processing system for image-based positioning in a quadrotor, as shown in [9].

This research project aims to design and implement algorithms that will be executed in a smartphone to estimate and control quadrotor dynamics by the Research Group in Industrial Control. This project confronts several challenges such as using a smartphone as a hardware development platform, trying to use a non real-time operating system for real-time applications, designing optimal and high order controllers using Java or C++, and executing that controllers in a smartphone.

This paper presents the design of an optimal and a robust controller, and a state estimator based on a Kalman filter for a smartphone-based quadrotor. The nonlinear and linerized model of the quadrotor is presented, the controller that allows the quadrotor to follow a trajectory reference using two approachs is designed. The two approaches are the LQG and H_∞ controllers. The quadrotor flight dynamics estimation strategies using sensor fusion algorithms are described.

In recent years, the interest in aerial robotics research has increased substantially. This is because this type of robotics offers several potential new services such as search and rescue, observation, mapping, inspection, etc. On the other hand, smartphones have become essential devices for humans and easily acquirable development tools. The interaction between these two technologies allow the development of low cost quadcopters based on an everyday item such as the smartphones, facilitating the distribution of the quadcopter control software and its implementation by other researchers.

In this paper, the implementation of a quadcopter with a smartphone acting as its flight controller while using exclusively the sensors and processor in the smartphone, is shown. The controller keeps the attitude of the quadcopter stabilized while making the quadcopter to hover at an altitude reference. It is presented the detailed composition of the test platform (quadcopter) used, integrating it with its dynamic

Figure 1.2: Assembled quadcopter used in this research with the on-board smartphone on the top center of it.

model in addition to the quadcopter altitude and attitude estimation strategies using sensor fusion algorithms.

1.3 Research Problem

The existing research challenges include how to develop and implement efficient control algorithms for smartphones using the Android operating system, and assess, adapt and develop the appropriate communication, sensing and performance technologies with smartphones in the execution of missions using quadrotors.

Then, the question to be answered is: how to develop control strategies in a smartphone in order to control the flight dynamics of a quadrotor so that it can develop flight missions while the instrumentation and computing capacity of the smartphone is used?

1.4 Objectives

In order to find a solution for the research problem, the following general and specific objectives are proposed:

General Objective

Design and implement algorithms for control and estimation of flight dynamics executed in a smartphone for the quadrotor of the Industrial Control Research Group.

Specific Objectives

1. Conduct a study and analysis of the state of the art related to the control and estimation of states of quadrotors.

2. Integrate the existing quadrotor with a smart phone that contains the appropriate sensors for the control and estimation of states.

3. Obtain a dynamic model of the quadrotor.

4. Design and implement the algorithms for control and estimation of states for the quadrotor.

5. Integrate the experimentation platform with the control and estimation algorithms in the smartphone.

6. Evaluate the performance of the control strategies.

1.5 Literature Review

1.5.1 Control Strategies in Quadrotors

1.5.2 State Estimation in Quadrotors

1.5.3 Quadrotor Flight Modes

In quadrotors, the on-board flight controllers keep some of the quadrotors DoF in a desired value autonomously in order to allow pilots to perform tasks during a flight. This controllers have different modes that can control from 3 to 6 DoF depending on the will of the pilot. Flight modes commonly found in commercial flight controllers may be as basic to only control its attitude or as complex to let the quadrotor follow a complex trajectory with multiple waypoints [10].

The main flight modes, widely used in commercial flight controllers, are described bellow according to the number of controlled DoF, in ascending order.

- **Stabilize Mode**

  This mode allows the pilot to fly the quadrotor manually while the flight controller self-levels the quadrotor attitude and regulate its current heading. Thus, the stabilize mode attempts to control three DoF of the quadrotor.

  The attitude references can be set or changed by the pilot using the remote control, but their default value is 0 rad. On the other hand, the quadrotor heading is simply set to be regulated in its current state, enabling its rate using the remote control.

  As the stabilize mode do not take into account the control of the quadrotor position, the pilot needs to regularly change the attitude references manually to keep the quadrotor in a desired position, as it is affected by wind disturbances. Also, the pilot needs to regularly adjust the quadrotor thrust, so that a desired altitude is maintained.

- **Altitude Hold Mode**

  The altitude hold mode adds automatic elevation control to the Stabilize mode. This way, in addition to controlling the attitude, the quadrotor thrust is set by the flight controller in order to maintain the quadrotor in a desired altitude, getting four controlled DoF.

  In this mode, the pilot can remotely control the rate of change of the elevation (with a default value of 0 m/s), as well as the attitude references.

- **Loiter Mode**

  Loiter mode automatically attempts to regulate the six DoF of the quadrotor in order to maintain a desired position, heading and altitude during a flight. Here, the quadrotor attitude is self-leveled, while the position and altitude reference can be modified by the pilot using the remote controller. The position and altitude references are initialized using the current quadrotor position and altitude when this mode is set. This is a GNSS-Dependent flight mode.

- **Auto Mode**

  The Auto mode attempts to make a quadrotor follow automatically a preprogrammed path connecting multiple position and heading waypoints, while receiving . This mode use the same controller as the Loiter mode, but its references are set automatically following the waypoints list. As in Auto mode the quadrotor must follow geo-located waypoints, this is a GNSS-Dependent flight mode.

- **Return-To-Launch Mode**

  During a flight mission, the home location is set as the position and altitude where the quadrotor took off. The Return-To-Launch mode is used in case of emergency or when a the last waypoint is reached within a flight mission. This mode is equivalent to the Auto mode, but only has two waypoints. The first waypoint consists in the position of the home location with a previously set altitude (RTL altitude) greater than the home altitude. When this waypoint is reached, the home location is set as the following waypoint so that the quadrotor starts its landing keeping the position controlled.

1.5.4 Smartphones in Control Systems

Current smartphone processors are able to perform complex calculations such as those required in the implementation of real time control strategies. There are many ongoing research related to the possibility of using smartphones to implement control strategies, such as [11], as configuration and monitoring interfaces in control systems as seen in [12, 13], and as a tool in both education and design of control strategies seen in [14, 15]. Following this trend, in the Universidad del Valle, it was developed a smartphone-based platform for monitoring, control and communication in portable laboratories, where a controller for a pendulum based in the Lego Mindstorms EV3 platform was implemented [16].

[17] [18] [19] [20] [21] [22] [23] [24] [25] [26]

1.5.5 Smartphone-based Quadrotors

[27] [28] [29] [30] [31]

In the University of Pennsylvania, in [9], was developed a quadcopter using a last generation smartphone as a flight controller and an additional processing system for image-based positioning. The state estimation algorithms, control and planning were firstly implemented in a ODROID-XU board with additional sensors, but then, in [32], this algorithms were ported to the Qualcomm processor in the phone due to the Qualcomm colaboration in that project.

Current research focuses on the development of aerial robots potentiated by the use of smartphones, as seen in [7, 8, 33, 34]. In the last years, computing capacity and sensor technology in smartphones has decreased in price but increased in performance. Smartphones have become an inexpensive tool capable of commanding an UAV. The challenge then, is to use smartphones as quadcopter flight controllers for autonomous flights following specific missions, taking advantage of the fact that

the phones today are very powerful computers that include elements of sensing, processing and signal communication.

Smartphone-based Quadrotor Limitations

The idea of using a smartphone as flight controller in an UAV opens the possibility of a quick and inexpensive development [33]. A smartphone offers other advantages compared with off-the-shelf flight controllers, for instance its powerful quad, hexa or octa-core processors and communications interfaces. However, smartphones and the Android operating system have some limitations that set challenges when implementing a control system in it.

Android is not a real-time operating system and thus can not assure execution of algorithms, like estimation and control, with a constant sample time. Furthermore, the sensors embedded in commercial smartphones are made for applications that do not have high requirements of accuracy nor precision and therefore may not be appropriate for sensing quadrotor dynamics. Nonetheless, as explained by [34], due to its computing capabilities, smartphones can overcome this limitations while using a temporized thread to execute the control system algorithms and implementing a sensor fusion technique to improve the states estimation reliability. This thread must be executed with a lower sample time compared to the one of the sensors embedded in the smartphone. This will ensure that the execution is not delayed by the sensors acquisition process.

1.6 Outline

This thesis is organized as follows.

In Chapter 1,

2 3 4 5 6

In this chapter an introduction is given to this project and the quadrotor platform. Chapter 2 contains a overview of other main quadrotor platforms. In the next part the dynamic model of the quadrotor is described, which is used to test the indoor navigation and control in simulation. Because the quadrotor is open-loop unstable system, a controller is required to do flight tests in simulation. This controller is designed in chapter 4. In part III first the selected sensors that are used in the sensor integration are modelled. Chapter 6 explains the state estimation method used to integrate the sensors of the IMU and the IR sensors and discusses

the results of this state estimation in simulation. To perform flight tests with actual sensors, a quadrotor platform is designed. Part IV starts with the development of this platform in chapter 7. Because without any control feedback the quadrotor cannot fly, onboard filtering and control is required, which is described in chapter 8. Next in chapter 9 the real data from the quadrotor UAV is used to test the state estimation method developed. Finally, a conclusions and discussion of the work and recommendations for future work are given.

Chapter 2

Dynamic Model of the Quadrotor

In this chapter, the dynamics of the quadcopter and the calculation of the mathematical model through two different approaches is provided. In addition, the consequences of choosing the geometry of a quadrotor between two configurations for the inputs setting in a real implementation are put into consideration. This dynamic model and its inputs setting are necessary for the design of the controllers that will make it possible to maintain the quadrotor in a desired state.

In Section 2.1: Quadrotors Configurations, a description of two of the mainly used quadrotor geometry configurations and what it means to choose one or the other for the setting of the quadrotor inputs, is provided.

Section (2.2: Nonlinear Model) shows the dynamic model of the quadrotor obtained using the Newton-Euler and Euler-Lagrange approaches, neglecting the gyroscopic effects produced by the propellers.

Finally, an overview of the Jacobian linearization method applied to the quadrotor dynamic model and the thrust compensation needed to be implemented in a real quadrotor, is exposed in Section 2.3: Linearized Model.

2.1 Quadrotors Configurations

As the term 'quadrotor' refers to a multirotor whose thrust is generated from four motors and propellers, quadrotors can be built in multiple ways as long as they comply with the established definition. There are some configurations that are already standardized being widely used by commercial manufacturers and hobbyists, such as the '+' and 'X' configurations.

2.1.1 The '+' Configuration

The geometry used in quadrotors built in '+' configuration is shown in Fig. 2.1.

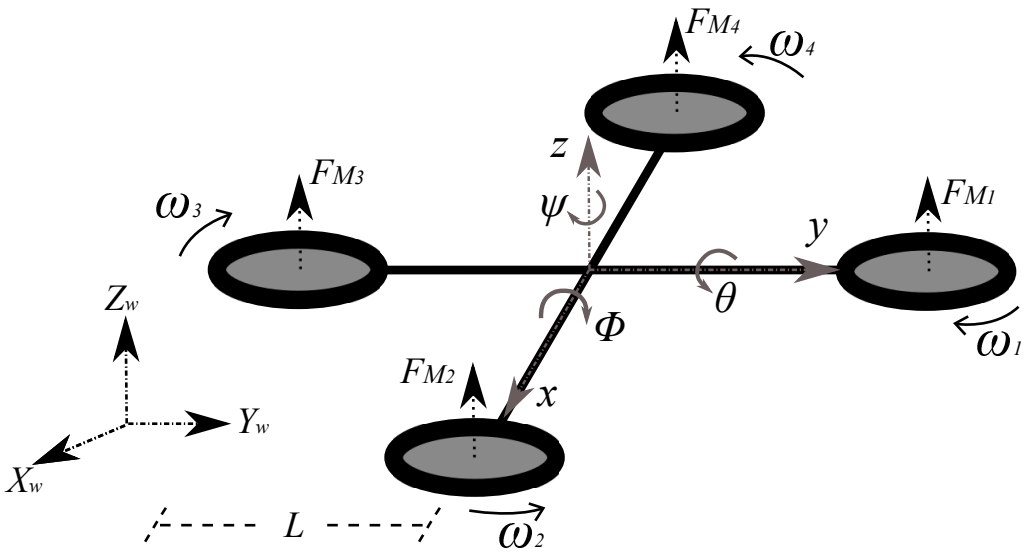


Figure 2.1: Quadrotor geometry in '+' configuration

Where, (ϕ, θ, ψ) are the angular deviations (pitch, roll and yaw, respectively) of the quadrotor about the body-frame, (x, y, z) are the body-frame axes, (X_W, Y_W, Z_W) are the earth-frame axes, L is the distance from the quadrotor center of gravity (CoG) to the motors center, F_{M_i} is the thrust force exerted by each motor M_i, and ω_i is the angular velocity of M_i, with $i = 1, 2, 3, 4$. In this configuration, the body-frame axes x and y, coincide with the lines that connect motors of opposite sides in the quadrotor frame.

Although the quadrotor is a system with 6 degrees of freedom (DoF), the fact of having only four actuators, makes it an underactuated system. Therefore, it is only possible to reach a desired state for 4 degrees of freedom. However, it is possible to choose four control signals, or inputs, that represent the quadrotor basic movements and allow the quadrotor to achieve a desired position and attitude. This inputs are described below.

- **Thrust** T_u $[N]$

  The thrust u is the total thrust force exerted parallel to the body-frame z-axis by the four motors. When the θ and ϕ angles are different from zero, the thrust generates accelerations, not only on the Z_W-axis, but also on the X_W

and Y_W axes, as the body-frame z-axis and the Z_W do not coincide.

This control signal affects the speed of rotation of all motors, and therefore their thrust, in equal magnitude, and is set as

$$T_u = \sum_{i=1}^{4} F_{M_i}. \tag{2.1}$$

- **Yaw Torque** τ_ψ $[N \cdot m]$

  As can be seen in Fig. 2.1, M_1 and M_3 have a clockwise rotation while M_2 and M_4 rotate counter-clockwise. This configuration of opposite pairs rotational directions allows the system to control its conservation of momentum, and thus change its yaw angle in a controlled manner, just unbalancing the total momentum around the z-axis and without the need of a tail rotor used in the standard helicopter structure ([35]). In '+' configuration, the fact that the pitch and roll angles are controlled using only two motors that rotate in the same direction, leads to large changes in the thrust force of the other two motors to achieve conservation of momentum.

