ISSN 0280-5316
ISRN LUTFD2/TFRT--5823--SE

Modelling, Identification and Control of a Quadrotor Helicopter

Tommaso Bresciani

Department of Automatic Control
Lund University
October 2008

Lund University
Department of Automatic Control
Box 118
SE-221 00 Lund Sweden

Document name
MASTER THESIS

Date of issue
October 2008

Document Number
ISRN LUTFD2/TFRT--5823--SE

Author(s)
Tommaso Bresciani

Supervisor
Daniele Caltabiano and Roberto Sannino,
STMicroelectronics in Milano, Italy
Anders Rantzer Automatic Control, Lund (Examiner)

Sponsoring organization

Title and subtitle
Modelling, Identification and Control of a Quadrotor Helicopter (Modellering, identifiering och reglering av en quadrotor helikopter)

Abstract
This thesis work focused on the study of a quadrotor helicopter. The dynamic system modelling and the control algorithm evaluation were carried out. To test the results, a simulator and a real platform were developed.
The Newton-Euler formalism was used to model the dynamic system. Particular attention was given to the group composed of the DC-motor, the gear box and the propeller which needed also the estimation of aerodynamic lift and torque to reach better accuracy. PID control algorithms were compared. The first stage tests were performed on a simulated model where it was easy to evaluate the performance with a mathematical approach. The second stage tests were carried out on the quadrotor platform to evaluate the behavior of the real system. A simulator based on Matlab-Simulink was developed. With this program it was possible to test the accuracy of the model and the robustness of the control algorithms. Furthermore a 3D graphic output and a joystick interface made easier the testability and the observability of the system.
A quadrotor platform was developed. The electronics was composed of a Micro Control Unit (MCU) interfaced with several devices: the power supply, the receiving unit, the DC-motor power boards, the Inertial Measurement Unit (IMU), the SONAR and the IR modules. Thanks to these devices and the MCU software, both guided and autonomous flights were possible.

Keywords
Quadrotor, UAV, VTOL, PID Control, System Identification

Classification system and/or index terms (if any)

Supplementary bibliographical information

ISSN and key title
0280-5316

ISBN

Language
English

Number of pages
180

Recipient's notes

Security classification

http://www.control.lth.se/publications/

"Everybody knows that some things are simply impossible
until somebody, who doesn't know that,
makes them possible."

A. Einstein

"Alla inser att några saker är helt omöjliga
tills någon, som inte vet,
gör dom eventuella."

A. Einstein

"Tutti sanno che una cosa è impossibile da realizzare
finchè arriva uno sprovveduto che non lo sa
e la inventa."

A. Einstein

Acknowledgments

- I would like to thank my thesis supervisors, Prof. Andrea Bonarini and Prof. Anders Rantzer, for their insight and support throughout the duration of this thesis. I would also like to thank Prof. Marco Lovera for his advice on flight dynamics.

- Thank you to my company supporting supervisors, Eng. Daniele Caltabiano and Eng. Roberto Sannino, for their helpful solutions and competence in this project. Furthermore I want to thank ST Microelectronics for giving me the chance to carry out this experience.

- Many thanks to my family for the patience and the support in all my choices, no matter what they were. In particular, I thank my partner Eleonora for her company and understanding during this thesis activity.

- Last but not least, I would also like to thank my friends and fellow students for the great time spent together and the experiences shared.

Contents

Abstract **I**

Populärvetenskaplig sammanfattning **III**

Estratto in lingua italiana **V**

Acknowledgments **VII**

Contents **IX**

Preface **XIII**

1 Introduction **1**

2 State of the art **5**

3 Quadrotor model & system **7**

3.1 Basic concepts 8

3.2 Newton-Euler model 12

3.3 DC-motor 22

3.4 System architecture 28

4 Control algorithms **31**

4.1 Control modelling 32

4.2 PID techniques 36

5 Quadrotor simulator **41**

5.1 System structure 42

5.2 Blocks implementation 44

5.3 3D visualization 48

6 Identification of the constants **49**
6.1 Basic measurements 50
6.2 Geometry and dynamics derivation 53
6.3 Geometry and aerodynamics derivation 62
6.4 Datasheet data extrapolation 64
6.5 Experimental data extrapolation 66
6.6 Constants comparison 78

7 Sensors: SONAR, IR & IMU **79**
7.1 SONAR module 80
7.2 IR module 83
7.3 IMU 87

8 Actuators: motor drivers **91**
8.1 Schematic design 92
8.2 Simulations 96
8.3 PCB design 98
8.4 Realization & final tests 99

9 STR730 Micro Control Unit **101**
9.1 STR730 MCU overview 102
9.2 Peripherals organization 106
9.3 Software architecture 109

10 Experimental results **111**

11 Conclusion **117**

A Kinematics and Dynamics **119**
A.1 Kinematics 119
A.2 Dynamics 124

B Linear regression **127**

C Aerodynamics calculation **129**
C.1 Momentum theory 129
C.2 Blade element theory 132

D Moment of inertia calculation **135**

E Acronyms and abbreviations **143**

F Units of measurement **147**

G List of prefixes **149**

H List of constants **151**

I List of variables **159**

Bibliography **165**

Index **169**

Preface

In this study, development and modelling of a quadrotor helicopter were performed. The main activities can be divided into four groups. The *dynamic system modelling* needed to be examined to understand the evolution of the forces in play. The *control algorithm evaluation* pointed out the stability and robustness using several control laws. The *Matlab simulator* was a good tool to test the correctness and the accuracy of the model and the control algorithms. Finally the *real platform* gave the chance to verify the calculations and simulations on a mechanical structure. This activities were conducted from October 2007 to October 2008.

The project was carried out by myself, Tommaso Bresciani, as thesis trainee, Eng. Daniele Caltabiano and Eng. Roberto Sannino as Company supporting supervisors. Most of the studies were performed in the AST Robotics group, ST Microelectronics, Agrate, Italy. Two universities cooperated in this activity: Politecnico di Milano, Italy and Lunds Tekniska Högskola, Sweden. Many thanks to Prof. Andrea Bonarini (my thesis supervisor from Italy) and Prof. Anders Rantzer (my thesis supervisor from Sweden) for their contribution. I would also like to thank Prof. Marco Lovera for his advices on flight dynamics.

Agrate, October 2008.

Tommaso Bresciani

Chapter 1

Introduction

This thesis work focuses on the study of a Vertical Take-Off and Landing (VTOL) Unmanned Aerial Vehicle (UAV). The proposed structure is a four-propeller helicopter called quadrotor.

In these last years, a growing interest has been shown in robotics. In fact, several industries (automotive, medical, manufacturing, space, ...), require robots to replace men in dangerous, boring or onerous situations. A wide area of this research is dedicated to aerial platform.

Several structures and configurations have been developed to allow 3D movements. For example, there are blimps, fixed-wing planes, single rotor helicopters, bird-like prototypes, quadrotors, ... Each of them has advantages and drawbacks. The Vertical Take-Off and Landing requirement of this project exclude some of the previous configuration. However, the platforms which show this characteristic have unique ability for vertical, stationary and low speed flight. The quadrotor architecture has been chosen for this research for its low dimension, good maneuverability, simple mechanics and payload capability. As main drawback, the high energy consumption can be mentioned. However, the trade-off results very positive.

This structure can be attractive in several application, in particular for surveillance, imaging, dangerous environments, indoor navigation and mapping. The goals of this thesis are the system modelling, the control algorithm evaluation, the simulator design and the real platform development.

The study of the kinematics and dynamics helps to understand the physics of the quadrotor and its behavior. Together with the modelling, the determination of the control algorithm structure is very important to achieve a better stabilization. The whole system can be tested thanks to a Matlab-Simulink program which is interfaced with the remote controller. This software provides a 3D graphic output as well as status data, for the debugging of the system performance.

The real platform has been developed by creating a system of interconnected devices. Two types of sensors have been used for measuring the robot attitude and its height from the ground. For the first purpose, an Inertial Measurement Unit (IMU) have been adopted, while the distance has been estimated with a SOund Navigation And Ranging (SONAR) and an InfraRed (IR) modules. The data processing and the control algorithm are handled in the Micro Control Unit (MCU) STR730 which provides the signals to the motors. Actually, four motor driver boards are needed to amplify the power delivered to the motors. Their rotation is transmitted to the propellers which move the whole structure.

According to the goals of this project, the research has been very detailed in both modelling and simulation. Thanks also to the identification process, the performance of the real platform has been satisfactory. The quadrotor tests shows roll and pitch errors always less than one degree. The yaw error has values less than two degrees under static condition and less than four degrees under dynamic tracking. The height stabilization has an error of just two centimeters.

To improve this quadrotor project, several modifications can be done. For example, a high level controller can be implemented to follow position requirements, obstacle avoidance and trajectory planning. Furthermore a camera can be used for mobile targets tracking or environment mapping ...

This thesis is structured as follows:

Chapter 2 gives an overview of the state of the art of the research area. Other related works are cited to show what has already been done in this field.

Chapter 3 provides the derivation of the quadrotor model. The dynamics is explained from the basic concepts to the Newton-Euler formalism. Particular attention is given to the motor-gears-propeller system and to the whole quadrotor architecture.

Chapter 4 focuses on the control algorithms needed to stabilize the quadrotor. The model of the helicopter is simplified to be able to use an easier controller and to lower the algorithm complexity. PID techniques is adopted in this work. The different phases of the control structure are presented.

Chapter 5 shows the quadrotor simulator. It is a Matlab-Simulink program used to verify the correctness of the helicopter dynamic model and to test the control algorithm performance. The system structure, block implementations and 3D visualization are deepened to better explain the power of this tool.

Chapter 6 deals with several approaches used to estimate constants, according to the known information. Some of these are: basic measurements, dynamic derivation and experimental data extrapolation.

Chapter 7 presents the main quadrotor sensors. The height of the robot is identified thanks to a SONAR and a IR modules. Particular attention is given to their performance and interfaces. To measure the attitude of the helicopter an IMU is chosen. Its inner sensors and communication frame are presented.

Chapter 8 focuses on the quadrotor motor drivers. They are fundamental to interface the Micro Controller Unit (MCU) outputs to the motors (connected with the propellers). All the developing process is reported: schematic design, simulation, PCB design, realization and final tests.

Chapter 9 gives an overview of the STR730 Micro Control Unit. The characteristics of its peripherals, their use and their interconnections are presented. In addition, the software structure shows the main control cycle and its timing.

Chapter 10 shows the developing process of the quadrotor platform. A few photos are attached to provide a visual description. In addition, the experimental results of the stabilization algorithm are reported.

Chapter 11 summarizes the goals of this thesis, evaluates the performance and the results of the project and proposes solutions to improve this quadrotor platform.

Appendix A describes the basic equations which identify a 6 DOF rigid body with the Newton-Euler formulation.

Appendix B shows the linear regression method in which data are fitted with a straight line according to the ordinary least squares.

Appendix C explains the determination of the thrust and the drag coefficients thanks to aerodynamic considerations.

Appendix D presents the moment of inertia theory which describes the dynamic behavior of a body in rotation around a defined axis.

Appendix E gives a list of acronyms and abbreviations used in this thesis (with the respective descriptions), to better clarify their meaning.

Appendix F shows a table with several physic units of measurement. symbols, names, quantities and equivalences are reported.

Appendix G presents a table of prefixes used in engineering. Each of them has a specific factor, name, symbol and equivalence.

Appendix H lists all the constants used in this thesis with their symbols, units, values and descriptions.

Appendix I provides a table with symbols, units and descriptions of all the variables used in this thesis.

Chapter 2

State of the art

In the last few years, the state of the art in Vertical Take-Off and Landing (VTOL) Unmanned Aerial Vehicle (UAV) has received several contributes. Moreover, most of the attention has been focused on, the quadrotor structure. Some projects are based on commercially available platforms like Draganflyer [1], X-UFO [2] and MD4-200 [3]. Other researchers prefer instead to build their own structure. A few examples are the mesicopter, the X4-Flyer and the STARMAC.

There are articles which present hybrid configuration such as structure with non-symmetric rotation directions or with two directional rotors [4, 5]. A few other works focus instead on modelling derivation [6, 7] and efficient configurations [8]. Multi-agent task is also an interesting field for VTOL UAV [9].

Even though there are a lot of different topics about the qudrotor structure, that one on which most of the publications have focused on is the control algorithm. It can be stated that the 85% of the articles propose a control low or compare the performance of few of them. The most important techniques and the respective publications are now presented:

The first control is done using Lyapunov Theory [10, 11, 12, 13]. According to this technique, it is possible to ensure, under certain condition, the asymptotical stability of the helicopter.

The second control is provided by PD^2 feedback and PID structures [14, 15, 16]. The strength of the PD^2 feedback is the exponential convergence property mainly due to the compensation of the Coriolis and gyroscopic terms. On the contrary a PID structure does not require some specific model parameters and the control law is much simpler to implement.

The third control uses adaptive techniques [17, 18]. These methods provide good performance with parametric uncertainties and unmodeled dynamics.

The fourth control is based on Linear Quadratic Regulator (LQR) [14, 19]. The main advantage of this technique is that the optimal input signal turns out to be obtainable from full state feedback (by solving the Ricatti equation). On the other hand the analytical solution to the Ricatti equation is difficult to compute.

The fifth control is done with backstepping control [20, 21, 22]. In the respective publications the convergence of the qudrotor internal states is guaranteed, but a lot of computation is reqired.

The sixth control is provided by dynamic feedback [23, 24]. This technique is implemented in a few quadrotor projects to transform the closed loop part of the system into a linear, controllable and decoupled subsystem.

The seventh control is based on visual feedback. The camera used for this purpose can be mounted on-board [25, 26, 27] (fixed on the helicopter) or off-board [28, 29] (fixed on the ground).

Other control algorithms are done with fuzzy techniques [30], neural networks [31] and reinforcement learning [32].

The contribution of this thesis lies mainly in four fields:

- accurate dynamic and aerodynamic modelling
- easy and robust control structure
- powerful and interactive simulator
- system identification and design of a real platform

Chapter 3

Quadrotor model & system

In this chapter, the derivation of the quadrotor model is provided. This result is very important because it describes how the helicopter moves according to its inputs. Thanks to these equations it is possible to define and predict the positions reached by the helicopter by investigating just the four motor speeds. The model equations will be "inverted" in the next chapter (Control algorithms) to identify which inputs are needed to reach a certain position.

The first section (3.1: Basic concepts) shows the main idea of the quadrotor dynamics and describes intuitively which movements are allowed and how it manages to perform stationary flight (hovering).

The second section (3.2: Newton-Euler model) provides the model information with physics and mathematical derivations. In this work, the Newton-Euler formalism and the Euler angles theories have been chosen.

In the third section (3.3: DC-motor), additional information was added to the model taking into account the whole motor system which is composed of the motor itself, the reduction gears and the propeller.

The last section (3.4: System architecture) provides an overview of the architecture: connections between devices and abstraction of the software and task.

3.1 Basic concepts

The quadrotor is very well modeled with a four rotors in a cross configuration. This cross structure is quite thin and light, however it shows robustness by linking mechanically the motors (which are heavier than the structure). Each propeller is connected to the motor through the reduction gears. All the propellers axes of rotation are fixed and parallel. Furthermore, they have fixed-pitch blades and their air flows points downwards (to get an upward lift). These considerations point out that the structure is quite rigid and the only things that can vary are the propeller speeds.

In this section, neither the motors nor the reduction gears are fundamental because the movements are directly related just to the propellers velocities. The others parts will be taken into account in the following sections. Another neglected component is the electronic box. As in the previous case, the electronic box is not essential to understand how the quadrotor flies. It follows that the basic model to evaluate the quadrotor movements it is composed just of a thin cross structure with four propellers on its ends.

The front and the rear propellers rotate counter-clockwise, while the left and the right ones turn clockwise. This configuration of opposite pairs directions removes the need for a tail rotor (needed instead in the standard helicopter structure). Figure 3.1 shows the structure model in hovering condition, where all the propellers have the same speed.

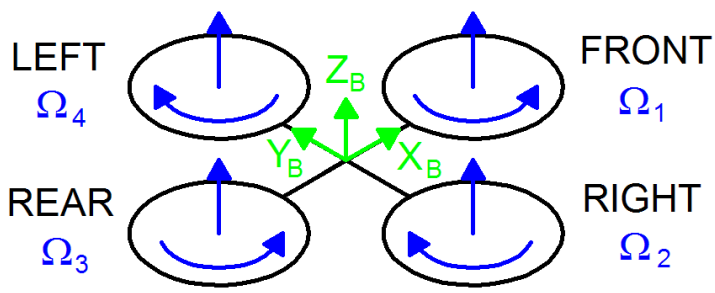


Figure 3.1: *Simplified quadrotor motor in hovering*

In figure 3.1 a sketch of the quadrotor structure is presented in black. The fixed-body B-frame is shown in green and in blue is represented the angular speed of the propellers. In addition to the name of the velocity variable, for each propeller, two arrows are drawn: the curved one represents the direction of rotation,

the other one represents the velocity. This last vector always points upwards hence it doesn't follow the right hand rule (for clockwise rotation) because it also models a vertical thrust and it would be confusing to have two speed vectors pointing upwards and the other two pointing downwards.

In the model of figure 3.1 all the propellers rotate at the same (hovering) speed Ω_H $[rad\ s^{-1}]$ to counterbalance the acceleration due to gravity. Thus, the quadrotor performs stationary flight and no forces or torques move it from its position.

Even though the quadrotor has 6 DOF, it is equipped just with four propellers, hence it is not possible to reach a desired set-point for all the DOF, but at maximum four. However, thanks to its structure, it is quite easy to chose the four best controllable variables and to decouple them to make the controller easier. The four quadrotor targets are thus related to the four basic movements which allow the helicopter to reach a certain height and attitude. It follows the description of these basic movements:

- Throttle (U_1 $[N]$)
  This command is provided by increasing (or decreasing) all the propeller speeds by the same amount. It leads to a vertical force WRT body-fixed frame which raises or lowers the quadrotor. If the helicopter is in horizontal position, the vertical direction of the inertial frame and that one of the body-fixed frame coincide. Otherwise the provided thrust generates both vertical and horizontal accelerations in the inertial frame. Figure 3.2 shows the throttle command on a quadrotor sketch.

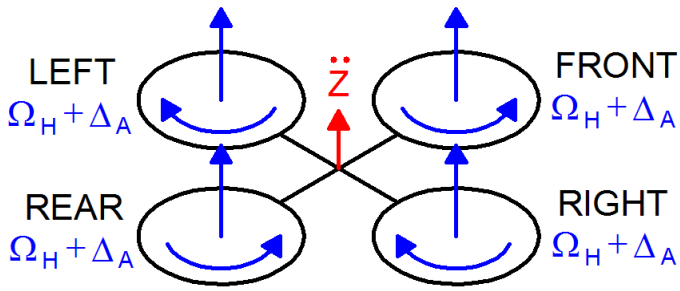


Figure 3.2: *Throttle movement*

In blue it is specified the speed of the propellers which, in this case, is equal to $\Omega_H + \Delta_A$ for each one. Δ_A $[rad\ s^{-1}]$ is a positive variable which represents an increment respect of the constant Ω_H. Δ_A can't be too large because the model would eventually be influenced by strong non linearities or saturations.

