MASTER THESIS

MASTER DEGREE IN AUTOMATIC CONTROL AND ROBOTICS

A robust predictive control for a quadrotor with an airsoft marker

Author: Sergio Moyano Diaz

Supervisor: Dr. Antoni Grau

Co-Supervisor: Dr. Yolanda Bolea

Call: April 2018

ETSEIB

ESCUELA TECNICA SUPERIOR DE INGENERIA INDUSTRIAL DE BARCELONA

UPC

Abstract

This project is part of a more ambitious European project which consists on the autonomous inspection of petrol pipes. This inspection is done using a custom design drone equipped with a direction system, called gimbal, that allows changing the direction of a set of tools attached to the drone. These tools are a stereo vision system and an airsoft marker, which have special bullets that are used to detect the possible leaks in the pipes in order to fix them. Moreover, the shooting power of the marker can change to adjust according the distance of the pipe. However, the marker introduce into the system an unknown disturbance that can destabilize the drone.

Therefore, the main goal of the project is to propose a model predictive controller for the drone.

As a first step needed, the project come up with a mathematical model of the drone, specifically a Quadrotor. Also, it is described a way of modeling the Airsoft marker as a disturbance. Then, this model is used in the MPC to calculate the exact control action for satisfying the optimization law, that it is tuned to find a balance between the desired performance and the energy used by the actuators.

Finally, a case scenarios is simulated to show how perform the designed controller and the different commands (roll, pitch and yaw).

Contents

ETSEIB

ETSEIB

List of Figures

Chapter 1

Introduction

1.1 Context

This project is part of an European project, called EuroArms, whose objective is the autonomous inspection of oil pipelines. At the beginning of the project, a feasibility study was made to evaluate the possible solutions and the technologies to carry out the inspection. After several days of discussing and contributing ideas, several solutions were reached:

The first solution was to do the manual inspection. However, it would need an operator to continuously observe the pipelines. Therefore, this was discarded from the start, since there are many kilometres of pipe.

The second solution was to attach a robot to the pipes in such a way that it would be able to move along the pipes, and by means of an electromagnetic sensor to be able to detect if there are leaks or not in the pipe.

The third and most valued option was to use a drone that was able to identify leaks using a stereo system cameras. To repair the pipelines, it is used Airsoft marker with an special type of bullets. Moreover, the bullets help to identify if there is a leak or not, because it add more visual characteristics to pipe, like a very high contrast between the oil and the pipe.

Therefore, it was need also a directional system, called gimbal, where will be attached the stereo camera system and the Airsoft marker. It will be helpful to avoid to move the drone along the whole pipe, and observe the position in a quick and easy way. Additionally, it will allow to do a manual inspection, but only in case of very complex and unclear situation. To perform this, it will be able to use a virtual reality glasses, like the Samsung Gear with a Samsung S7. The design of this custom drone was developed by the author of this thesis, in the master thesis of computer engineering (MEI). But in the final deployment was observed that shooting of the marker introduce an unknown disturbances into the system. The drone has implemented a PID controller that can not compensate in acceptable way the disturbances.

1.2 Objectives

This project propose to change the classical PID controller implemented in the drone, for a model predictive controller to be able to improve the performance of the actual controller. Therefore, to achieve this main objective it is needed:

- A mathematical model of the quadrotor
- An estimation of the parameters of the model
- A mathematical model of the gimbal with the Airsoft marker
- Propose a way to be robust against disturbances
- Simulation and validation of the proposed controller.

1.3 Thesis Outline

Chapter 2: Description of the quadrotor
This chapter introduce the relevant information to understand how the quadrotor works, and the main components that are involved.

Chapter 3: Mathematical Model
This chapter shows the main equations that describe the movement of the quadrotor, and propose a formulation in state space.

Chapter 4: Identification
This chapter estimate each of the parameters that mathematical model and propose a way to verify the results.

Chapter 5: Model predictive controller
This chapter review the theory behind the MPC, and present the selected controller. Moreover, a method to be robust against unknown disturbances is presented.

Chapter 6: Simulation
This chapter shows how the selected control works against different cases like shooting the marker or against wind. Additionally, it presented a real scenario where is tested the controller.

Chapter 7: Conclusion
This chapter summarizes the major contributions made for this work, review the controller performance and discuss several ways to improve the performance and robustness.

Chapter 2

Description of the quadrotor

2.1 Movement and operation of the quadrotor

A very simple way and realistic to model the quadrotor only for the explanation of their movements is a simple solid rigid which have four rotors with asymmetric rotation of adjacent propellers. This means that one pair of the propellers rotate clockwise, while the other pair rotates counter-clockwise, as it shown in the figure 2.1, but all propellers generate an airflow for lift the quadrotor. Moreover, this configuration helps to remove side propeller needed in standard helicopter.

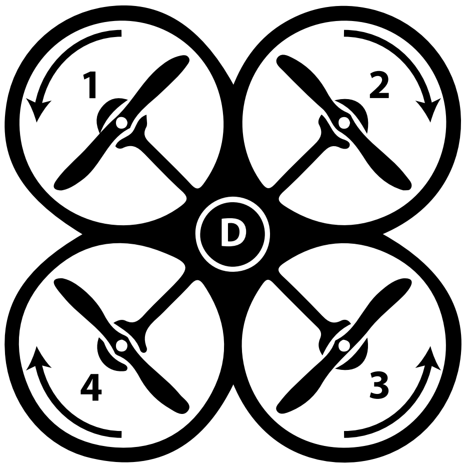


FIGURE 2.1: Simplified schematic of the quadrotor

Another interesting property of a quadrotor is that it is an underacted system. This means that the number of degrees of freedom of the system (in total 6: 3 of position and 3 of rotation) are more than the number of actuators (4 rotors). Therefore, it is not possible achieve the desired state for all degrees of freedom. Despite this, the particular geometric configuration of a quadrotor helps decouple control variables and selecting a simple controller. The basic movements are related to the way in which the spacecraft rotates, ie Euler angles. The figure 2.2 shows the assignment of each euler angle to each axis of the quadrotor: Pitch, roll and yaw.

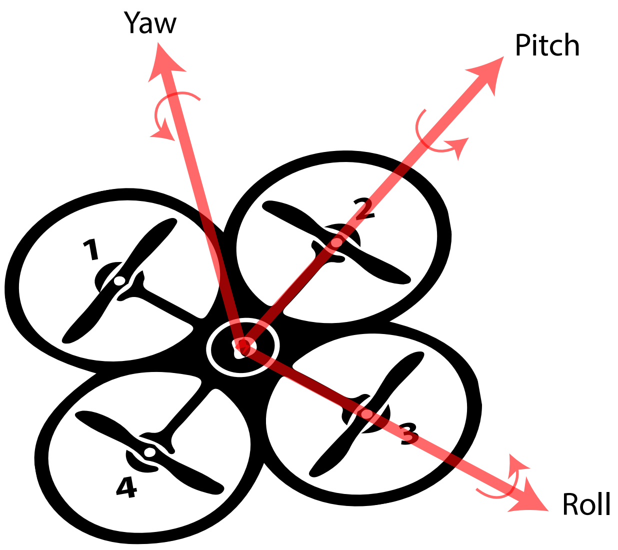


FIGURE 2.2: Euler angles for a quadrotor

Therefore there are four basic movements on a quadrotor:

- Roll

  This movement is done by decreasing (or increasing) the rotation speed of the rotor 4 and increasing (or decreasing) the rotation speed of the rotor 2. Additionally, it is performed with the particularity that the same quantity is added and subtracted to the rotors respectively. However, if the quantity is not the same, can lead to a movement on other axis. So, this produce only a torque in the roll axis, as a result of the unbalance forces in that axis. Meanwhile, the total thrust remains constant at all the time.

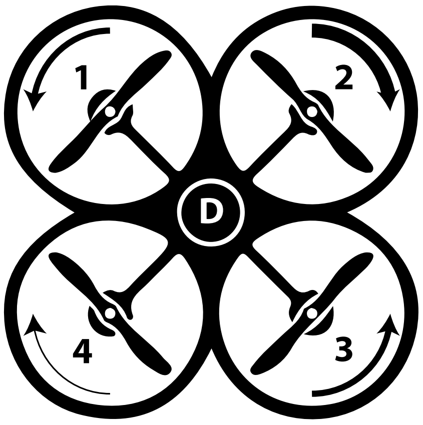


FIGURE 2.3: Roll movement

- Pitch

  This movement is quite similar to the roll and it is done by decreasing (or increasing) the rotation speed of the rotor 3 and increasing (or decreasing) the rotation speed of the rotor 1. Additionally, it is performed with the particularity that the same quantity is added and subtracted to the rotors respectively. However, if the quantity is not the same, can lead to a movement on other axis. So, this produce only a torque in the pitch axis, as a result of the unbalance forces in that axis. Meanwhile, the total thrust remains constant at all the time.

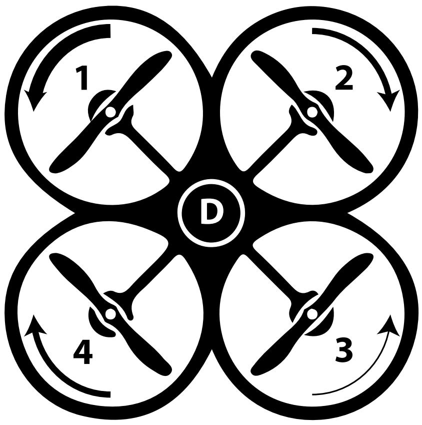


FIGURE 2.4: Pitch movement

- Yaw

  This movement is done by decreasing the rotation speed of the rotor 1 and 3, and increasing the rotation speed of the rotor 2 and 4, with the particularity that the same quantity is added and subtracted to the rotors respectively. This is possible thanks to opposite rotation of the propellers, and generating an opposite torque in the yaw axis. Meanwhile, the total thrust remains constant at all the time.

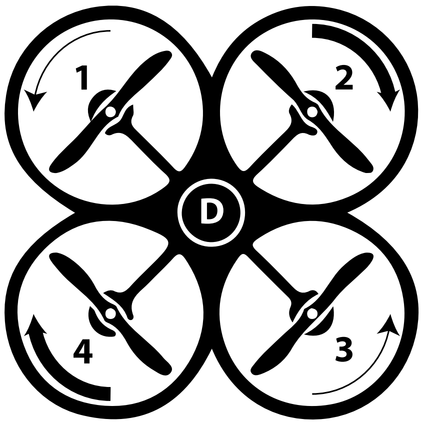


FIGURE 2.5: Yaw movement

- Throttle

  This movement is done by adding (or subtracting) the same quantity to the rotation speed of each rotor. Therefore, this leads to a vertical force in the body frame that lifts the quadrotor.