  This conservation of momentum is controlled using the torque generated by each motor (τ_{M_i}) around the body-frame z-axis, causing a turn in the direction opposite to the rotation of the motor. Taking into account that the torque τ_ψ is positive when it generates a clock-wise rotation around the z-axis, only M_2 and M_4 contribute positively to it, while the others have a negative contribution to τ_ψ. Thus, the total torque around the z-axis is set as

  $$\tau_\psi = \tau_{M_2} + \tau_{M_4} - \tau_{M_1} - \tau_{M_3}. \tag{2.2}$$

  Each torque τ_{M_i} has a linear relationship with the thrust applied by the motor, with K_M being the proportional constant and

  $$\tau_{M_i} = K_M F_{M_i}. \tag{2.3}$$

  Replacing (2.3) in (2.2) the total torque τ_ψ dependence on the forces F_{M_i} is got as

  $$\tau_\psi = K_m(F_{M_2} + F_{M_4} - F_{M_1} - F_{M_3}). \tag{2.4}$$

- **Roll Torque** τ_θ $[N \cdot m]$

  The roll torque τ_θ is the torque exerted on the x-axis and about the y-axis. In the '+' configuration, the only motors that affect the quadrotor rotation with respect to the x-axis are M_2 and M_4. These in turn do not affect the rotation with respect to the y-axis. In this case, the torques are generated by the forces

F_{M_2} and F_{M_4} being applied at a distance L from the quadrotor CoG. Considering a positive τ_θ as the one causing a counter clock-wise rotation about the y-axis, the roll torque in '+' is set as

$$\tau_\theta = L(F_{M_4} - F_{M_2}). \tag{2.5}$$

- **Pitch Torque** τ_ϕ $[N \cdot m]$

  Unlike the roll torque, the pitch torque τ_ϕ is the one exerted on the y-axis about the x-axis. As exposed in Fig. 2.1, M_1 and M_3 are the motors applying the forces that create this torque. Being τ_ϕ positive when clock-wise rotation about the x-axis is caused, it is defined for a '+' configuration as

$$\tau_\phi = L(F_{M_3} - F_{M_1}). \tag{2.6}$$

Inputs Setting in '+' Configuration

Summarizing the description of the four inputs in the quadrotor represented in a '+' configuration, these are established as linear combinations of the motors forces F_{M_i} and can be represented as a system of linear equations in matrix representation based on equations (2.1), (2.4), (2.5) and (2.6). Thus, the vector of inputs $U_{(+)}$ is defined as

$$\mathbf{U}_{(+)} = \begin{bmatrix} T_u \\ \tau_\psi \\ \tau_\theta \\ \tau_\phi \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ -K_m & K_m & -K_m & K_m \\ 0 & -L & 0 & L \\ -L & 0 & L & 0 \end{bmatrix} \begin{bmatrix} F_{M_1} \\ F_{M_2} \\ F_{M_3} \\ F_{M_4} \end{bmatrix}. \tag{2.7}$$

2.1.2 The 'X' Configuration

Following the same nomenclature used in the '+' configuration, in 'X' configuration, the quadrotor frame is rotated $\pi/4\ rad$ about the z-axis in the body-frame, as shown in Fig. 2.2.

In this case, the front-line in the quadrotor is set between M_1 and M_4, while M_2 and M_3 define its back line. Using the 'X' configuration, the roll and pitch torques are applied using the forces exerted by all the motors at a distance $L_X = L \cdot \cos(\pi/4)$. The geometry used in this configuration is shown in Fig. 2.2.

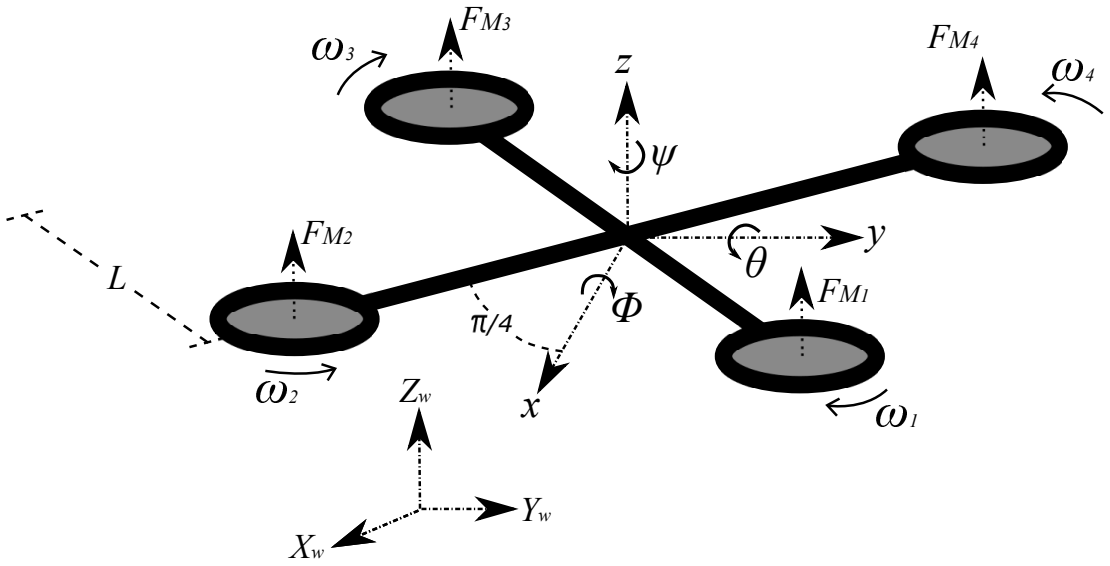


Figure 2.2: Quadrotor geometry in 'X' configuration

The quadrotor inputs in 'X' configuration do not change with respect to the + configuration, but it does change the way τ_θ and τ_ϕ are defined with respect to the forces F_{M_i}. The inputs description for the quadrotor 'X' configuration is defined as follows.

- **Thrust** T_u $[N]$

  The total thrust exerted by the four motors in the quadrotor is not changed between the '+' and 'X' configurations. Hence, the thrust u is defined in the same way as in '+' configuration as

$$T_u = \sum_{i=1}^{4} F_{M_i}. \tag{2.8}$$

- **Yaw Torque** τ_ψ $[N \cdot m]$

  As the z-axis is not modified between the '+' and 'X' configurations, and the motors rotate in the same direction as in the other configuration, the yaw torque τ_ψ is set as

$$\tau_\psi = K_m(F_{M_2} + F_{M_4} - F_{M_1} - F_{M_3}). \tag{2.9}$$

- **Roll Torque** τ_θ $[N \cdot m]$

  In 'X' configuration, $L_X = L \cdot \cos(\pi/4)$ is the real distance between the point of application of the roll and pitch torques and the quadrotor center of mass along the x and y axes ([36]). Also, as shown in Fig. 2.2, each of the four motors affects these torques. For the roll torque τ_θ, rhe motors M_3 and M_4 contribute positively to τ_θ, while M_1 and M_2 have a negative contribution to this torque. Then, for the 'X' configuration, the roll torque is set as

$$\tau_\theta = L_X(F_{M_3} + F_{M_4} - F_{M_2} - F_{M_1}). \tag{2.10}$$

- **Pitch Torque** τ_ϕ $[N \cdot m]$

  As in the roll torque, the pitch torque τ_ϕ is affected by the forces exerted by the four motors in the quadrotor in 'X' configuration. Due to the clockwise positive direction of the ϕ angle, and keeping L_X as the distance between the quadrotor CoG and the point of application of the force on each axis, the pitch torque for the 'X' configuration is defined as

$$\tau_\phi = L_X(F_{M_2} + F_{M_3} - F_{M_1} - F_{M_4}). \tag{2.11}$$

Inputs Setting in 'X' Configuration

In the 'X' configuration, the quadrotor inputs remain the same with respect to the '+' configuration. However the torques τ_θ and τ_ϕ are affected by the interaction of all the motors and the change in the distance of the point of application of the forces F_{M_i} on the x and y axes. The system of linear equations that shows the inputs setting for a quadrotor in 'X' configuration is based on equations (2.8), (2.9), (2.10) and (2.11), and is defined as

$$\mathbf{U}_{(\mathbf{X})} = \begin{bmatrix} T_u \\ \tau_\psi \\ \tau_\theta \\ \tau_\phi \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ -K_m & K_m & -K_m & K_m \\ -L_X & -L_X & L_X & L_X \\ -L_X & L_X & L_X & -L_X \end{bmatrix} \begin{bmatrix} F_{M_1} \\ F_{M_2} \\ F_{M_3} \\ F_{M_4} \end{bmatrix}. \tag{2.12}$$

Maximum Torque About x and y axes in 'X' Configuration

In '+' configuration, the torques around the x and y axes are set using just two motors applying a force at a distance L from the quadrotor CoG. This implies that, for example in the case of roll torque τ_θ, the maximum torque $\tau_{\theta max(+)}$ is achieved when $F_{M_4} = F_{M_i max}$ and $F_{M_3} = 0$, where $F_{M_i max}$ is the maximum thrust of the

motors. Then, in '+' configuration, the $\tau_{\theta max(+)}$ is set as

$$\tau_{\theta max(+)} = L \cdot F_{M_i max} \tag{2.13}$$

On the other hand, the torques around the x and y axes in 'X' configuration depend on the forces of the four motors in the quadrotor, which are applied at a distance $L_X = L \cdot \cos(\pi/4)$ from the quadrotor CoG. Continuing with the example of the roll torque, in 'X' configuration the maximum torque $\tau_{\theta max(X)}$ is achieved when $F_{M_3} = F_{M_4} = F_{M_i max}$ and $F_{M_1} = F_{M_2} = 0$. Hence, the maximum torque about the y-axis in 'X' configuration is

$$\begin{aligned} \tau_{\theta max(X)} &= L \cdot \cos(\pi/4) \cdot 2 \cdot F_{M_i max} \\ \tau_{\theta max(X)} &= 2 \cdot \cos(\pi/4) \cdot \tau_{\theta max(+)} \end{aligned} \tag{2.14}$$

Thereby, the quadrotor in 'X' configuration has $2 \cdot \cos(\pi/4)$ times more available torque to rotate about the x and y axes, when compared with the '+' configuration, and therefore it can achieve 41.42 % more rotational acceleration about the x and y axes.

2.2 Nonlinear Model

This section describes the dynamic modeling used to develop the quadrotor control, based on the study carried out in [35] and [37]. This model represents the quadrotor as a solid symmetrical object subject to a total thrust (u) and three torques (τ_ψ, τ_θ and τ_ϕ), assuming that the quadrotor CoG coincides with the origin of the body-frame and without considering the dynamics of the actuators . The modeling of the quadrotor system is done by two different methods. The Newton-Euler approach is based on the quadrotor body-frame, while the Euler-Lagrange approach bases its translational equations in the earth-frame while keeping its rotational equations related to the body-frame.

2.2.1 Newton-Euler Approach

Following the quadrotor geometry shown in Fig. 2.2, the quadrotor position vector ($\mathbf{\Xi}$), composed by the translational position ($\mathbf{\Gamma_W}$ $[m]$) and rotational position ($\mathbf{\Theta_W}$ $[rad]$) with respect to the earth-frame, is defined as

$$\mathbf{\Xi} = \begin{bmatrix} \mathbf{\Gamma_W} \\ \mathbf{\Theta_W} \end{bmatrix} = \begin{bmatrix} X_W \\ Y_W \\ Z_W \\ \psi \\ \theta \\ \phi \end{bmatrix}. \tag{2.15}$$

On the other hand, the quadrotor velocity vector (ν) is composed by the translational ($\mathbf{V_B}$ $[m{\cdot}s^{-1}]$) and rotational ($\mathbf{\Omega_B}$ $[rad{\cdot}s^{-1}]$) velocities with respect to the body-frame as

$$\nu = \begin{bmatrix} \mathbf{V_B} \\ \mathbf{\Omega_B} \end{bmatrix} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\psi} \\ \dot{\theta} \\ \dot{\phi} \end{bmatrix}. \tag{2.16}$$

There exists a generalized matrix

$$\zeta_{\mathbf{\Theta}} = \begin{bmatrix} \mathbf{R_b^w} & \mathbf{0_{3\times3}} \\ \mathbf{0_{3\times3}} & \mathbf{T_b^w} \end{bmatrix}, \tag{2.17}$$

that ensures that

$$\dot{\mathbf{\Xi}} = \zeta_{\mathbf{\Theta}}\nu, \tag{2.18}$$

where $\mathbf{0_{3\times3}}$ is a 3×3-matrix filled with zeros, $\mathbf{R_b^w}$ is the rotation matrix and $\mathbf{T_b^w}$ the transfer matrix from the body to the world-frame, defined as

$$\mathbf{R_b^w} = \begin{bmatrix} c_\theta c_\psi & c_\psi s_\theta s_\phi - c_\phi s_\psi & s_\phi s_\psi + c_\phi c_\psi s_\theta \\ c_\theta s_\psi & s_\psi s_\theta s_\phi + c_\phi c_\psi & c_\phi s_\psi s_\theta - s_\phi c_\psi \\ -s_\theta & c_\theta s_\phi & c_\theta c_\phi \end{bmatrix}, \tag{2.19}$$

$$\mathbf{T_b^w} = \begin{bmatrix} 1 & s_\phi t_\theta & c_\phi t_\theta \\ 0 & c_\phi & -s_\phi \\ 0 & s_\phi/c_\theta & c_\phi/c_\theta \end{bmatrix}, \tag{2.20}$$

with $s_\theta = \sin(\theta)$, $c_\theta = \cos(\theta)$, and $t_\theta = \tan(\theta)$.