- Roll (U_2 $[N\ m]$)
  This command is provided by increasing (or decreasing) the left propeller speed and by decreasing (or increasing) the right one. It leads to a torque with respect to the x_B axis which makes the quadrotor turn. The overall vertical thrust is the same as in hovering, hence this command leads only to a roll angle acceleration (in first approximation). Figure 3.3 shows the roll command on a quadrotor sketch.

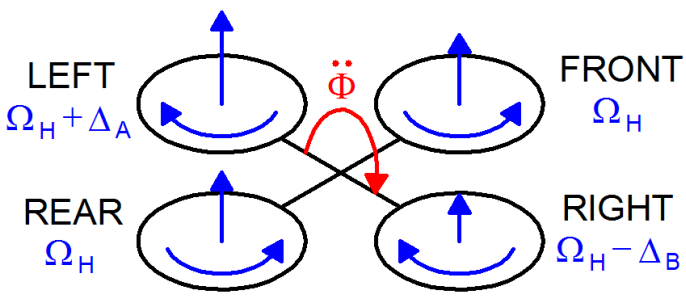


Figure 3.3: *Roll movement*

The positive variables Δ_A and Δ_B $[rad\ s^{-1}]$ are chosen to maintain the vertical thrust unchanged. It can be demonstrated that for small values of Δ_A, $\Delta_B \approx \Delta_A$. As in the previous case, they can't be too large because the model would eventually be influenced by strong non linearities or saturations.

- Pitch (U_3 $[N\ m]$)
  This command is very similar to the roll and is provided by increasing (or decreasing) the rear propeller speed and by decreasing (or increasing) the front one. It leads to a torque with respect to the y_B axis which makes the quadrotor turn. The overall vertical thrust is the same as in hovering, hence this command leads only to a pitch angle acceleration (in first approximation). Figure 3.4 shows the pitch command on a quadrotor sketch. As in

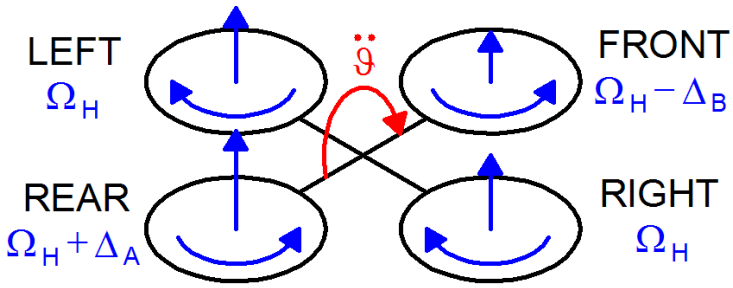


Figure 3.4: *Pitch movement*

the previous case, the positive variables Δ_A and Δ_B are chosen to maintain the vertical thrust unchanged and they can't be too large. Furthermore, for small values of Δ_A, it occurs $\Delta_B \approx \Delta_A$.

- Yaw (U_4 $[N\ m]$)
  This command is provided by increasing (or decreasing) the front-rear propellers' speed and by decreasing (or increasing) that of the left-right couple. It leads to a torque with respect to the z_B axis which makes the quadrotor turn. The yaw movement is generated thanks to the fact that the left-right propellers rotate clockwise while the front-rear ones rotate counterclockwise. Hence, when the overall torque is unbalanced, the helicopter turns on itself around z_B. The total vertical thrust is the same as in hovering, hence this command leads only to a yaw angle acceleration (in first approximation). Figure 3.5 shows the yaw command on a quadrotor sketch. As in the pre-

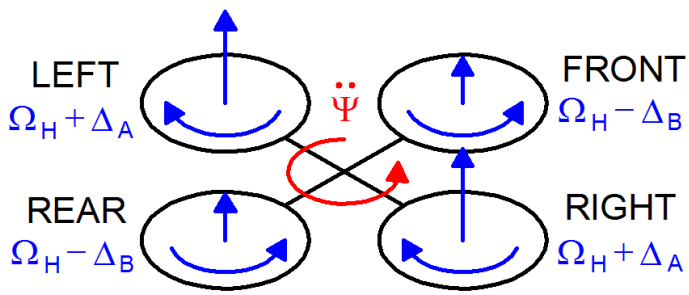


Figure 3.5: *Yaw movement*

vious two cases, the positive variables Δ_A and Δ_B are chosen to maintain the vertical thrust unchanged and they can't be too large. Furthermore it maintains the equivalence $\Delta_B \approx \Delta_A$ for small values of Δ_A.

3.2 Newton-Euler model

This section provides the specific model information of the quadrotor architecture starting from the generic 6 DOF rigid-body equation derived with the Newton-Euler formalism in appendix A.

Two frames have to be defined:

- the earth inertial frame (E-frame)
- the body-fixed frame (B-frame)

The equations of motion are more conveniently formulated in the body-fixed frame because of the following reasons [33]:

- The inertia matrix is time-invariant.
- Advantage of body symmetry can be taken to simplify the equations.
- Measurements taken on-board are easily converted to body-fixed frame.
- Control forces are almost always given in body-fixed frame.

Equation (3.1) describes the kinematics of a generic 6 DOF rigid-body.

$$\dot{\boldsymbol{\xi}} = \boldsymbol{J}_{\boldsymbol{\Theta}}\, \boldsymbol{\nu} \tag{3.1}$$

where $\dot{\boldsymbol{\xi}}$ $[+]$ is the generalized velocity vector WRT E-frame, $\boldsymbol{\nu}$ $[+]$ is the generalized velocity vector WRT B-frame and $\boldsymbol{J}_{\boldsymbol{\Theta}}$ $[-]$ is the generalized matrix.

$\boldsymbol{\xi}$ $[+]$ is composed of the quadrotor linear $\boldsymbol{\Gamma}^E$ $[m]$ and angular $\boldsymbol{\Theta}^E$ $[rad]$ position vectors WRT E-frame as shown in equation (3.2).

$$\boldsymbol{\xi} = \begin{bmatrix} \boldsymbol{\Gamma}^E & \boldsymbol{\Theta}^E \end{bmatrix}^T = \begin{bmatrix} X & Y & Z & \phi & \theta & \psi \end{bmatrix}^T \tag{3.2}$$

Similarly, $\boldsymbol{\nu}$ $[+]$ is composed of the quadrotor linear $\boldsymbol{V}^B$ $[m\ s^{-1}]$ and angular $\boldsymbol{\omega}^B$ $[rad\ s^{-1}]$ velocity vectors WRT B-frame as shown in equation (3.3).

$$\boldsymbol{\nu} = \begin{bmatrix} \boldsymbol{V}^B & \boldsymbol{\omega}^B \end{bmatrix}^T = \begin{bmatrix} u & v & w & p & q & r \end{bmatrix}^T \tag{3.3}$$

In addition, the generalized matrix $\boldsymbol{J}_{\boldsymbol{\Theta}}$ is composed of 4 sub-matrix according to equation (3.4).

$$\boldsymbol{J}_{\boldsymbol{\Theta}} = \begin{bmatrix} \boldsymbol{R}_{\boldsymbol{\Theta}} & \boldsymbol{0}_{3\times3} \\ \boldsymbol{0}_{3\times3} & \boldsymbol{T}_{\boldsymbol{\Theta}} \end{bmatrix} \tag{3.4}$$

The notation $\boldsymbol{0}_{3\times3}$ means a sub-matrix with dimension 3 times 3 filled with all zeros while the rotation $\boldsymbol{R}_{\boldsymbol{\Theta}}$ $[-]$ and the transfer $\boldsymbol{T}_{\boldsymbol{\Theta}}$ $[-]$ matrices are defined according to equations (3.5) and (3.6).

$$\boldsymbol{R}_{\boldsymbol{\Theta}} = \begin{bmatrix} c_\psi c_\theta & -s_\psi c_\phi + c_\psi s_\theta s_\phi & s_\psi s_\phi + c_\psi s_\theta c_\phi \\ s_\psi c_\theta & c_\psi c_\phi + s_\psi s_\theta s_\phi & -c_\psi s_\phi + s_\psi s_\theta c_\phi \\ -s_\theta & c_\theta s_\phi & c_\theta c_\phi \end{bmatrix} \tag{3.5}$$

$$\boldsymbol{T}_{\boldsymbol{\Theta}} = \begin{bmatrix} 1 & s_\phi t_\theta & c_\phi t_\theta \\ 0 & c_\phi & -s_\phi \\ 0 & s_\phi / c_\theta & c_\phi / c_\theta \end{bmatrix} \tag{3.6}$$

In the previous two equations (and in the following) this notation has been adopted: $c_k = \cos k$, $s_k = \sin k$, $t_k = \tan k$.

The dynamics of a generic 6 DOF rigid-body takes into account the mass of the body m $[kg]$ and its inertia matrix $\boldsymbol{I}$ $[N\ m\ s^2]$. $\boldsymbol{I}$ is calculated in this work according to appendix D. The dynamics is described by equation (3.7).

$$\begin{bmatrix} m\ \boldsymbol{I}_{3\times3} & \boldsymbol{0}_{3\times3} \\ \boldsymbol{0}_{3\times3} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{V}}^B \\ \dot{\boldsymbol{\omega}}^B \end{bmatrix} + \begin{bmatrix} \boldsymbol{\omega}^B \times \left(m\ \boldsymbol{V}^B\right) \\ \boldsymbol{\omega}^B \times \left(\boldsymbol{I}\ \boldsymbol{\omega}^B\right) \end{bmatrix} = \begin{bmatrix} \boldsymbol{F}^B \\ \boldsymbol{\tau}^B \end{bmatrix} \tag{3.7}$$

Where the notation $\boldsymbol{I}_{3\times3}$ means a 3 times 3 identity matrix. $\dot{\boldsymbol{V}}^B$ $[m\ s^{-2}]$ is the quadrotor linear acceleration vector WRT B-frame while $\dot{\boldsymbol{\omega}}^B$ $[rad\ s^{-2}]$ is the quadrotor angular acceleration vector WRT B-frame. In addition, $\boldsymbol{F}^B$ $[N]$ is the quadrotor forces vector WRT B-frame and $\boldsymbol{\tau}^B$ $[N\ m]$ is the quadrotor torques vector WRT B-frame.

Two assumptions have been done in this approach:

- The first one states that the origin of the body-fixed frame o_B is coincident with the center of mass (COM) of the body. Otherwise, another point (COM) should have been taken into account and it would have considerably complicated the body equations.
- The second one specifies that the axes of the B-frame coincide with the body principal axes of inertia. In this case the inertia matrix $\boldsymbol{I}$ is diagonal and, once again, the body equations become easier.

A generalized force vector $\boldsymbol{\Lambda}$ can be defined according to equation (3.8).

$$\boldsymbol{\Lambda} = \begin{bmatrix} \boldsymbol{F}^B & \boldsymbol{\tau}^B \end{bmatrix}^T = \begin{bmatrix} F_x & F_y & F_z & \tau_x & \tau_y & \tau_z \end{bmatrix}^T \tag{3.8}$$

Therefore it is possible to rewrite equation (3.7) in a matrix form [33].

$$\boldsymbol{M}_B\ \dot{\boldsymbol{\nu}} + \boldsymbol{C}_B(\boldsymbol{\nu})\ \boldsymbol{\nu} = \boldsymbol{\Lambda} \tag{3.9}$$

Where $\dot{\boldsymbol{\nu}}$ is the generalized acceleration vector WRT B-frame. $\boldsymbol{M}_B$ [+] is the system inertia matrix and $\boldsymbol{C}_B(\boldsymbol{\nu})$ [+] is the Coriolis-centripetal matrix, both WRT B-frame. Equation (3.10) shows the system inertia matrix.

$$\boldsymbol{M}_B = \begin{bmatrix} m\ \boldsymbol{I}_{3\times3} & \boldsymbol{0}_{3\times3} \\ \boldsymbol{0}_{3\times3} & \boldsymbol{I} \end{bmatrix} = \begin{bmatrix} m & 0 & 0 & 0 & 0 & 0 \\ 0 & m & 0 & 0 & 0 & 0 \\ 0 & 0 & m & 0 & 0 & 0 \\ 0 & 0 & 0 & I_{XX} & 0 & 0 \\ 0 & 0 & 0 & 0 & I_{YY} & 0 \\ 0 & 0 & 0 & 0 & 0 & I_{ZZ} \end{bmatrix} \tag{3.10}$$

It's easy to see that $\boldsymbol{M}_B$ is diagonal and constant (thanks to the assumptions made above). Equation (3.11) shows the Coriolis-centripetal matrix.

$$\boldsymbol{C}_B(\boldsymbol{\nu}) = \begin{bmatrix} \boldsymbol{0}_{3\times3} & -m\ \boldsymbol{S}(\boldsymbol{V}^B) \\ \boldsymbol{0}_{3\times3} & -\boldsymbol{S}(\boldsymbol{I}\ \boldsymbol{\omega}^B) \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & m\ w & -m\ v \\ 0 & 0 & 0 & -m\ w & 0 & m\ u \\ 0 & 0 & 0 & m\ v & -m\ u & 0 \\ 0 & 0 & 0 & 0 & I_{ZZ}\ r & -I_{YY}\ q \\ 0 & 0 & 0 & -I_{ZZ}\ r & 0 & I_{XX}\ p \\ 0 & 0 & 0 & I_{YY}\ q & -I_{XX}\ p & 0 \end{bmatrix} \tag{3.11}$$

In the previous equation the skew-symmetric operator $\boldsymbol{S}(\cdot)$ $[+]$ has been adopted. For a generic three dimension vector $\boldsymbol{k}$ $[-]$, the skew-symmetric matrix of $\boldsymbol{k}$ ($\boldsymbol{S}(\boldsymbol{k})$) is defined according to equation (3.12).

$$\boldsymbol{S}(\boldsymbol{k}) = -\boldsymbol{S}^T(\boldsymbol{k}) = \begin{bmatrix} 0 & -k_3 & k_1 \\ k_3 & 0 & -k_1 \\ -k_2 & k_1 & 0 \end{bmatrix} \qquad \boldsymbol{k} = \begin{bmatrix} k_1 \\ k_2 \\ k_3 \end{bmatrix} \tag{3.12}$$

Equation (3.9) is totally generic and is valid for all the rigid body which obeys to the hypothesis (or simplifications) previously made. However, it was used in this work to model the quadrotor helicopter, hence the last vector contains specific information about its dynamics. $\boldsymbol{\Lambda}$ can be divided in three components according to the nature of the quadrotor contributions.

The first contribution is the gravitational vector $\boldsymbol{G}_B(\boldsymbol{\xi})$ $[+]$ given from the acceleration due to gravity g $[m\ s{-}2]$. It's easy to understand that it affects just the linear and not the angular equations since it's a force and not a torque. Equation (3.13) shows the transformations to get $\boldsymbol{G}_B(\boldsymbol{\xi})$.

$$\boldsymbol{G}_B(\boldsymbol{\xi}) = \begin{bmatrix} \boldsymbol{F}_G^B \\ \boldsymbol{0}_{3\times1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{R}_{\boldsymbol{\Theta}}^{-1}\ \boldsymbol{F}_G^E \\ \boldsymbol{0}_{3\times1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{R}_{\boldsymbol{\Theta}}^T \begin{bmatrix} 0 \\ 0 \\ -m\ g \end{bmatrix} \\ \boldsymbol{0}_{3\times1} \end{bmatrix} = \begin{bmatrix} m\ g\ s_\theta \\ -m\ g\ c_\theta s_\phi \\ -m\ g\ c_\theta s_\phi \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{3.13}$$

Where $\boldsymbol{F}_G^B$ $[N]$ is the gravitational force vector WRT B-frame and $\boldsymbol{F}_G^E$ $[N]$ is that one WRT E-frame. $\boldsymbol{0}_{3\times1}$ $[-]$ is a vertical vector filled with three zeros. Furthermore, since $\boldsymbol{R}_{\boldsymbol{\Theta}}$ is an orthogonal normalized matrix, its inverted $\boldsymbol{R}_{\boldsymbol{\Theta}}^{-1}$ is equal to the transposed one $\boldsymbol{R}_{\boldsymbol{\Theta}}^T$.

The second contribution takes into account the gyroscopic effects produced by the propeller rotation. Since two of them are rotating clockwise and the other two counterclockwise, there is a overall imbalance when the algebraic sum of the rotor speeds is not equal to zero. If, in addition, the roll or pitch rates are also different than zero, the quadrotor experiences a gyroscopic torque according to equation (3.14).

$$\begin{aligned}
\boldsymbol{O}_B(\boldsymbol{\nu})\,\boldsymbol{\Omega} &= \begin{bmatrix} \mathbf{0}_{3\times1} \\ -\sum_{k=1}^{4} J_{TP} \left(\boldsymbol{\omega}^B \times \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \right) (-1)^k \Omega_k \end{bmatrix} = \\
&= \begin{bmatrix} \mathbf{0}_{3\times1} \\ J_{TP} \begin{bmatrix} -q \\ p \\ 0 \end{bmatrix} \Omega \end{bmatrix} = J_{TP} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ q & -q & q & -q \\ -p & p & -p & p \\ 0 & 0 & 0 & 0 \end{bmatrix} \boldsymbol{\Omega}
\end{aligned} \tag{3.14}$$

$\boldsymbol{O}_B(\boldsymbol{\nu})$ $[+]$ is the gyroscopic propeller matrix and J_{TP} $[N\ m\ s^2]$ is the total rotational moment of inertia around the propeller axis calculated in the next section. It's easy to see that the gyroscopic effects produced by the propeller rotation are just related to the angular and not the linear equations.

Equation (3.15) defines the overall propellers' speed Ω $[rad\ s^{-1}]$ and the propellers' speed vector $\boldsymbol{\Omega}$ $[rad\ s^{-1}]$ used in equation (3.14).

$$\Omega = -\Omega_1 + \Omega_2 - \Omega_3 + \Omega_4 \qquad \boldsymbol{\Omega} = \begin{bmatrix} \Omega_1 \\ \Omega_2 \\ \Omega_3 \\ \Omega_4 \end{bmatrix} \tag{3.15}$$

Where Ω_1 $[rad\ s^{-1}]$ is the front propeller speed, Ω_2 $[rad\ s^{-1}]$ is the right propeller speed, Ω_3 $[rad\ s^{-1}]$ is the rear propeller speed. Ω_4 $[rad\ s^{-1}]$ is the left propeller speed.

The third contribution takes into account the forces and torques directly produced by the main movement inputs. From aerodynamics consideration, it follows that both forces and torques are proportional to the squared propellers' speed. Therefore the movement matrix $\boldsymbol{E}_B$ [+] is multiplied by $\boldsymbol{\Omega}^2$ to get the movement vector $\boldsymbol{U}_B(\boldsymbol{\Omega})$ [+]. The derivation of the aerodynamic contributions (thrust b $[N\ s^2]$ and drag d $[N\ m\ s^2]$ factors) is treated in deep in Appendix C.