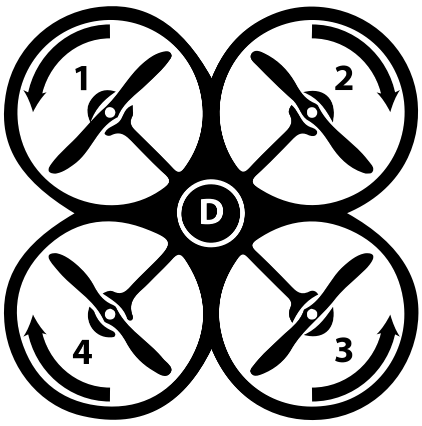


FIGURE 2.6: Thrust movement

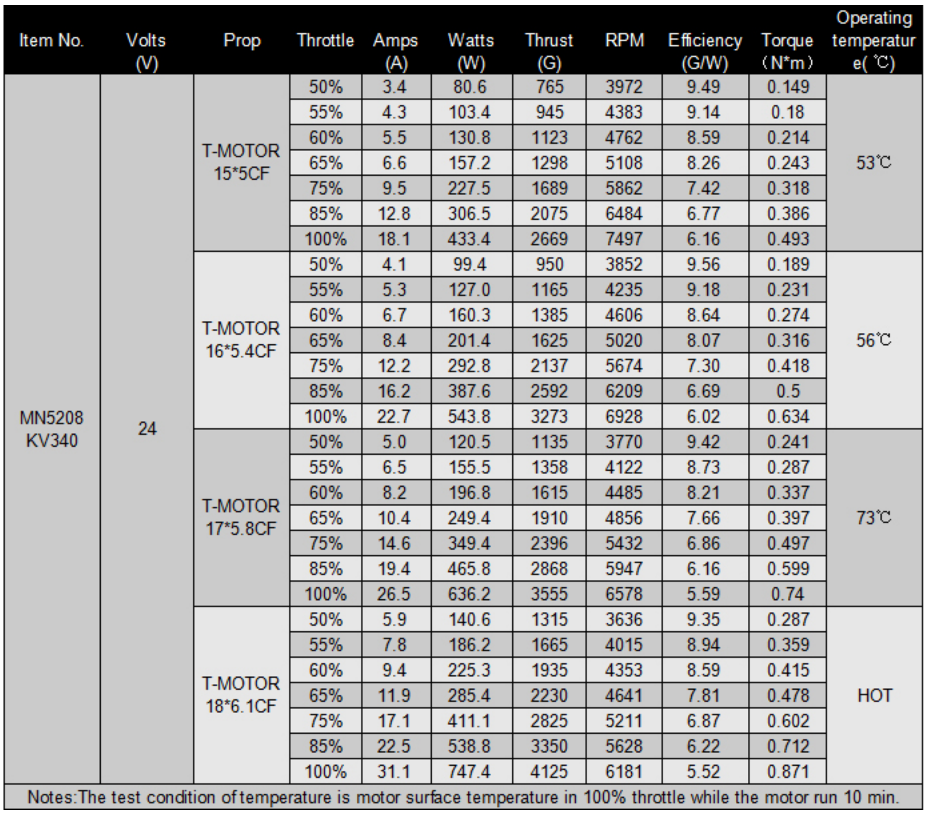
2.2 Hardware

In this section, the most relevant parts for the operation of the quadrotor is described, doing special emphasis on the electronic parts, and also it is presented a diagram to sum up the components used and the connections between them.

2.2.1 Rotor

In this quadrotor, it is used three-phase outrunner brushless motors, which have the advantage of delivering high power at very low weight. However, the control of these rotors involves a much more complicated procedure, as an electronic controller that are charge to regulate the speed of rotation. In one hand, the DC motors are much easier to control, but usually need an additional set of gears to achieve the desired speed and torque, which introduce into the system an extra weight, complexity, inefficient and high noise. In another hand, the inrunner usually have less poles, so they spin much faster than the outrunners, which generates less torque, that for a quadrotor is critical.

The selected model is a T-Motor MN5208, which generate a maximum thrust of 3273 grams with 16 inches propellers. The figure 2.7 shows the main features of this rotor like the maximum thrust, the constant K-V, the drop intensity at different rotation speed and the propellers that are compatible.



Item No.	Volts (V)	Prop	Throttle	Amps (A)	Watts (W)	Thrust (G)	RPM	Efficiency (G/W)	Torque (N*m)	Operating temperature(℃)
MN5208 KV340	24	T-MOTOR 15*5CF	50%	3.4	80.6	765	3972	9.49	0.149	53℃
			55%	4.3	103.4	945	4383	9.14	0.18	
			60%	5.5	130.8	1123	4762	8.59	0.214	
			65%	6.6	157.2	1298	5108	8.26	0.243	
			75%	9.5	227.5	1689	5862	7.42	0.318	
			85%	12.8	306.5	2075	6484	6.77	0.386	
			100%	18.1	433.4	2669	7497	6.16	0.493	
		T-MOTOR 16*5.4CF	50%	4.1	99.4	950	3852	9.56	0.189	56℃
			55%	5.3	127.0	1165	4235	9.18	0.231	
			60%	6.7	160.3	1385	4606	8.64	0.274	
			65%	8.4	201.4	1625	5020	8.07	0.316	
			75%	12.2	292.8	2137	5674	7.30	0.418	
			85%	16.2	387.6	2592	6209	6.69	0.5	
			100%	22.7	543.8	3273	6928	6.02	0.634	
		T-MOTOR 17*5.8CF	50%	5.0	120.5	1135	3770	9.42	0.241	73℃
			55%	6.5	155.5	1358	4122	8.73	0.287	
			60%	8.2	196.8	1615	4485	8.21	0.337	
			65%	10.4	249.4	1910	4856	7.66	0.397	
			75%	14.6	349.4	2396	5432	6.86	0.497	
			85%	19.4	465.8	2868	5947	6.16	0.599	
			100%	26.5	636.2	3555	6578	5.59	0.74	
		T-MOTOR 18*6.1CF	50%	5.9	140.6	1315	3636	9.35	0.287	HOT
			55%	7.8	186.2	1665	4015	8.94	0.359	
			60%	9.4	225.3	1935	4353	8.59	0.415	
			65%	11.9	285.4	2230	4641	7.81	0.478	
			75%	17.1	411.1	2825	5211	6.87	0.602	
			85%	22.5	538.8	3350	5628	6.22	0.712	
			100%	31.1	747.4	4125	6181	5.52	0.871	
Notes:The test condition of temperature is motor surface temperature in 100% throttle while the motor run 10 min.										

FIGURE 2.7: Specification MN5208

2.2.2 Electronic speed controller

The ESC is a circuit that is responsible for generating a three-phase signal that power the rotor. In particular, a digital open-source ESC has been selected. The vast majority control the speed of rotation by a signal supplied by the PWM. However, it can be controlled by: PWM (RC servo), analogue, UART, I2C, USB or CAN-bus. It also allows to read all rotor variables: power, consumption, speed, etc . It also has other advantages, such as regenerative braking, which allows recharging the battery as the motor is braked, real time system (ChibiOS / RT).

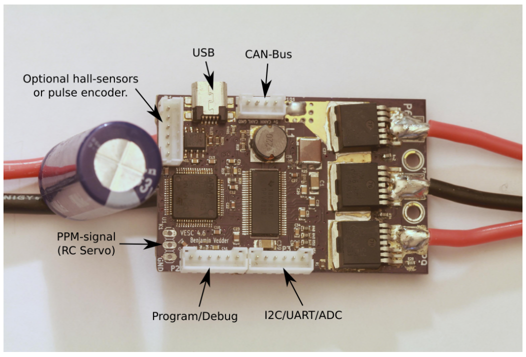


FIGURE 2.8: Electronic speed controller

2.2.3 Second processing unit

The second processing unit is a board that is responsible for controlling in short term the state of the quadrotor. Therefore, it has the responsibility to send control signals to the ESC, read the current orientation and position from the sensors, and handles all basic communication using radio frequency to connect to the base station. Additionally, it also sends the gimbal position and activates the airsoft marker.

The control signal of each ESC is generated using a PID controller. The main objective of this controller is to adjust to system imperfections such as the different length of the propellers, the unique response of each rotor, etc. This allow to be more tolerant against disturbances like wind or the effect of shooting the marker. Furthermore, each euler angle has a separate controller to achieve the desired angle. This is given by the main processing unit or radio frequency from the ground station.

The current board is a pixhack 2.8.4 which integrates the necessary sensors (accelerometer, gyroscope and magnetometer) to determine the Euler angles and its angular velocity. All the sensors are decoupled from the noisy vibration from the rotors using a damping platform inside the controller. Moreover, it contains the powerful 32 bit microcontroller STM32F427 that have 168 of frequency, 256 KB of RAM, 1 port I2c, 2 telemetry ports and 8 ports PWM and 1 port PPM. Since this board is based on the same components that the ardupilot module, it is fully compatible with the arducopter firmware and you can use the Mission Planner software (PC) to install and upgrade the firmware as well as planning your drone missions.

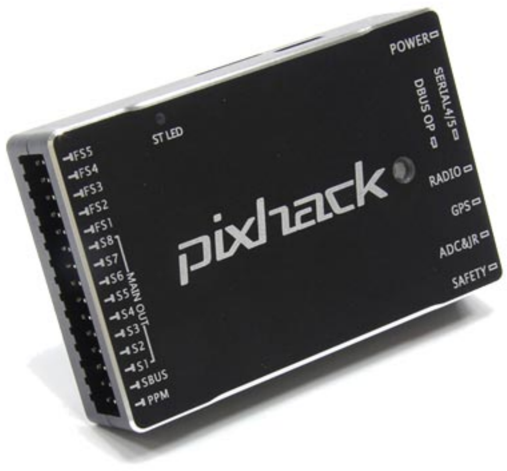


FIGURE 2.9: Pixhack

2.2.4 Main processing unit

The main processing unit is the board that is responsible for monitoring in long term the status of the quadrotor, like generate a trajectory knowing the current position and a map of the area or processing information from cameras to avoid obstacles. Also, it is capable of running the optimization toolbox like ACADO, since it is a C++ code that can compile on the board. Moreover, it has also the capability of running in real-time some optimization procedure, like the predictive control.

The current controller is an ODROID XU4 which is a single board computer developed by the company Hardkernel. The design integrates an octa-core processor Exynos 422 at 2GHz, 2 GB of DRR2 memory and a GPU Mali DRR2 440 MHz. It comes with the possibility to install Xubuntu or Android, but in this project it is used Xubuntu with the framework ROS.

ROS is a middleware that complement very well the operating system, providing hardware abstraction, control at low level of devices, implementation of usual functionalities in the robotic environment. Despite the importance of reactivity and low latency in robot control, ROS is not a real-time OS (RTOS), although it is possible to use it with a real-time code. Furthermore, the next generation of ROS (2.0) address this kind of problems.

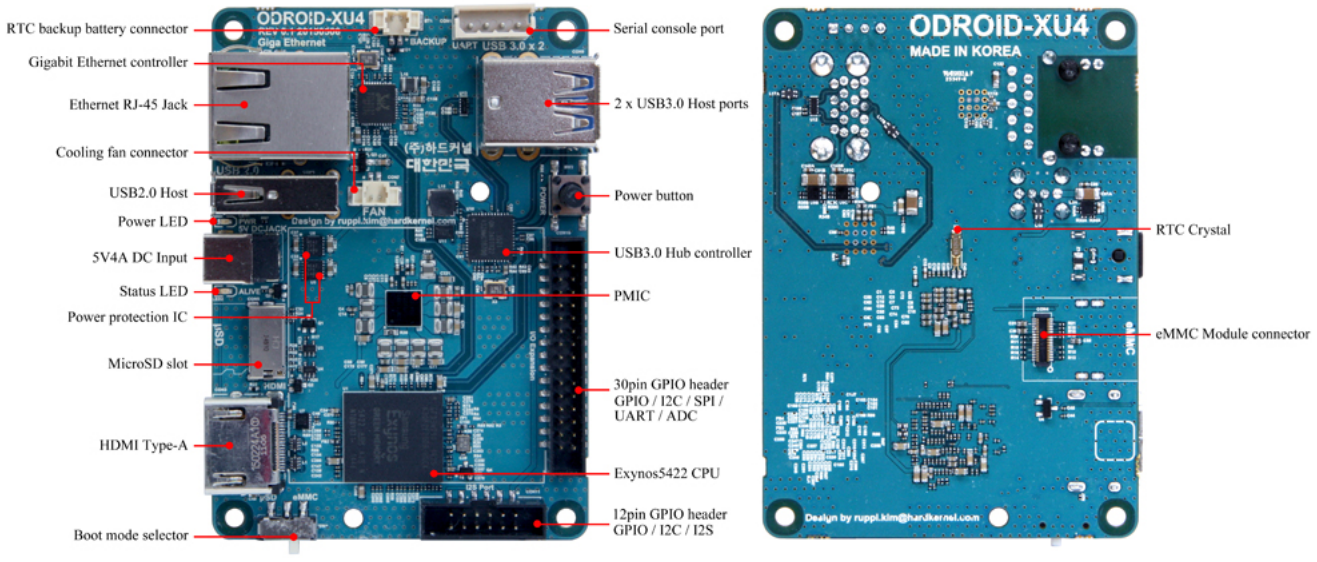


FIGURE 2.10: Odroid XU4

2.2.5 Gimbal

The gimbal is a system specially designed to orientate a set of tools in the space. It is designed to be modular and be able to adapt the size of the gimbal to make fit the tools on it. Moreover, the system has been damped using a set of antivibration rubbers.