As the quadrotor is assumed to be a rigid body of 6 DoF, its dynamics consider the mass (m $[kg]$) and the inertia matrix ($\mathbf{J}$ $[kg \cdot m^2]$) of it, and are described as

$$\mathbf{M_B}\dot{\nu} + \mathbf{S}_\nu = \mathbf{\Lambda} \tag{2.21}$$

where

$$\begin{aligned}
\mathbf{M_B} &= \begin{bmatrix} m\mathbf{I_{3\times 3}} & \mathbf{0_{3\times 3}} \\ \mathbf{0_{3\times 3}} & \mathbf{J} \end{bmatrix} = \begin{bmatrix} m & 0 & 0 & 0 & 0 & 0 \\ 0 & m & 0 & 0 & 0 & 0 \\ 0 & 0 & m & 0 & 0 & 0 \\ 0 & 0 & 0 & J_{zz} & 0 & 0 \\ 0 & 0 & 0 & 0 & J_{yy} & 0 \\ 0 & 0 & 0 & 0 & 0 & J_{xx} \end{bmatrix}, \\
\dot{\nu} &= \begin{bmatrix} \dot{\mathbf{V}}_\mathbf{B} \\ \dot{\mathbf{\Omega}}_\mathbf{B} \end{bmatrix}, \\
\mathbf{S}_\nu &= \begin{bmatrix} \mathbf{\Omega_B} \times m\mathbf{V_B} \\ \mathbf{\Omega_B} \times \mathbf{J\Omega_B} \end{bmatrix} = \begin{bmatrix} m(\ddot{x} + \dot{z}\dot{\theta} - \dot{y}\dot{\phi}) \\ m(\ddot{y} + \dot{x}\dot{\phi} - \dot{z}\dot{\psi}) \\ m(\ddot{z} + \dot{y}\dot{\psi} - \dot{x}\dot{\theta}) \\ J_{zz}\dot{\psi} + \dot{\theta}\dot{\phi}(J_{xx} - J_{yy}) \\ J_{yy}\dot{\theta} + \dot{\psi}\dot{\phi}(J_{zz} - J_{xx}) \\ J_{xx}\dot{\phi} + \dot{\psi}\dot{\theta}(J_{yy} - J_{zz}) \end{bmatrix}, \\
\mathbf{\Lambda} &= \begin{bmatrix} \mathbf{F_B} \\ \tau_\mathbf{B} \end{bmatrix} = \begin{bmatrix} F_x & F_y & F_z & \tau_z & \tau_y & \tau_x \end{bmatrix}^T,
\end{aligned} \tag{2.22}$$

being $\mathbf{J}$ diagonal due to the assumption of a perfectly symmetric quadrotor body, $\mathbf{I}$ the identity matrix, $\dot{\mathbf{V}}_\mathbf{B}$ is the quadrotor translational acceleration in the body-frame, $\dot{\mathbf{\Omega}}_\mathbf{B}$ is the quadrotor angular acceleration in the body-frame, $\mathbf{F_B}$ is the quadrotor force vector and $\tau_\mathbf{B}$ is the quadrotor torques vector.

The forces and torques vector $\mathbf{\Lambda}$ results from the effect of the gravitational force (represented in $\mathbf{G_\Lambda}$), the quadrotor inputs created by the motor forces (represented in $\mathbf{U_\Lambda}$), and the gyroscopic effects produced when the motors propellers are rotating (represented in $\mathbf{P_\Lambda}$). However in this project, for simplicity in the dynamic model, it is not taken into account the $\mathbf{P_\Lambda}$ contribution to the $\mathbf{\Lambda}$ vector and thus $\mathbf{P_\Lambda} \approx \mathbf{0_{6\times 1}}$.

The gravitational force affects just the $\mathbf{F_B}$ component in $\mathbf{\Lambda}$, proportionally to its magnitude $|\vec{g}| = g = 9.807\ m/s^2$. $\mathbf{G_\Lambda}$ is the contribution of the gravitational force to the vector $\mathbf{\Lambda}$ and is expressed as

$$\mathbf{G_\Lambda} = \begin{bmatrix} \hat{\mathbf{F}}_\mathbf{Gb} \\ \mathbf{0_{3\times 1}} \end{bmatrix} = \begin{bmatrix} \mathbf{R_b^w F_{G\xi}} \\ \mathbf{0_{3\times 1}} \end{bmatrix} = \begin{bmatrix} \mathbf{R_w^b} \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} \\ \mathbf{0_{3\times 1}} \end{bmatrix} = \begin{bmatrix} mg\sin(\theta) \\ mg\cos(\theta)\sin(\phi) \\ -mg\cos(\theta)\sin(\phi) \\ \mathbf{0_{3\times 1}} \end{bmatrix} \tag{2.23}$$

were $\mathbf{F_{G\xi}} = \mathbf{R_b^w}\hat{\mathbf{F}}_\mathbf{Gb}$ is the translational force due to gravity in the earth-frame, and $\mathbf{R_w^b} = (\mathbf{R_b^w})^{-1}$ is the rotation matrix from the world to the body-frame.

From Section 2.1, it follows that the quadrotor inputs are proportional to the motor forces F_{M_i}, and do not affect the x and y components of $\mathbf{F_B}$. This inputs, depend on the quadrotor configuration ('+' or 'X') and are set as

$$\mathbf{U_\Lambda} = \begin{bmatrix} 0 \\ 0 \\ u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ T_u \\ \tau_\psi \\ \tau_\theta \\ \tau_\phi \end{bmatrix} \tag{2.24}$$

where T_u, τ_ψ, τ_θ and τ_ϕ depend on the configuration of the quadrotor as exposed in (2.7) and (2.12).

Thus, (2.21) can be redefined as

$$\mathbf{M_B}\dot{\nu} + \mathbf{S}_\nu = \mathbf{G_\Lambda} + \mathbf{U_\Lambda}. \tag{2.25}$$

The quadrotor acceleration vector $\dot{\nu}$ in the body-frame is then isolated from (2.25), getting

$$\dot{\nu} = \mathbf{M_B}^{-1}(-\mathbf{S}_\nu + \mathbf{G_\Lambda} + \mathbf{U_\Lambda}), \tag{2.26}$$

or expressed as a system of equations

$$\begin{aligned}
\ddot{x} &= \dot{y}\dot{\phi} - \dot{z}\dot{\theta} + g\sin(\theta), \\
\ddot{y} &= \dot{z}\dot{\psi} - \dot{x}\dot{\phi} + g\cos(\theta)\sin(\phi), \\
\ddot{z} &= \dot{x}\dot{\theta} - \dot{y}\dot{\psi} - g\sin(\theta) + \frac{u_1}{m}, \\
\ddot{\psi} &= \dot{\phi}\dot{\theta}\frac{J_{xx} - J_{yy}}{J_{zz}} + \frac{u_2}{J_{zz}}, \\
\ddot{\theta} &= \dot{\phi}\dot{\psi}\frac{J_{zz} - J_{xx}}{J_{yy}} + \frac{u_3}{J_{yy}}, \\
\ddot{\phi} &= \dot{\theta}\dot{\psi}\frac{J_{yy} - J_{zz}}{J_{xx}} + \frac{u_4}{J_{xx}},
\end{aligned} \tag{2.27}$$

with the inputs set as

$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} = \begin{bmatrix} T_u \\ \tau_\psi \\ \tau_\theta \\ \tau_\phi \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ -K_m & K_m & -K_m & K_m \\ 0 & -L & 0 & L \\ -L & 0 & L & 0 \end{bmatrix} \begin{bmatrix} F_{M_1} \\ F_{M_2} \\ F_{M_3} \\ F_{M_4} \end{bmatrix} \tag{2.28}$$

for quadrotors in '+' configuration, and

$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} = \begin{bmatrix} T_u \\ \tau_\psi \\ \tau_\theta \\ \tau_\phi \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ -K_m & K_m & -K_m & K_m \\ -L_X & -L_X & L_X & L_X \\ -L_X & L_X & L_X & -L_X \end{bmatrix} \begin{bmatrix} F_{M_1} \\ F_{M_2} \\ F_{M_3} \\ F_{M_4} \end{bmatrix} \tag{2.29}$$

for quadrotors in 'X' configuration.

Defining the state vector as

$$\mathbf{x} = \begin{bmatrix} x & \dot{x} & y & \dot{y} & z & \dot{z} & \psi & \dot{\psi} & \theta & \dot{\theta} & \phi & \dot{\phi} \end{bmatrix}^T, \tag{2.30}$$

the quadrotor non-linear dynamics model $\mathbf{f}(\mathbf{x}, \mathbf{u})$ got with a Newton-Euler approach is

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}) = \begin{bmatrix} \dot{x} \\ \dot{y}\dot{\phi} - \dot{z}\dot{\theta} + g\sin(\theta) \\ \dot{y} \\ \dot{z}\dot{\psi} - \dot{x}\dot{\phi} + g\cos(\theta)\sin(\phi) \\ \dot{z} \\ \dot{x}\dot{\theta} - \dot{y}\dot{\psi} - g\sin(\theta) + \dfrac{u_1}{m} \\ \dot{\psi} \\ \dot{\phi}\dot{\theta}\dfrac{J_{xx} - J_{yy}}{J_{zz}} + \dfrac{u_2}{J_{zz}} \\ \dot{\theta} \\ \dot{\phi}\dot{\psi}\dfrac{J_{zz} - J_{xx}}{J_{yy}} + \dfrac{u_3}{J_{yy}} \\ \dot{\phi} \\ \dot{\theta}\dot{\psi}\dfrac{J_{yy} - J_{zz}}{J_{xx}} + \dfrac{u_4}{J_{xx}} \end{bmatrix} \tag{2.31}$$

2.2.2 Euler-Lagrange Approach

The general coordinates representing the position and attitude of the quadrotor are defined as

$$\mathbf{\Xi} = \begin{bmatrix} \xi_\mathbf{W} & \mathbf{\Theta_B} \end{bmatrix}^T, \tag{2.32}$$

where $\xi_\mathbf{W} = \begin{bmatrix} X_W & Y_W & Z_W \end{bmatrix}^T$ is the vector representing the position of the CoG of the quadrotor relative to the earth-frame shown in Fig. 2.2 and $\mathbf{\Theta_B} = \begin{bmatrix} \psi & \theta & \phi \end{bmatrix}^T$ represent the quadrotor attitude.

The Lagrangian of the quadrotor is defined by

$$L(\boldsymbol{\Xi}, \dot{\boldsymbol{\Xi}}) = K_{trans} + K_{rot} - E_{pot}, \tag{2.33}$$

where K_{trans} is the quadrotor translational kinetic energy, K_{rot} is the quadrotor rotational kinetic energy, E_{pot} is the quadrotor potential energy, and z is the quadrotor elevation. Hence, the Lagrangian in (2.33) is rewritten as

$$L(\boldsymbol{\Xi}, \dot{\boldsymbol{\Xi}}) = \frac{m}{2}\dot{\xi}_{\mathbf{W}}^{T}\dot{\xi}_{\mathbf{W}} + \frac{1}{2}\dot{\boldsymbol{\Theta}}_{\mathbf{B}}^{T} J \dot{\boldsymbol{\Theta}}_{\mathbf{B}} - mgz. \tag{2.34}$$

The dynamic model of the quadrotor is derived from the Euler-Lagrange equation

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{\boldsymbol{\Xi}}} - \frac{\partial L}{\partial \boldsymbol{\Xi}} = \begin{bmatrix} F_{\xi} \\ \tau \end{bmatrix}, \tag{2.35}$$

where $F_{\xi} = R_b^w \hat{F}_b$ is the translational force applied to the quadrotor by its four motors, and $\tau = \begin{bmatrix} \tau_{\psi} & \tau_{\theta} & \tau_{\phi} \end{bmatrix}^T$.

In the quadrotor body-frame, the translational force $\hat{F}_b$ is only applied in the z-axis, and thus it is represented by

$$\hat{F}_b = \begin{bmatrix} 0 \\ 0 \\ T_u \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \sum_{i=1}^{4} F_{M_i} \end{bmatrix}. \tag{2.36}$$

The system of equations that represent the dynamics of the quadrotor got using the Euler-Lagrange approach (2.35) is

$$\begin{aligned}
\ddot{X}_W &= \frac{u_1}{m}(\cos(\phi)\sin(\theta)\cos(\psi) + \sin(\phi)\sin(\psi)), \\
\ddot{Y}_W &= \frac{u_1}{m}(\cos(\phi)\sin(\theta)\sin(\psi) - \sin(\phi)\cos(\psi)), \\
\ddot{Z}_W &= \frac{u_1}{m}(\cos(\phi)\cos(\theta)) - g, \\
\ddot{\psi} &= \dot{\phi}\dot{\theta}\frac{J_{xx} - J_{yy}}{J_{zz}} + \frac{u_2}{J_{zz}}, \\
\ddot{\theta} &= \dot{\phi}\dot{\psi}\frac{J_{zz} - J_{xx}}{J_{yy}} + \frac{u_3}{J_{yy}}, \\
\ddot{\phi} &= \dot{\theta}\dot{\psi}\frac{J_{yy} - J_{zz}}{J_{xx}} + \frac{u_4}{J_{xx}},
\end{aligned} \tag{2.37}$$

with the inputs set as

$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} = \begin{bmatrix} T_u \\ \tau_\psi \\ \tau_\theta \\ \tau_\phi \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ -K_m & K_m & -K_m & K_m \\ 0 & -L & 0 & L \\ -L & 0 & L & 0 \end{bmatrix} \begin{bmatrix} F_{M_1} \\ F_{M_2} \\ F_{M_3} \\ F_{M_4} \end{bmatrix} \tag{2.38}$$

for quadrotors in '+' configuration, and

$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} = \begin{bmatrix} T_u \\ \tau_\psi \\ \tau_\theta \\ \tau_\phi \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ -K_m & K_m & -K_m & K_m \\ -L_X & -L_X & L_X & L_X \\ -L_X & L_X & L_X & -L_X \end{bmatrix} \begin{bmatrix} F_{M_1} \\ F_{M_2} \\ F_{M_3} \\ F_{M_4} \end{bmatrix} \tag{2.39}$$

for quadrotors in 'X' configuration ([38, 39]).