Equation (3.16) shows the action of the movement vector on the quadrotor helicopter dynamics.

$$\boldsymbol{U}_B(\boldsymbol{\Omega}) \ = \boldsymbol{E}_B\ \boldsymbol{\Omega}^2 = \begin{bmatrix} 0 \\ 0 \\ U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ b\,({\Omega_1}^2 + {\Omega_2}^2 + {\Omega_3}^2 + {\Omega_4}^2) \\ b\ l\,({\Omega_4}^2 - {\Omega_2}^2) \\ b\ l\,({\Omega_3}^2 - {\Omega_1}^1) \\ d\,({\Omega_2}^2 + {\Omega_4}^2 - {\Omega_1}^2 - {\Omega_3}^2) \end{bmatrix} \tag{3.16}$$

Where l $[m]$ is the distance between the center of the quadrotor and the center of a propeller. U_1, U_2, U_3 and U_4 are the movement vector components introduced in the previous section. Their relation with the propellers' speeds comes from aerodynamic calculus (see appendix C). The expression of the torque produced by U_4 has been simplified by neglecting its $\dot{\boldsymbol{\Omega}}$ component. Therefore all the movements have a similar expression and are easier to control.

As stated before (and shown in the previous equation), it is possible to identify a constant matrix $\boldsymbol{E}_B$ which multiplied by the squared propellers' speed $\boldsymbol{\Omega}^2$ produces the movement vector $\boldsymbol{U}_B(\boldsymbol{\Omega})$. Equation (3.17) shows the movement matrix.

$$\boldsymbol{E}_B = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ b & b & b & b \\ 0 & -b\ l & 0 & b\ l \\ -b\ l & 0 & b\ l & 0 \\ -d & d & -d & d \end{bmatrix} \tag{3.17}$$

From equation (3.9) it is possible to describe the quadrotor dynamics considering these last three contributions according to equation (3.18).

$$\boldsymbol{M}_B\ \dot{\boldsymbol{\nu}} + \boldsymbol{C}_B(\boldsymbol{\nu})\ \boldsymbol{\nu} = \boldsymbol{G}_B(\boldsymbol{\xi}) + \boldsymbol{O}_B(\boldsymbol{\nu})\ \boldsymbol{\Omega} + \boldsymbol{E}_B\ \boldsymbol{\Omega}^2 \tag{3.18}$$

By rearranging equation (3.18) it is possible to isolate the derivate of the generalized velocity vector WRT B-frame $\dot{\boldsymbol{\nu}}$.

$$\dot{\boldsymbol{\nu}} = \boldsymbol{M}_B{}^{-1}\left(-\boldsymbol{C}_B(\boldsymbol{\nu})\ \boldsymbol{\nu} + \boldsymbol{G}_B(\boldsymbol{\xi}) + \boldsymbol{O}_B(\boldsymbol{\nu})\ \boldsymbol{\Omega} + \boldsymbol{E}_B\ \boldsymbol{\Omega}^2\right) \tag{3.19}$$

Equation (3.20) shows the previous expression not in a matrix form, but in a system of equations.

$$\begin{cases} \dot{u} &= (v\ r - w\ q) + g\ s_\theta \\ \dot{v} &= (w\ p - u\ r) - g\ c_\theta s_\phi \\ \dot{w} &= (u\ q - v\ p) - g\ c_\theta s_\phi + \dfrac{U_1}{m} \\ \dot{p} &= \dfrac{I_{YY} - I_{ZZ}}{I_{XX}}\ q\ r - \dfrac{J_{TP}}{I_{XX}}\ q\ \Omega + \dfrac{U_2}{I_{XX}} \\ \dot{q} &= \dfrac{I_{ZZ} - I_{XX}}{I_{YY}}\ p\ r + \dfrac{J_{TP}}{I_{YY}}\ p\ \Omega + \dfrac{U_3}{I_{YY}} \\ \dot{r} &= \dfrac{I_{XX} - I_{YY}}{I_{ZZ}}\ p\ q + \dfrac{U_4}{I_{ZZ}} \end{cases} \tag{3.20}$$

Where the propellers' speed inputs are given through equation (3.21).

$$\begin{cases} U_1 &= b\ ({\Omega_1}^2 + {\Omega_2}^2 + {\Omega_3}^2 + {\Omega_4}^2) \\ U_2 &= l\ b\ (-{\Omega_2}^2 + {\Omega_4}^2) \\ U_3 &= l\ b\ (-{\Omega_1}^2 + {\Omega_3}^2) \\ U_4 &= d\ (-{\Omega_1}^2 + {\Omega_2}^2 - {\Omega_3}^2 + {\Omega_4}^2) \\ \Omega &= -\Omega_1 + \Omega_2 - \Omega_3 + \Omega_4 \end{cases} \tag{3.21}$$

The quadrotor dynamic system in equation (3.20) is written in the body-fixed frame. As stated before, this reference is widely used in 6 DOF rigid-body equations. However in this case it can be useful to express the dynamics with respect to a hybrid system composed of linear equations WRT E-frame and angular equations WTR B-frame. Therefore the following equations will be expressed in the new "hybrid" frame called H-frame. This new reference is adopted because it's easy to express the dynamics combined with the control (in particular for the vertical position in the earth inertial frame). Equation (3.22) shows the quadrotor generalized velocity vector WRT H-frame ($\boldsymbol{\zeta}$ [+]).

$$\boldsymbol{\zeta} = \begin{bmatrix} \dot{\boldsymbol{\Gamma}}^E & \boldsymbol{\omega}^B \end{bmatrix}^T = \begin{bmatrix} \dot{X} & \dot{Y} & \dot{Z} & p & q & r \end{bmatrix}^T \tag{3.22}$$

The dynamics of the system in the H-frame can be rewritten in a matrix form according to equation (3.23).

$$\boldsymbol{M}_H \, \dot{\boldsymbol{\zeta}} + \boldsymbol{C}_H(\boldsymbol{\zeta}) \, \boldsymbol{\zeta} = \boldsymbol{G}_H + \boldsymbol{O}_H(\boldsymbol{\zeta}) \, \boldsymbol{\Omega} + \boldsymbol{E}_H(\boldsymbol{\xi}) \, \boldsymbol{\Omega}^2 \tag{3.23}$$

Where $\dot{\boldsymbol{\zeta}}$ [+] is the quadrotor generalized acceleration vector WRT H-frame. It now follows the definitions of all the matrices and vectors used in equation (3.23). The system inertia matrix WRT H-frame $\boldsymbol{M}_H$ [+] is equal to that one WRT B-frame and defined according to equations (3.10) and (3.24).

$$\boldsymbol{M}_H = \boldsymbol{M}_B = \begin{bmatrix} m \, \boldsymbol{I}_{3\times3} & \boldsymbol{0}_{3\times3} \\ \boldsymbol{0}_{3\times3} & \boldsymbol{I} \end{bmatrix} = \begin{bmatrix} m & 0 & 0 & 0 & 0 & 0 \\ 0 & m & 0 & 0 & 0 & 0 \\ 0 & 0 & m & 0 & 0 & 0 \\ 0 & 0 & 0 & I_{XX} & 0 & 0 \\ 0 & 0 & 0 & 0 & I_{YY} & 0 \\ 0 & 0 & 0 & 0 & 0 & I_{ZZ} \end{bmatrix} \tag{3.24}$$

On the contrary, the Coriolis-centripetal matrix WRT H-frame $\boldsymbol{C}_H(\boldsymbol{\zeta})$ [+] is not equal to that one WRT B-frame and defined according to equation (3.25).

$$\boldsymbol{C}_H(\boldsymbol{\zeta}) = \begin{bmatrix} \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & -\boldsymbol{S}(\boldsymbol{I}\ \boldsymbol{\omega}^B) \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & I_{ZZ}\ r & -I_{YY}\ q \\ 0 & 0 & 0 & -I_{ZZ}\ r & 0 & I_{XX}\ p \\ 0 & 0 & 0 & I_{YY}\ q & -I_{XX}\ p & 0 \end{bmatrix} \tag{3.25}$$

The gravitational vector WRT H-frame $\boldsymbol{G}_H$ [+] is defined in equation (3.26). It can be seen that it affects all the three linear equations instead of just the third as in the previous case.

$$\boldsymbol{G}_H = \begin{bmatrix} \boldsymbol{F}_G^E \\ \mathbf{0}_{3\times1} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -m\ g \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{3.26}$$

The gyroscopic effects produced by the propeller rotation is unvaried because it affects only the angular equations referred to the B-frame. Then, the gyroscopic propeller matrix WRT H-frame $\boldsymbol{O}_H(\boldsymbol{\zeta})$ [+] is defined according to equations (3.14) and (3.27).

$$\boldsymbol{O}_H(\boldsymbol{\zeta})\ \boldsymbol{\Omega} = \boldsymbol{O}_B(\boldsymbol{\nu})\ \boldsymbol{\Omega} = \begin{bmatrix} \mathbf{0}_{3\times1} \\ J_{TP} \begin{bmatrix} -q \\ p \\ 0 \end{bmatrix} \Omega \end{bmatrix} = J_{TP} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ q & -q & q & -q \\ -p & p & -p & p \\ 0 & 0 & 0 & 0 \end{bmatrix} \boldsymbol{\Omega} \tag{3.27}$$

The movement matrix WRT H-frame $\boldsymbol{E}_H(\boldsymbol{\xi})$ [+] is different from that one in the B-frame because the input U_1 affects all the three linear equations through the rotation matrix $\boldsymbol{R}_{\boldsymbol{\theta}}$. The product between the movement matrix and the squared propellers' speed vector is shown in equation (3.28).

$$\boldsymbol{E}_H(\boldsymbol{\xi})\ \boldsymbol{\Omega}^2 = \begin{bmatrix} \boldsymbol{R}_{\boldsymbol{\Theta}} & \boldsymbol{0}_{3\times3} \\ \boldsymbol{0}_{3\times3} & \boldsymbol{I}_{3\times3} \end{bmatrix} \boldsymbol{E}_B\ \boldsymbol{\Omega}^2 = \begin{bmatrix} (s_\psi s_\phi + c_\psi s_\theta c_\phi)\ U_1 \\ (-c_\psi s_\phi + s_\psi s_\theta c_\phi)\ U_1 \\ (c_\theta c_\phi)\ U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} \tag{3.28}$$

By rearranging equation (3.23) it is possible to isolate the derivate of the generalized velocity vector WRT H-frame $\dot{\boldsymbol{\zeta}}$ [+].

$$\dot{\boldsymbol{\zeta}} = \boldsymbol{M}_H{}^{-1} \left(-\boldsymbol{C}_H(\boldsymbol{\zeta})\ \boldsymbol{\zeta} + \boldsymbol{G}_H + \boldsymbol{O}_H(\boldsymbol{\zeta})\ \boldsymbol{\Omega} + \boldsymbol{E}_H(\boldsymbol{\xi})\ \boldsymbol{\Omega}^2\right) \tag{3.29}$$

Equation (3.30) shows the previous expression not in a matrix form, but in a system of equations.

$$\begin{cases} \ddot{X} &= (\sin\psi\sin\phi + \cos\psi\sin\theta\cos\phi)\,\dfrac{U_1}{m} \\ \ddot{Y} &= (-\cos\psi\sin\phi + \sin\psi\sin\theta\cos\phi)\,\dfrac{U_1}{m} \\ \ddot{Z} &= -g + (\cos\theta\cos\phi)\,\dfrac{U_1}{m} \\ \dot{p} &= \dfrac{I_{YY} - I_{ZZ}}{I_{XX}}\ q\ r - \dfrac{J_{TP}}{I_{XX}}\ q\ \Omega + \dfrac{U_2}{I_{XX}} \\ \dot{q} &= \dfrac{I_{ZZ} - I_{XX}}{I_{YY}}\ p\ r + \dfrac{J_{TP}}{I_{YY}}\ p\ \Omega + \dfrac{U_3}{I_{YY}} \\ \dot{r} &= \dfrac{I_{XX} - I_{YY}}{I_{ZZ}}\ p\ q + \dfrac{U_4}{I_{ZZ}} \end{cases} \tag{3.30}$$

Where the propellers' speed inputs are the same as in the system WRT B-frame and given through equation (3.21).

The system in equation (3.30) will be simplified and widely used in the Control algorithms chapter. Furthermore the angular equations will be related to angular fixed references through the kinematic equation (3.1).

3.3 DC-motor

The DC-motor is an actuator which converts electrical energy into mechanical energy (and vice versa). It is composed of two interactive electromagnetic circuits. The first one (called rotor) is free to rotate around the second one (called stator) which is fixed instead. In the rotor, several groups of copper windings are connected in series and are externally accessible thanks to a device called commutator. In the stator, two or more permanent magnets impose a magnetic field which affects the rotor. By applying a DC-current flow into the windings, the rotor turns because of the force generated by the electrical and magnetic interaction. Thanks to the rotor and the commutator geometries, the motor keeps turning while supplied by a DC-voltage on its terminals. These are just a few basic concepts to understand the following section, for more information please refer to [34].

The DC-motor has a well known model which binds electrical and mechanical quantities. This model is composed of the series of a resistor R [Ω] , an inductor L [H] and a voltage generator e [V]. The resistor represents the Joule loss due to the current flow into the copper conductor. Its value depends on geometric and material characteristics such as wire resistivity, length and section. The inductor behavior derives from the shape of the motor wires which are winded in the middle of the rotor. Lastly, the generator e (called also BEMF) supplies a voltage proportional to the motor speed. The model is represented in figure 3.6.

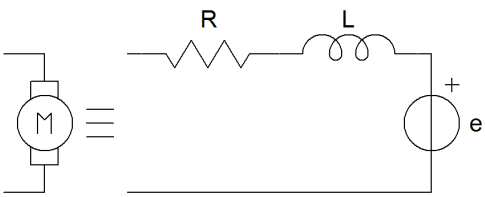


Figure 3.6: *Motor model*

The circuit of the DC-motor is controlled by a real voltage generator v [V] which gives the control input. In theory, another resistor should be added in series of the voltage generator representing the driver losses. However, in a good project, the generator losses are kept low therefore it is possible to neglect them in the model. The basic electrical circuit is shown in figure 3.7.

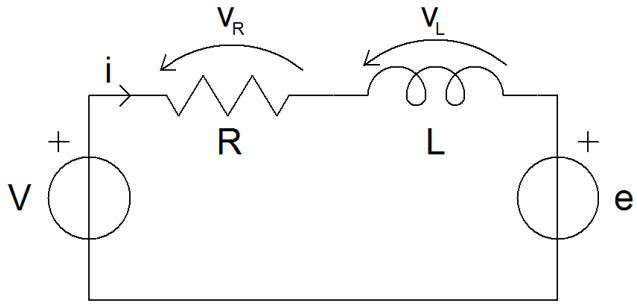


Figure 3.7: *Motor circuit*

By applying the Kirchhoff's voltage law, it follows equation (3.31).

$$v = v_R + v_L + e \tag{3.31}$$

Where v_R $[V]$ is the voltage across the resistor R and v_L $[V]$ is the voltage across the inductor L. Equation (3.31) can be rewritten as in the next equation.

$$v = R\, i + L\, \frac{\partial i}{\partial t} + K_E\, \omega_M \tag{3.32}$$

i $[A]$ is the motor current, K_E $[V\ s\ rad^{-1}]$ is called the motor constant and ω_M $[rad\ s^{-1}]$ is the motor angular speed. The first addend has been changed using the Ohm's law $v_R = R\, i$, while the second one using the inductor differential equation $v_L = L\, \frac{\partial i}{\partial t}$. The last member of equation (3.32) shows that mechanics and electrics are correlated.

The contribution of the inductor part is important to determine the characteristic of the DC-motor driver. However is often neglected in the mechanics computation because of three main aspects:

- Most of the motors used in robotics show small inductance thanks to construction optimization.
- The pole (response time) of the electrical part is always much faster than the mechanic one, therefore the speed of the overall system will be determined just by the slowest contribution.
- It's much easier to solve a first order differential equation rather than a second order one.

For all these reasons the inductor effect was neglected in this work too. Therefore equation (3.32) can be simplified according to equation (3.33).

$$v = R\ i + K_E\ \omega_M \tag{3.33}$$

The dynamics of the motor is described by the following equation.

$$J_{TM}\ \dot{\omega}_M = T_M - T_L \tag{3.34}$$

Where J_{TM} $[N\ m\ s^2]$ is the total motor moment of inertia, $\dot{\omega}_M$ $[rad\ s^{-2}]$ is the motor angular acceleration, T_M $[N\ m]$ is the motor torque and T_L $[N\ m]$ is the load torque. Equation (3.34) states that when the motor torque T_M and the load torque T_L are not equal, there is an acceleration (or deceleration) of the motor angular speed ω_M. This variation of speed depends also on the total motor moment of inertia J_{TM}: the smaller the value of J_{TM}, the higher the acceleration. Figure 3.8 shows a sketch of the simplified mechanic structure.

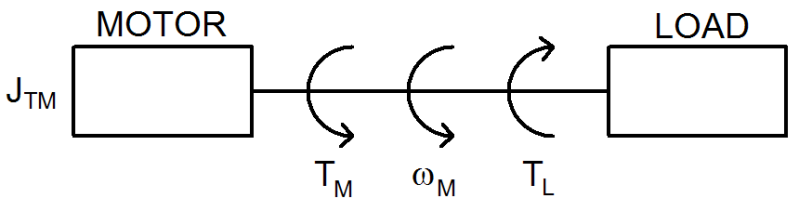


Figure 3.8: *Simplified motor system*

The motor torque T_M is proportional to the electrical current i through K_M $[N\ m\ A^{-1}]$: $T_M = K_M\ i$. Hence equation (3.34) can be rewritten according to equation (3.35).

$$J_{TM}\ \dot{\omega}_M = K_M\ i - T_L \tag{3.35}$$

By connecting equations (3.33) and (3.35) a differential equation in ω_M can be derived.

$$J_{TM}\ \dot{\omega}_M = -\frac{K_E\ K_M}{R}\ \omega_M - T_L + \frac{K_M}{R}\ v \tag{3.36}$$

It must be pointed out that the two constants K_E and K_M have the same value even though the units of measurement differ. This mismatch comes from the electric P_E $[W]$ and mechanic P_M $[N\ m\ s^{-1}]$ power balance.

$$P_E = P_M \left\{ \begin{array}{ccccc} P_E & = & e\ i & = & K_E\ i\ \omega_M \\ P_M & = & T_M\ \omega_M & = & K_M\ i\ \omega_M \end{array} \right\} \Rightarrow K_E = K_M \tag{3.37}$$

The real motor system is composed of the motor itself, the gear box and the propeller. A few equations must be added to take into account these connections. Figure 3.9 shows the structural system.

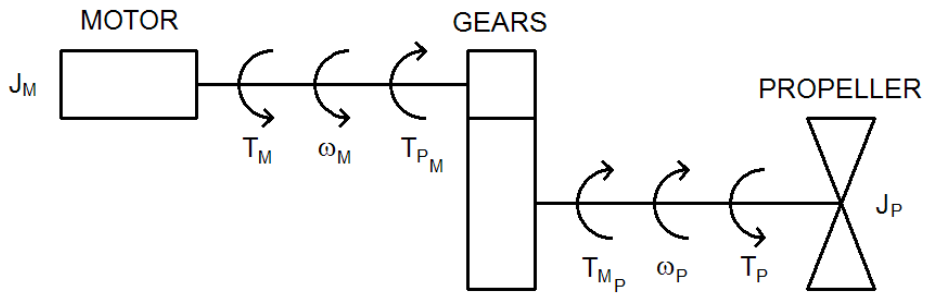


Figure 3.9: *Motor system*

In figure 3.9 several variables have been introduced: J_M $[N\ m\ s^2]$ is the rotor moment of inertia around the motor axis, J_P $[N\ m\ s^2]$ is the rotor moment of inertia around the propeller axis, ω_P $[rad\ s^{-1}]$ is the propeller angular speed, T_{P_M} $[N\ m]$ is the propeller torque in the motor axis, T_P $[N\ m]$ is the propeller torque and T_{M_P} $[N\ m]$ is the motor torque in the propeller axis.