The gimbal has its own controller that uses three rotors GB-42, one for each euler angle of the gimbal. The controller is a Basecam SimpleBGC that combine two external sensors to calculate very precisely the pose of the tool. Also, it receives the desired orientation from the second processing unit. Furthermore, it is possible to control manually the orientation using a joystick. The actual gimbal has 2 tools: an airsoft marker and stereo camera system.

The airsoft marker is responsible for launching the bullets to make and repair the oil lines. The selected marker is a Delta Yakuza tactics. The most important part is that have micro-mosfet that allows electronic activation, so it is directly connected to a digital port of the micro-controller to activate the release of the bullet.

The stereo camera system has two webcameras Trust 720p, because it is a cheap setup that give really good results. However, it will be change for a professional cameras once the quadrotor became stable with the marker.

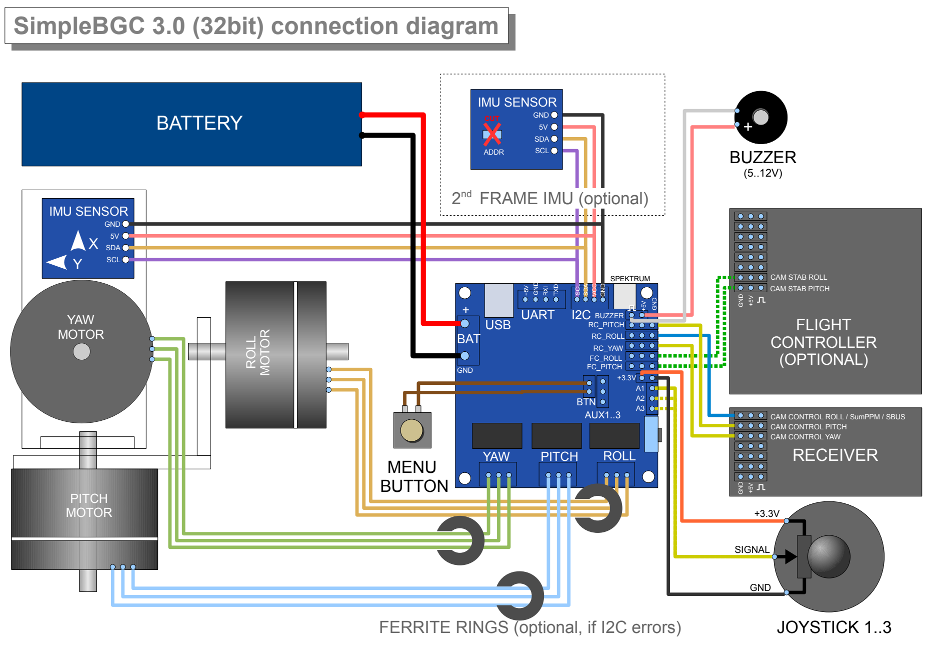


FIGURE 2.11: Gimbal Connections

2.2.6 Block diagram

The figure 2.12 shows a block diagram where it show the components involve and how interact each other.

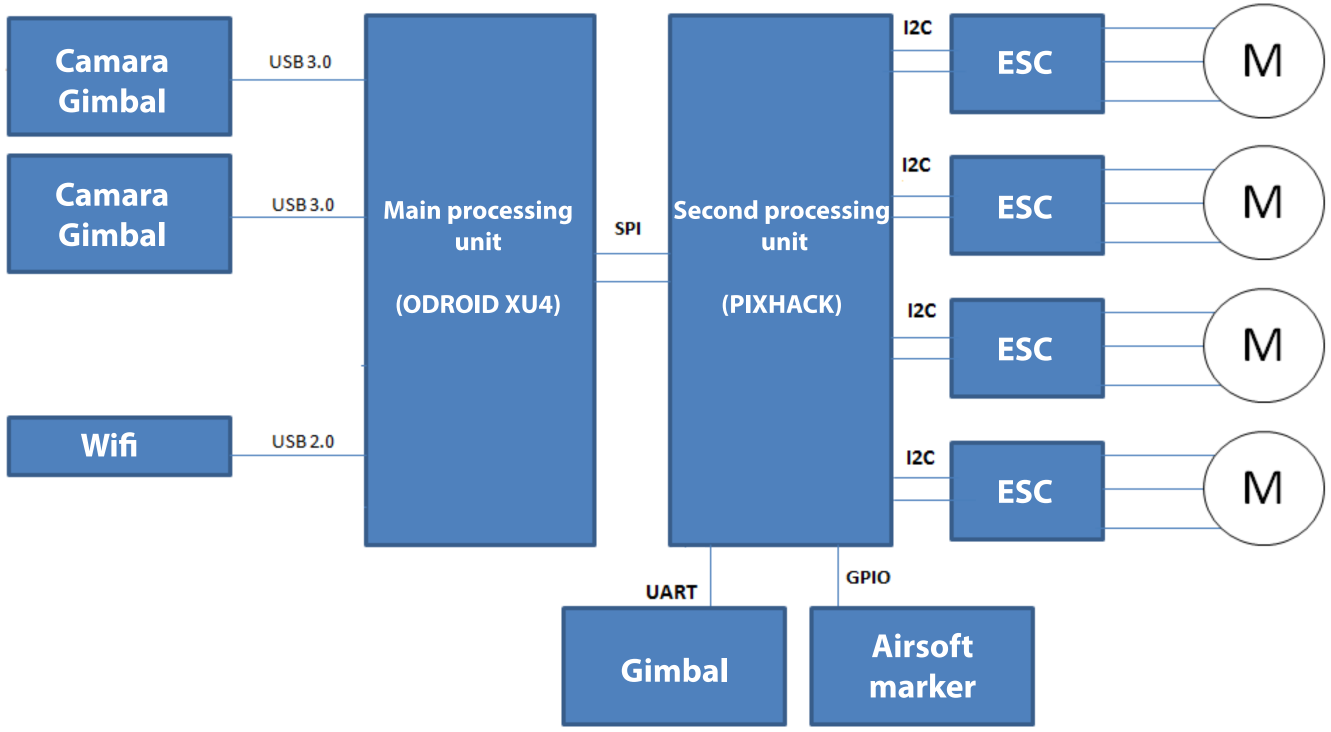


FIGURE 2.12: Block diagram

Chapter 3

Mathematical Model

In this chapter, the kinematic model of each part of the quadrotor will be derived using the Euler angles, and also the dynamics model of the quadrotor will calculated based on the first principles of a rigid body using the Newtom-Euler formulation. Furthermore, the following assumptions are used :

- The structure is completely rigid and perfect symmetric
- The center of mass is in the origin of the quadrotor fixed frame.
- The thrust are proportional to the square of the motors rotational speed.

Finally, the chapter will end with the formulation in the space space of the equations of the quadrotor.

3.1 Kinematics

First of all, it is needed to define the coordinate frame that will be used. Mainly, it is used 2 different frames: the Earth frame and the quadrotor frame.

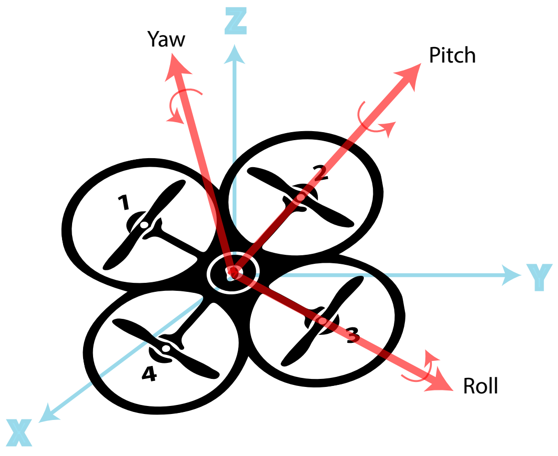


FIGURE 3.1: Euler angles for a quadrotor with inertial frame

The figure 3.1 shows the Earth frame with axis X , Y , Z and the body frame with axis Pitch , Roll and Yaw. In one hand, the Earth is an inertial frame situated in a fixed place in the ground. In another hand, the body frame is in the geometric center of the quadrotor, the pitch is pointing to propeller 2 , Roll axis pointing to propeller 3 and the Yaw axis is cross product of Roll and Pitch. So, it is used this frames to express the relative position of the center of mass of the quadrotor with respect the inertial frame and the orientation with respect the quadrotor frame. The relative distance is directly the difference between the Earth frame and the quadrotor frame r = [x,y,z]. The orientation is described by the Euler angles : roll , pitch and yaw (φ, θ, ψ).

An interesting information will be to relate the orientation in euler angles in the body frame, with the orientation in the earth frame. The order to apply the rotations is the Euler ZYX angle convention. This means that first it is rotated the roll , after the pitch and finally the yaw.

Therefore, the rotation matrix is given by :

$$R = RotZ(\psi)RotY(\theta)RotX(\phi) \tag{3.1}$$

The result of perform the matrix multiplication is :

$$R = \begin{bmatrix} c_\theta c_\psi & s_\phi s_\theta c_\psi - c_\psi s_\phi & c_\phi s_\theta c_\psi + s_\phi s_\psi \\ c_\theta c_\psi & s_\phi s_\theta s_\psi + c_\theta c_\psi & c_\phi s_\theta s_\psi - s_\theta c_\psi \\ s_\theta & s_\phi c_\theta & c_\phi c_\theta \end{bmatrix} \tag{3.2}$$

This rotation matrix will be used to express forces that are more convient to define in the quadrotor frame into the Earth frame (e.g the thrust forces) .

Another interesting transformation will be to relate the angular velocities η= [φ, θ, ψ] in the quadrotor frame, with the angular velocities w = [p,q,r] in the Earth frame. Usually, the information about the angular speed in the quadrotor frame is provided by a IMU. So, the transformation matrix is the following :

$$w = R_r\text{J} \tag{3.3}$$

where R_r is the rotation matrix that relate the angular velocities in the quadrotor frame with the angular velocities in earth frame.

$$R_r = \begin{bmatrix} 1 & 0 & -sin\theta \\ 0 & cos\phi & sin\phi cos\theta \\ 0 & -sin\phi & cos\phi cos\theta \end{bmatrix} \tag{3.4}$$

3.2 Dynamic Model

The motion of the quadrotor can be divided into two subsystems: the rotational system (roll, pitch, yaw) and the translation system (x, y, z). The rotational is a system fully actuated and the translation is a system underacted.