Defining the state vector as

$$\mathbf{x} = \begin{bmatrix} X_W & \dot{X_W} & Y & \dot{Y_W} & Z_W & \dot{Z_W} & \psi & \dot{\psi} & \theta & \dot{\theta} & \phi & \dot{\phi} \end{bmatrix}^T, \tag{2.40}$$

the quadrotor non-linear dynamics model got with an Euler-Lagrange approach, can be represented as

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}) = \begin{bmatrix} \dot{X}_W \\ \frac{u_1}{m}(\cos(\phi)\sin(\theta)\cos(\psi) + \sin(\phi)\sin(\psi)) \\ \dot{Y}_W \\ \frac{u_1}{m}(\cos(\phi)\sin(\theta)\sin(\psi) - \sin(\phi)\cos(\psi)) \\ \dot{Z}_W \\ \frac{u_1}{m}(\cos(\phi)\cos(\theta)) - g \\ \dot{\psi} \\ \dot{\phi}\dot{\theta}\frac{J_{xx} - J_{yy}}{J_{zz}} + \frac{u_2}{J_{zz}} \\ \dot{\theta} \\ \dot{\phi}\dot{\psi}\frac{J_{zz} - J_{xx}}{J_{yy}} + \frac{u_3}{J_{yy}} \\ \dot{\phi} \\ \dot{\theta}\dot{\psi}\frac{J_{yy} - J_{zz}}{J_{xx}} + \frac{u_4}{J_{xx}} \end{bmatrix} \tag{2.41}$$

2.3 Linearized Model

2.3.1 Jacobian Linearization

The linearization of a non-linear system is done about an equilibrium point $\overline{\mathbf{x}}$ achieved with a specific input called equilibrium input $\overline{\mathbf{u}}$ where $\mathbf{f}(\overline{\mathbf{x}}, \overline{\mathbf{u}}) \approx \mathbf{0}$. In the quadrotor, such an equilibrium point can be the hover state where $\mathbf{\Omega_B} \to \mathbf{0_{3\times 1}}$, $\dot{\mathbf{\Omega}}_\mathbf{B} \to \mathbf{0_{3\times 1}}$, and $\mathbf{V_B} \to \mathbf{0_{3\times 1}}$. This is

$$\begin{aligned}\overline{\mathbf{x}} &= \begin{bmatrix}\overline{x} & \dot{\overline{x}} & \overline{y} & \dot{\overline{y}} & \overline{z} & \dot{\overline{z}} & \overline{\psi} & \dot{\overline{\psi}} & \overline{\theta} & \dot{\overline{\theta}} & \overline{\phi} & \dot{\overline{\phi}}\end{bmatrix}^T \\ &= \begin{bmatrix}\overline{x} & 0 & \overline{y} & 0 & \overline{z} & 0 & 0 & 0 & 0 & 0 & 0 & 0\end{bmatrix}^T,\end{aligned} \tag{2.42}$$

where $\overline{x}$, $\overline{y}$ and $\overline{z}$ define a constant desired position in the body and earth-frame. When the quadrotor state is near the equilibrium point, the body and earth frames are assumed to coincide and thus $\dot{\xi}_\mathbf{W} \to \mathbf{V_B}$, $X_W \to x$, $Y_W \to y$, and $Z_W \to z$ ([40]).

The equilibrium point shown in 2.42 is obtained using a constant input value $\overline{\mathbf{u}}$ where only the thrust that overcomes gravity is applied, as shown below.

$$\overline{\mathbf{u}} = \begin{bmatrix}T_u & \tau_\psi & \tau_\theta & \tau_\phi\end{bmatrix}^T = \begin{bmatrix}mg & 0 & 0 & 0\end{bmatrix}^T \tag{2.43}$$

The Jacobian linearization is based on the fact that if the quadrotor is not exactly at the equilibrium point, but close to it with a small deviation $\delta_\mathbf{x} = \mathbf{x} - \overline{\mathbf{x}}$ due to an input deviation $\delta_\mathbf{u} = \mathbf{u} - \overline{\mathbf{u}}$ ([41]), the non-linear system will be represented by

$$\dot{\delta_\mathbf{x}} = \mathbf{f}(\overline{\mathbf{x}} + \delta_\mathbf{x}, \overline{\mathbf{u}} + \delta_\mathbf{u}). \tag{2.44}$$

Using the first order Taylor polynomial from (2.44) and considering that $\mathbf{f}(\overline{\mathbf{x}}, \overline{\mathbf{u}}) \approx \mathbf{0}$, the following expression is obtained.

$$\dot{\delta_\mathbf{x}} \approx \frac{\partial f(\mathbf{x}, \mathbf{u})}{\partial \mathbf{x}}\Big|_{\substack{\mathbf{x}=\overline{\mathbf{x}} \\ \mathbf{u}=\overline{\mathbf{u}}}} \delta_\mathbf{x} + \frac{\partial f(\mathbf{x}, \mathbf{u})}{\partial \mathbf{u}}\Big|_{\substack{\mathbf{x}=\overline{\mathbf{x}} \\ \mathbf{u}=\overline{\mathbf{u}}}} \delta_\mathbf{u} \tag{2.45}$$

The equation (2.45) describes a linear time-invariant representation of the non-linear dynamics of the quadrotor near an equilibrium point $\overline{\mathbf{x}}$ and with an input that tends to be $\overline{\mathbf{u}}$, which is established as a state space model

$$\dot{\mathbf{x}}(t) = A\mathbf{x}(t) + B\mathbf{u}(t) \tag{2.46}$$

where

$$\mathbf{x} = \begin{bmatrix} x & \dot{x} & y & \dot{y} & z & \dot{z} & \psi & \dot{\psi} & \theta & \dot{\theta} & \phi & \dot{\phi} \end{bmatrix}^T$$

$$A = \frac{\partial f(\mathbf{x}, \mathbf{u})}{\partial \mathbf{x}}\Big|_{\substack{\mathbf{x}=\bar{\mathbf{x}} \\ \mathbf{u}=\bar{\mathbf{u}}}} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & g & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & g & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \tag{2.47}$$

$$B = \frac{\partial f(\mathbf{x}, \mathbf{u})}{\partial \mathbf{u}}\Big|_{\substack{\mathbf{x}=\bar{\mathbf{x}} \\ \mathbf{u}=\bar{\mathbf{u}}}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & \frac{1}{m} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{J_{zz}} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{J_{yy}} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{J_{xx}} \end{bmatrix}^T.$$

2.3.2 Thrust Compensation

Given the fact that the forces F_{M_i} applied by the motors in the quadrotor are parallel to the body-frame z-axis and not to the earth-frame Z_W-axis, a compensated thrust input u_1^* must be set so that its projection u_1 (third component of F_ξ) on the Z_W-axis is such that it allows keeping the inputs of the system close to the equilibrium input, even though there are deviations in θ or ϕ.

Given the linearization consideration where the z and Z_W axes coincide, if a linear controller is designed to control the quadrotor, it takes the projection u_1 of $u_1^* = T_u$ on the Z_W-axis as its thrust control signal. From linear algebra, and following the geometry seen in Fig. 2.2, the projection u_1 is defined as

$$u_1 = u_1^* \cos(\theta) \cos(\phi), \tag{2.48}$$

for any $|\theta|$ and $|\phi|$ magnitudes greater than zero. Thus, after each iteration of the control algorithm the real thrust input u_1^* is calculated as

$$u_1^* = \frac{u_1}{\cos(\theta)\cos(\phi)}, \tag{2.49}$$

keeping the desired vertical thrust, despite any deviation about the x or y axes.

For simplicity, the rest of the document assumes that $u_1 = u_1^*$, however during the real implementation the thrust compensation is taken into account.

2.4 Conclusions

This chapter described the dynamic model of the quadrotor system, based on its reference frame and geometry configuration. The '+' and 'X' quadrotor configurations are detailed, including the setting of the quadrotor inputs and its dependence on the quadrotor motors forces. The dynamic model is obtained using the Newton-Euler approach, for a body-frame base model, and an Euler-Lagrange approach, for a hybrid earth-frame-position/body-frame-attitude model. In order to design linear controllers for the flight dynamics of the quadrotor, a Jacobian linearization about an equilibrium point is implemented. This linearized model is valid for low translational and rotation speed flights with small angular deviations, where both non-linear models, got from different approaches, tend to be equivalent. This equilibrium point is achieved by an equilibrium input, which must be added to any control action set by the controller during its implementation. Finally, it is considered the decompensation of the force set by the controller to be exerted by the motors on the earth-frame Z_W-axis, so that it can be compensated in the real quadrotor implementation.

Chapter 3

Smartphone-based Quadrotor Prototype

In this chapter, the quadrotor prototype is presented. All the components that are used to build the prototype are described with the purpose of detailing the function that each component fulfils within the quadrotor. Furthermore, the specific parameters of the built quadrotor are shown and the procedure carried out to find them experimentally is explained. This is of great importance for the design of the controllers that allow the flight of the built system.

In Fig. 3.1, a small overview of the hardware related to the electrical signals within the quadrotor is presented. This overview shows the interrelation between the main components detailed below.

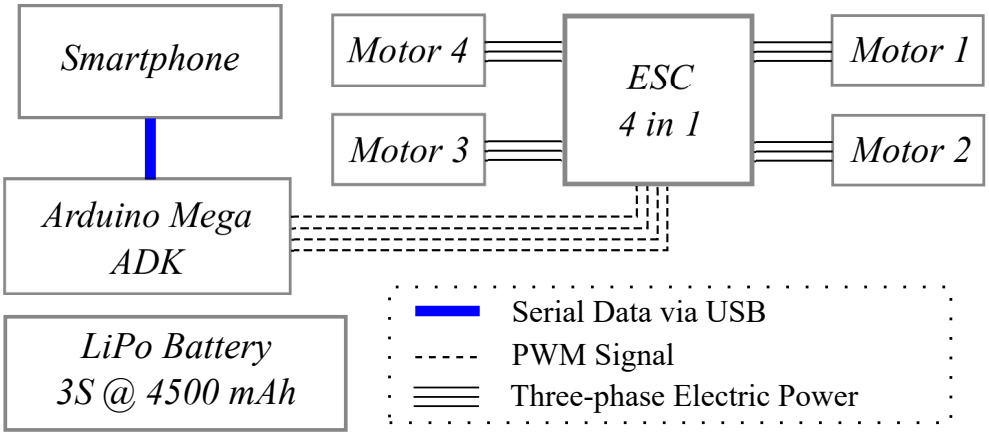


Figure 3.1: Quadrotor prototype's hardware overview

3.1 Quadrotor Components

The hardware used to build the quadrotor prototype is detailed in this section. The main goal of this research project is to control a quadrotor using just a smartphone on-board. This smartphone executes the state estimation algorithms and defines the controls signals that will command the actuators. The control signals are processed in the smartphone and then sent, through a gateway, to the Electronic Speed Controllers (ESCs) that will set the angular speed of the motors depending on the control signal they receive.

3.1.1 Frame

The quadrotor's frame is responsible for carrying all the other components needed in the system. Additionally, it must be able to support the weight of any possible payload that the quadrotor will carry. The multirotors frames are usually built using fiberglass or carbon fiber as main material, being the carbon fiber frames stiffer and lighter than equally built fiberglass frames.

For this project, the requirement to build a quadrotor between 0.20 and 0.30 cm in radius was defined. This requirement is based on the fact that the quadrotor must carry a smartphone and possibly some mission-related payloads, which weight can be greater than 100 g. Taking that requirement and the literature review into account, in addition to the weight and stiffness advantages of the carbon fiber, a LJI 500-X4 carbon fiber frame was selected.

Figure 3.2: LJI 500-X4 carbon fiber frame[1]

This frame (Fig. 3.2) has a weight of 431 g and a radius L of 0.244 m measured between the frame center and each of the four rotor axis holes. Its landing gear

[1]LJI 500-X4 frame image taken from https://goo.gl/hHfHQR

has a height of 0.170 m, allowing the payload to be placed in the lower part of the quadrotor.

3.1.2 Smartphone

In this project, the smartphone takes the place of the quadrotor's flight controller. The basic instrumentation needed for a flight controller includes a triaxial accelerometer, a triaxial gyroscope, a triaxial magnetometer, a barometer and a GNSS receiver that can process signals from the GPS and GLONASS satellites constellations. On the other side, the processor of a flight controller must be powerful enough to execute the control and estimation algorithms within the sample time of the control system.

The LG Nexus 5X, shown in Fig. 3.3, is a smartphone developed by Google and assembled by LG, released in 2015 with the Android 6.0.1 operating system. This phone has a Hexa-core Qualcomm MSM8992 Snapdragon 808 CPU that includes four Cortex-A53 and two Cortex-A57 cores with 1.4 GHz and 1.8 GHz clock rate respectively and an Adreno 418 GPU. Its features also include 2 GB of RAM memory, 32 GB of Flash memory, total mass of 136 g and a 12.3 MP camera.

Figure 3.3: LG Nexus 5X, smartphone used as flight controller[2]

This smartphone features all the needed instrumentation for a flight controller, specified previously. The maximum sample rates of the instrumentation contained in the LG Nexus 5X are described in Table 3.1.

[2]LG Nexus 5X image taken from https://goo.gl/RxyDJd

Table 3.1: Sample Rates of the Sensors in the Smartphone

Sensor	Sample rate $[Hz]$
Triaxis accelerometer	400
Triaxis gyroscope	200
Triaxis magnetometer	50
Barometer	10
GNSS	1

The LG Nexus 5X smartphone was selected in this project mainly due to its computational and instrumentation capabilities in addition to its communication interfaces (Bluetooth 4.0, Wireless LAN, USB Type-C and GSM). Its powerful CPU handling a GPU enables the possibility of developments using the camera for visual odometry, and controllers whose execution represents a high computational load, which exceeds the limits of a standard microcontroller such as the ATmega 2560.

As the quadrotor's flight controller, the smartphone receives remote orders from the quadrotor's pilot using wireless communication, while executing the sensorial, estimation and control algorithms. After the control signals are calculated, they are sent to the actuators as a serial data frame using the USB interface between the smartphone and a gateway that converts the serial data into PWM signals.

3.1.3 Motors and Electronic Speed Controllers

The actuation system of the quadrotor is composed by R/C-type brushless motors, propellers and ESCs. The LJI 500-X4 frame manufacturer recommends using motors with a RPM constant of 810 KV, which means that these motors can rotate at a speed of 810 RPM for each Volt applied to it. This kind of motors are used in medium to big sized multirotors due to its thrust efficiency and thrust capacity.