The gear box has a reduction ratio N $[-]$ equal to the motor speed ω_m divided by the propeller speed ω_p: $N = \frac{\omega_M}{\omega_P}$. N can be also calculated as the ratio of the number of teeth of the propeller gear to the number of teeth of the motor gear. Another parameter of the gear box is the conversion efficiency η $[-]$ which binds the mechanical power of the motor axis P_M to the propeller axis one P_P $[N\ m\ s^{-1}]$.

$$\begin{aligned} P_M\ \eta &= P_P \\ \omega_M\ T_{P_M}\ \eta &= \omega_P\ T_{M_P} \end{aligned} \tag{3.38}$$

Therefore, it is possible to calculate the dynamics of the gear box system according to the following steps.

$$J_M\ \dot{\omega}_M = T_M - T_{P_M} \tag{3.39}$$

$$J_P\ \dot{\omega}_P = T_{M_P} - T_P \tag{3.40}$$

Where $\dot{\omega}_P$ $[rad\ s^{-2}]$ is the propeller angular acceleration. Equation (3.38) can be rewritten (taking into account equations (3.39) and (3.40)) as follows.

$$\begin{aligned} \omega_M \left(T_M - J_M\ \dot{\omega}_M\right) \eta &= \omega_P \left(T_P + J_P\ \dot{\omega}_P\right) \\ &\vdots \\ \left(J_M + \frac{J_P}{\eta\ N^2}\right) \omega_M &= T_M - \frac{T_P}{\eta\ N} \end{aligned} \tag{3.41}$$

In equation (3.41) the following two substitutions were done.

$$\dot{\omega}_P = \frac{\omega_P}{\omega_M}\ \dot{\omega}_M \qquad \text{and} \qquad \frac{\omega_M}{\omega_P} = N$$

Equation (3.41) is very important because it has the same structure of equation (3.34). Comparing the two equations, the real values of the load torque T_L and the total motor moment of inertia J_{TM} can be derived according to equations (3.42) and (3.43).

$$T_L = \frac{T_P}{\eta\ N} \tag{3.42}$$

$$J_{TM} = J_M + \frac{J_P}{\eta\ N^2} \tag{3.43}$$

The parameters in equations (3.42) and (3.43) can be calculated by geometric, dynamic and aerodynamic evaluation of the mechanic structure. Detailed analysis is reported in chapter 6; section Geometry and aerodynamics derivation 6.3 regarding equation (3.42) and section Geometry and dynamics derivation 6.2 regarding equation (3.43).

Equation (3.44) follows from aerodynamic calculus (see appendix C).

$$T_P = d\ \omega_P^2 = \frac{d\ \omega_M^2}{N^2} \tag{3.44}$$

Where d $[N\ m\ s^2]$ is the aerodynamic drag factor. From equations (3.42) and (3.44) the final load torque expression can be derived.

$$T_L = \frac{d\ \omega_M^2}{\eta\ N^3} \tag{3.45}$$

Equations (3.36) can be rewritten according to equations (3.43) and (3.45).

$$\left(J_M + \frac{J_P}{\eta\ N^2}\right)\ \dot{\omega}_M = -\frac{K_E\ K_M}{R}\ \omega_M - \frac{d}{\eta\ N^3}\ {\omega_M}^2 + \frac{K_M}{R}\ v \tag{3.46}$$

All these previous equations can be reformulated with reference to the propeller axis (and not the motor one). Therefore the motor system can be also modeled with the following differential equation.

$$\left(J_P + \eta\ N^2\ J_M\right)\ \dot{\omega}_P = -\frac{K_E\ K_M}{R}\ \eta\ N^2\ \omega_P - d\ {\omega_P}^2 + \frac{K_M}{R}\ \eta\ N\ v \tag{3.47}$$

Where the $\dot{\omega}_P$ coefficient is the total rotational moment of inertia around the propeller axis (J_{TP} $[N\ m\ s^2]$) as shown in equation (3.48).

$$J_{TP} = J_P + \eta\ N^2\ J_M \tag{3.48}$$

Since the differential equation equation (3.47) is non linear, a good approach is to linearize it around its working point. The first order Taylor series method has been adopted to derive equation (3.49).

$$\dot{\omega}_P = A_P\ \omega_P + B_P\ v + C_P \tag{3.49}$$

In the previous equation A_P $[rad\ s^{-1}]$ is the linearized propeller's speed coefficient, B_P $[rad^2\ s^{-2}\ V^{-1}]$ is the linearized input voltage coefficient and C_P $[rad^2\ s^{-2}]$ is the linearized constant coefficient. Their values are defined according to equations (3.50), (3.51) and (3.52).

$$A_P = \left.\frac{\partial\ \dot{\omega}_P}{\partial \omega_P}\right|_{\omega_P\ =\ \omega_H} = -\frac{K_E\ K_M\ \eta\ N^2}{J_{TP}\ R} - \frac{2\ d}{J_{TP}}\ \omega_H = -22.5\ [rad\ s^{-1}] \tag{3.50}$$

$$B_P = \left.\frac{\partial\ \dot{\omega}_P}{\partial v}\right|_{\omega_P\ =\ \omega_H} = \frac{K_M\ \eta\ N}{J_{TP}\ R} = 509\ [rad^2\ s^{-2}\ V^{-1}] \tag{3.51}$$

$$C_P = \left.(\dot{\omega}_P - (A_P\ \omega_P + B_P\ v))\right|_{\omega_P\ =\ \omega_H} = \frac{d}{J_{TP}}\ {\omega_H}^2 = 489\ [rad^2\ s^{-2}] \tag{3.52}$$

Therefore, with just these three parameters it is possible to describe the dynamics of all the four motors systems. Equation (3.53) shows the differential equation in a matrix form.

$$\dot{\boldsymbol{\Omega}} = A_P\ \boldsymbol{\Omega} + B_P\ \boldsymbol{v} + C_P \tag{3.53}$$

Where $\boldsymbol{\Omega}$ $[rad\ s^{-1}]$ is the propellers' speed vector, $\dot{\boldsymbol{\Omega}}$ $[rad\ s^{-2}]$ is the propellers' acceleration vector and $\boldsymbol{v}$ $[V]$ is the inputs' voltage vector.

This last equation will be used in more detail in the next chapter.

3.4 System architecture

This last section provide an overview of the architecture presenting the devices and their connections.

The quadrotor structure is equipped with a RF receiver, a STR730 Micro Controller Unit (MCU), four motor's power boards and several sensors to provide a stable autonomous system. A general block diagram of the architecture is provided by figure 3.10.

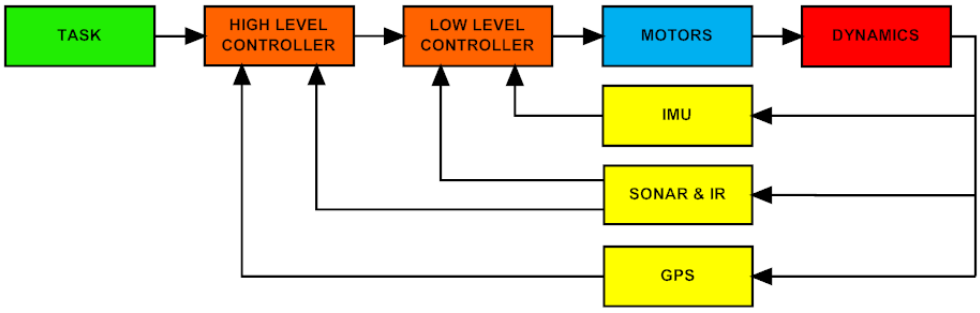


Figure 3.10: *Control architecture*

The task block represents the goal of the quadrotor. It can be burned in the MCU's ROM or provided by the user through a wireless communication. In the first case the robot will behave to follow its (fixed) high level task and no external communication is required. In the second case, the task can be changed by the user. This configuration is more flexible and is the one implemented in this work. The interface between the user and the quadrotor is fulfilled by a Remote Controller (RC) equipped with a RF transmitter which sends the task information to a RF receiver mounted on the quadrotor.

The high level controller block represents an algorithm on the MCU which provide the mapping, the trajectory planning and the obstacle avoidance. Its task is not to stabilize the quadrotor, but to make it navigate with safety. From figure 3.10 it shows that the input of this block is not only the task, but also the GPS and the SONAR & IR. With this information the algorithm is able to

provide a trajectory, send the references to the low level controller and verify the progress of the motion through the sensors in feedback. In this work, the task is directly given in a low level format, therefore this block is not implemented. However it can be added in future through software upgrades.

Even the low level controller block represents an algorithm on the MCU, but its purpose is to stabilize the quadrotor. This program is also indirectly composed of the description of the dynamics of the robot. Hence it is possible to define a control law which bind the motors' inputs to the quadrotor linear and angular position. Two control laws have been investigated as shown in the next chapter. Furthermore the dynamics of the motors has been taken into account and a proportional-integral controller was developed to provide a specific propeller's speed from its voltage input. The inputs of this block are the high level controller, the Inertial Measurement Unit (IMU) and the SONAR & IR blocks. Through the sensors it is possible to have a feedback on the position and provide autonomous stabilization.

The motors block is composed of the motor's power boards and the motor-propeller systems. The power boards are essential to supply the voltage and current needed by the motor. It also provide a current feedback to a MCU's ADC to observe the state of the motor. This circuit is analyzed in chapter 8. The motor-propeller system was instead presented in the previous section. It can be stated that the motors block binds the MCU algorithms to the quadrotor dynamics through electronics and mechanics components.

The dynamics block represents how the quadrotor behave by varying the propellers' speeds. Its parameters can not be changed (in first approximation) because they are referred to the physical structure. It is ruled by equation (3.30) in section 3.2 and equation (3.47) in section 3.3. It is very important to model the quadrotor with high accuracy because all the model parameters are used in the low level controller to provide a good control. Therefore, a detailed work of constant identification has been done and reported in chapter 6.

The IMU block represents the physical device which provides information about the quadrotor's attitude and heading. It is composed of three accelerometers, three magnetometers, three gyroscopes, a barometer and a temperature sensor. Thanks to all these components, the IMU calculates the roll-pitch-yaw angles (ϕ, θ, ψ defined in section 3.2) and it sends them to a MCU's UART. The communication is provided through a digital RS-232 interface. Furthermore, thanks to the three gyroscopes, it is possible to have a feedback about the quadrotor's angular speeds in the fixed-body frame (p, q, r defined in section 3.2). The IMU is therefore an essential device for UAVs. Its capabilities and interfaces are shown, with a higher level of detail, in chapter 7.

The GPS block represents the device which provides information about the quadrotor's global position. It is physically built in the IMU, but its information is not provided by inertial sensors. The GPS has a position accuracy of about 10 meters, hence it is not used in the low level controller because the error is too large. Furthermore the GPS doesn't work properly in indoor applications. However it can be connected to the high level controller as a feedback for the desired outdoor global trajectory.

The SONAR & IR block represents the devices involved to estimate the distance from the quadrotor to an obstacle in a certain direction. Two different systems have been mounted: the SONAR and the IR module. The SONAR detect the distance of an obstacle thanks to ultrasound waves. It's beam width is quite wide: 55 degrees. Furthermore the SONAR is mounted on a PCB which provide an I^2C digital communication to the MCU. The IR module uses instead light waves and has a narrower beam. It is also mounted on a PCB, but it is connected to the MCU thanks to a MCU's ADC since the information is analog. These devices are used to estimate the height of the quadrotor from the ground and to provide information on the availability of the space around itself. Modules' performance and communications are shown in chapter 7.

Chapter 4

Control algorithms

The control algorithms tested in this work are presented in this chapter. The first stage tests were performed on the Matlab simulated model where it was easy to evaluate the performance with a mathematical approach. The second stage tests were carried out on the quadrotor platform to evaluate the behavior of the real system. This chapter is strictly connected with the previous one (3), because it analyzes the quadrotor model and tries to "invert" it to reach a certain attitude and height.

The first section (4.1: Control modelling) shows the basic quadrotor model simplifications. These must be done to be able to use an easier controller and to lower the algorithm complexity. In addition, thanks to the parameters determined in chapter 6 (Identification of the constants), further reductions were possible in the control chain.

The second section (4.2: PID techniques) introduces the PID theory and its strengths. After that it shows and explains in detail the four inner control diagrams. Their goal is to determine the basic movement signals from attitude and height data (sensors) and from task references (remote controller). According to the controlled variable, an enhanced PID structure has been implemented.

4.1 Control modelling

The dynamics of the quadrotor is well described in the previous chapter. However the most important concepts can be summarized in equations (4.1), (4.2) and (4.3). The first one shows how the quadrotor accelerates according to the basic movement commands given.

$$\begin{cases} \ddot{X} &= (\sin\psi\sin\phi + \cos\psi\sin\theta\cos\phi)\,\dfrac{U_1}{m} \\ \ddot{Y} &= (-\cos\psi\sin\phi + \sin\psi\sin\theta\cos\phi)\,\dfrac{U_1}{m} \\ \ddot{Z} &= -g + (\cos\theta\cos\phi)\,\dfrac{U_1}{m} \\ \dot{p} &= \dfrac{I_{YY} - I_{ZZ}}{I_{XX}}\; q\; r - \dfrac{J_{TP}}{I_{XX}}\; q\;\Omega + \dfrac{U_2}{I_{XX}} \\ \dot{q} &= \dfrac{I_{ZZ} - I_{XX}}{I_{YY}}\; p\; r + \dfrac{J_{TP}}{I_{YY}}\; p\;\Omega + \dfrac{U_3}{I_{YY}} \\ \dot{r} &= \dfrac{I_{XX} - I_{YY}}{I_{ZZ}}\; p\; q + \dfrac{U_4}{I_{ZZ}} \end{cases} \tag{4.1}$$

The second system of equations explains how the basic movements are related to the propellers' squared speed.

$$\begin{cases} U_1 = b\;({\Omega_1}^2 + {\Omega_2}^2 + {\Omega_3}^2 + {\Omega_4}^2) \\ U_2 = l\;b\;(-{\Omega_2}^2 + {\Omega_4}^2) \\ U_3 = l\;b\;(-{\Omega_1}^2 + {\Omega_3}^2) \\ U_4 = d\;(-{\Omega_1}^2 + {\Omega_2}^2 - {\Omega_3}^2 + {\Omega_4}^2) \\ \Omega \;= -\Omega_1 + \Omega_2 - \Omega_3 + \Omega_4 \end{cases} \tag{4.2}$$

The third equation takes into accounts the motors dynamics and shows the relation between propellers' speed and motors' voltage.

$$\left(J_P + \eta\; N^2\; J_M\right)\;\dot{\boldsymbol{\Omega}} = -\frac{K_E\; K_M}{R}\;\eta\; N^2\;\boldsymbol{\Omega} - d\;\boldsymbol{\Omega}^2 + \frac{K_M}{R}\;\eta\; N\;\boldsymbol{v} \tag{4.3}$$

With this approach it is possible (in theory) to determine the quadrotor position by double integrating its accelerations (linear and angular). To do this operation, just the internal state and the four motor voltages must be managed. This process is also known as direct kinematics and direct dynamics.

The goal of the quadrotor stabilization is to find those value of the motors' voltage which maintains the helicopter in a certain position required in the task. This process is also known as inverse kinematics and inverse dynamics. Unlike the direct ones, the inverses operations are not always possible and not always unique. For these reasons their consideration is much more complicated.

The quadrotor dynamics must be simplified a lot to provide an easy inverse model which can be implemented in the control algorithms. Equation (4.1) can be rearranged according three considerations:

- The angular contributes are quite complex because several variables have been taken into account. Most of those come from cross coupling of angular speeds (gyroscopic effects and Coriolis-centripetal form). Since the motion of the quadrotor can be assumed close to the hovering condition, small angular changes occur (especially for roll and pitch). It follows that these terms can be simplified because smaller than the main ones.
- The angular accelerations are referred to the angles of the quadrotor measured in its fixed frame. They are not equal to the acceleration of the Euler angles which determines the attitude in the earth frame. The transfer matrix $\boldsymbol{T_\Theta}$ (reported in equation (A.14)) defines the relation between the angular velocities in the earth frame and those ones in the body-fixed frame. Since in hovering condition it is close to the identity matrix, the acceleration equations have been referred directly to the Euler angle accelerations.
- The whole control algorithm is used to give the right signals to the propellers. Since they are four, no more than four variables can be controlled in the loop. From the beginning of the project, it has been decided to stabilize attitude (Euler angles) and height. According to this choice, the equations which describe the X and Y position have been deleted.

Equation (4.4) shows the quadrotor dynamics used in the control.

$$\begin{cases} \ddot{Z} = -g + (\cos\theta\cos\phi)\,\dfrac{U_1}{m} \\ \ddot{\phi} = \dfrac{U_2}{I_{XX}} \\ \ddot{\theta} = \dfrac{U_3}{I_{YY}} \\ \ddot{\psi} = \dfrac{U_4}{I_{ZZ}} \end{cases} \tag{4.4}$$

The control algorithm receives, as inputs, the data from the sensors and from the task. During the computation it uses a lot of constants and variables which describe the dynamics and the quadrotor states. The output of the algorithm is the code which determine the PWM signal of the four motors. The controller can be divided in four components according to figure 4.1.

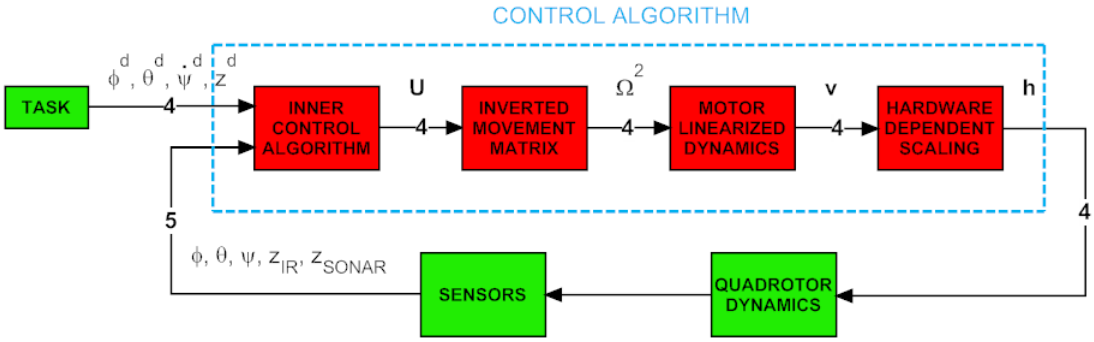


Figure 4.1: *Control block diagram*

- "INNER CONTROL ALGORITHMS" represents the core of the control algorithms. It processes the task and the sensors data and provides a signal for each basic movements which balances the position error. Equation (4.4) is used in this block to transfer an acceleration command to a basic movement one. The control rules used to estimate the acceleration commands are PID techniques. The implementation of this block will be explained with better accuracy in the next section.