3.2.1 Rotational Equations of Motion

The rotational equations of motion are derived in the quadrotor frame using the first law of Newton-Euler formulation

$$J\dot{w} + w \times Jw = -M_G + M_B \tag{3.5}$$

The first two terms of Equation 3.5 represent the rate of change of the angular momentum in the quadrotor frame. M_g represent the gyroscope moments due the rotors inertia J_r and M_b represent the moments applied to the quadrotor in the quadrotor frame.

The most important reason to deriving the rotational equation of motion in the quadrotor frame and not in the Earth frame , is that the inertial matrix is independent on time and the moments are expressed in the quadrotor frame in which they are known.

The gyroscopic moment of a rotor is a physical effect in which it tries to align the spin axis of the rotor along the Earth Z axis. They are defined as :

$$M_G = \omega \times [00J_r\omega_r] \tag{3.6}$$

where the ω is the relative speed of the rotors calculated as $\omega_r = \omega_1 + \omega_2 + \omega_3 + \omega_4$, and the J_r is the inertia of the rotors.

The inertial matrix of the quadrotor is a diagonal matrix because the symmetry produce that the off-diagonal elements are zero.

$$J = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \tag{3.7}$$

where the I_{xx} is the inertia with respect the pitch axis of quadrotor frame, the I_{yy} is the inertia with respect the roll axis of quadrotor frame and the I_{zz} is the inertia with respect the yaw axis of quadrotor frame.

The most important physical effects are the aerodynamic forces and moments produced by the rotors as a consequence of the rotation. The equations 3.8 and 3.9 show the force F_i and the moment M_i produced by the i rotor.

$$F_i = 0.5\rho A C_T r^2 \omega^2 \tag{3.8}$$

$$M_i = 0.5\rho A C_D r^2 \omega^2 \tag{3.9}$$

where the ρ is the air density, the A is the blade area, the C_T and the C_D are the aerodynamic coefficients, the r is the radius of blade and the ω_i is the angular velocity of rotor i.

This equations can be simplified using the assumption that the air density is constant which is true if the altitude is not so high.

$$F_i = k_F \omega^2 \tag{3.10}$$

$$M_i = K_M \omega^2 \tag{3.11}$$

where K_F is constant of the force and K_M constant of the moment, both can be determined experimentally for each motor. So the forces produce a moment in the X and Y axis of the quadrotor, it can be calculated as :

$$M_X = -F_2 l + F_4 l = l K_F(-\omega_2^2 + \omega_4^2) \tag{3.12}$$

$$M_Y = -F_1 l + F_3 l = l K_F(-\omega_1^2 + \omega_3^2) \tag{3.13}$$

The moments of the motors produce a moment in yaw axis of the quadrotor , it can be calculated as :

$$M_Z = M_1 - M_2 + M_3 - M_4 = K_M(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2) \tag{3.14}$$

Finally, combining the equations 3.12, 3.13 and 3.14 in vector form, the result is :

$$M_B = \begin{bmatrix} l K_F(-\omega_2^2 + \omega_4^2) \\ l K_F(\omega_1^2 - \omega_3^2) \\ K_M(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2) \end{bmatrix} \tag{3.15}$$

where l is the distance to motor with respect the center of the quadcopter frame.

3.2.2 Translational Equations of Motion

The translation equations of motion for the quadrotor are based on the second law of the Newton-Euler formulation. They are derived in the Earth frame :

$$m\ddot{r} = \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} + RF_B \tag{3.16}$$

where the $r = [x, y, z]^T$ is the quadrotor position, the m_q is quadrotor mass, the g is gravitational acceleration (g = 9.81 m/s^2), the F_B are the forces applied to the quadrotor.

The force applied are basically the sum of the thrust of the motors in yaw axis and it can be calculated as :

$$F_B = \begin{bmatrix} 0 \\ 0 \\ K_F(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2) \end{bmatrix} \tag{3.17}$$

F_B is multiplied by the rotation matrix R to obtain the force in the Earth axis, so it can be applied to any orientation of the quadrotor.

3.3 Space State

Formulating the mathematical model for the quadrotor into space state will help to simulate and make the control problem easier to manage.

3.3.1 State Vector

The state vector of the quadrotor is formulated as :

$$X = [x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}, x_{12}]^T \tag{3.18}$$

which is mapped to the degree of freedom of the quadrotor as the following :

$$X = [x, y, z, \phi, \theta, \psi, \dot{x}, \dot{y}, \dot{z}, \dot{\phi}, \dot{\theta}, \dot{\psi}] \tag{3.19}$$

The state vector define the position orientation linear velocity and angular velocity of the quadrotor.

3.3.2 Control Inputs

The control inputs are defined as the following :

$$U = [U_1, U_2, U_3, U_4] \tag{3.20}$$

where $U_1 = \omega_1^2$, $U_2 = \omega_2^2$, $U_3 = \omega_3^2$, $U_4 = \omega_4^2$

The ω_{MAX} is the maximum velocity achieve by the motors and it is divided to normalize the inputs, so the U_i is between 0 and ω_{MAX}^2.

3.3.3 State Space Representation

Using the equation of the rotational and lineal acceleration , it is possible to transform to the state space variable using the equations 3.5 to 3.20 . The result is the following :

$$\begin{aligned}
\dot{x_1} &= x_7 \\
\dot{x_2} &= x_8 \\
\dot{x_3} &= x_9 \\
\dot{x_4} &= x_{10} \\
\dot{x_5} &= x_{11} \\
\dot{x_6} &= x_{12} \\
\dot{x_7} &= K_F(U_1+U_2+U_3+U_4)(sin(x_4)sin(x_6)+cos(x_4)sin(x_5)cos(x_6))/m_q \\
\dot{x_8} &= K_F(U_1+U_2+U_3+U_4)(sin(x_4)cos(x_6)-cos(x_4)sin(x_5)sin(x_6))/m_q \\
\dot{x_9} &= -g+K_F(U_1+U_2+U_3+U_4)(cos(x_4)cos(x_5))/m_q \\
\dot{x_{10}} &= \frac{x_{11}x_{12}(I_{yy}-I_{zz})-x_{11}\omega_r(J_r)+lK_F(-U_2+U_4)}{I_{xx}} \\
\dot{x_{11}} &= \frac{x_{10}x_{12}(I_{zz}-I_{xx})+x_{10}\omega_r(J_r)+lK_F(U_1-U_3)}{I_{yy}} \\
\dot{x_{12}} &= \frac{x_{10}x_{11}(I_{xx}-I_{yy})+lK_M(U_1-U_2+U_3-U_4)}{I_{zz}}
\end{aligned} \tag{3.21}$$

3.4 Linearization of a quadrotor

The model deduced in 3.21 present several no linearities that are produced by three factors: the orientation (Pitch, Roll and Yaw), the control action and the angular velocities. To be able to work with the system in manageable way, it necessary to linearize the system. Therefore, it is proposed two method: a classical linearization around several set-points and a LPV linearization.

3.4.1 Linearization around set-points

This linearization require that the model can be expressed in the following way:

$$\begin{aligned}\dot{x} = A_{sp}(x - x_{sp}) + B_{sp}(u - u_{sp}) + G_{sp}(d - d_{sp}) + \dot{x}_{sp} \\ y = C_{sp}(x - x_{sp})\end{aligned} \tag{3.22}$$

Where the x is the state of the quadrotor, u is the control action, d is the disturbances and y is the output variable. The matrices $A_{sp}, B_{sp}, C_{sp}, G_{sp}$ are the linealized equation from 3.21. The x_{sp} is the set-point state about around is linealized the system.

The total system has been simplified removing the effects of the angular forces like the gyroscopic and the Coriolis-centripetal effects, since the motion of the quadrotor is assumed to be close the hovering position, this terms are much smaller than the main ones. The orientation variables are bounded from 0 to 360 degrees. So it possible to have all the no linear space discretized around a set-Keypoint which are selected to be uniformly distributed into the space state. However, the total number of subsystems will increase very quickly with the discretization size(n) which in a first approach is n^3 models. Also, it is possible to discretized in a smart way. For example, only the models that change the derivative in a significant way. Therefore, the matrices of the subsystems are:

$$A_{sp} = \left[\begin{array}{c|c} 0_{6x6} & I_{6x6} \\ \hline 0_{6x6} & 0_{6x6} \end{array}\right] \tag{3.23}$$

$$B_{sp} = \left[\begin{array}{c} 0_{6x4} \\ \hline B_1 \end{array}\right] \tag{3.24}$$

$$B_1 = \left[\begin{array}{c|c|c|c} \frac{K_F}{m_q}(s_4s_6 + c_4s_5c_6) & \frac{K_F}{m_q}(s_4s_6 + c_4s_5c_6) & \frac{K_F}{m_q}(s_4s_6 + c_4s_5c_6) & \frac{K_F}{m_q}(s_4s_6 + c_4s_5c_6) \\ \frac{K_F}{m_q}(s_4c_6 + c_4s_5s_6) & \frac{K_F}{m_q}(s_4c_6 + c_4s_5s_6) & \frac{K_F}{m_q}(s_4c_6 + c_4s_5s_6) & \frac{K_F}{m_q}(s_4c_6 + c_4s_5s_6) \\ \frac{K_F}{m_q}(c_4c_5) & \frac{K_F}{m_q}(c_4c_5) & \frac{K_F}{m_q}(c_4c_5) & \frac{K_F}{m_q}(c_4c_5) \\ 0 & \frac{-lK_F}{I_{xx}} & 0 & \frac{lK_F}{I_{xx}} \\ \frac{lK_F}{I_{yy}} & 0 & \frac{-lK_F}{I_{yy}} & 0 \\ \frac{lK_M}{I_{zz}} & \frac{-lK_M}{I_{zz}} & \frac{lK_M}{I_{zz}} & \frac{-lK_M}{I_{zz}} \end{array}\right] \tag{3.25}$$

$$C = \left[\ I\ \right] \tag{3.26}$$

3.4.2 Linearization LPV

This linearization aims to transform the nonlinear system into a linear system which depends on some measurable scheduling variables. Therefore, the system can be formulated as:

$$\dot{x} = A(\rho(t))x + B(\rho(t))u \tag{3.27}$$

where ρ(t) is the variable parameters.

The variable parameters ρ(t) are in function of scheduling variables p(t), which is recommend to be external to the system. However, the system variables are the scheduling variables, it is called quasi-LPV. The task of finding such functions for building a LPV systems are a complex task. The typical method is to hide the nonlinearities in parameters, known as the nonlinear embedding. As a consequence the nonlinear system are equivalent to the LPV system. The variation of the parameters ρ(t) is bounded using a bounded box that set the maximum and minimum of a function, which normally are called vertex on a polytopic representation. Moreover, it is possible to represent the system as a interpolation of the matrices evaluated in each vertex.

$$A(\rho(t)) = \sum_{i=1}^{N_v} \pi_i(\rho(t))A_i \tag{3.28}$$

Where π_i are the polytopic interpolators of each submodel A_i that depends on the scheduling variables and it weight distance between the vertexes. Moreover, the a of all π_i are equal to 1.