Following this recommendation, the EMAX MT2216II motors were selected. These motors, shown in Fig. 3.4a, are labelled by its manufacturer as 810 KV motors with a maximum current consumption of 9.8 A and a maximum thrust of 6.6 N, when using a 10-inch propeller.

(a) EMAX MT2216II motor

(b) EMAX 4in1 ESC

Figure 3.4: Motors and ESC used in the Quadrotor[3]

The quadrotor's brushless motors are powered using a three-phase electric signal generated by the ESC. Each ESC sets one motor rotational velocity depending on a PWM signal input, and must be able to handle the maximum current consumption of the motor. Mainly due to the ease of handling and positioning within the frame, the EMAX 4in1 ESC was selected for this project (Fig. 3.4b). This 4in1 ESC contains four ESCs with a maximum current supply of 30 A each. It also has a DC-DC converter that outputs an isolated 5 V DC supply for any additional electronics. The power supply of the ESC is fully delivered by the quadrotor's battery.

3.1.4 Smartphone-to-ESC Gateway

As the smartphone can not generate any PWM signal and each ESC needs a PWM signal input to set the rotational velocity of a motor, a gateway between the smartphone and the ESCs must be used. In order to avoid interference in the electromagnetic spectrum and delays that could lead the control system to instability while using wireless communications, the smartphone wired serial bus (USB interface) is defined as the only channel of communication through which the control signals are sent to the gateway.

The Arduino Mega ADK, is shown in Fig. 3.5, is a development board based on the Atmel 8-bit AVR RISC-based ATmega2560 microcontroller. This board supports the Android Open Accesory (AOA) protocol, which allows external hardware to exchange data with Android devices.

[3]Taken from https://goo.gl/6qD6Qg and https://goo.gl/fDHiUp

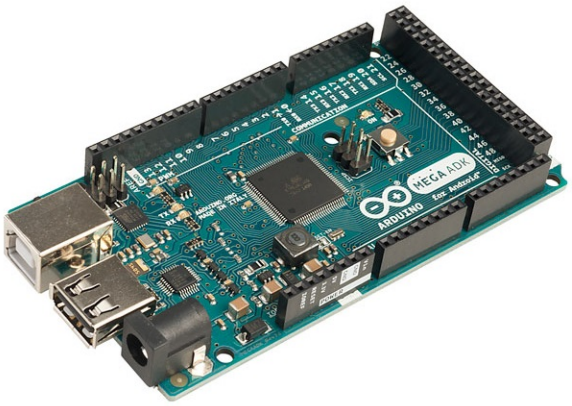


Figure 3.5: Arduino Mega ADK[4]

Although there are other boards that support the AOA protocol, as the IOIO board, the Arduino Mega ADK offers the possibility of executing tasks in the ATmega2560 microcontroller without any intervention of the Android device. This feature is very useful in case of future developments that include additional hardware to the smartphone-based quadrotor.

The Arduino Mega ADK is selected as the smartphone-to-ESC gateway. The workflow of this board is set as follows: the microcontroller receives a serial data frame from the smartphone through the USB port, each of the four PWM signal widths is extracted from the data frame, the PWM signals are sent to the ESCs, finally the cycle starts again. This workflow is executed each 2 ms and if the gateway does not receive any updated data frame from the smartphone, it keeps the last PWM signals set.

3.1.5 Battery

The quadrotor's battery must supply power to all the active components in the quadrotor. The maximum power consumption of these components, based on the nominal voltage value of a 3-cells lithium-ion polymer (LiPo) battery (11.1 V), is shown below.
The maximum current consumption in the quadrotor reaches 42.08 A, so the battery must be able to deliver current amplitudes greater than that value. A Floureon 3-cells LiPo battery with a nominal capacity of 4500 mAh and a nominal voltage of 11.1 V, was selected for this prototype and is shown in Fig. 3.6. This battery can deliver up to 30 times its nominal current, this is 135 A, continuously.

[4]Arduino Mega ADK image taken from https://goo.gl/ejeQeX

Table 3.2: Maximum Power Consumption of the Quadrotor's Components

Component	Voltage $[V]$	Max. Current $[A]$	Max. Power $[W]$
Smartphone	5	0.75	3.75
Arduino Mega ADK	11.1	0.75	8.33
4 x ESCs	11.1	1.38	15.32
4 x Motors	11.1	39.2	435.12

Figure 3.6: LiPo battery that powers the Quadrotor[5]

3.1.6 3D-printed Parts

The smartphone, the Arduino Mega ADK and the 4in1 ESC need to be easily placed and protected when installed on the frame. For that reason, multiple support components and one case were designed and 3D-printed using polylactic acid (PLA) filament. These 3D-printed objects are detailed below.

Smartphone Support

In order to place the smartphone near the *CoG* in the quadrotor and enable the possibility of capturing nadir photos or videos, the smartphone support is designed in such a way that it can be placed under the quadrotor's *CoG* but above the landing gear of the frame. Additionally, the support includes a free area that allows to use the complete field of view of the camera without being obstructed by it. The smartphone support is shown in Fig. 3.7.

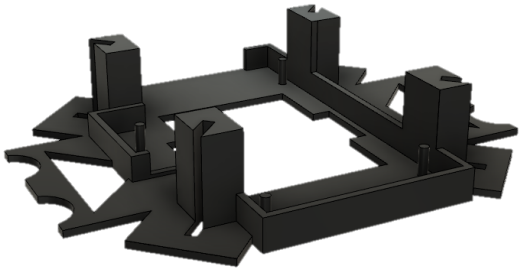
Arduino Mega ADK and ESC supports

The Arduino Mega ADK and the Emax 4in1 ESC are placed on the frame, above the quadrotor's *CoG*. Given the limited space available to locate these components

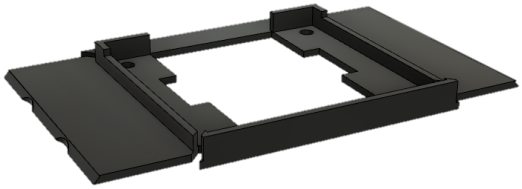
[5]Floureon 3S LiPo battery image taken from https://goo.gl/anC9M2

Figure 3.7: Smartphone support

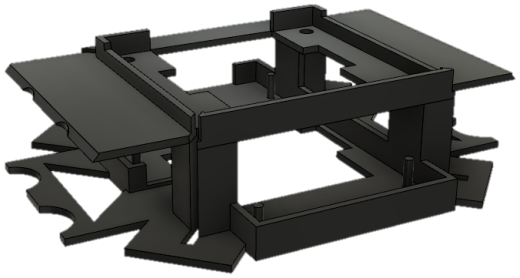
on the top of the frame, their supports are designed to be placed one on top of the other, as shown in Fig. 3.8.



(a) Arduino Mega ADK support



(b) ESCs support



(c) Arrangement of the Gateway and ESCs Support as a Whole

Figure 3.8: Arduino Mega ADK and EMAX 4in1 ESCs designed supports

Dome

To protect the Arduino Mega ADK and the ESCs, a dome that covers and encloses them is designed. This dome, shown in Fig. 3.9, totally encloses the Mega ADK and

ESCs supports restricting their movement and protecting them in case of a shock or hit.

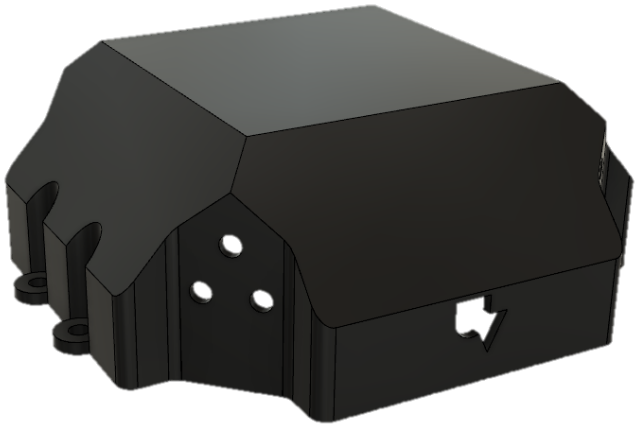


Figure 3.9: 3D Designed Dome

3.1.7 Assembled Smartphone-based Quadrotor

The smartphone-based quadrotor prototype is assembled using all the components exposed in this section, and is shown in Fig. 3.10.

Figure 3.10: Assembled Smartphone-based Quadrotor Prototype

3.2 Quadrotor Parameters

In this section, the quadrotor parameters are detailed. These parameters define the specific dynamic model for the smartphone-based quadrotor prototype and the conversion of control signals to correctly set the motors thrust.

3.2.1 Mass

The mass of each quadrotor's component is measured using a kitchen scale that has an uncertainty of ± 1 g. The results are shown in Table 3.3.

The total mass of the quadrotor m is then calculated as the sum of the weight of each of the components in the quadrotor, being $m = 1.568$ kg.

Table 3.3: Mass values of all the Quadrotor's components

Component	Mass $[kg]$
Smartphone	0.136
Frame	0.431
Battery	0.351
ESC	0.110
Arduino Mega ADK	0.330
Motors (4)	0.140
Propellers (4)	0.380
Smartphone support	0.105
ADK support	0.590
ESC support	0.350
Dome	0.130
Total	1.568

3.2.2 Moments of Inertia

As exposed by [42], in a quadrotor, the moment of inertia matrix $\mathbf{J}$ is set as

$$\mathbf{J} = \begin{bmatrix} J_{xx} & -J_{xy} & -J_{xz} \\ -J_{yx} & J_{yy} & -J_{yz} \\ -J_{zx} & -J_{zy} & J_{zz} \end{bmatrix}. \tag{3.1}$$

Taking into account that the symmetry of the quadrotor with respect to the x and y axes is assumed, the $\mathbf{J}$ matrix can be approximated to

$$\mathbf{J} \approx \begin{bmatrix} J_{xx} & 0 & 0 \\ 0 & J_{yy} & 0 \\ 0 & 0 & J_{zz} \end{bmatrix}. \tag{3.2}$$

The J_{xx}, J_{yy} and J_{zz} values can be obtained using multiple methods including: using a CAD model of the quadrotor and obtaining the inertia values from a 3D design software ([43]), approximating the shape of the quadrotor components to cylinders, cubes and other basic shapes to simplify the mathematical calculation of inertia ([44]), and developing the bifilar pendulum experiment on the quadrotor ([45]). For this project, it was decided to obtain the quadrotor parameters experimentally, so the bifiliar pendulum experiment was developed.

In the bifilar pendulum experiment, an object is hung from two parallel ropes of length r and separated by a distance $2l$, being allowed to rotate freely around a the

axis that is parallel to the ropes. In Fig. 3.11, the geometry of the experiment to get J_{zz}, is shown. For the J_{xx} and J_{yy} inertias experiment, it is necessary to place the quadrotor hanging in such a way that the x and y axes are pointing paraller to the ropes, respectively.

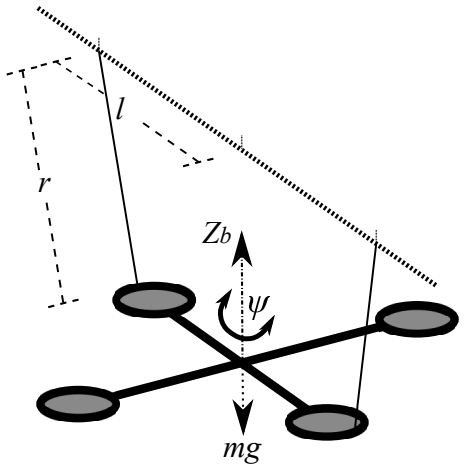


Figure 3.11: Bifilar pendulum experiment geometry for inertia identification

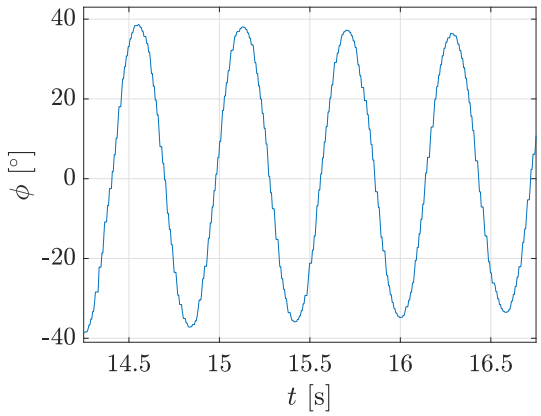
Once the quadrotor is hanging from the ropes, a small torque is applied manually to the quadrotor making it rotate about the vertical axis. The tension in the ropes makes the quadrotor swing, with a period of T_{osc} $[s]$, about the rotation axis.

The moment of inertia is then calculated using the bifiliar equation

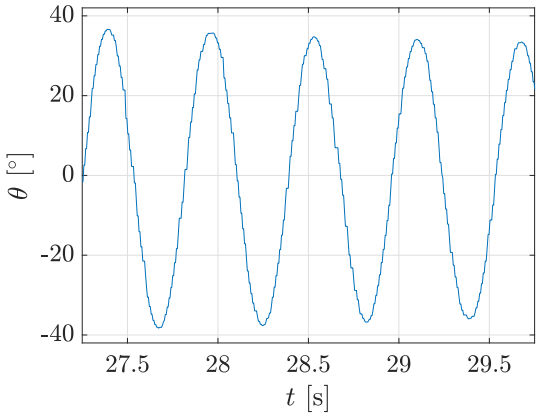
$$J.. = \frac{mgT_{osc}^2 l^2}{4\pi^2 r} \; [kg \cdot m^2], \tag{3.3}$$

where $m = 1.568\ kg$ is the quadrotor's total mass and $g = 9.807\ m/s^2$ is the gravity acceleration magnitude ([46]).

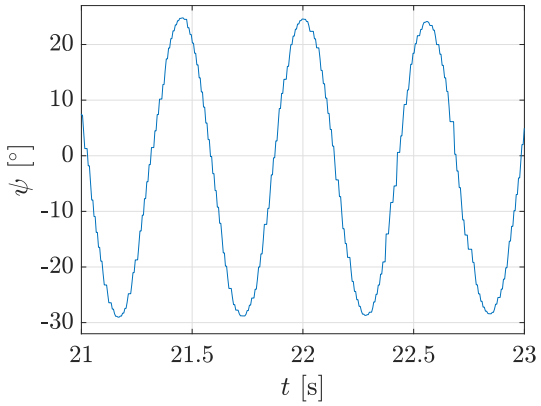
In Fig. 3.12, it is shown the excursion of the rotation angles, ϕ, θ and ψ, about the x, y and z axes respectively, while performing the bifilar pendulum experiment separately.