- "INVERTED MOVEMENTS MATRIX" is the second block in the control chain. It is used to compute the propellers' squared speed from the four basic movement signals. Since the determinant of the movement matrix is different than zero, it can be inverted to find the relation $\boldsymbol{U}$ to $\boldsymbol{\Omega}^2$. The block computation is shown in equation (4.5).

$$\begin{cases} {\Omega_1}^2 = \frac{1}{4\,b}\,U_1 - \frac{1}{2\,b\,l}\,U_3 - \frac{1}{4\,d}\,U_4 \\ {\Omega_2}^2 = \frac{1}{4\,b}\,U_1 - \frac{1}{2\,b\,l}\,U_2 + \frac{1}{4\,d}\,U_4 \\ {\Omega_3}^2 = \frac{1}{4\,b}\,U_1 + \frac{1}{2\,b\,l}\,U_3 - \frac{1}{4\,d}\,U_4 \\ {\Omega_4}^2 = \frac{1}{4\,b}\,U_1 + \frac{1}{2\,b\,l}\,U_2 + \frac{1}{4\,d}\,U_4 \end{cases} \tag{4.5}$$

- "MOTOR LINEARIZED DYNAMICS" is the third block in the control chain. The motor dynamics is a non linear differential equation represented in equation (4.3). Several approaches can be followed in this block.

  The first one is simply to solve numerically the equation to obtain the right voltage values.

  The second one is to linearize the equation to reduce the computation (in comparison with the previous option). The linearized method is discussed with a higher level of detail in the section DC-motor of chapter 3.

  The third method is to characterize experimentally the motor behavior and use the deduced relation. This method is discussed instead in the section Experimental data extrapolation of chapter 6.

  This third method has been adopted because of its simplicity and the certainty of the model description. Even though the relation $\boldsymbol{\Omega}^2$ to $\boldsymbol{v}$ shows a first order (one pole) transfer function, it has been decided to characterize it just with a DC gain to reduce even further the complexity. This radical simplification is however not that influent because of the strength of the closed loop performance.

- "HARDWARE DEPENDENT SCALING" is the last block in the control chain. It process the inputs voltage vector ($\boldsymbol{v}$) and provide the PWM code vector ($\boldsymbol{h}$) to the hardware. The control algorithm before this block doesn't take into account the battery voltage. This variable is instead very important to determine the right commands. Therefore a scaling must be added to consider the voltage.

  Furthermore this block is used to avoid distortion in case of saturation. This means that if the required voltage is above the battery one, the algorithms gives priority to the attitude over the height. Without this operations an undesired behavior can occur.

4.2 PID techniques

In the industrial area the most used liner regulators are surely the PID. The reasons of this success are mainly three [35]:

- simple structure,
- good performance for several processes,
- tunable even without a specific model of the controlled system.

In robotics, PID technique represents the basics of control. Even though a lot of different algorithms provide better performance than PID, this last structure is often chosen for the reasons expressed above.

The traditional PID structure is composed of the addition of three contributes, as shown in figure 4.2 and equation (4.6).

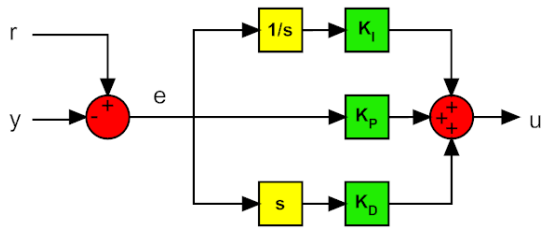


Figure 4.2: *Traditional PID structure*

The blocks "1/s" and "s" represents the integration and derivation operations.

$$u(t) = K_P\ e(t) + K_I \int_0^t e(\tau)\ d\tau + K_Z\ \frac{de(t)}{dt} \tag{4.6}$$

Where u is a generic controlled variable, e is the error between the task r and the process output y, K_P is the proportional coefficient, K_I is the integral coefficient and K_D is the derivative coefficient.

The first contribute (P) is proportional to the error and define the proportional bandwidth. Inside this interval the output will be proportional to the error while outside the output will be minimum or maximum. The second contribute (I) varies according to the integral of the error. Even though this component increases the overshoot and the settling time, it has a unique propriety: it eliminates the steady state error. The third contribute (D) varies according to the derivate of the error. This component help to decrease the overshoot and the settling time.

In the Laplace domain, the traditional PID structure can be rewritten according to equation (4.7).

$$u(s) = \left(K_P + \frac{K_I}{s} + s\ K_D \right) e(s) \tag{4.7}$$

Since this function is improper, it is not physically feasible (because of the derivative therm). After a certain frequency, the D contribute must be attenuated to filter the off-band noise. For this reason, in the real derivator a pole is added as shown in equation (4.8)

$$u(s) = \left(K_P + \frac{K_I}{s} + \frac{s\ K_D}{1 + s\ K_D/(k\ K_P)} \right) e(s) \tag{4.8}$$

The traditional PID structure presents two main drawbacks:

- the derivate action is calculated from the error. If the task adds a step in the reference, the output of the derivator would present an impulse. This sharp movement can saturate the actuators and push away the system from the linear zone. For this reasons most of the PID architecture presents the derivate action of the process output only.
- The integral action combined with an actuator saturation can provide a non linear effect which can decrease the performance of the control system. When the integral value is large and the error changes sign it is necessary to wait a lot of time before the system restores its linear behavior (after the "discharging" of the integral action). This phenomenon is called integral wind-up. To avoid it, a saturator is added after the integral to limit its maximum and minimum vaues.

Figure 4.3 shows the enhanced PID structure.

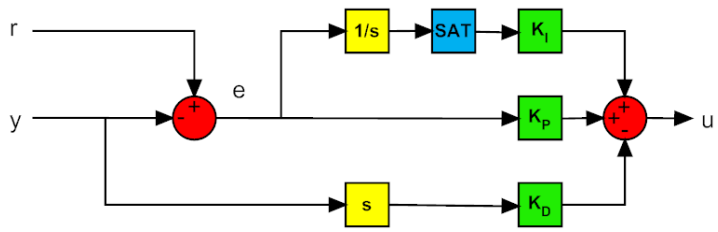


Figure 4.3: *Enhanced PID structure*

The block "SAT" represent the saturator.

The description of the four inner control algorithms for the height and attitude stabilization is now presented.

- Roll control:

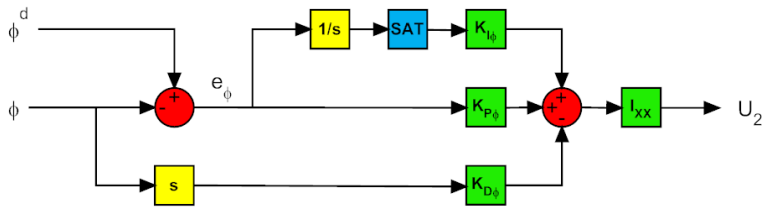


Figure 4.4: *Block diagram of the roll control*

ϕ^d $[rad]$ represents the desired roll angle, ϕ $[rad]$ is the measured roll angle, e_ϕ $[rad]$ is the roll error and U_2 $[N\ m]$ is the required roll torque. $K_{P\phi}$ $[s^{-2}]$, $K_{I\phi}$ $[s^{-3}]$ and $K_{D\phi}$ $[s^{-1}]$ are the three control parameters. At last I_{XX} $[N\ m]$ is the body moment of inertia around the x-axis. It can be noted that the roll stabilization has a structure very close to that one explained before. The only difference is that there is the block "I_{XX}" after the sum of the three main components. This contribute comes from equation (4.4) and is necessary to relate the roll control to U_2.

- Pitch control:

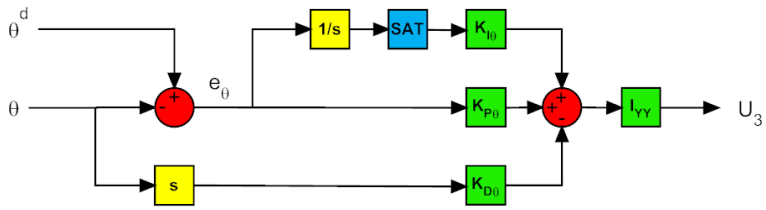


Figure 4.5: *Block diagram of the pitch control*

θ^d $[rad]$ represents the desired pitch angle, θ $[rad]$ is the measured pitch angle, e_θ $[rad]$ is the pitch error and U_3 $[N\ m]$ is the required pitch torque. $K_{P\theta}$ $[s^{-2}]$, $K_{I\theta}$ $[s^{-3}]$ and $K_{D\theta}$ $[s^{-1}]$ are the three control parameters. At last I_{YY} $[N\ m]$ is the body moment of inertia around the y-axis. It is easy to see that the pitch stabilization has a structure very close to the roll one. The only difference is that the roll acts around the x axis while the pitch acts around the y axis, according to equation (4.4).

- Yaw control:

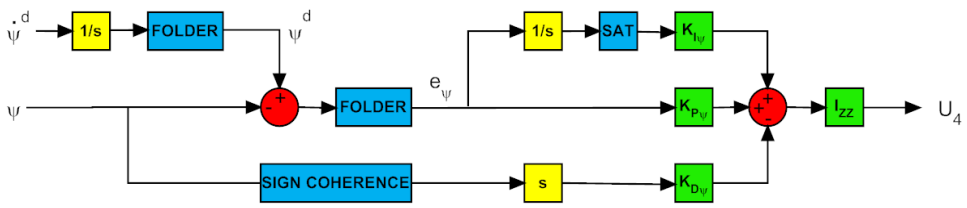


Figure 4.6: *Block diagram of the yaw control*

$\dot{\psi}^d$ $[rad\ s^{-1}]$ represents the desired yaw angle velocity, ψ^d $[rad]$ represents the desired yaw angle, ψ $[rad]$ is the measured yaw angle, e_ψ $[rad]$ is the yaw error and U_4 $[N\ m]$ is the required yaw torque. $K_{P\psi}$ $[s^{-2}]$, $K_{I\psi}$ $[s^{-3}]$ and $K_{D\psi}$ $[s^{-1}]$ are the three control parameters. At last I_{ZZ} $[N\ m]$ is the body moment of inertia around the z-axis. The "I_{ZZ}" block is needed to relate the yaw control to U_4, according to equation (4.4).

This structure presents a few differences from the previous two.

The new block "FOLDER" represents a code which allow to take into account the discontinuity of the yaw angle at $\pm\pi$. Therefore if the input of the block is equal to $3\pi/2$, it outputs $-\pi/2$.

The new block "SIGN COHERENCE" is instead a little bit more complex. It is needed to consider the sign of the two sample for the derivate computation. Once again, the purpose of this algorithm is used to avoid error during the transition π to $-\pi$ and vice versa.

Another big difference from the previous structure is that the task gives a reference on the derivate of the yaw angle and not directly on the yaw angle. This configuration has been chosen to be able to fly the helicopter with a higher degree of freedom. With a position reference, it would be impossible to make the quadrotor turn on itself more than one time (without losses of accuracy). Furthermore the positioning for the take-off sequence could not be arbitrarily chosen. By integrating the derivate of the yaw angle reference, it is easy to obtain the yaw angle reference and to process it, as in the previous structures.

- height control:

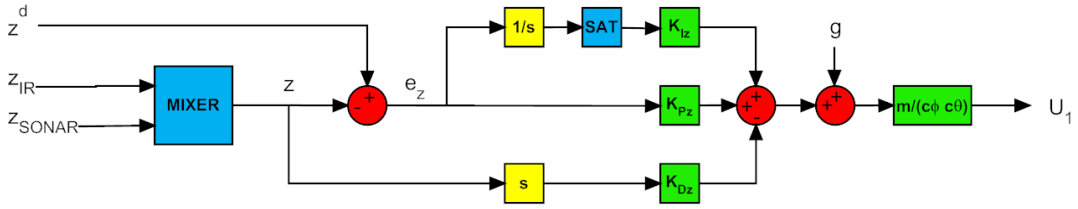


Figure 4.7: *Block diagram of the height control*

z^d $[m]$ represents the desired height, z_{IR} $[m]$ is the height measured by the IR module, z_{SONAR} $[m]$ is the height measured by the SONAR module, z $[m]$ is the height estimated from the sensors, e_z $[m]$ is the height error and U_1 $[N]$ is the required thrust. K_{Pz} $[s^{-2}]$, K_{Iz} $[s^{-3}]$ and K_{Dz} $[s^{-1}]$ are the three control parameters. At last g $[m\ s^{-2}]$ is the acceleration due to gravity, m $[kg]$ is the mass of the quadrotor, $c\phi$ $[-]$ is the roll angle cosine and $c\theta$ $[-]$ is the pitch angle cosine.

Respect to the enhanced PID architecture, the block diagram of the height control presents two main differences:

The "MIXER" block must be added to process the height data from the IR module and the SONAR. Its purpose is to calculate the height of the quadrotor with the best accuracy it can achieve from the two sensors. Furthermore, IR and SONAR have different ranges and, for specific heights, one of the two sensors can provide wrong data. Thanks to the "MIXER" block, the algorithm estimates the real height and avoids ambiguity.

According to equation (4.4) the height dynamics is more complex than the other three. In fact, it also depends from the roll and pitch angles. Furthermore the acceleration due to gravity must be compensated. The quadrotor mass (m) has the same role as the moments of inertia in the angular case.

Chapter 5

Quadrotor simulator

In this chapter, the quadrotor simulator is presented. This tool is very helpful to verify the correctness of the helicopter dynamic model and to test the control algorithms performance. Furthermore, thanks to a real-time interface with the remote controller, it is possible to evaluate the behavior of the quadrotor through a 3D view. This simulator has been developed with the Matlab tool Simulink.

The first section (5.1: System structure) introduces the simulation tools Matlab and Simulink. Furthermore it provides an overview of the quadrotor system architecture and it gives a brief description of blocks and commands.

The second section (5.2: Blocks implementation) shows the implementation of the previously introduced blocks. Particular attention is given to the model of the dynamics, the quantization (in both time and frequency) of the sensors signals, the interface of the inputs and the portability of the control algorithm.

The third section (5.3: 3D visualization) underlines the importance of a real-time platform and shows the 3D interface. Thanks to this visualization, it is easy to analyze the position and the orientation of the quadrotor, hence verify its performance.

5.1 System structure

Matlab is a high-level technical computing language and interactive environment for algorithm development, data visualization, data analysis and numeric computation [36]. It is widely used in engineering and science because of its easy interface and powerful commands.

The main strengths of Matlab are [37]:

- Is relatively easy to learn.
- Optimized code to be relatively quick when performing matrix operations.
- May behave like a calculator or as a programming language.
- Is an interpreted language, errors are easier to fix.

Matlab main weakness is instead its slowness: it is almost always much slower than a compiled language such as C (since it is an interpreted language).

Simulink is an environment for multidomain simulation and Model-Based Design for dynamic and embedded systems. It provides an interactive graphical environment and a customizable set of block libraries which allow to design, simulate, implement, and test a variety of time-varying systems [36]. Simulink has been chosen in this work for its easy and clear graphic interface.

The model of the whole system is composed of several interconnected blocks in a classic feedback structure. "dynamics" represents the physics of the quadrotor and provides the position, velocity and acceleration of both linear and angular quantities. The actuators dynamic is also modeled in this block. The input of this block is the voltage given to the motor drivers (Vcontrol) from the "control". "sensors" models the IMU, IR and SONAR. It gives its data directly to the "control", while it receives information about the quadrotor motion from "dynamics". "inputs" acquires the task references directly form the helicopter remote controller or from a signal builder. As for the "sensors", the output of "inputs" is directly connected with "control". This last block includes the control algorithm for the helicopter stabilization (the same as the microcontroller code on the real platform). "3d view" implements the code that allow to see the quadrotor in a 3D

visualization during the real-time simulation. The two blocks "pos-vel-acc scope" and "Vcontrol scope" are useful in the test phase to analyze (once the simulation is stopped) the waveforms of position, velocity, acceleration and Vcontrol of the helicopter. Figure 5.1 shows a snapshot of the system structure in Simulink.

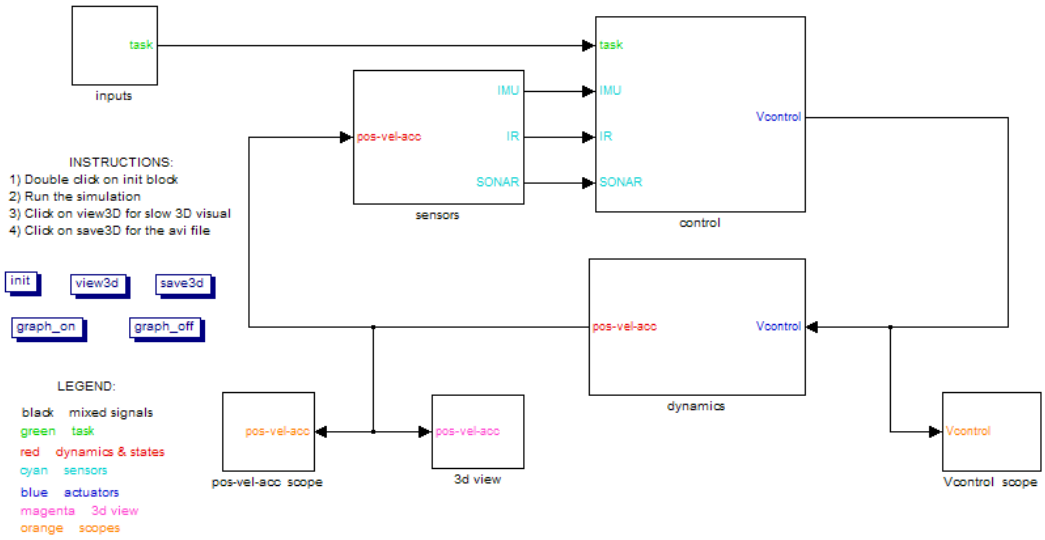


Figure 5.1: *System structure*

On the left side of the schematic, the instructions specify how to run the simulation, manage the 3D visualization and save it in an AVI file. The following five buttons are located under the instruction:

- "init" initializes all the constants and variables.
- "view3d" shows the last simulation data with a 3D visualization.
- "save3d" saves the 3D visualization with the last data in a AVI video.
- "graph_on" enables the 3D interface during the simulation (visual feedback).
- "graph_off" disables the 3D interface during the simulation (low complexity).

Furthermore, several colors are used in this scheme to identify the signals characteristic (as shown in the legend).

5.2 Blocks implementation

"dynamics" represents the physics of the quadrotor and provides the position, velocity and acceleration of both linear and angular quantities. The input of this block is the voltage given to the motor drivers (Vcontrol) from the "control". Figure 5.2 shows a snapshot of the "dynamics" implementation in Simulink.