For this linearization, it is simplified only the centripetal force. So, the system equations for the quadrotor are the following:

$$\rho_1 = s_4s_6 + c_4s_5c_6, \rho_2 = s_4c_6 + c_4s_5s_6, \rho_3 = c_4c_5, \rho_4 = \dot{\phi}, \rho_5 = \dot{\theta}, \rho_6 = \dot{\psi} \tag{3.29}$$

$$A(\rho(t)) = \left[\begin{array}{c|c} 0_{6x6} & I_{6x6} \\ \hline 0_{6x6} & A_1 \end{array}\right] \tag{3.30}$$

$$A_1 = \left[\begin{array}{c|c} 0_{3x3} & A_2 \\ \hline 0_{3x3} & A_3 \end{array}\right] \tag{3.31}$$

$$A_2 = \left[\begin{array}{c|c|c} 0 & 0 & 0 \\ 0 & 0 & 0 \\ \dfrac{-g}{2max(|\rho_4|,\epsilon)sign(\rho_4)} & \dfrac{-g}{2max(|\rho_5|,\epsilon)sign(\rho_5)} & 0 \end{array}\right] \tag{3.32}$$

$$A_3 = \left[\begin{array}{c|c|c} 0 & \rho_6 \dfrac{I_{yy} - I_{zz}}{I_{xx}} & \rho_5 \dfrac{I_{yy} - I_{zz}}{I_{xx}} \\ \rho_6 \dfrac{I_{zz} - I_{xx}}{I_{yy}} & 0 & \rho_4 \dfrac{I_{zz} - I_{xx}}{I_{yy}} \\ \rho_5 \dfrac{I_{xx} - I_{yy}}{I_{zz}} & \rho_4 \dfrac{I_{xx} - I_{yy}}{I_{zz}} & 0 \end{array}\right] \tag{3.33}$$

$$B(\rho(t)) = \left[\begin{array}{c} 0_{6x4} \\ \hline B_1 \end{array}\right] \tag{3.34}$$

$$B_1 = \left[\begin{array}{c|c|c|c} \dfrac{K_F}{m_q}(\rho_1) & \dfrac{K_F}{m_q}(\rho_1) & \dfrac{K_F}{m_q}(\rho_1) & \dfrac{K_F}{m_q}(\rho_1) \\ \dfrac{K_F}{m_q}(\rho_2) & \dfrac{K_F}{m_q}(\rho_2) & \dfrac{K_F}{m_q}(\rho_2) & \dfrac{K_F}{m_q}(\rho_2) \\ \dfrac{K_F}{m_q}(\rho_3) & \dfrac{K_F}{m_q}(\rho_3) & \dfrac{K_F}{m_q}(\rho_3) & \dfrac{K_F}{m_q}(\rho_3) \\ 0 & \dfrac{-lK_F}{I_{xx}} & 0 & \dfrac{lK_F}{I_{xx}} \\ \dfrac{lK_F}{I_{yy}} & 0 & \dfrac{-lK_F}{I_{yy}} & 0 \\ \dfrac{lK_M}{I_{zz}} & \dfrac{-lK_M}{I_{zz}} & \dfrac{lK_M}{I_{zz}} & \dfrac{-lK_M}{I_{zz}} \end{array}\right] \tag{3.35}$$

$$C = \left[\begin{array}{c} I \end{array}\right] \tag{3.36}$$

3.5 Modelization of the airsoft marker

The launching of the bullets can be simplified as a force that has been position by the gimbal following the euler angles of it. Moreover, it is assumed that the launching don't affect to the orientation of the gimbal. The force introduce into the system a moment for the misalignment with the center of mass of the quadrotor. Therefore, the figure 3.2 show the euler angles (Roll, Pitch and Yaw) used for describe the orientation of the gimbal, hence, of the force also. The axis are only shifted in the yaw axis the distance l_g.

The disturbances generated by the force can be introduce into the space state system deduced in 3.21. The resultant matrix is the following:

$$G_{sp} = \left[\begin{array}{c} 0_{6x1} \\ \hline G_1 \end{array}\right] \tag{3.37}$$

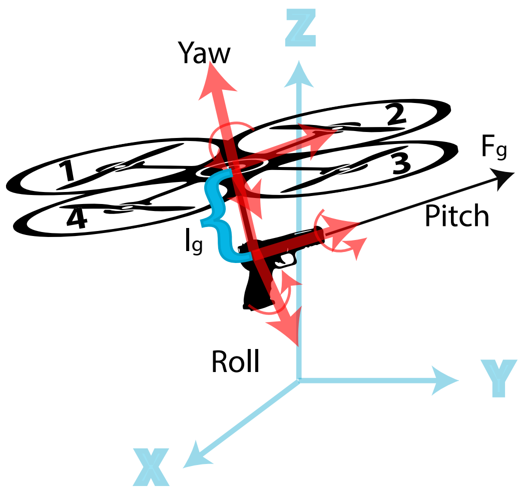


FIGURE 3.2: Gimbal schematic

$$G_1 = \begin{bmatrix} \frac{F_g}{m_q}(sin(x_4+\phi_g)sin(x_6+\psi_g)+cos(x_4+\phi_g)sin(x_5)cos(x_6+\psi_g)) \\ \frac{F_g}{m_q}(sin(x_4+\phi_g)cos(x_6+\psi_g)+cos(x_4+\phi_g)sin(x_5)sin(x_6+\psi_g)) \\ \frac{F_g}{m_q}(cos(x_4+\phi_g)cos(x_5)) \\ \frac{l_gF_g}{I_{xx}}sin(\phi_g)cos(\psi_g) \\ \frac{l_gF_g}{I_{yy}}sin(\phi_g)sin(\psi_g) \\ 0 \end{bmatrix} \tag{3.38}$$

The marker is capable of generate that force by creating a differential pressure against the atmosphere in the launching chamber. The pressure of the chamber, which is the generated by a compressor in the marker, is responsible of the dynamic of the force applied to the bullet. Moreover, it is normally very complex to evaluate, however it can be determined experimentally using a pressure sensor or an accelerometer. Therefore, the equation that relate the force with the pressure is the following:

$$F_g = (p_{chamber} - p_{atmosphere})A_{chamber} \tag{3.39}$$

Where the F_g is force applied to the bullet, the $p_{chamber}$ is the pressure of the chamber, the $p_{atmosphere}$ is the pressure of the atmosphere and the $A_{chamber}$ is the inner area of the canon.

Chapter 4

Parameters estimation

This chapter will explain the process to estimate each parameter of the model according the known information and their nature. Theses can group in the following categories:

- Basic measurements
- Geometry derivation
- Experimental data extrapolation

Additionally, it will be tested the model with the estimated parameters to validate if the model is similar to the reality.

4.1 Basic measurements

This part illustrate how are done the basic measurements for the estimation of the parameters. Mainly, it was used two kinds of instruments for measure the mass and length. The mass was estimated using a digital balance ”Anpro" due to the size of the quadrotor was quite complex to use the standard balance available in the laboratory. This kind of balance allow to attach for one side the drone and measure with a precision of 10 g, which was completely reasonable.

FIGURE 4.1: Balance digital

The length was estimated using measure tape, due to the measure range needed. The distance to measure are two: the distance to the center of mass of the quadrotor to the rotors and the distance from the center of the quadrotor to the center of mass of the gimbal.

Parameter	Value
m_q	4,23 kg
l	332 mm
l_g	274 mm

TABLE 4.1: Basic Parameters

4.2 Geometric derivation

This part show the process to obtaining the position of the center of mass and the values of the inertia matrices. There are two different approaches for estimate these values, either theoretically or experimentally. The experimental approach should be more accurate, since it gives the values for the real quadrotor. However, performing the necessary experiments to estimate these parameters on a full size quadrotor can be complex and non cost-effective. Therefore, the theoretical approach was preferred. The procedure which is more effective is to use a commercial CAD software packages.

The selected software to create the CAD model is SolidWorks 2017, which allow to define each component with their corresponding material. Therefore, the program is able to calculate the mass, volume, surface area, centre of mass, the principal moments of inertia and their orientation. Additionally, the components created can be assembled to create different components that can be calculate the previous properties. However the mayor disadvantage is to generate good results is to have and accurate known of the dimensions and their material. This information was obtain thanks several measurements on every component of the quadrotor.

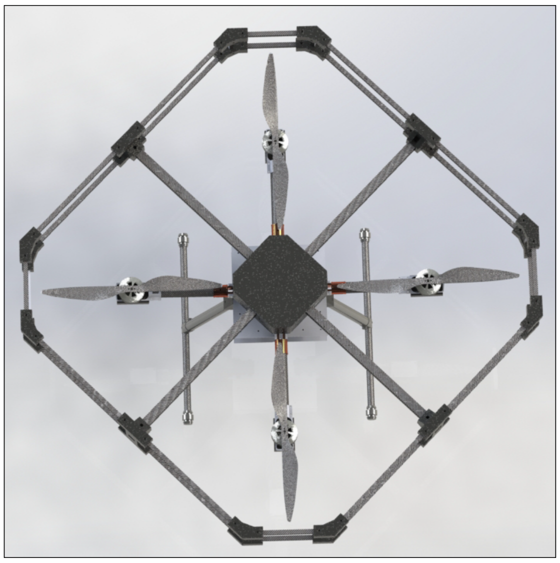


FIGURE 4.2: Quadrotor CAD

Parameter	Value
I_{xx}	287885,411 Kg mm^2
I_{yy}	500747,953 Kg mm^2
I_{zz}	292895,098 Kg mm^2
J_M	645 $Kgmm^2$

TABLE 4.2: Geometric derivation parameters

4.3 Experimental derivation

This part show the process to obtaining the constant of the force and the moment of the rotors. Also, it detail the set-up used to carry out to experimental tests, which are based in collected data for estimating the constant that express the forces and the moments. Additionally, the experiment was done only in one rotor, although the uniques of each motor, to avoid the stress of others rotors and for simplicity. Therefore, it was used two different set-ups: one for estimate the constant force, and the other for estimate the constant of the moment.

To carry out the force test, a test bench has used, consisting of a cylindrical bar and a strain gage. The data collected here is the thrust of the rotors while they are regulated by the ESC, to know what relation of the thrust there is for each of the ESC inputs.

To carry out the moment test, a similar set-up has used, with the difference that the force that it is need to measure is the lateral one. Therefore, it is used the test bench than the force test, but the bench is connected through a bar that is doing like a lever. The distance to the center of the lever is the same for both rotor and test bench.

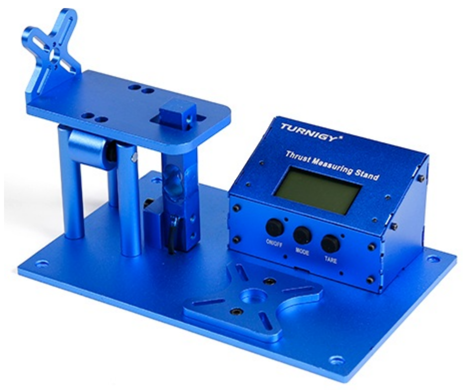


FIGURE 4.3: Test bench

Another important condition is that all the test receive the same voltage. For this, a 24 V power supply is used, which is capable of feeding the motors with the maximum intensity, that is 23A. Most commercial sources that can be found that only generate between 2-3 A, but a LED source is cheaper and is able to give more than 15 A (500 W source). So it is chose to use a led source. This avoids the use of the battery, since as it is used, its voltage decreases and makes it difficult to guarantee the same conditions for all tests.