(a) Rotation about x axis, J_{xx} experiment



(b) Rotation about y axis, J_{yy} experiment



(c) Rotation about z axis, J_{zz} experiment

Figure 3.12: Rotation about x, y and z axes during the bifilar pendulum experiments

The resulting data got after the execution of the experiment around the three components of the quadrotor body frame, are shown in Table 3.4.

Table 3.4: Bifilar pendulum experiment results

Rotation Axis	r $[m]$	l $[m]$	T_{osc} $[s]$	Inertia value $[kg \cdot m^2]$
x	1.18	0.173	1.168	$J_{xx} = 0.0135$
y	1.10	0.173	1.080	$J_{yy} = 0.0124$
z	1.025	0.265	1.122	$J_{zz} = 0.0336$

3.2.3 Motors Thrust

As seen in Section 3.1, the motors rotational velocity ω_i is set by the ESC, which receives a PWM signal input. However, the inputs of the quadrotor (T_u, τ_ψ, τ_θ, and τ_ϕ) depend on the thrust force F_{M_i} applied by each quadrotor motor, as exposed in Section 2.1.

In order to correctly set the desired force F_{M_i} in each motor during a flight, it is necessary to characterize the motor; and thus know how the force applied by the motor behaves with respect to the input PWM signal. This characterization is carried out by means of a thrust test.

In the thrust test, the motor and its corresponding propeller are fixed pointing up over a calibrated scale, as shown in Fig. 3.13, while connected to the ESC, and then the PWM signal is increased with steps of 10, with 0 being the minimum width of PWM signal and 255 the maximum, until reaching the maximum PWM width.

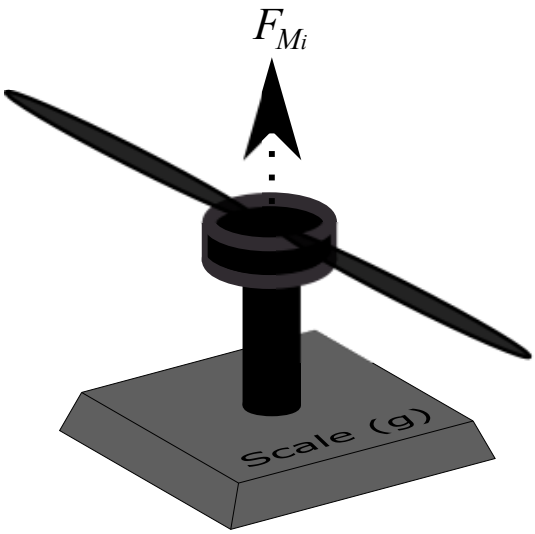
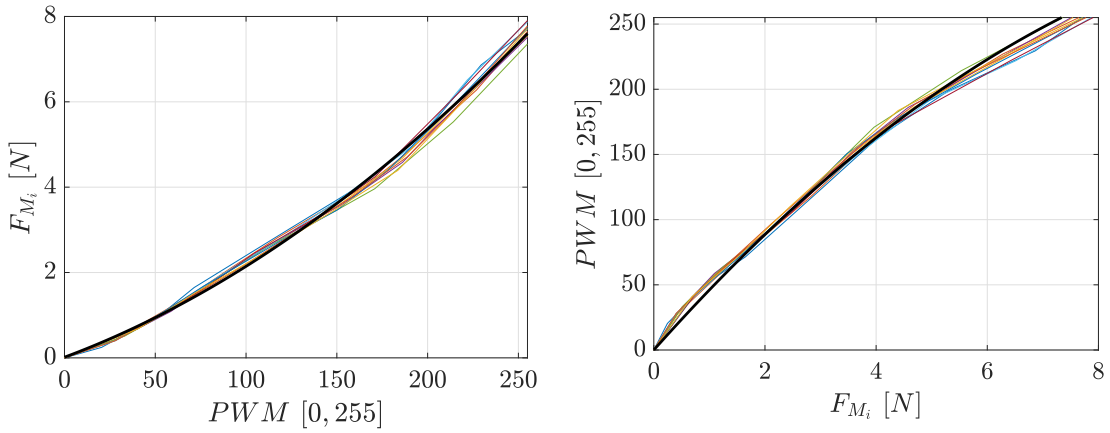


Figure 3.13: Thrust test configuration

With each PWM signal width increment, a scale reading is made. Since the readings m_s are obtained in units of mass (kg), the F_{M_i} must be calculated as

$$F_{M_i} = m_s g \ [N]. \tag{3.4}$$

This test was developed twice with each motor and its results are shown in Fig. 3.14a.



(a) Motors characterization

(b) Inverse motors characterization

Figure 3.14: Motors thrust test results

In Fig. 3.14b, the inverse characterization is exposed. In this case, the PWM signal is defined as the dependent variable, and F_{M_i} as independent. This inverse characterization defines the PWM width setting that is sent from the smartphone to the Arduino Mega ADK.

The trend equations resulting from the thrust test are

$$F_{M_i} = (5.441 \times 10^{-5})(PWM)^2 + 0.01586(PWM) + 0.014808, \tag{3.5}$$

$$PWM = -1.983F_{M_i}^2 + 47.84F_{M_i} + 3.835, \tag{3.6}$$

where F_{M_i} is given in N, and PWM is a value between 0 and 255.

3.2.4 Motors Torque

The quadrotor suffers the application of a torque τ_{M_i} around the z axis when a motor M_i rotates. This torque affects the ψ angle indirectly by adding to the τ_ψ torque proportionally to the thrust force F_{M_i}, being $\tau_{M_i} = K_M F_{M_i}$, as shown in (2.3).

The torque τ_{M_i} is generated due to the conservation of momentum, and its unbalance is used to rotate the quadrotor about the z axis in the opposite direction to the rotation of the motor that generates it as exposed in Section 2.1. In order to properly control the mentioned unbalance, it is necessary to find the value of the constant K_m.

The constant K_m can be found through a steady-state torque experiment, as stated by [47]. In this experiment, the z axis in the quadrotor is located parallel to the ground, leaving the rotation about this axis as the only unblocked DoF. Using the

geometry seen in Fig. 3.15, it is necessary to measure the force F_s that is generated when the motor M_i is rotating.

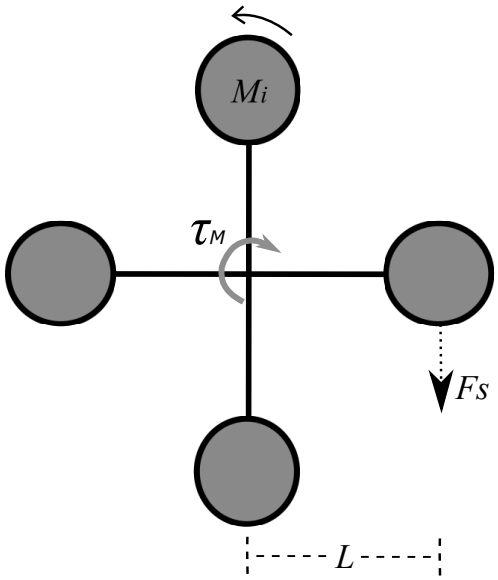


Figure 3.15: Motors torque experiment configuration

Taking into account that when the motor rotates clockwise, the quadrotor will tend to rotate counter-clockwise and vice versa, a scale is located to indirectly obtain the generated force F_s, using the reading of the weight m_s as $F_s = m_s|\vec{g}|\ [N]$. Using (3.5) and the PWM width steps used in the motors thrust experiment, multiple F_{M_i}-dependent F_s readings are obtained. The torques τ_M are then calculated by

$$\tau_M = F_s L\ [N \cdot m], \tag{3.7}$$

and their results are shown in Fig. 3.16.

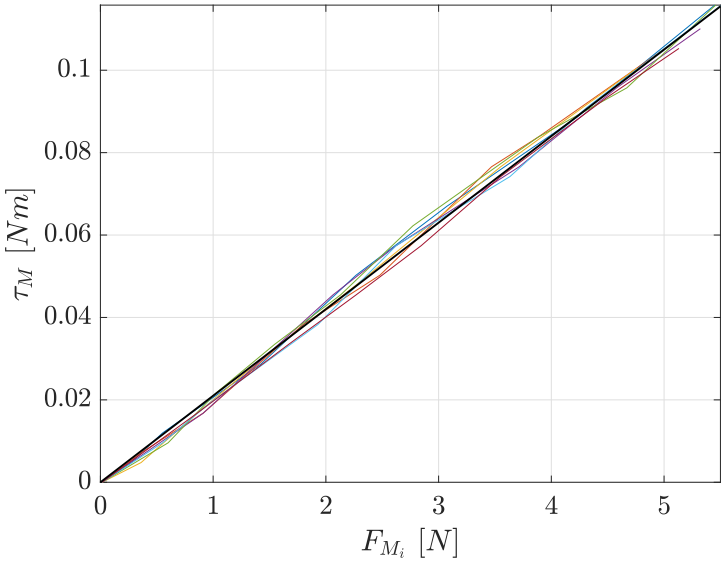


Figure 3.16: Motors torque experiment results

The relation between τ_M and F_{M_i} is linear as expected, and its slope defines the variable K_m as

$$K_m = 0.021\ [m]. \tag{3.8}$$

3.3 Conclusions

This chapter presented the physical composition and parametrization of the smartphone-based quadrotor prototype. In the first part, all the components of the quadrotor are detailed. A medium-size carbon fiber frame is used to support the complete quadrotor system. The LG Nexus 5X is selected as the smartphone where the control and estimation algorithms will be executed, which sends the control signals to an Arduino Mega ADK in order to generate PWM signals that set the motors rotational velocities through the ESC. These components are supported and protected by 3D-printed objects that are attached to the carbon fiber frame. On the other hand, the system is parameterized in its entirety using experimental methods. Here, the total mass and moments of inertia of the system are defined using a scale and the bifilar pendulum experiment respectively, while the motor's thrust and torque about the z-axis are found applying multiple PWM signals to the ESCs and obtaining the corresponding variable with the help of the scale.

Chapter 4

Control Strategies and State Estimation

rtrtrtrtr

4.1 Controllability and Observability

The controllability and observability features of the linearized system are checked using the controllability and observability Grammians W_c and W_o defined as

$$W_c = \int_0^\infty e^{At} B B^T e^{A^T t} dt, \tag{4.1}$$

$$W_o = \int_0^\infty e^{A^T t} C^T C e^{At} dt. \tag{4.2}$$

As the matrices W_c and W_o are positive definite matrices, then the system is both controllable and observable [48].

4.2 Control Strategies

This section exposes the controllers design procedure. Here, the mathematical procedure to design a linear quadratic Gaussian controller is described. It includes a regulator and an estimator, in addition to a gain compensator, allowing the system to track a trajectory. The design process of a H_∞ controller is shown, taking into account some weighting sensitivities.

4.2.1 Linear Quadratic Regulator

Designing an infinite-time regulator, a linear quadratic Gaussian (LQG) controller is set while looking for the minimization of the cost function V as

$$V = \int_0^\infty (X^T Q X + U^T R U) dt. \tag{4.3}$$

This minimization is achieved using Q and R as penalization matrices for the states and the inputs respectively, and the optimal input U^* is found as

$$\begin{aligned} U^* &= FX, \\ F &= -R^{-1} B^T P, \end{aligned} \tag{4.4}$$

such that P is a solution of the Algebraic Riccati Equation [49]

$$0 = PA + A^T P + Q - PBR^{-1}B^T P. \tag{4.5}$$

For this quadrotor controller, the tuning matrices Q and R were set as $Q = C^T C$ and $R = 0.7 * \mathcal{I}_{4\times4}$.

To find the linear quadratic estimator (LQE) gain F_e it is used the duality principle and then, F_e is defined as

$$F_e = -P_e C^T R_e^{-1} \tag{4.6}$$

such that P_e is a solution of the Filter Algebraic Riccati Equation

$$0 = P_e A^T + A P_e + Q_e - P_e C^T R_e^{-1} C P_e, \tag{4.7}$$

setting the tuning matrices as $Q_e = BB^T$ and $R_e = 0.9 * \mathcal{I}_{4\times4}$.

Thus, the LQG controller will be a mixture between the LQR and LQE where the closed-loop system is defined as

$$\begin{aligned} \dot{X} &= (A + BF + F_e C)X + BU, \\ Y &= CX, \end{aligned} \tag{4.8}$$

and is shown in Fig. 4.1.
As the LQR is not made for reference tracking, to let the quadrotor track a defined trajectory, it is necessary to include a gain filter v for the reference that compensates the total gain of the closed-loop system and enables the tracking of a reference input r. This gain filter is defined as

$$v = -(C * (A + BF + F_e C)^{-1} * B)^{-1}, \tag{4.9}$$

and its placing in the system is shown in Fig. 4.2.
where G_{cl} is the closed-system with the LQG controller described in (4.8).

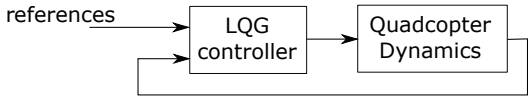


Figure 4.1: Closed-loop of the controlled system with an LQG controller.

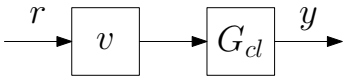


Figure 4.2: Closed-loop system with gain compensation for the LQG controller aiming to track a reference.

4.2.2 H_∞ Controller

For the design of the $H\infty$ controller, a generalized plant like the shown in the Fig. 4.3, was built.

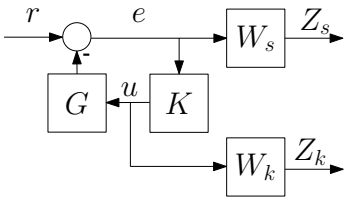


Figure 4.3: Generalized plant with the weighting filters W_s and W_k.