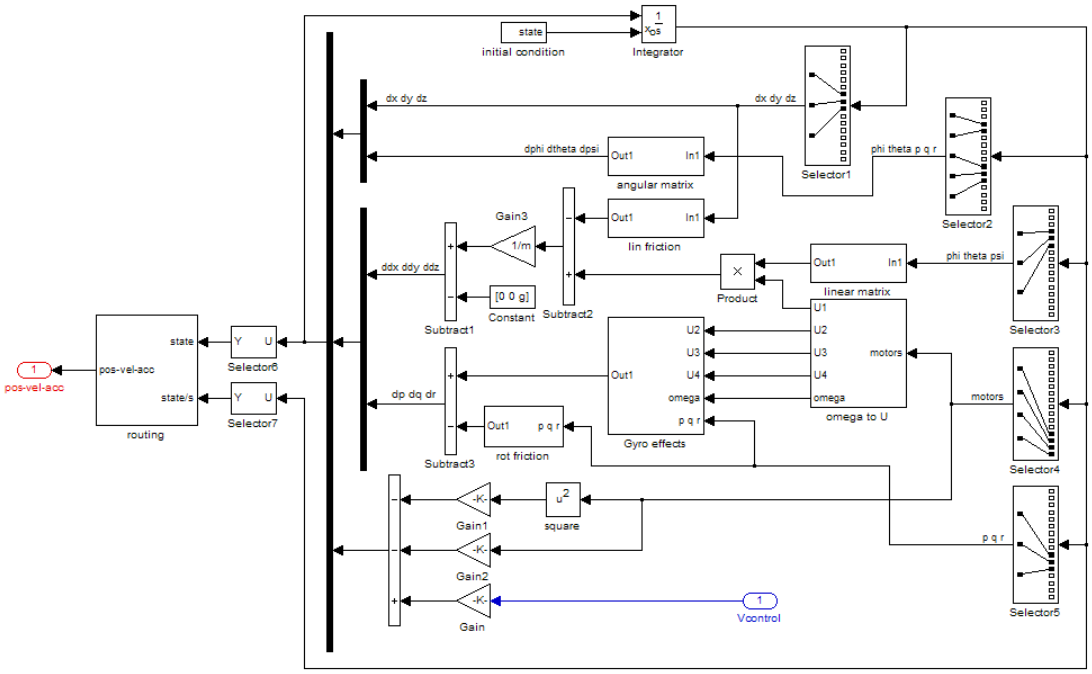


Figure 5.2: *Dynamics implementation*

This block (as most of the following) is implemented with a lot of hardware blocks instead of a software code. The reason of this choice is that a lot of time is saved for the computation and the simulation is much faster.
The propellers' speed is calculated from its past values and from the input "Vcontrol" according to the theory presented in chepter 3.
The linear velocity vector "dx dy dz" is obtained directly from the integral of the linear acceleration vector "ddx ddy ddz".
The angular velocity vector "dphi dtheta dpsi" is obtained from the integral of both the angular acceleration vector "dp dq dr" and itself through the "angular matrix" transformation.
The linear acceleration vector "ddx ddy ddz" is composed of two components: the "linear friction" (depending on the linear velocity) and the product between the "linear matrix" and the "U1" command (obtained from the propellers' speed).

The angular acceleration vector "dp dq dr" is also composed of two components: the "rot friction" (depending on the angular velocity vector) and the sum of the "gyro effects" and the "omega to U" (obtained from both the propellers' speed and the angular velocity vector).

"sensors" models the IMU, IR and SONAR. It receives information about the quadrotor motion from "dynamics", while it gives its data directly to the "control". It is composed of three sub-models which identifies with a good level of detail the IMU, the IR module and the SONAR. In each sensor implementation four main operation are applied:

- A delay to model the time behavior of the sensor.
- A rate transition block which sample the sensor at the real rate.
- A noise generator to model accuracy imprecision.
- A conversion block to represent the sensor data in their real format

The output format of the IMU is represented (in degrees) with four bytes in the fixed point signed 12.20 format. Its sample period is 0.01 [s]. The "yaw_limiter" block is used to limit the yaw value between 180 and -180 [deg]. Figure 5.3 shows a snapshot of the "IMU sensor" implementation in Simulink.

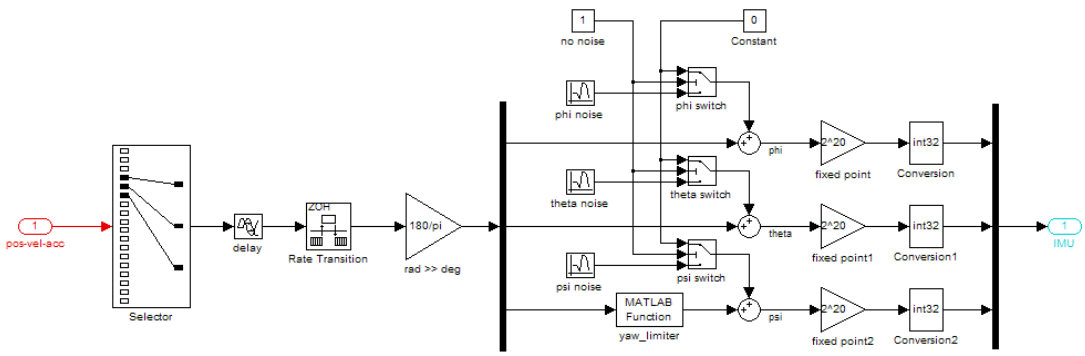


Figure 5.3: *IMU implementation*

The output format of the IR is represented (in centimeters) with one byte (actually, since the IR is analogue, this format is provided after a brief microcontroller computation in the real platform). Its sample period is 0.04 [s]. Figure 5.4 shows a snapshot of the "IR sensor" implementation in Simulink.

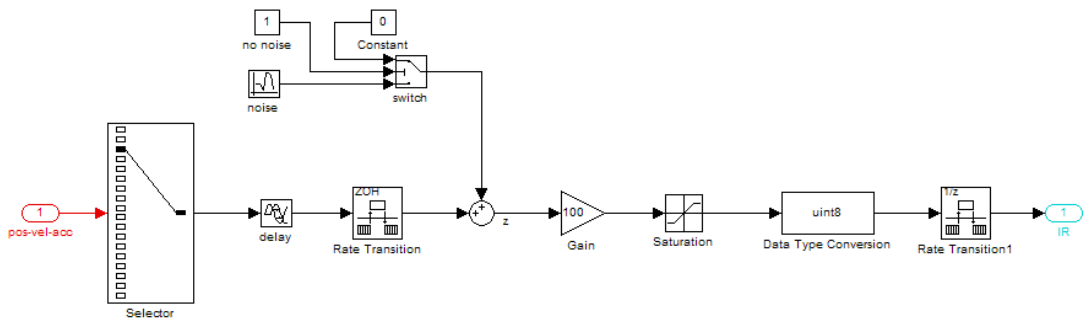


Figure 5.4: *IR implementation*

The output format of the SONAR is represented (in centimeters) with two bytes. Its sample period is 0.07 [s]. Figure 5.5 shows a snapshot of the "SONAR sensor" implementation in Simulink.

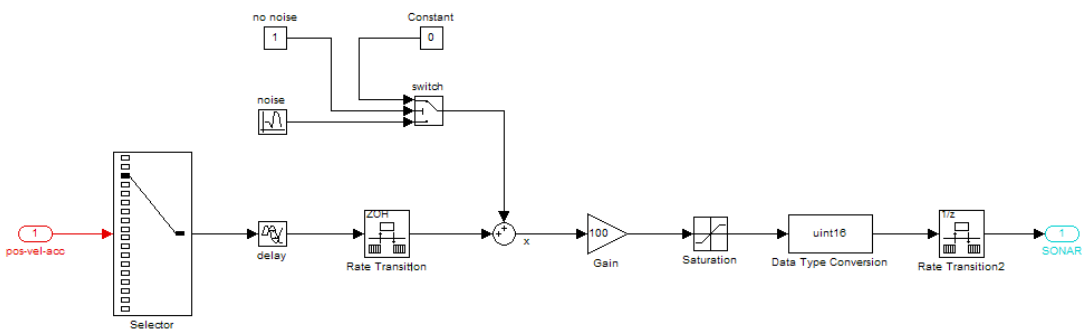


Figure 5.5: *SONAR implementation*

"inputs" acquires the task references directly form the helicopter remote controller or from a signal builder. The output of this block is directly connected with "control". The block "Joystick inputs" provide the interface between the remote controller and Simulink. In the "inputs" block, a lot of mixer are connected because of the signals must be correctly routed according to the specified options. The block "button" is used to define the 3D visual. Figure 5.6 shows a snapshot of the "SONAR sensor" implementation in Simulink.

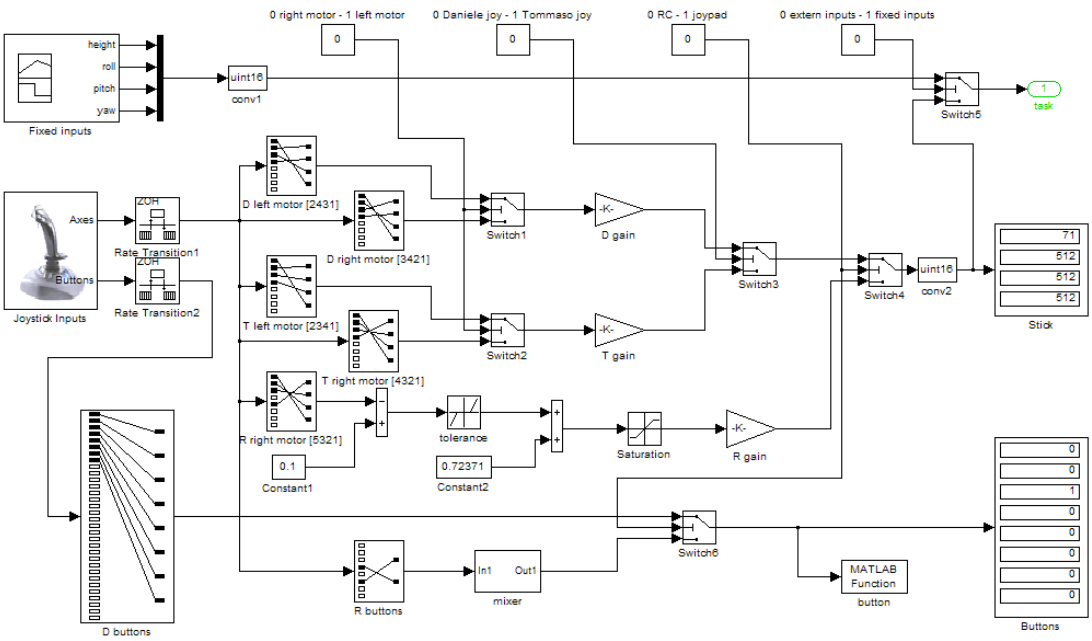


Figure 5.6: *Inputs implementation*

"control" includes the control algorithm for the helicopter stabilization hence determines the motors' voltage "Vcontrol". These data can be generated from a signal builder or from the S-Function Builder. This last block processes the sensors and task signals, hence provides the actuators output through a C code. Figure 5.7 shows a snapshot of the "SONAR sensor" implementation in Simulink.

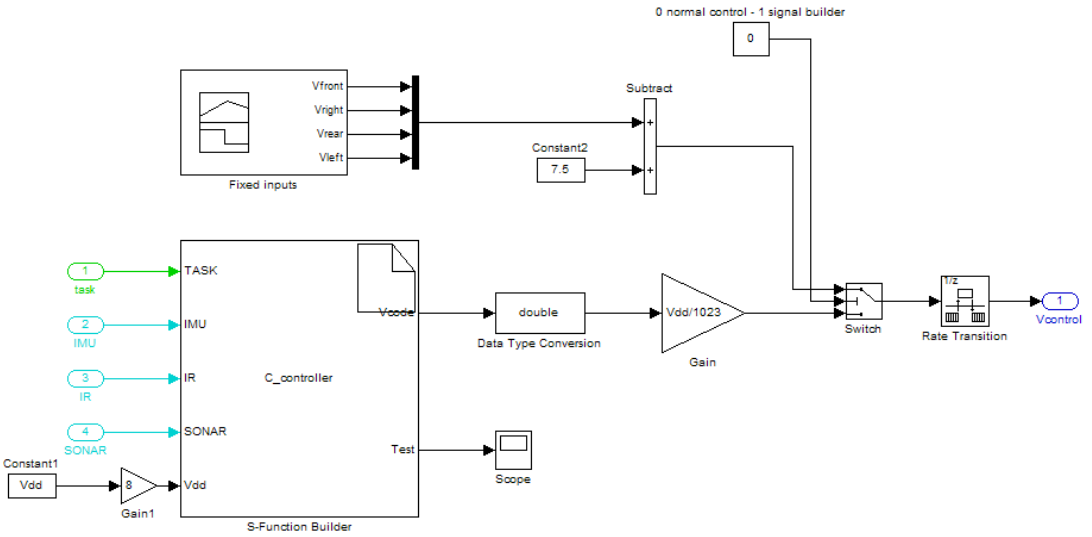


Figure 5.7: *Control implementation*

5.3 3D visualization

To evaluate the stability of the quadrotor using the remote controller, it is very important that the simulator runs in real-time. This means that the simulated time must be equal to the real time. This equivalence is possible thanks to a block called Simulink Execution Control V2.0.

Furthermore the 3D visualization allows to analyze the position and the orientation of the quadrotor, hence verify its performance. The helicopter is represented with four circles (propellers rotation), each at one end of a cross structure. The front circle is red (instead of blue) to mark the orientation. A yellow line shows the path done by the quadrotor from its start position (0,0,0). Three segments represent the three axes direction from the origin (x red, y black, z green). An additional square centered in the xy plane is placed in the 3D structure.

By switching the position of two buttons on the remote controller it is possible to change the 3D visual. Four different configuration of (dynamic) camera position and target are selectable. Figure 5.8 shows a snapshot of the 3D visualization during a simulation.

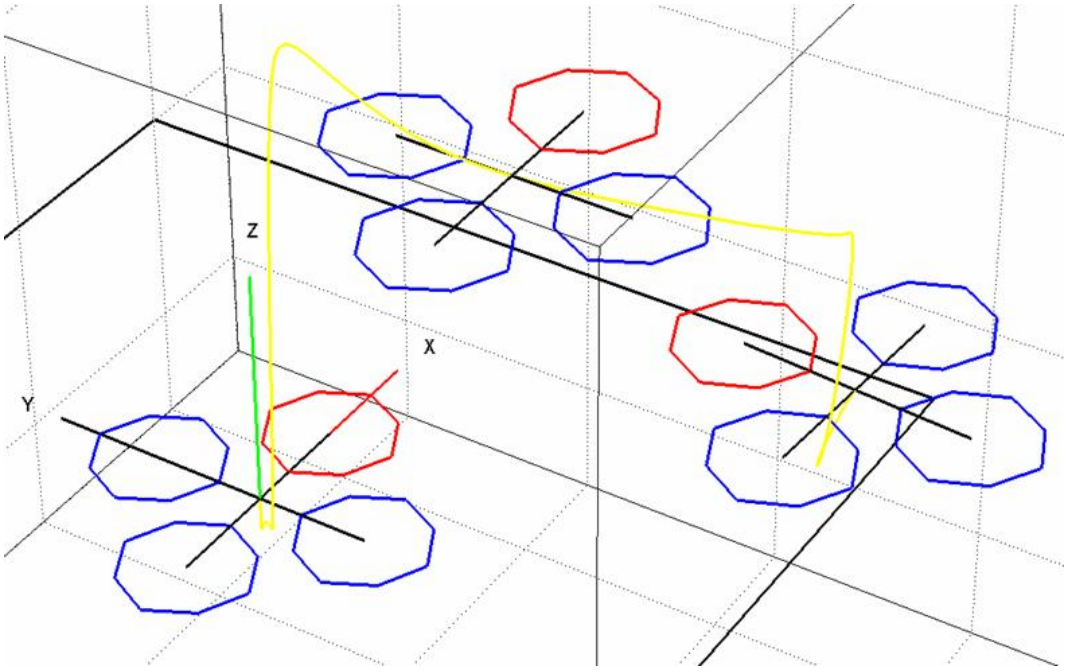


Figure 5.8: *3D visualization*

Chapter 6

Identification of the constants

There are different approaches to estimate a constant. In this work, several methods were applied according to the known information. This chapter shows the techniques used to find all the constants of this project. These can be divided in the following categories:

1. Basic measurements (section 6.1)

2. Geometry and dynamics derivation (section 6.2)

3. Geometry and aerodynamics derivation (section 6.3)

4. Datasheet data extrapolation (section 6.4)

5. Experimental data extrapolation (section 6.5)

6. Constants comparison (section 6.6)

6.1 Basic measurements

This section shows the basic measurements done in the identification process. Three groups of instruments were used:

- mass (weight) measuring
- length measuring
- electric measuring

The mass was estimated thanks to a laboratory digital balance "Kern 440-45N" (full scale 1 kg, accuracy 0.2×10^{-3} kg). The length was instead estimated with two instruments according to the measurement range: an electronic caliper "Mitutoyo 500-181-20"(full scale 153.65×10^{-3} m, accuracy 0.02×10^{-3} m) and a measuring tape (full scale 2 m, accuracy 1×10^{-3} m). All the parts of the quadrotor were weighed and measured to be able to calculate more complex quantities. Figures 6.1, 6.2 and 6.3 show the photos of the previously mentioned measurement instruments.

Figure 6.1: *Electronic balance Kern 440-45N*

Figure 6.2: *Electronic caliper Mitutoyo 500-181-20*

Figure 6.3: *Measuring tape Assist 3M*

To measure electric quantities, two instruments were used: a digital multimeter "GBC KDM-350C" and a digital RLC meter "Agilent 4263B". The RLC meter is a device which can measure resistance (R), inductance (L), and capacitance (C) with great accuracy. The multimeter instead has less accuracy but can measure more electrical quantities in addition to continuity, diode and transistors tests. Table 6.1 shows the comparison of the multimeter and LRC meter features (in the table, NA means not available).

	Mutimeter		**RLC-meter**	
Measure	**range**	**accuracy**	**range**	**accuracy**
Resistance	$200\ \Omega \div 200\ M\Omega$	$\pm$ 0.8 %	$1\ m\Omega \div 100\ M\Omega$	$\pm$ 0.1 %
Inductance	NA	NA	$10\ nH \div 100\ kH$	$\pm$ 0.1 %
Capacitance	$200\ pF \div 20\ \mu F$	$\pm$ 2.5 %	$1\ pF \div 1\ F$	$\pm$ 0.1 %
DC voltage	$200\ mV \div 1000\ V$	$\pm$ 0.5 %	NA	NA
AC voltage	$2\ V \div 700\ V$	$\pm$ 0.8 %	NA	NA
DC current	$2\ mA \div 20\ A$	$\pm$ 0.8 %	NA	NA
AC current	$20\ mA \div 20\ A$	$\pm$ 1.2 %	NA	NA
Frequency	$20\ kHz$	$\pm$ 1 %	NA	NA
Temperature	-20 °C $\div$ 1000 °C	$\pm$ 0.75 %	NA	NA

Table 6.1: multimeter - LRC meter comparison

The RLC meter was mainly used to determine the resistance R and inductance L of the motors. The multimeter was used instead for a lot of electronic circuit measures during identification, developing and testing. Figures 6.4 and 6.5 show the photos of the two previously mentioned instruments.