FIGURE 4.4: Power source led

With this set-ups, the table 4.3 was generated :

Input (%)	Force bench (N)	Moment bench (N)	Angular speed (rpm)
0	0	0	0
5	1501	27	1409
10	3178	57	2017
15	4650	84	2492
20	6004	108	2781
25	7887	142	3068
30	9035	163	3295
35	10919	197	3430
40	12047	217	3775
45	13969	251	4062
50	15784	284	4338
55	16049	289	4591
60	18443	332	5075
65	18933	341	5421
70	21798	393	5652
75	22033	397	5829
80	23622	425	6136
85	25349	457	6301
90	27458	494	6502
95	28537	514	6674
100	31343	564	6841

TABLE 4.3: Rotors experimentation

The figure 4.5 show the regression done to calculate the constants K_F, which is deduced from the equations seen in chapter 3. The vertical force generated for the rotor is proportional to squared angular speed. Therefore, it is directly the slope of the regression.

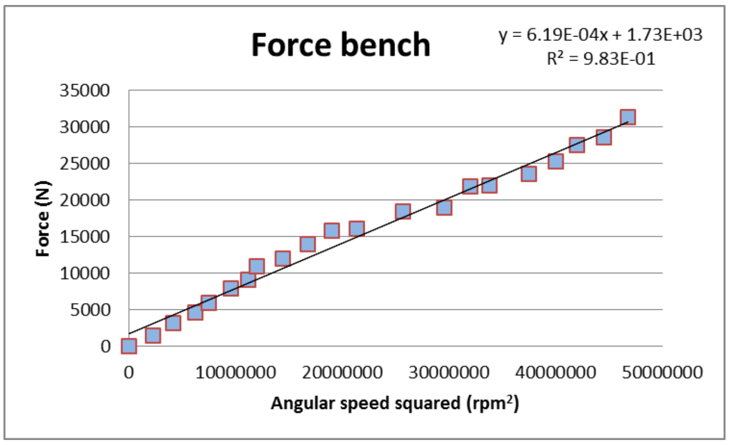


FIGURE 4.5: Regression force bench

In the same way, it is calculated the constant K_M using the regression obtained from the data of the moment bench, which is presented in figure 4.6

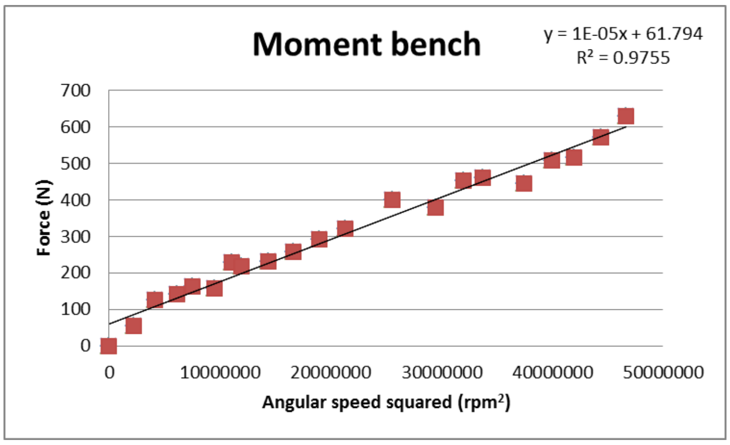


FIGURE 4.6: Regression moment bench

Parameter	Value
K_F	6.19 10^{-4} N$/rpm^2$
K_M	10^{-5} N$/rpm^2$

TABLE 4.4: Experimental parameters

4.3.1 Airsoft marker

The produce to estimate the force generate by the marker was to attach a accelerometer to measure their evolution. However, due to the fast dynamics of a marker it is necessary an accelerometer that can work at least 200 Hz. Fortunately, it was accessible an MTI-100 of xsens capable of work at 2 Khz, although it was necessary a deal between accuracy and sampling rate. At 200 Hz is enough as it is shown in the figure 4.7. The set-up was simple, only attach the accelerometer to the marker and align it with the canon axis.

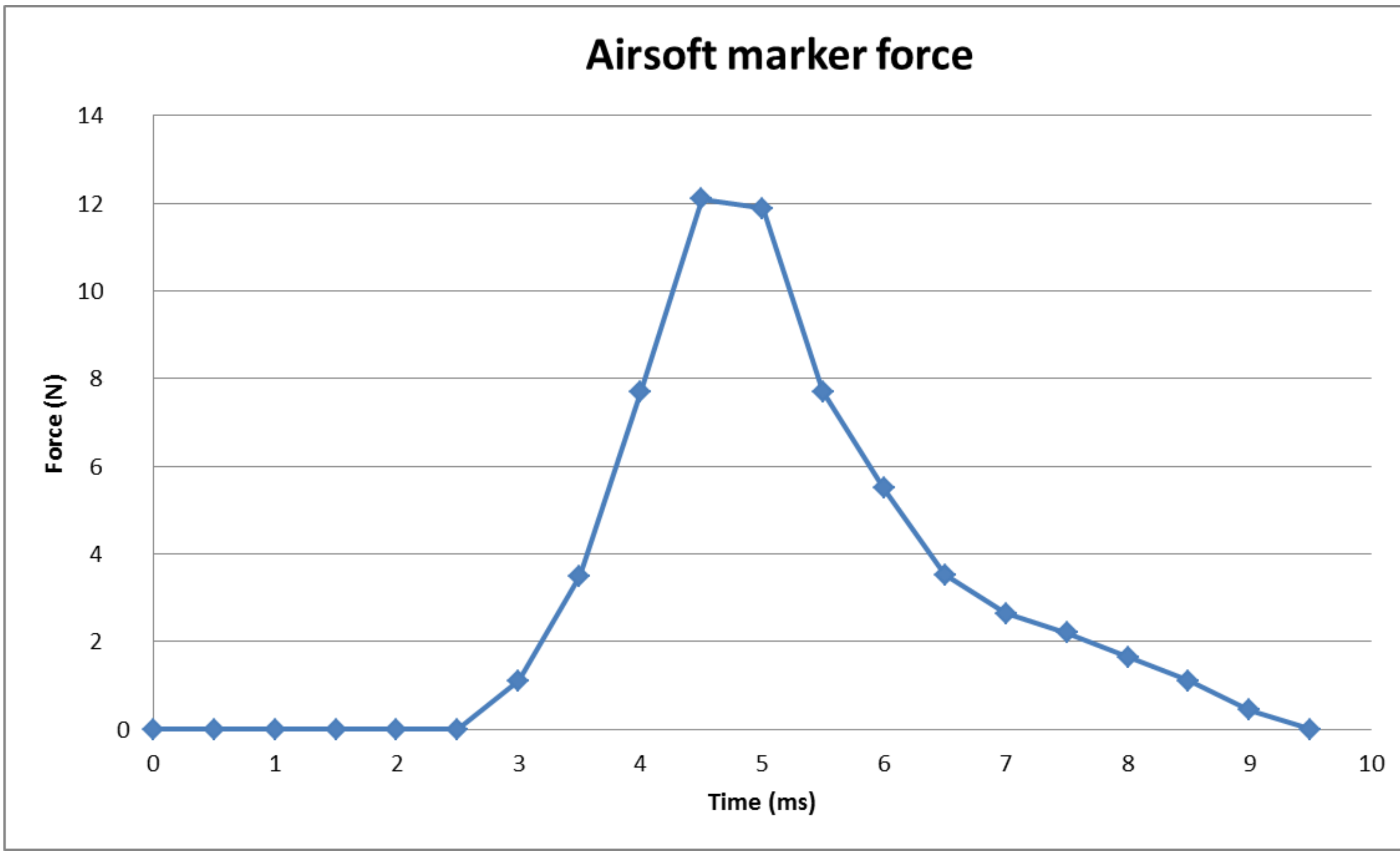


FIGURE 4.7: Force generated by the marker

4.4 Validation of the models

This section try to verify if the models proposed in the chapter 3 with the parameters estimated are close to the reality. However, this verification need that the quadrotor works without any controller in open loop. To achieve this task, was build a bench test that allow the quadrotor only to rotate in one axis. The main disadvantage was that probe was only possible to roll and pitch and not for yaw. Moreover, the bench structure add more inertia to the system leading a behavior more distance from the reality. The test bench has a encoder that allow to register the current angle of the test with great accuracy (0.1º). Additionally, the test has to be attached to ground with screws, because the moment of the quadrotor generate a dangerous vibrations.

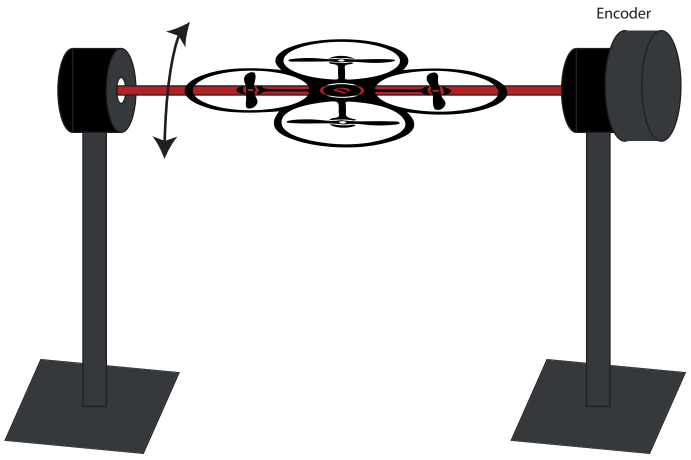


FIGURE 4.8: Test bench quadrotor

The models compared with the real behavior are the linearized, because are the ones which are going to be used in the controller implemented. The PWA (set-points) is discretized using 400 points for angle and the inputs value is assigned to get maximum angular velocity in each axis:

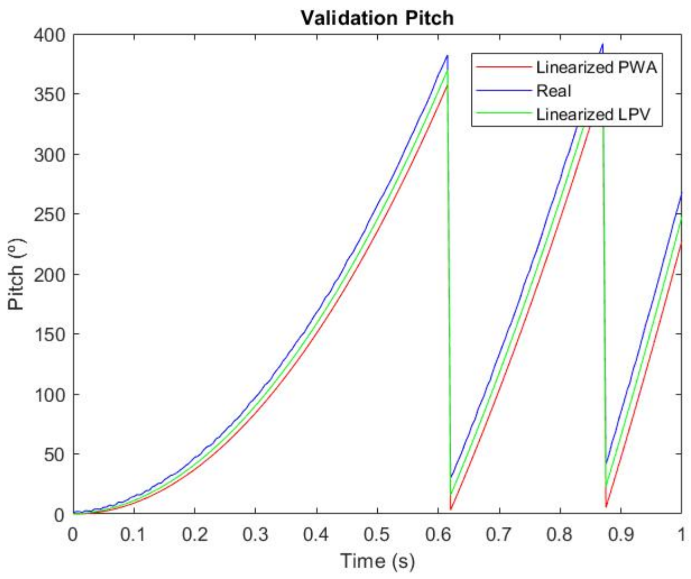


FIGURE 4.9: Validation pitch

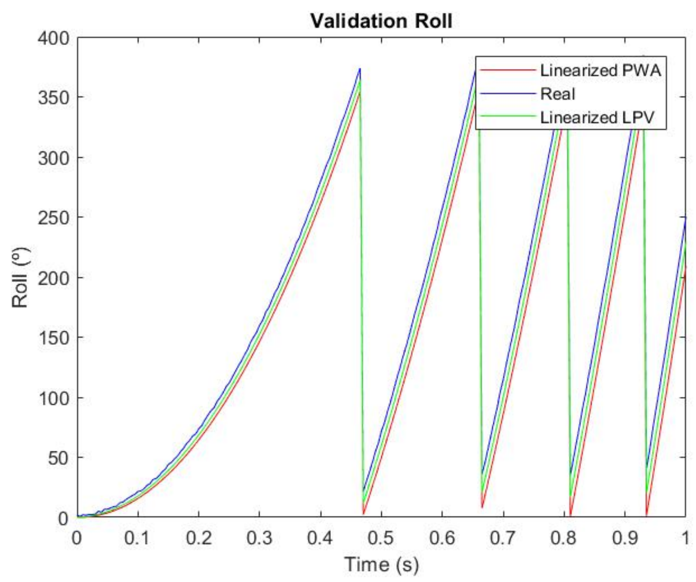


FIGURE 4.10: Validation roll

The validation show that both models are good enough, although the quadrotor was only simulated 1 second due to the know fact that drag force was not included in the model and in the end the data start to diverge for both models. In real operation is not an important fact, due to the fact that it need slow velocities in quadrotor to be able to perform their activities. It is important to remark that the PWA is good with a lot of points, however the price to pay is the quantity of models needed to achieved. Otherwise, the data obtained is stepped and very inaccurate.