In this generalized plant, G represents the quadrotor dynamics, K the controller, r is the system reference, e is the error, u is the control input and W_s, W_k are weighting filters that must satisfy

$$\gamma = \left|\left| \begin{bmatrix} W_s S \\ W_k K S \end{bmatrix} \right|\right|_\infty < 1, \tag{4.10}$$

where S is the sensitivity function and KS is the control sensitivity defined as

$$S = (\mathcal{I} + GK)^{-1}, \quad KS = K(\mathcal{I} + GK)^{-1}. \tag{4.11}$$

The weighting filters for the sensitivity and the control sensitivity were chosen as

$$\begin{aligned} W_S &= \frac{w_s/M_s}{s + w_s} * \mathcal{I}_{4\times4} \ , \\ W_K &= \frac{c}{M_k} \frac{s + w_k}{s + c w_k} * \mathcal{I}_{4\times4} \ , \end{aligned} \tag{4.12}$$

where $w_s = 10^{-4}$, $M_s = 10^{-4}$, $w_k = 20$, $M_k = 20$, and $c = 10^3$. With these weighting filters, the value of γ is greater than one, and then it is necessary to rebuild the generalized plant using normalized weighting filters.

The H_∞ controller can be simulated as shown in Fig. 4.4.

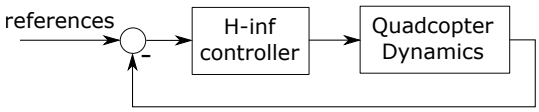


Figure 4.4: Closed-loop of the controlled system with an H_∞ controller.

H_∞ Controller Order Reduction

The designed H_∞ controller for our quadrotor, will have 4 outputs (the quadrotor has 4 control inputs), 4 inputs (the quadrotor has 4 measured outputs) and 20 states (due to the 12 order plant, 4 order W_k and 4 order W_s). To be able to implement this controller in a smartphone running Android, it it necessary to find a reduced order controller that behaves similarly to the full order controller K. To reduce the order of the controller, the singular values of the Hankel matrix

$$H_k = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{k-1} \end{bmatrix} \begin{bmatrix} B \\ AB \\ \vdots \\ A^{k-1}B \end{bmatrix}^T, \tag{4.13}$$

are analysed. The energy of the Hankel singular values is shown in Fig. 4.5.

As shown in Fig. 4.5, the last ordered four states have unnoticeable energy when it is plotted; that means that these four states can be truncated from the controller without modifying its dynamics [50].

4.3 Controllers Design

fhfgfhfh

4.3.1 Stabilize Mode

fghgfh

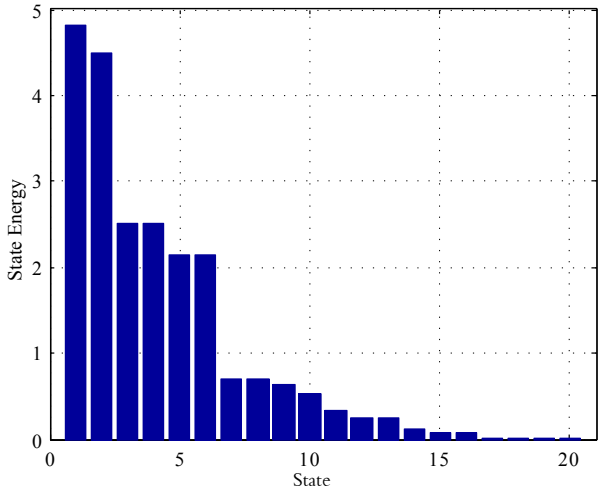


Figure 4.5: Hankel singular values energy histogram of the designed controller.

Dynamic Model

rtrterere

Linear Quadratic Regulator

rtrterere

H_∞ Controller

rtrtererre

4.3.2 Altitude Hold Mode

Dynamic Model

rtrterere

Linear Quadratic Regulator

rtrterere

H_∞ Controller

rtrtererre

4.3.3 GNSS Dependent Flight Modes

Dynamic Model

rtrterere

Linear Quadratic Regulator

rtrterere

H_∞ Controller

rtrtererre

4.4 State Estimation Through Kalman Filter

The quadrotor dynamics are sensed exclusively using the on-board smartphone's sensors. Considering that these sensors have different sample frequencies, and poor accuracy, it is necessary to use estimation algorithms, as a Kalman filter, to get reliable state data with constant sample frequency and improved accuracy and precision when compared with the raw data from the sensors.

4.4.1 Attitude Estimation

The Android API implements a Kalman filter for attitude estimation using the raw data deliverd by the smartphone's accelerometer, gyroscope and magnetometer, as exposed in [51]. Using the quaternion Q_s delivered by the Rotation virtual sensor included in the Android SDK, it is obtained an absolute orientation representation with respect to the Earth frame [52], with

$$\begin{aligned} Q_s &= \mathrm{e}^{(\alpha/2)(u\vec{i}+v\vec{j}+w\vec{k})} \\ &= \begin{bmatrix} u\sin(\alpha/2) \\ v\sin(\alpha/2) \\ w\sin(\alpha/2) \\ \cos(\alpha/2) \end{bmatrix} = \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} \end{aligned} \tag{4.14}$$

where α is the amount of degrees the quaternion is rotated around the axis $u\vec{i} + v\vec{j} + w\vec{k}$. The rotation matrix R_b^w can be defined using Q_s as

$$R_b^w = \begin{bmatrix} 1-2(q_2^2+q_3^2) & 2(q_1q_2-q_0q_3) & 2(q_0q_2+q_1q_3) \\ 2(q_1q_2-q_0q_3) & 1-2(q_1^2+q_3^2) & 2(q_2q_3+q_0q_1) \\ 2(q_1q_3+q_0q_2) & 2(q_0q_1+q_2q_3) & 1-2(q_1^2+q_2^2) \end{bmatrix}. \tag{4.15}$$

Comparing the terms in the two representations of R_b^w, the Euler angles are obtained as

$$\begin{bmatrix} \psi \\ \theta \\ \phi \end{bmatrix} = \begin{bmatrix} atan2(2(q_3q_2+q_0q1), 1-2(q_1^2+q_2^2)) \\ arcsin(2(q_3q_1-q_2q_0)) \\ atan2(2(q_3q_0+q_1q2), 1-2(q_0^2+q_1^2)) \end{bmatrix}. \tag{4.16}$$

4.4.2 Position Estimation

Taking into account that the global navigation satellite systems (GNSS) receivers in smartphones have an accuracy of around 3 m and a sampling frequency of 1 Hz, a Kalman filter for position tracking is designed. In order to keep the estimation system independent from the application of controlling the quadrotor, this filter is based on the dynamics of a moving particle with constant acceleration between two samples

$$\xi(k+1) = \xi(k) + \dot{\xi}(k)t_k + 0.5\ddot{\xi}(k)t_k^2, \tag{4.17}$$

where t_k is the sample time.

Using $E_k = \begin{bmatrix} \xi & \dot{\xi} & \ddot{\xi} \end{bmatrix}^T = \begin{bmatrix} x_k & y_k & z_k & \dot{x}_k & \dot{y}_k & \dot{z}_k & \ddot{x}_k & \ddot{y}_k & \ddot{z}_k \end{bmatrix}^T$ as state vector and being

$$\Gamma = \begin{bmatrix} 1 & 0 & 0 & t_k & 0 & 0 & 0.5t_k^2 & 0 & 0 \\ 0 & 1 & 0 & 0 & t_k & 0 & 0 & 0.5t_k^2 & 0 \\ 0 & 0 & 1 & 0 & 0 & t_k & 0 & 0 & t_k^2 \\ 0 & 0 & 0 & 1 & 0 & 0 & t_k & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & t_k & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & t_k \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \tag{4.18}$$

the matrix that satisfies $E_{k+1} = \Gamma E_k$, the prediction of the state $\hat{E}_k^-$ and its covariance P_k^- are obtained as

$$\hat{E}_k^- = \Gamma \hat{E}_{k-1}, \tag{4.19}$$

$$P_k^- = \Gamma P_{k-1} \Gamma^T + Q_k, \tag{4.20}$$

where $\hat{E}_{k-1}$ is the previous estimated state, P_{k-1} the previous error covariance matrix and Q_k the process variance. The state prediction is then corrected calculating the

Kalman gain vector K_k as

$$K_k = P_k^- H^T (H P_k^- H^T + R)^{-1}, \tag{4.21}$$

and updating the state estimation $\hat{E}_k$ and its covariance P_k, based on the measurements ζ_k as

$$\begin{aligned} \hat{E}_k &= \hat{E}_k^- + K_k(\zeta_k - H\hat{E}_k^-), \\ P_k &= (\mathcal{I} - K_k H)P_k^-, \end{aligned} \tag{4.22}$$

where R is the measurement covariance matrix, $\mathcal{I}$ is the identity matrix and H is the matrix that relate ζ_k and E_k. H is defined as

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \tag{4.23}$$

considering that $\zeta_k \in R^6$ contains the GNSS measurements of x_m and y_m, the barometer measurements of z_m (in m), and the measurements of $\ddot{x}_m$, $\ddot{y}_m$ and $\ddot{z}_m$ in the Earth frame (in m/s^2).

The x_m and y_m position measurements are acquired using the GNSS receiver in the smartphone. Each sample of these coordinates are initially set in an ellipsoidal representation of decimal degrees by the receiver, and then converted to a bi-dimensional representation in meter units using the cartographic projection Magna-Sirgas.
The z_m measurements are acquired using the barometric pressure sensor which delivers the pressure value p_k in hPa. This signal is converted to meters as

$$z_m = 44330 \left(1 - \frac{p_k}{p_0}^{1/5.255}\right), \tag{4.24}$$

where p_0 is the atmospheric pressure at sea level [53].

The remaining measurement signals are the accelerations with respect to the Earth frame, $\ddot{\xi}_m$. These signals are calculated using the raw acceleration measurements from the smartphone a_b, which are represented with respect to the smartphone's body frame, and the attitude quaternion Q_s as

$$\ddot{\xi}_m = \begin{bmatrix} \ddot{x}_m & \ddot{y}_m & \ddot{z}_m \end{bmatrix}^T = Q_s a_b Q_s', \tag{4.25}$$

where Q_s' is the quaternion conjugate of Q_s.

The vector ζ_k is then set as

$$\zeta_k = \begin{bmatrix} x_m & y_m & z_m & \ddot{x}_m & \ddot{y}_m & \ddot{z}_m \end{bmatrix}^T. \tag{4.26}$$

4.4.3 Particle Model

rtrtrtrtrt

4.4.4 Quadrotor Model

ytytyttytt

4.5 Conclusions

This chapter presented the simulation and the testing of five control techniques for the attitude control of a quadrotor. The first technique is based on Lyapunov theory, it proved to be very reactive, especially for the yaw angle control. However, the stabilization in the direct neighborhood of the equilibrium point was not rigid enough to permit hover flight. The second one is a PID controller, it proved to be well adapted to the quadrotor when flying near hover. It was possible using this technique to successfully perform the first autonomous flight. The PID controller was only able to control the quadrotor in near hover and absence of large disturbances. The third one is an LQ controller, it displayed average stabilization results. It showed to be less dynamic than the PID. The fourth control technique is the Backstepping, its ability to control the orientation angles in presence of relatively high perturbations is very interesting. The sliding-mode technique is the fifth approach, it did not provide excellent results. The switching nature of the controller seems to be ill adapted to the dynamics of the quadrotor. The results of all these control approaches conducted to a combination of PID and Backstepping into the so-called Integral Backstepping. This was proposed as a single tool to design attitude, altitude and position controllers. The experiment has shown that OS4 is currently able to take-off, hover, land and avoid collisions automatically. As far as we know, OS4 is the first quadrotor practically capable of a collision avoidance maneuver.

Chapter 5

Implementation and Results

rgrtgrtgrgtrgrrg

5.1 Android Application

sdfdsfdsfds

5.2 Ground Control Station

sdfsdfdsd

5.3 Linear Quadratic Regulator Results

grtgrgtrgtrgtrg

5.3.1 Simple Translational Movements (LQR)

grtgtrgrtgrg

5.3.2 Trajectory Tracking (LQR)

trgrgrtgtrgr

5.4 H∞ Regulator Results

grgtrgrgtrg

5.4.1 Simple Translational Movements (H∞)

grtgrgrtgrtgtrg

5.4.2 Trajectory Tracking (H∞)

rgrtgrtgrgtrgrg

5.5 Conclusions

rgrgrgtrgrtgr

Chapter 6

Conclusions and Outlook

In this thesis distributed algorithms

Publications

A. Astudillo, P. Muñoz, F. Alvarez and E. Rosero, "Altitude and attitude cascade controller for a smartphone-based quadcopter," in *2017 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, jun 2017, pp. 1447–1454. [Online]. Available: http://ieeexplore.ieee.org/document/7991400/

A. Astudillo, B. Bacca and E. Rosero, "Optimal and robust controllers design for a smartphone-based quadrotor," in *2017 IEEE 3rd Colombian Conference on Automatic Control (CCAC)*

(Paper Submitted to Journal) A. Astudillo, P. Muñoz and E. Rosero, "Cascade Controller for Autonomous Flight of a Smartphone-based Quadrotor," in *Journal of Intelligent & Robotic Systems, SI: UAS-2017.*

Supplementary Material

All supplementary material including code, simulations, 3D objects designs, etc., is hosted in: https://goo.gl/U43bB6

Bibliography

[1] H. Liu and X. Wang, "Quaternion-based robust attitude control for quadrotors," in *2015 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, jun 2015, pp. 920–925. [Online]. Available: http://ieeexplore.ieee.org/document/7152379/

[2] R. Lopez, I. Gonzalez-Hernandez, S. Salazar, A. E. Rodriguez, J. J. Ordaz, and A. Osorio, "Disturbance rejection for a Quadrotor aircraft through a robust control," in *2015 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, jun 2015, pp. 409–415. [Online]. Available: http://ieeexplore.ieee.org/document/7152317/

[3] J. Jung, Y. Jung, D. You, and D. H. Shim, "A flight control system design for highly unstable unmanned combat aerial vehicles," in *2014 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, may 2014, pp. 1117–1125. [Online]. Available: http://ieeexplore.ieee.org/document/6842365/

[4] S. Kohno and K. Uchiyama, "Design of robust controller of fixed-wing UAV for transition flight," in *2014 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, may 2014, pp. 1111–1116. [Online]. Available: http://ieeexplore.ieee.org/document/6842364/

[5] B. Shang, J. Liu, T. Zhao, and Y. Chen, "Fractional order robust visual servoing control of a quadrotor UAV with larger sampling period," in *2016 International Conference on Unmanned Aircraft Systems (ICUAS)*, vol. 1, no. 209. IEEE, jun 2016, pp. 1228–1234. [Online]. Available: http://ieeexplore.ieee.org/document/7502645/

[6] S. Salazar, I. Gonzalez-Hernandez, R. Lopez, and R. Lozano, "Simulation and robust trajectory-tracking for a Quadrotor UAV," in *2014 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, may 2014, pp. 1167–1174. [Online]. Available: http://ieeexplore.ieee.org/document/6842371/

[7] C. Pearce, M. Guckenberg, B. Holden, A. Leach, R. Hughes, C. Xie, M. Hassett, A. Adderley, L. E. Barnes, M. Sherriff, and G. C. Lewin, "Designing a spatially aware, automated quadcopter using an Android control

system," in *2014 Systems and Information Engineering Design Symposium (SIEDS)*, vol. 00, no. c. IEEE, apr 2014, pp. 23–28. [Online]. Available: http://ieeexplore.ieee.org/document/6829921/

[8] A. Bjälemark and H. Bergkvist, "Quadcopter control using Android based sensing," *Advances in Electrical and Computer Engineering*, pp. 15–21, 2014.