Figure 6.4: *Digital multimeter GBC KDM-350C*

Figure 6.5: *Digital RLC meter Agilent 4263B*

Another basic measure consists of the determination of the gear box reduction ratio N. This quantity doesn't need any instrument to be estimated. It is equal to the ratio of the number of teeth of the propeller gear to the number of teeth of the motor gear. Another way to determine the same quantity (with probably less accuracy) is to compute the ratio of the propeller gear radius to the motor gear radius.

The last group of basic measures are those determined by others. A few measurements have already been estimated from previous work and experiments with good accuracy. It was a good idea though to use them instead of recalculate their values. The "borrowed" constants are:

- a, lift slope [38]
- g, acceleration due to gravity [39]
- C_D, drag coefficient [38]
- I_{XY}, I_{YX}, I_{XZ}, I_{ZX}, I_{YZ}, I_{ZY}, off-diagonal elements of the body moment of inertia tensor [16]
- η, gear box efficiency [40]
- ρ_A, air density [39]

6.2 Geometry and dynamics derivation

This section uses the data derived by basic measurements to identify the body and rotational moments of inertia. To deepen the basics of the moment of inertia theory see appendix D. This identification can be divided in 2 groups:

- Rotational moment of inertia calculation
- Body moment of inertia calculation

In the first part, the main goal is to derive the total rotational moment of inertia around the motor axis. It is composed of two constants: the rotational moment of inertia around the motor axis J_M $[N\ m\ s^2]$ and the rotational moment of inertia around the propeller axis J_P $[N\ m\ s^2]$. Figure 6.6 shows all the rotational moment of inertia involved in the motor - gear box - propeller system.

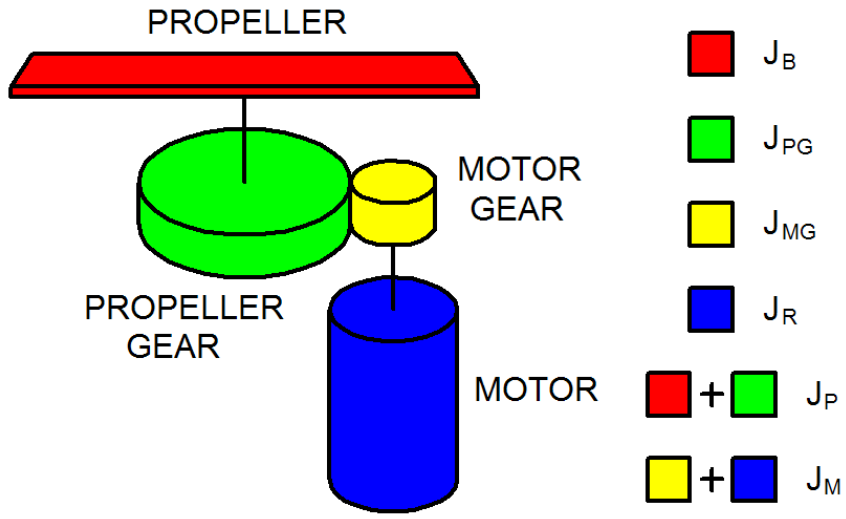


Figure 6.6: *Rotor inertia*

J_M itself is also composed of two contributions: the rotor and the motor gear rotational moments of inertia (J_R $[N\ m\ s^2]$ and J_{MG} $[N\ m\ s^2]$). Both can be modeled as a solid cylinder, therefore their moments of inertia are equal to (from appendix D):

$$J_R = \frac{1}{2}\ M_R\ {R_R}^2 = 1.08 \times 10^{-6}\ [N\ m\ s^2] \tag{6.1}$$

$$J_{MG} = \frac{1}{2}\ M_{MG}\, {R_{MG}}^2 = 12.8 \times 10^{-9}\ [N\ m\ s^2] \tag{6.2}$$

$$J_M = J_R + J_{MG} = 1.1 \times 10^{-6}\ [N\ m\ s^2] \tag{6.3}$$

Even J_P is composed of two contributions: the blade and the propeller gear rotational moments of inertia (J_B $[N\ m\ s^2]$ and J_{PG} $[N\ m\ s^2]$). The first one can be modeled as a flat plate while the second one as a solid cylinder. Their moments of inertia are equal to (from appendix D):

$$J_B = \frac{1}{12}\ M_B\ ({W_B}^2 + {L_B}^2) = 70.8 \times 10^{-6}\ [N\ m\ s^2] \tag{6.4}$$

$$J_{PG} = \frac{1}{2}\ M_{PG}\ {R_{PG}}^2 = 1.38 \times 10^{-6}\ [N\ m\ s^2] \tag{6.5}$$

$$J_P = J_B + J_{PG} = 72.2 \times 10^{-6}\ [N\ m\ s^2] \tag{6.6}$$

According to equations (3.43) and (3.48), it is possible to derive the total motor inertia around the motor axis J_{TM} $[N\ m\ s^2]$ and the propeller J_{TP} $[N\ m\ s^2]$.

$$J_{TM} = J_M + \frac{J_P}{\eta\ N^2} = 3.66 \times 10^{-6}\ [N\ m\ s^2] \tag{6.7}$$

$$J_{TP} = J_P + \eta\ N^2\ J_M = 104 \times 10^{-6}\ [N\ m\ s^2] \tag{6.8}$$

In the body moment of inertia calculation, the main task is to identify the dynamic behavior of the whole quadrotor in rotation around a defined axis. Since the structure shows a high degree of symmetry, the body of inertia tensor (see appendix D) can be simplified as a diagonal matrix. Hence, in this part, the main goal is to derive the three body moments of inertia I_{XX}, I_{YY} and I_{ZZ}.
Firs of all, the (complex) structure of the quadrotor has to be modeled as several components with easier geometry [7]. A good approximation is to define the following parts:

- 1 cross structure, modeled as two solid narrow cylinders fastened in the middle forming a cross
- 1 electronics box, modeled as a rectangular parallelepiped
- 4 motors, modeled as solid cylinders
- 4 motor gears, modeled as solid cylinders
- 4 propeller gears, modeled as solid cylinders
- 4 propellers, modeled as flat cylinders with variable density (the geometry comes from a rotating blade)

For each of the previous parts, three moments of inertia have to be computed: one around the x-axis, one around the y-axis and one around the z-axis. It now follows the body moments of inertia computation:

The narrow cylinders of the cross structure model have a radius R_C of 2.5×10^{-3} $[m]$, a length L_C of 470×10^{-3} $[m]$ and a mass M_C of 25×10^{-3} $[kg]$ each. Figure 6.7 shows the geometry of the cross structure.

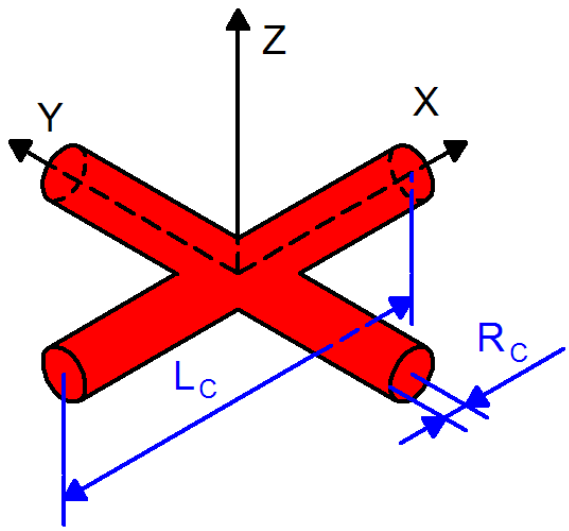


Figure 6.7: *Cross structure*

Thanks to its symmetry, the cross structure moment of inertia around the x-axis I_{CX} $[N\ m\ s^2]$ is equal to that one around the y-axis I_{CY} $[N\ m\ s^2]$:

$$\begin{aligned} I_{CX} = I_{CY} &= M_C \left(\frac{{R_C}^2}{4} + \frac{{L_C}^2}{12} \right) + \frac{1}{2}\ M_C\ {R_C}^2 = \\ &= 460 \times 10^{-6}\ [N\ m\ s^2] \end{aligned} \tag{6.9}$$

$$\begin{aligned} I_{CZ} &= M_C \left(\frac{{R_C}^2}{4} + \frac{{L_C}^2}{12} \right) + M_C \left(\frac{{R_C}^2}{4} + \frac{{L_C}^2}{12} \right) = \\ &= 920 \times 10^{-6}\ [N\ m\ s^2] \end{aligned} \tag{6.10}$$

The rectangular parallelepiped of the electronics box model has a length L_E of 200×10^{-3} $[m]$, a width W_E of 200×10^{-3} $[m]$, a height H_E of 20×10^{-3} $[m]$, a distance from the COM around the z-axis D_E of 70×10^{-3} $[m]$ and a mass M_E of 200×10^{-3} $[kg]$. Figure 6.8 shows the geometry of the electronics box.

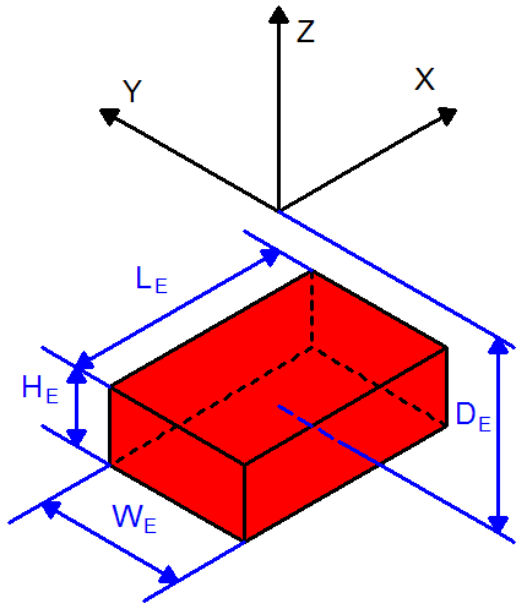


Figure 6.8: *Electronics box*

Its moments of inertia around the x-axis I_{EX} $[N\ m\ s^2]$, y-axis I_{EY} $[N\ m\ s^2]$ and z-axis I_{EY} $[N\ m\ s^2]$ are defined according to equations (6.11), (6.12) and (6.13).

$$I_{EX} = M_E \left(\frac{{W_E}^2}{12} + \frac{{H_E}^2}{12} + {D_E}^2 \right) = 1.7 \times 10^{-3}\ [N\ m\ s^2] \tag{6.11}$$

$$I_{EY} = M_E \left(\frac{{L_E}^2}{12} + \frac{{H_E}^2}{12} + {D_E}^2 \right) = 1.7 \times 10^{-3}\ [N\ m\ s^2] \tag{6.12}$$

$$I_{EZ} = M_E \left(\frac{{L_E}^2}{12} + \frac{{W_E}^2}{12} \right) = 1.3 \times 10^{-6}\ [N\ m\ s^2] \tag{6.13}$$

The four motors show a high degree of symmetry, therefore the computation were done just on the front motor and then transposed on the other ones. The solid cylinder of the motor model has a radius R_M of 12×10^{-3} $[m]$, a height H_M of 30×10^{-3} $[m]$, a distance from the COM around the x-axis L_M of 210×10^{-3} $[m]$, a distance from the COM around the z-axis D_M of 12×10^{-3} $[m]$ and a mass M_M of 46×10^{-3} $[kg]$. Figure 6.9 shows the geometry of the motor.

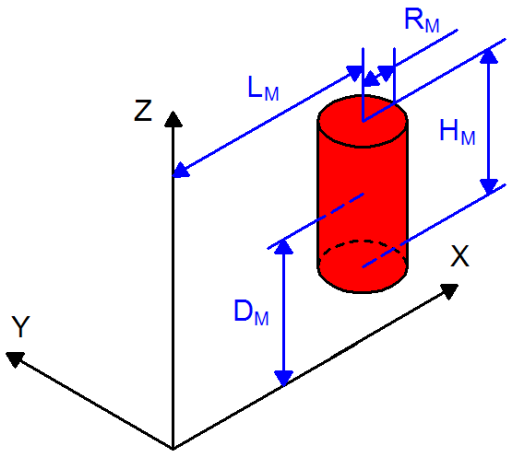


Figure 6.9: *Motor*

Its moments of inertia around the x-axis I_{M1X} $[N\ m\ s^2]$, y-axis I_{M1Y} $[N\ m\ s^2]$ and z-axis I_{M1Z} $[N\ m\ s^2]$ are defined in equations (6.14), (6.15) and (6.16).

$$I_{M1X} = M_M \left(\frac{{R_M}^2}{4} + \frac{{H_M}^2}{12} + {D_M}^2 \right) = 11.7 \times 10^{-6}\ [N\ m\ s^2] \tag{6.14}$$

$$I_{M1Y} = M_M \left(\frac{{R_M}^2}{4} + \frac{{H_M}^2}{12} + {L_M}^2 + {D_M}^2 \right) = 2 \times 10^{-3}\ [N\ m\ s^2] \tag{6.15}$$

$$I_{M1Z} = M_M \left(\frac{{R_M}^2}{2} + {L_M}^2 \right) = 2 \times 10^{-3}\ [N\ m\ s^2] \tag{6.16}$$

According to the geometry, equations (6.17), (6.18) and (6.19) show the derivation of the three moments of inertia of the remaining three motors.

$$I_{M1X} = I_{M2Y} = I_{M3X} = I_{M4Y} \tag{6.17}$$

$$I_{M1Y} = I_{M2X} = I_{M3Y} = I_{M4X} \tag{6.18}$$

$$I_{M1Z} = I_{M2Z} = I_{M3Z} = I_{M4Z} \tag{6.19}$$

Like the motors, even the four motor gears show a high degree of symmetry. Therefore the computation were done just on the front motor gear and then transposed on the other ones. The solid cylinder of the motor gear model has a radius R_{MG} of 4×10^{-3} $[m]$, a height H_{MG} of 5.2×10^{-3} $[m]$, a distance from the COM around the x-axis L_{MG} of 210×10^{-3} $[m]$, a distance from the COM around the z-axis D_{MG} of 10.6×10^{-3} $[m]$ and a mass M_{MG} of 1.6×10^{-3} $[kg]$. Figure 6.10 shows the geometry of the motor gear.

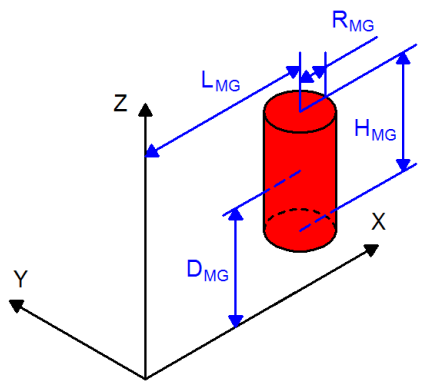


Figure 6.10: *Motor gear*

Its moments of inertia around the x-axis $I_{MG1X}[N\ m\ s^2]$, y-axis $I_{MG1Y}[N\ m\ s^2]$ and z-axis I_{MG1Z} $[N\ m\ s^2]$ are defined in equations (6.20), (6.21) and (6.22).

$$I_{MG1X} = M_{MG}\left(\frac{{R_{MG}}^2}{4} + \frac{{H_{MG}}^2}{12} + {D_{MG}}^2\right) = 0.19 \times 10^{-6}\,[N\ m\ s^2] \quad (6.20)$$

$$I_{MG1Y} = M_{MG}\left(\frac{{R_{MG}}^2}{4} + \frac{{H_{MG}}^2}{12} + {L_{MG}}^2 + {D_{MG}}^2\right) = 70.8\,[N\ m\ s^2] \quad (6.21)$$

$$I_{MG1Z} = M_{MG}\left(\frac{{R_{MG}}^2}{2} + {L_{MG}}^2\right) = 70.6 \times 10^{-3}\,[N\ m\ s^2] \quad (6.22)$$

According to the geometry, equations (6.23), (6.24) and (6.25) show the derivation of the three moments of inertia of the remaining three motor gears.

$$I_{MG1X} = I_{MG2Y} = I_{MG3X} = I_{MG4Y} \quad (6.23)$$

$$I_{MG1Y} = I_{MG2X} = I_{MG3Y} = I_{MG4X} \quad (6.24)$$

$$I_{MG1Z} = I_{MG2Z} = I_{MG3Z} = I_{MG4Z} \quad (6.25)$$

The four propeller gears show a high degree of symmetry (as the motors), therefore the computation were done just on the front propeller gear and then transposed on the other ones. The solid cylinder of the propeller gear model has a radius R_{PG} of 22×10^{-3} $[m]$, a height H_{PG} of 4.5×10^{-3} $[m]$, a distance from the COM around the x-axis L_{PG} of 235×10^{-3} $[m]$, a distance from the COM around the z-axis D_{PG} of 10.6×10^{-3} $[m]$ and a mass M_{PG} of 5.7×10^{-3} $[kg]$. Figure 6.11 shows the geometry of the propeller gear.

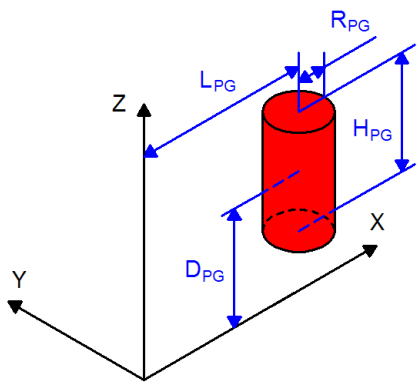


Figure 6.11: *Propeller gear*

Its moments of inertia around the x-axis I_{PG1X} $[N\ m\ s^2]$, y-axis I_{PG1Y} $[N\ m\ s^2]$ and z-axis I_{PG1Z} $[N\ m\ s^2]$ are defined in equations (6.26), (6.27) and (6.28).

$$I_{PG1X} = M_{PG}\left(\frac{{R_{PG}}^2}{4} + \frac{{H_{PG}}^2}{12} + {D_{PG}}^2\right) = 1.3 \times 10^{-6}\,[N\ m\ s^2] \tag{6.26}$$

$$I_{PG1Y} = M_{PG}\left(\frac{{R_{PG}}^2}{4} + \frac{{H_{PG}}^2}{12} + {L_{PG}}^2 + {D_{PG}}^2\right) = 316 \times 10^{-6}\,[N\ m\ s^2] \tag{6.27}$$

$$I_{PG1Z} = M_{PG}\left(\frac{{R_{PG}}^2}{2} + {L_{PG}}^2\right) = 316 \times 10^{-3}\,[N\ m\ s^2] \tag{6.28}$$

According to the geometry, equations (6.29), (6.30) and (6.31) show the derivation of the three moments of inertia of the remaining three propeller gears.

$$I_{PG1X} = I_{PG2Y} = I_{PG3X} = I_{PG4Y} \tag{6.29}$$

$$I_{PG1Y} = I_{PG2X} = I_{PG3Y} = I_{PG4X} \tag{6.30}$$

$$I_{PG1Z} = I_{PG2Z} = I_{PG3Z} = I_{PG4Z} \tag{6.31}$$

The four propellers, as with the previous three different components, show a high degree of symmetry. Therefore the computation were done just on the front propeller and then transposed on the other ones. They have been modeled as flat cylinders with variable density. This shape has been chosen because it comes from a rotating (flat plate) propeller, furthermore the density depends on the radius and it decreases by getting far from the center. The flat cylinder of the propeller model has a radius R_P of 155×10^{-3} $[m]$, a height H_P of 2×10^{-3} $[m]$, a distance from the COM around the x-axis L_P of 235×10^{-3} $[m]$, a distance from the COM around the z-axis D_P of 20×10^{-3} $[m]$ and a mass M_P of 8.8×10^{-3} $[kg]$. Figure 6.12 shows the geometry of the propeller.