Chapter 5

Control algorithms

This chapter review the literature about the proposed controllers for the quadrotor that are able to compensate the effects of shooting the marker: a PID feed-forward and a model predictive controller.

First of all, it is explained the actual PID with the feed-forward improvement for the four control loops that have the quadrotor. Secondly, it is explained the proposed cost function and the restrictions for the model predictive controller. Additionally, it is presented a method to make it robust against unknown bounded disturbances.

5.1 PID

The most popular algorithm to control in the industrial are the PID. Mainly, because it has a lot of advantages:

- Good performance for a lot of process.
- Relative simple architecture.
- Tunning the parameters without knowing the mathematical model of the system.

In the control field, the PID is most simple technique that you can use. Therefore, it exist a lots of different techniques that give better performance and stability, but are far more complex.

The main equation of a PID is the following:

$$u(t) = K_p e(t) + K_I \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt} \tag{5.1}$$

Where u(t) is the control variable, e is the the error between the desired state r(t) and the actual state, K_P is the proportional constant, K_D is the derivative constant and K_I is the integrative constant.
The figure 5.1 show the basic diagram of the 3 components of a PID and how is connected to

the the process.

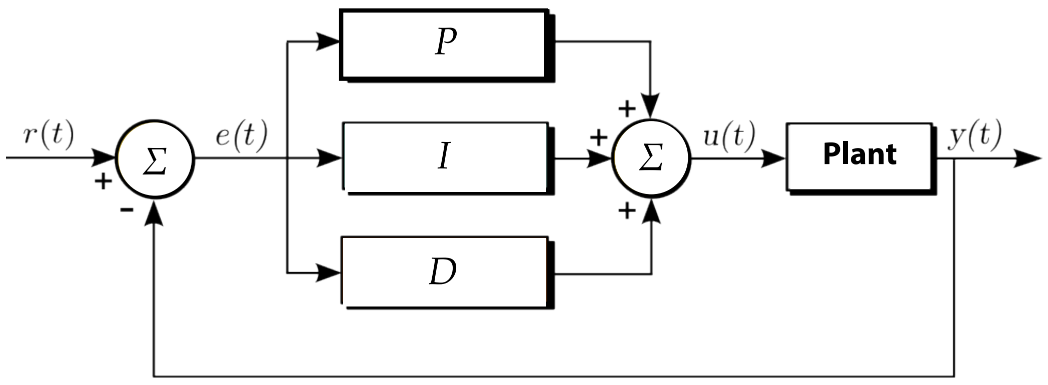


FIGURE 5.1: PID architecture

The PID has three components: the first term is proportional (P), that is proportional to the magnitude of the angular error, and helps achieve the desired angle. The second term is integral (I), that is proportional to the amount of angular error accumulated over time, and helps compensate for disturbances such as wind, although introduce into the system overshoot and increase the setting time. The third term is the derivative (D), that is proportional to the rate of change of angular error and resists quadrotor movements, it help to reduce the overshoot and reduce the setting time.

The complex in this type of controller is adjusting the three constants. Normally, flight controllers have different sets of precalculated constants that allow different fight mode. Additionally, the manual tuning for these constants in a quadrotor can be a bit complicated and requires a lot experience in the field. Probably, it will help to use a methodology like the proposed by the Ziegler–Nichols, which is based to put the system in a oscillatory mode, only tuning the proportional constant and the rest equal to 0. However, this has the problem of dangerous oscillations. this can be approach in the laboratory with a fix setup, that allow only to rotate in one axis the quadrotor.

The classical approach to deal with disturbances in the PID, as to two main ways: the feedback and feedforward. The feedback is the classical one, when you are not able to know anything about the disturbances, so it is impossible to know the model of the disturbance, so it is managed as a error seen in 5.1. However, if it is known the model of the disturbance, it is possible to compensate the effects of it. The figure 5.2 shows the architecture of the PID with feedback compensation. The d(t) represent the activation signal from the second processing unit which directly excites the airsoft marker and their modelization.

Therefore, this technique of compensation is applied to each controller of the different axis:

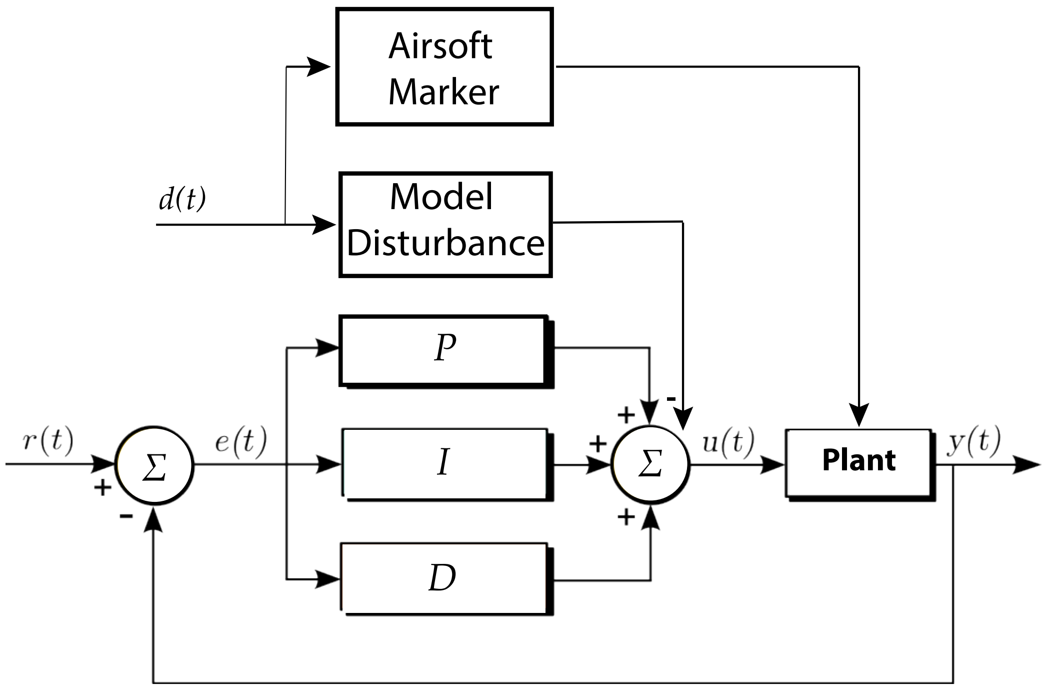


FIGURE 5.2: PID with feedforward compensation

The change with respect the standard PID are simple, the derivative signal dy(t) is directly get from the sensors, so there is no need to derivative it, the integral is saturated to avoid no linearities. If it is not saturated, it can provoke a decrease of performance, waiting to "discharge" the integral part, when the integral value is large and a change of sign in the error happen. Finally, it is subtracted the effect of the disturbance from the control signal.

5.2 Model predictive controller

The model predictive controller is a technique that has gain a lot popularity with the years, due to the capabilities to deal with complex dynamical systems in order to integrate constrains in a easy way and define multi-objective cost function. Using this, it is possible to describe the desired behavior of the system. However, it is critical to have a very accurate model of the system, that could be lineal or no lineal, deterministic or stochastic, or continuous, discrete or hybrid. Another important point to remark, it is the convexity of the cost function, because it make the optimization problem to converge easier and faster.

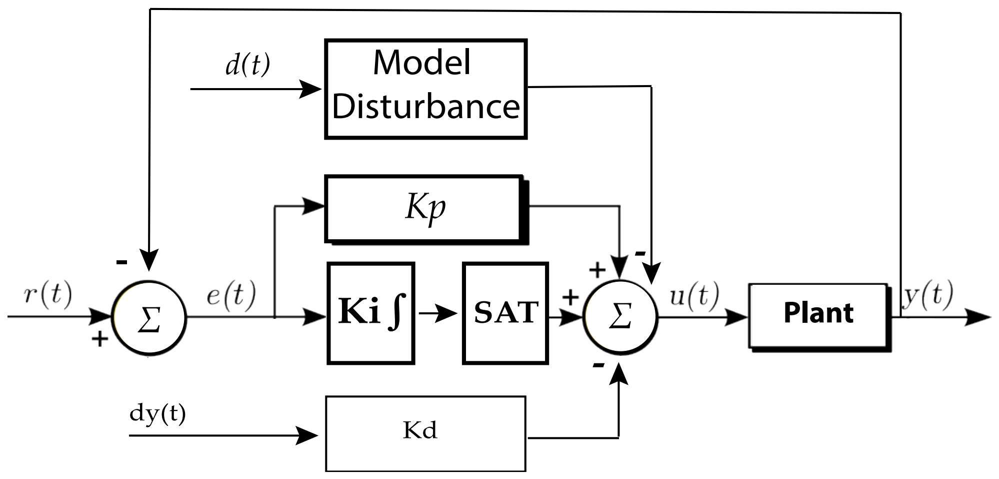


FIGURE 5.3: PID implemented in each axis

5.2.1 Problem statement

The standard MPC uses the past and present information about the system measures and control inputs over a finite period of time solve a open-loop optimization problem to get the sequence of inputs that make minimum the cost function respecting the constrains. After solving the problem, it only used the first control action and the rest is discarded. Afterwards, the produce start again.

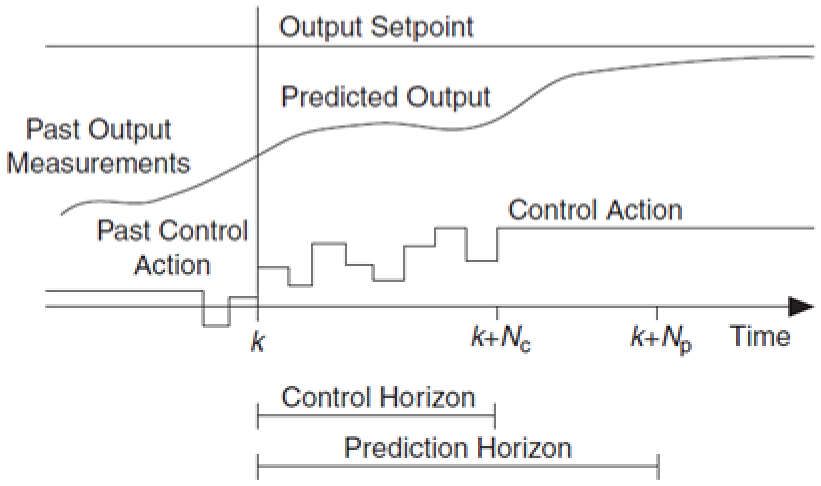


FIGURE 5.4: MPC architecture

The cost function is presented as the following:

$$J* = min \sum_{i=0}^{Np-1} F(x_{k+i}, u_{k+i}) \tag{5.2}$$

Where the F function describe how good if achieve the desired state in function of the state and the input.