[9] G. Loianno, G. Cross, C. Qu, Y. Mulgaonkar, J. A. Hesch, and V. Kumar, "Flying Smartphones: Automated Flight Enabled by Consumer Electronics," *IEEE Robotics & Automation Magazine*, vol. 22, no. 2, pp. 24–32, jun 2015. [Online]. Available: http://ieeexplore.ieee.org/document/7105372/

[10] ArduPilot Dev Team, "Copter - Flight Modes," 2016. [Online]. Available: http://ardupilot.org/copter/docs/flight-modes.html

[11] A. Drumea, "Control of Industrial Systems Using Android-Based Devices," *36th Int. Spring Seminar on Electronics Technology*, pp. 405–408, 2013.

[12] K.-w. Lin, Z.-h. Wu, and Y.-l. Lin, "Applications of Temperature Control Based on Android Platform," *Proceedings of the International MultiConference of Engineers and Computer Scientists*, vol. II, 2014.

[13] N.-V. Truong and D.-L. Vu, "Remote monitoring and control of industrial process via wireless network and Android platform," *2012 International Conference on Control, Automation and Information Sciences (ICCAIS)*, no. 1, pp. 340–343, 2012.

[14] L. F. Aristizabal, D. F. Almario, and J. A. Lopez, "Development of an Android App as a learning tool of dynamic systems and automatic control," in *2014 III International Congress of Engineering Mechatronics and Automation (CIIMA)*. IEEE, oct 2014, pp. 1–5. [Online]. Available: http://ieeexplore.ieee.org/document/6983438/

[15] S. Wu Wu, "Sintonización de controladores PI/PID mediante el desarrollo de una aplicación en Android," Ph.D. dissertation, Universidad de Costa Rica, 2013.

[16] J. García Téllez and D. F. Ochoa Calambás, "Desarrollo de una plataforma de monitorización, control y comunicación con teléfonos inteligentes, para laboratorios portátiles de soporte al área de señales y sistemas," Bachelor Thesis, Universidad del Valle, Cali, 2015.

[17] M. A. Gunawan, F. Sulaiman, T. A. Putra, C. Daniel Hasudungan, and R. F. Sari, "Object-following robot using adaptive cruise control algorithm with IOIO," in *2014 International Conference on Intelligent Green Building and Smart Grid (IGBSG)*. IEEE, apr 2014, pp. 1–5.

[Online]. Available: http://ieeexplore.ieee.org/xpls/abs{_}all.jsp?arnumber=6835251http://ieeexplore.ieee.org/document/6835251/

[18] P. Stefka and K. Zakova, "Mobile application for remote control of thermo-optical plant," in *2016 13th International Conference on Remote Engineering and Virtual Instrumentation (REV)*, no. February. IEEE, feb 2016, pp. 435–439. [Online]. Available: http://ieeexplore.ieee.org/document/7444519/

[19] A. Mora, H. Yagama, D. Zorro, and L. F. R. Jimenez, "Speed digital control for scale car via Bluetooth and Android," in *2015 CHILEAN Conference on Electrical, Electronics Engineering, Information and Communication Technologies (CHILECON)*. IEEE, oct 2015, pp. 129–134. [Online]. Available: http://ieeexplore.ieee.org/document/7400364/

[20] Y. Luo, J. Liu, Y. Gao, and Z. Lu, "Smartphone-Controlled Robot Snake for Urban Search and Rescue," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2014, vol. 8917, pp. 352–363. [Online]. Available: http://www.engineeringvillage.com/blog/document.url?mid=cpx{_}4bcb64f714a3ae9b8caM668310178163125{&}database=cpxhttp://link.springer.com/10.1007/978-3-319-13966-1{_}35

[21] Z. Lu, F. Nagata, K. Watanabe, and M. K. Habib, "iOS application for quadrotor remote control," *Artificial Life and Robotics*, vol. 22, no. 3, pp. 374–379, sep 2017. [Online]. Available: http://link.springer.com/10.1007/s10015-017-0372-3

[22] G. Chen, S. A. King, and M. Scherger, "Robot Remote Control Using Bluetooth and a Smartphone Augmented System," in *Lecture Notes in Electrical Engineering*, 2011, vol. 133 LNEE, no. VOL. 2, pp. 453–460. [Online]. Available: http://link.springer.com/10.1007/978-3-642-25992-0{_}63

[23] T. Tetzlaff, R. Zandian, L. Drüppel, and U. Witkowski, *Robot Intelligence Technology and Applications 2012*, ser. Advances in Intelligent Systems and Computing, J.-H. Kim, E. T. Matson, H. Myung, and P. Xu, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, vol. 208. [Online]. Available: http://link.springer.com/10.1007/978-3-642-37374-9

[24] A. Geissbühler, J. Demongeot, M. Mokhtari, B. Abdulrazak, and H. Aloulou, *Inclusive Smart Cities and e-Health*, ser. Lecture Notes in Computer Science, A. Geissbühler, J. Demongeot, M. Mokhtari, B. Abdulrazak, and H. Aloulou, Eds. Cham: Springer International Publishing, 2015, vol. 9102. [Online]. Available: http://link.springer.com/10.1007/978-3-319-19312-0

[25] N. Oros and J. L. Krichmar, "Smartphone Based Robotics: Powerful, Flexible and Inexpensive Robots for Hobbyists, Educators, Students and

Researchers," 2013. [Online]. Available: https://pdfs.semanticscholar.org/1e4f/371b9509dac7b649a473a0b95c39f0bfd63a.pdf

[26] J. P. de A. Barbosa, F. do P. de C. Lima, L. dos S. Coutinho, J. P. R. Rodrigues Leite, J. Barbosa Machado, C. Henrique Valerio, and G. Sousa Bastos, "ROS, Android and cloud robotics: How to make a powerful low cost robot," in *2015 International Conference on Advanced Robotics (ICAR)*. IEEE, jul 2015, pp. 158–163. [Online]. Available: http://ieeexplore.ieee.org/document/7251449/

[27] M. A. Alsharif, Y. E. Arslantas, and M. S. Holzel, "A comparison between advanced model-free PID and model-based LQI attitude control of a quadcopter using asynchronous android flight data," in *2017 25th Mediterranean Conference on Control and Automation (MED)*. IEEE, jul 2017, pp. 1023–1028. [Online]. Available: http://ieeexplore.ieee.org/document/7984252/

[28] M. A. Alsharif and M. S. Holzel, "Estimation of a drone's rotational dynamics with piloted Android flight data," in *2016 IEEE 55th Conference on Decision and Control (CDC)*, no. Cdc. IEEE, dec 2016, pp. 1199–1204. [Online]. Available: http://ieeexplore.ieee.org/document/7798429/

[29] M. A. Alsharif, Y. E. Arslantas, and M. S. Holzel, "Advanced PID attitude control of a quadcopter using asynchronous android flight data," in *2017 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, jun 2017, pp. 1602–1607. [Online]. Available: http://ieeexplore.ieee.org/document/7991325/

[30] G. Loianno, C. Brunner, G. McGrath, and V. Kumar, "Estimation, Control, and Planning for Aggressive Flight With a Small Quadrotor With a Single Camera and IMU," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 404–411, 2017. [Online]. Available: http://ieeexplore.ieee.org/document/7762111/

[31] V. A. Isuru, M. A. I. M. Dharmadasa, D. V. B. C. Jayasinghe, and C. I. Keppitiyagama, "Reusing discarded-smartphone capabilities on quadcopters: The rationale, benefits and issues," in *2016 Sixteenth International Conference on Advances in ICT for Emerging Regions (ICTer)*. IEEE, sep 2016, pp. 229–236. [Online]. Available: http://ieeexplore.ieee.org/document/7829923/

[32] G. Loianno, Y. Mulgaonkar, C. Brunner, D. Ahuja, A. Ramanandan, M. Chari, S. Diaz, and V. Kumar, "Smartphones power flying robots," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, sep 2015, pp. 1256–1263. [Online]. Available: http://ieeexplore.ieee.org/document/7353530/

[33] L. Aldrovandi, M. Hayajneh, M. Melega, M. Furci, R. Naldi, and L. Marconi, "A smartphone based quadrotor: Attitude and position

estimation," in *2015 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, jun 2015, pp. 1251–1259. [Online]. Available: http://ieeexplore.ieee.org/document/7152418/

[34] P. Bryant, G. Gradwell, and D. Claveau, "Autonomous UAS controlled by onboard smartphone," in *2015 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, jun 2015, pp. 451–454. [Online]. Available: http://ieeexplore.ieee.org/document/7152322/

[35] T. Bresciani, "Modelling, Identification and Control of a Quadrotor Helicopter," Master's thesis, Lund University, 2008.

[36] M. Faessler, D. Falanga, and D. Scaramuzza, "Thrust Mixing, Saturation, and Body-Rate Control for Accurate Aggressive Quadrotor Flight," *IEEE Robotics and Automation Letters*, vol. PP, no. 99, pp. 1–7, 2016. [Online]. Available: http://rpg.ifi.uzh.ch/docs/RAL16{_}Faessler.pdf{%}0Ahttp://ieeexplore.ieee.org/document/7784809/

[37] S. Bouabdallah, "Design and Control of Quadrotors With Application To Autonomous Flying," *École Polytechnique Fédérale De Lausanne, À La Faculté Des Sciences Et Techniques De L'Ingénieur*, vol. 3727, no. 3727, p. 61, 2007.

[38] M. Emam and A. Fakharian, "Attitude tracking of quadrotor UAV via mixed H2/H∞ controller: An LMI based approach," in *2016 24th Mediterranean Conference on Control and Automation (MED)*. IEEE, jun 2016, pp. 390–395. [Online]. Available: http://ieeexplore.ieee.org/document/7535919/

[39] S. Badr, O. Mehrez, and A. E. Kabeel, "A novel modification for a quadrotor design," in *2016 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, jun 2016, pp. 702–710. [Online]. Available: http://ieeexplore.ieee.org/document/7502536/

[40] F. Sabatino, *Quadrotor control : modeling, nonlinear control design, and simulation*. Stockholm, Sweden: KTH Royal Institute of Technology in Stockholm, 2015, no. June.

[41] R. Vepa, *Nonlinear Control of Robots and Unmanned Aerial Vehicles: An Integrated Approach*, 1st ed. London: CRC Press, 2016.

[42] K. U. Lee, Y. H. Yun, W. Chang, J. B. Park, and Y. H. Choi, "Modeling and Altitude Control of Quad-rotor UAV," in *Proceedings of International Conference on Control, Automation and Systems*, 2011, pp. 1897–1902.

[43] M. A. Khodja, M. Tadjine, M. S. Boucherit, and M. Benzaoui, "Experimental dynamics identification and control of a quadcopter," in *2017 6th International Conference on Systems and Control (ICSC)*. IEEE, may 2017, pp. 498–502. [Online]. Available: http://ieeexplore.ieee.org/document/7958668/

[44] T. Jiřinec, *Stabilization and control of unmanned quadcopter*. Prague, Czech Republic: Czech Technical University in Prague, 2011. [Online]. Available: https://support.dce.felk.cvut.cz/mediawiki/images/d/d4/Dp{_}2011{_}jirinec{_}tomas.pdf

[45] G. A. Garcia, A. R. Kim, E. Jackson, S. S. Kashmiri, and D. Shukla, "Modeling and flight control of a commercial nano quadrotor," in *2017 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, jun 2017, pp. 524–532. [Online]. Available: http://ieeexplore.ieee.org/document/7991439/

[46] Z. Mustapa, S. Saat, A. M. Darsono, and H. H. Yusof, "Experimental Validation of an Altitude Control for Quadcopter," *ARPN Journal of Engineering and Applied Sciences*, vol. 11, no. 6, pp. 3789–3795, 2016.

[47] M. D. L. Costa De Oliveira, *Modeling, Identification and Control of a Quadrotor Aircraft*. Czech Technical University in Prague, jun 2011.

[48] H. Werner, *Lecture Notes - Control Systems Theory and Design*. Hamburg: Hamburg University of Technology, 2012.

[49] ——, *Lecture Notes - Optimal and Robust Control*. Hamburg: Hamburg University of Technology, 2013.

[50] S. Skogestad and I. Postlethwaite, *Multivariable Feedback Control: Analysis and Design*. John Wiley and Sons, 2001, vol. 2.

[51] A. Astudillo, P. Munoz, F. Alvarez, and E. Rosero, "Altitude and attitude cascade controller for a smartphone-based quadcopter," in *2017 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, jun 2017, pp. 1447–1454. [Online]. Available: http://ieeexplore.ieee.org/document/7991400/

[52] Android Developer, "SensorEvent," 2015. [Online]. Available: https://developer.android.com/

[53] K. S. Lauszus, "Flight Controller for Quad Rotor Helicopter in X-configuration," Ph.D. dissertation, Kongens Lyngby, Denmark, 2015.