We consider a class of system described by the following nonlinear set of differential equation,

$$\dot{x} = f(x(k), u(k)) \tag{5.3}$$

The equation 5.3 is subject to input and state constraints in the way:

$$u_{min} <= u <= u_{max}$$

$$x_{min} <= x <= x_{max}$$

Where the u_{min}, u_{max} are the constant that bound the minimum and the maximum value of the control action. Also the x_{min}, x_{max} are the constant that bound the minimum and the maximum value of the state.

5.2.2 Quadrotor case

The quadrotor is a system that is no lineal as has deduced for the equations of the chapter 3. Therefore, to use the standard toolkits for model predictive controller, like yalmip, it is necessary to linealize the system around several set-points. The linealized version of the quadrotor is a more easy to deal with it and to compute. However, it is possible to directly work with the no linear system using another toolkit, called ACADO. It has the main disadvantage of the need speed to solve the system.

Therefore, it is need to define a function F which describe how good is achieve our objective that is minimum error, calculated as the desired state minus the actual state, without caring a lot about the energy wasted.

$$F = Q(x_d - x)^2 + Ru(k) \tag{5.4}$$

Where Q and R is the matrix that give the importance between the desired state achievement or the energy of the control input. The typical value for a quadrotor is Q = diag(0.9) and R = diag(0.1).

The system equation to use are peace-wise linearization around several set-points and the LPV linearization. The constrains of the system are given of the maximum rotation of the motors and the definition of the speed zone:

$$\begin{aligned} W_{MAX}^2 &\geq u_1 \geq 0 \\ W_{MAX}^2 &\geq u_2 \geq 0 \\ W_{MAX}^2 &\geq u_3 \geq 0 \\ W_{MAX}^2 &\geq u_4 \geq 0 \\ PitchVel_{max} &\geq x_{10} \geq -PitchVel_{max} \\ RollVel_{max} &\geq x_{11} \geq -RollVel_{max} \\ YawVel_{max} &\geq x_{12} \geq -YawVel_{max} \end{aligned} \tag{5.5}$$

The definition of the speed zone is necessary to be sure that the quadrotor is working in a zone which has been linearited with a maximum of error.

5.2.3 MPC LPV

The particularity of this kind of MPC is the need to know the scheduling variables to able to give the value of the matrices for the prediction horizon. The trick is assume that are in the trajectory that is the reference to known. However, it has the disadvantage that generate bad matrices if the system are no close to the reference trajectory, leading to a poor performance. An interesting property that can be probed is the stability of the MPC, due to the fact that general MPC don't have the guarantee of stability, they will try to calculate the best control actions, but without having a formal proof, as is explained in the paper of [1]

5.2.4 Robust MPC

The particularity of this kind of MPC is that they try to reformulate the problem as minmax problem. In a straightforward approach, it is proposed a metric, called MPPM (Minimum Peak Performance Measure), which ensures the minimization of the worst–case deviation along the predicted trajectory, the problem can be formulated as a minimization of the control signal to the maximum disturbances that is take into account. However, the open–loop formulation is intractable to solve exactly especially when the complexity increase. They proposed a method called, feedback predictions, that reduce the complexity of the problem giving a close approximation. The toolbox of yalmip has implemented some of their ideas. This is explained in detail in [2].

Chapter 6

Simulation

In this chapter, the controller explained in the previous chapter, the MPC LPV and a robust MPC, are going to be implemented in Matlab 2017 with Matlab programming language using the toolbox of optimization, called YALMIP. Moreover, it is necessary to select a solver to solve the optimization problem. For this, it is used SeDuMi. It is an open-source solver and it is free. However, it is necessary to compile with Matlab in c++.
A series of scenearios was proposed to test the performance of the controllers:

- A one dimension reference scenario.
- A two dimension reference scenario.

6.1 One dimension reference scenario

This scenario consist on a set of fixed reference on the Z axis that change with the time. The objective has to use a simple scenario to see the result of the controller and see the temporal response.

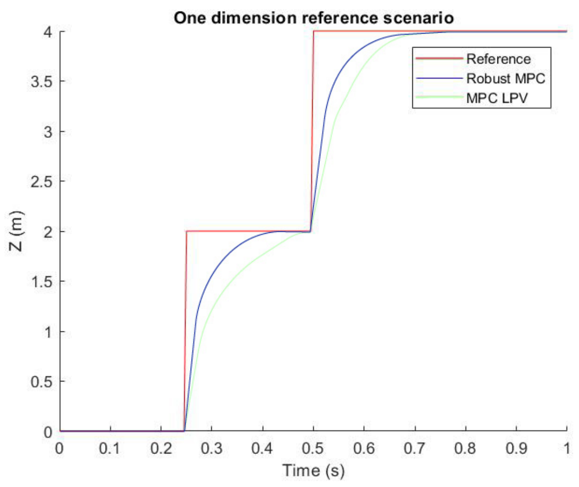


FIGURE 6.1: One dimension reference scenario

The result was quite predictive, the performance of the LPV MPC is better than the robust MPC, their setting time was faster and with and smooth response, due to the model that are using. Moreover, the Robust MPC need more time to compute the control actions around (10 min), however the LPV MPC only need 3 min.

6.2 Two dimension reference scenario

This scenario consist on circle reference in the plane XZ. The only way to make the circuit is having either pitch and yaw velocity or roll and yaw velocity, with the objective to see if the MPC robust was able to work due to an unmodeled dynamics.

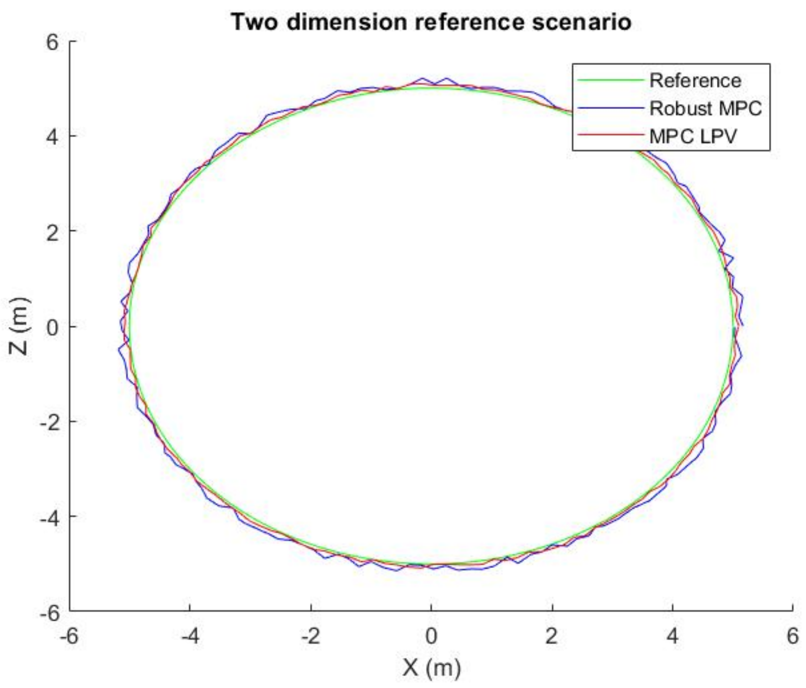


FIGURE 6.2: Two dimension reference scenario

The result has quite satisfactory, the MPC robust was capable of working with the cross product of the angular velocity and the LPV MPC also considering that their model already include this non lineality. Therefore, the performance of the LPV MPC is far better than the robust, although the overall results were close to the reference.

Chapter 7

Conclusions

In this chapter, it is sum up the main contributions of the thesis. Additionally, it also propose some of the possible improvement in a future research.

7.1 Conclusions

The main objective of the thesis was to build the mathematical model of the qudrotor, propose different types of linearization of the model in order in to use a MPC controller. Additionally, the parameters of the quadrotor was estimated using different methods depending their nature. Moreover, it presented different types of MPC controllers and it is simulated.

The mathematical model is derivaded from the Newtom-Euler formulation applying the standard forces acting above the quadrotor. However, the resultant model presented hard non-linealities. To fight against it, the mode is linealized using two different models: linealized and the linear parameter variant (LPV). Moreover, it has been proposed a modelization of the effects of the airsoft marker.

The estimation of the parameters of the quadrotor was tedious. The calculus of the inertia matrix was done using the CAD software, generating 36 different pieces. The experimental parameters require an specific set-up which is different for each parameter. Also, the validation require an additional set-up to check if the models with the estimated parameters are close to the reality.

The theory of the PID and MPC is reviewed to understand and select the correct type of the controller. The proposed MPC strategy are the Robust MPC and the MPC LPV. Both are implemented in matlab using the toolbox of YALMIP and they are testd to check their performance in different scenarios. The LPV was proved to have better performance.

7.2 Future Improvements

The proposed future work are the features that don't have enough time to do it in the scope of this project. Therefore, they are:

- It has been considered that all the states are measured. However, it is not true in a real quadrotor, so an observed should be designed.
- It should be implemented in the real system.
- Demonstrate the theoretical stability of the MPC LPV.

Bibliography

[1] Eugenio Alcala et al. “Gain Scheduling LPV Control for Autonomous Vehicles including Friction Force Estimation and Compensation Mechanism”. In: (Apr. 2018).

[2] Kostas Alexis et al. *Robust explicit model predictive flight control of unmanned rotorcrafts: Design and experimental evaluation*. June 2014.

[3] Inkyu Sa andPeter Corke. “System identification, estimation and control for a cost effectivevopen-source quadcopter”. In: *IEEE International Conference on Robotics and Automation, ICRA 2012, 14-18 May, 2012, St. Paul, Minnesota, USA*. 2012, pp. 2202–2209. URL: http://dx.doi.org/10.1109/ICRA.2012.6224896.

[4] S.S. Craciunas et al. “The JAviator: A High-Payload Quadrotor UAV with High-Level Programming Capabilities”. In: *Proc. of the AIAA Guidance, Navigation, and Control Conference (GNC)*. Honolulu, HI, USA, 2008.

[5] Zhijian He et al. “Ard-mu-Copter: A Simple Open Source Quadcopter Platform”. In: *11th International Conference on Mobile Ad-hoc and Sensor Networks, MSN 2015, Shenzhen, China, December 16-18, 2015*. 2015, pp. 158–164. DOI: 10.1109/MSN.2015.9. URL: http://dx.doi.org/10.1109/MSN.2015.9.

[6] Minh Quan Huynh, Weihua Zhao, and Lihua Xie. “L1 adaptive control for quadcopter: Design and implementation”. In: *13th International Conference on Control Automation Robotics & Vision, ICARCV 2014, Singapore, December 10-12, 2014*. 2014, pp. 1496–1501. DOI: 10.1109/ICARCV.2014.7064537. URL: http://dx.doi.org/10.1109/ICARCV.2014.7064537.

[7] Clemens D. Krainer. “JNavigator - An Autonomous Navigation System for the JAviator Quadrotor Helicopter”. Master’s Thesis. Salzburg, Austria: Department of Computer Sciences, University of Salzburg, 2009.

[8] Huu-Khoa Tran and Juing-Shian Chiou. “PSO-Based Algorithm Applied to Quadcopter Micro Air Vehicle Controller Design”. In: *Micromachines* 7.9 (2016), p. 168. DOI: 10.3390/mi7090168. URL: http://dx.doi.org/10.3390/mi7090168.

[9] Rainer Trummer. “Design and Implementation of the JAviator Quadrotor: An Aerial Software Testbed”. PhD Thesis. Salzburg, Austria: Department of Computer Sciences, University of Salzburg, 2011.