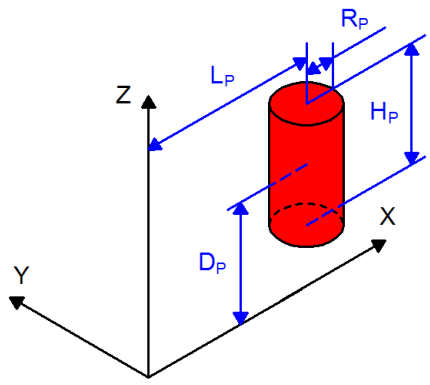


Figure 6.12: *Propeller*

Its moments of inertia around the x-axis I_{P1X} $[N\ m\ s^2]$, y-axis I_{P1Y} $[N\ m\ s^2]$ and z-axis I_{P1Z} $[N\ m\ s^2]$ are defined in equations (6.32), (6.33) and (6.34).

$$I_{P1X} = M_P \left(\frac{{R_P}^2}{6} + \frac{{H_P}^2}{12} + {D_P}^2 \right) = 38.8 \times 10^{-6}\ [N\ m\ s^2] \tag{6.32}$$

$$I_{P1Y} = M_P \left(\frac{{R_P}^2}{6} + \frac{{H_P}^2}{12} + {L_P}^2 + {D_P}^2 \right) = 525 \times 10^{-3}\ [N\ m\ s^2] \tag{6.33}$$

$$I_{P1Z} = M_P \left(\frac{{R_P}^2}{3} + {L_P}^2 \right) = 556 \times 10^{-3}\ [N\ m\ s^2] \tag{6.34}$$

According to the geometry, equations (6.35), (6.36) and (6.37) show the derivation of the three moments of inertia of the remaining three propellers.

$$I_{P1X} = I_{P2Y} = I_{P3X} = I_{P4Y} \tag{6.35}$$

$$I_{P1Y} = I_{P2X} = I_{P3Y} = I_{P4X} \tag{6.36}$$

$$I_{P1Z} = I_{P2Z} = I_{P3Z} = I_{P4Z} \tag{6.37}$$

At last it is possible to compute the three body moment of inertia I_{XX}, I_{YY} and I_{ZZ} by adding all the components around a defined axis.

$$\begin{aligned} I_{XX} = {} & I_{CX} + I_{EX} + I_{M1X} + I_{M2X} + I_{M3X} + I_{M4X} + I_{MG1X} + I_{MG2X} + \\ & + I_{MG3X} + I_{MG4X} + I_{PG1X} + I_{PG2X} + I_{PG3X} + I_{PG4X} + I_{P1X} + \\ & + I_{P2X} + I_{P3X} + I_{P4X} = 8.1 \times 10^{-3} \ [N \ m \ s^2] \end{aligned} \tag{6.38}$$

$$\begin{aligned} I_{YY} = {} & I_{CY} + I_{EX} + I_{M1Y} + I_{M2Y} + I_{M3Y} + I_{M4Y} + I_{MG1X} + I_{MG2X} + \\ & + I_{MG3X} + I_{MG4X} + I_{PG1X} + I_{PG2X} + I_{PG3X} + I_{PG4X} + I_{P1X} + \\ & + I_{P2X} + I_{P3X} + I_{P4X} = 8.1 \times 10^{-3} \ [N \ m \ s^2] \end{aligned} \tag{6.39}$$

$$\begin{aligned} I_{ZZ} = {} & I_{CZ} + I_{EX} + I_{M1Z} + I_{M2Z} + I_{M3Z} + I_{M4Z} + I_{MG1X} + I_{MG2X} + \\ & + I_{MG3X} + I_{MG4X} + I_{PG1X} + I_{PG2X} + I_{PG3X} + I_{PG4X} + I_{P1X} + \\ & + I_{P2X} + I_{P3X} + I_{P4X} = 14.2 \times 10^{-3} \ [N \ m \ s^2] \end{aligned} \tag{6.40}$$

From equations (6.38), (6.39) and (6.40) it shows that the moments of inertia around x and y axes has the same value. It means that the helicopter has a high degree of symmetry with respect to these axis. On the other hand, the moments of inertia around z is almost twice the other two. It follows that is easier to change the angular speed on x or y axes rather than change it on the z axis.

6.3 Geometry and aerodynamics derivation

This section gives a brief overview on aerodynamics and shows how to calculate the thrust b $[N\ s^2]$ and the drag d $[N\ m\ s^2]$ factors generated from the propellers motion using a combination of momentum and blade element theory (see appendix C).

The propellers' rotation creates aerodynamic and mechanical contributions to the structure [41]:

- Thrust is the aerodynamic force which is produced in the direction of the propellers. It is required to overcome the drag and weight forces, thus it sustains the forward flight of the vehicle.
- Drag torque is the results of the aerodynamic forces acting on the blade elements around the rotor shaft.

Figure 6.13 shows how these components act on a propeller. T_{BET} $[N]$ and Q_{BET} $[N\ m]$ are the thrust and the drag torque according to the blade element theory. ω_P $[rad\ s^{-1}]$ is the angular speed of the propeller.

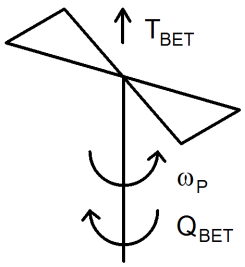


Figure 6.13: *Aero thrust torque*

The thrust factor b can be estimated with different approaches, thus the subscript of b identifies the method used. At the end of this chapter a mean of these components will be done to get the average value of the thrust factor.

To estimate b with blade element theory (b_{BET} $[N\ s^2]$), is necessary to describe the thrust T_{BET} in function of ω_P. Equation (6.41) shows this relation.

$$T_{BET} = N_B\ \rho_A\ a\ c\ {\omega_P}^2\ {R_P}^3\ \left(\frac{\theta_{I_0}}{6} - \frac{\theta_{I_{tw}}}{8} - \frac{\lambda}{4}\right) \tag{6.41}$$

To make the chapter more readable, the symbols of the previous (and following) equations are not reported in this section, but defined in the Aerodynamics

calculation appendix C.2.

Since the thrust T_{BET} is proportional to the square of the angular speed of the blade ω_P, it is possible to determine the ratio of T_{BET} to ${\omega_P}^2$, which is the definition of the thrust factor. The numeric result is provided by equation (6.42).

$$b_{BET} = \frac{T_{BET}}{{\omega_P}^2} = N_B\ \rho_A\ a\ c\ {R_P}^3 \left(\frac{\theta_{I_0}}{6} - \frac{\theta_{I_{tw}}}{8} - \frac{\lambda}{4}\right) = \\ = 53.8 \times 10^{-6}\ [N\ s^2] \tag{6.42}$$

The value of b can be also be evaluated by applying the momentum theory (C.1). Hence, the thrust factor with momentum theory b_{MT} $[N\ s^2]$ in hovering is simply the ratio of the weight carried from one propeller W_P to the squared propeller speed ${\omega_H}^2$ as shown in equation (6.43).

$$b_{MT} = \frac{W_P}{{\omega_H}^2} = 53.1 \times 10^{-6}\ [N\ s^2] \tag{6.43}$$

It can be verified that the two thrust factors evaluated in equations (6.42) and (6.43) are very similar. Furthermore, another estimation of b through an experimental setup is reported in the last section of this chapter (6.5).

To estimate the drag factor d, is necessary to describe the torque acting on the shaft Q_{BET} in function of the angular speed ω_P. Equation (6.44) shows this relation.

$$Q_{BET} = N_B\ \rho_A\ c\ {\omega_P}^2\ {R_P}^4 \left(\frac{C_D}{8} + a\ \lambda \left(\frac{\theta_{I_0}}{6} - \frac{\theta_{I_{tw}}}{8} - \frac{\lambda}{4}\right)\right) \tag{6.44}$$

As for the thrust T_{BET}, even the torque Q_{BET} is proportional to the square of the angular speed of the blade ω_P, then it is possible to determine the ratio of Q_{BET} to ${\omega_P}^2$, which is the definition of the drag factor d. The numeric result is provided by equation (6.45).

$$d = \frac{Q_{BET}}{{\omega_P}^2} = N_B\ \rho_A\ c\ {R_P}^4 \left(\frac{C_D}{8} + a\ \lambda \left(\frac{\theta_{I_0}}{6} - \frac{\theta_{I_{tw}}}{8} - \frac{\lambda}{4}\right)\right) = \\ = 1.1 \times 10^{-6}\ [N\ m\ s^2] \tag{6.45}$$

Unlike the b parameter, the drag coefficient d was estimated just with blade element theory, hence, no subscript was applied.

6.4 Datasheet data extrapolation

This section uses the datasheet information and identifies a few motor quantities. The motor model is show in chapter 3. Its parameters are:

- R [Ω] Motor resistance.
- L [H] Motor inductance.
- K_E [$V\ s\ rad^{-1}$] Electric motor constant.

The value of the inductance L was measured with the RLC meter introduced in section 6.1. The motor inductance has not a key role in the model, therefore its value was neglected and it was measured in just one way. Contrariwise the constants R and K_E are very important and several estimations were done to reach more reliable values. In this section, the datasheet data extrapolation method was applied. The information provided by the datasheet is the following:

- v_{RT} [V] Motor rated voltage: Tension used in the datasheet experiments.
- ω_{NL} [$rad\ s^{-1}$] Motor no load speed: motor velocity at no load torque.
- T_{NL} [$N\ m$] Motor no load torque: applied torque for the previous test.
- ω_{RL} [$rad\ s^{-1}$] Motor rated load speed: motor velocity at rated load torque.
- T_{RL} [$N\ m$] Motor rated load torque: applied torque for the previous test.

It now follows the derivation of the motor resistance R_{DAT} [Ω] and the electric motor constant $K_{E_{DAT}}$ [$V\ s\ rad^{-1}$] with the datasheet data extrapolation. Equation (6.46) shows the power balance in the motor, where the electric power is equal to the sum of the mechanic power and the losses.

$$\begin{aligned} v\ i &= e\ i + v_R\ i \\ v\ i &= K_E\ \omega_M\ i + R\ i^2 \\ v\ \frac{T_M}{K_M} &= K_E\ \omega_M\ \frac{T_M}{K_M} + R\left(\frac{T_M}{K_M}\right)^2 \\ \omega_M &= \frac{v}{K_E} - \frac{R}{{K_E}^2}\ T_M \end{aligned} \tag{6.46}$$

In the last step, K_M were replaced by K_E (numerically equal) to get a clearer representation.

According to equation (6.46) is easy to see that the speed ω_M varies linearly with the torque T_M. Two points of this characteristic are provided by the datasheet. Figure 6.14 shows the $\omega_M - T_M$ relation.

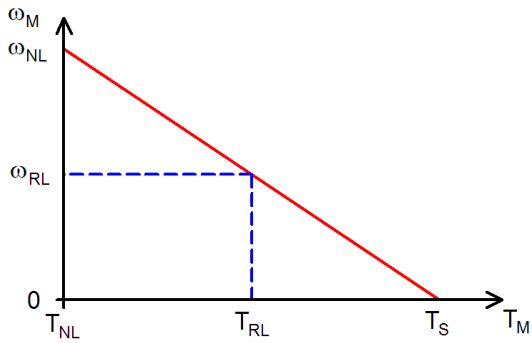


Figure 6.14: $\omega_M - T_M$ *function*

In the previous figure T_S $[N\ m]$ is called stall torque and identifies the maximum torque supplied by the motor. It is not directly provided by the datasheet, but it can be easily calculated from the other data. This parameter is useful to decide whether the motor is suitable or not for its load target.

Equation (6.46) can be rewritten according to the graphic representation of the characteristic $\omega_M - T_M$ as shown in figure 6.14. In equations (6.47), (6.48) and (6.49) the speed constant term ω_{M_0} $[rad\ s^{-1}]$ and the speed linear coefficient $\Delta\omega_M$ $[rad\ s^{-1}\ N^{-1}\ m^{-1}]$ were added.

$$\omega_{M_0} = \frac{v}{K_E} = \omega_{NL} = 2.07 \times 10^3\ [rad\ s^{-1}] \tag{6.47}$$

$$\Delta\omega_M = -\frac{R}{{K_E}^2} = \frac{\omega_{RL} - \omega_{NL}}{T_{RL} - T_{NL}} = -49 \times 10^3\ [rad\ s^{-1}\ N^{-1}\ m^{-1}] \tag{6.48}$$

$$\omega_M = \omega_{M_0} + \Delta\omega_M\ T_M = 2.07 \times 10^3 - 49 \times 10^3\ T_M\ [rad\ s^{-1}] \tag{6.49}$$

From the previous three equations, the constants T_S, $K_{E_{DAT}}$ and R_{DAT} can be derived according to equations (6.50), (6.51) and (6.52).

$$T_S = -\frac{\omega_{M_0}}{\Delta\omega_M} = 42.2 \times 10^{-3}\ [N\ m] \tag{6.50}$$

$$K_{E_{DAT}} = \frac{v_{RT}}{\omega_{M_0}} = 4.6 \times 10^{-3}\ [V\ s\ rad^{-1}] \tag{6.51}$$

$$R_{DAT} = -\Delta\omega_M \left(\frac{v_{RT}}{\omega_{M_0}}\right)^2 = 1.05\ [\Omega] \tag{6.52}$$

These last two quantities have the subscript DAT which identifies that their values come out from the datasheet data extrapolation. Further approaches were used, thus at the end of this chapter a mean of these components will be done to get the average value of K_E and R.

6.5 Experimental data extrapolation

This section describes the set-up used in this work to carry out several tests. It was possible to plot the collected data for estimating the relation between variables and for extrapolating a few constants.

To identify the characteristic of the motors, it was chosen to do the tests on just one motor to avoid losses of time and stress on the other motors. With this setup, the voltage supply and the codes (representing the duty cycle of the square waveform given to the motor) were given as inputs. In addition to these two variables, it was possible to measure the current through the motor, its speed and the lift generated.

The voltage supply V_{DD} $[V]$ was determined by a voltage supply generator equipped with an internal display. The code of the PWM was provided by a PC interfaced with the μC (through UART interface). In its turn, the μC was connected with the motor driver and varied the duty cycle of the waveform supplied to the motor. The current i $[A]$ was provided by reading the voltage on a motor series low resistor with a multimeter. A counter proof was done by sampling the same value with an ADC and reading the averaged data with the PC. Figure 6.15 shows the system connections between the PC and the motor.

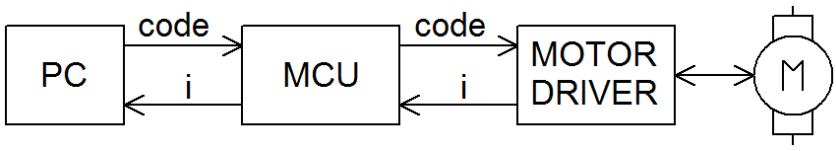


Figure 6.15: *PC - Motor connections*

The propeller speed ω_P $[rad\ s^{-1}]$ were measured with an oscilloscope connected to an optic board. This circuit drived an IR LED which lighted a phototransistor. In the space between the two optical devices the propeller blades were free to rotate. When a blade darkened the phototransistor (from the IR LED light), the voltage of the waveform on the oscilloscope decreased. With the rotation of the propeller, it was thus possible to determine the speed ω_P by calculating the frequency of the waveform on the oscilloscope f_{OLC} $[blade\ s^{-1}]$. Figure 6.15 shows the system for the rotating speed computation.

f_{OLC} must be multiplied by π to get the value in the units of measurement

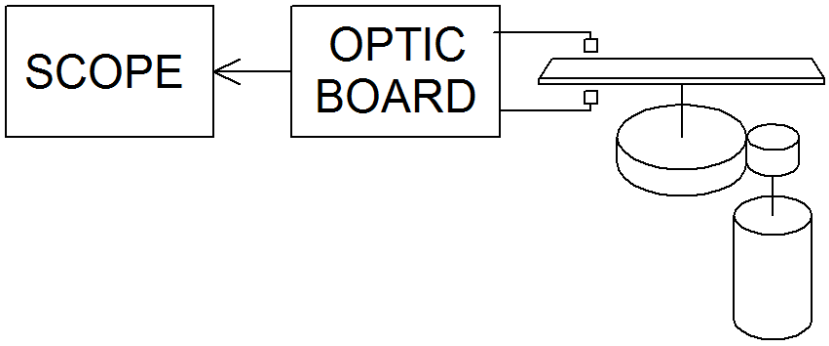


Figure 6.16: *Rotating speed computation*

$[rad\ s^{-1}]$, as shown in equation (6.53).

$$\omega_P \left[\frac{rad}{s}\right] = f_{OLC} \left[\frac{blade}{s}\right] \frac{1}{2} \left[\frac{turn}{blade}\right] 2\pi \left[\frac{rad}{turn}\right] = f_{OLC}\ \pi \left[\frac{rad}{s}\right] \tag{6.53}$$

According to the information needed, it is possible to consider the propeller ω_P or the motor ω_M speeds related by equation (6.54).

$$\omega_M = N\ \omega_P = 5.6\ \omega_P \tag{6.54}$$

In the following data tables, the considered speed will be that of the motor (ω_M), if not otherwise stated.

To estimate the lift T_N $[N]$, the electronic balance described in section 6.1 was used. A rigid structure with one degree of freedom was built. A hinge (fixed on a table) was connected to one end of a horizontal light stick. Its other end was tied to the motor. Therefore, the propeller motion could produce a vertical force strong enough to lift the motor system (actually, it was a rotation around the hinge). An additional weight was added to the motor to avoid the structure being lifted by the vertical force of the propeller. By placing the motor system on the electronic balance, it was possible to measure the propeller lift through differential reading. Figure 6.17 shows the set-up.

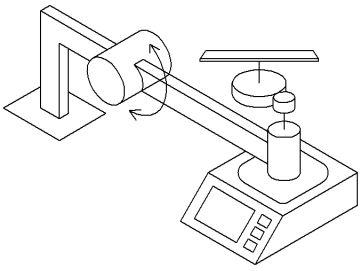


Figure 6.17: *Lift set-up*

The mass lifted by the propeller T_G $[g]$ was equal to the difference between the balance value at zero motor speed and the balance value at the desired speed. In this way the weight tare is subtracted. In the following data tables the lift is expressed in grams $[g]$ (not in kilograms $[kg]$) and in Newtons $[N]$. To obtain the lift information in Newtons (T_N), the grams value T_G need to be multiplied by 1000 and by the acceleration due to gravity g, according to equation (6.55).

$$T_N\,[N] = T_G\,[g]\;\frac{1}{1000}\left[\frac{kg}{g}\right]\;g\left[\frac{m}{s^2}\right] = T_G\;9810\,[N] \tag{6.55}$$

It now follows a list of tables with the acquired datasets.

The first test was carried out by varying the code, maintaining the supply voltage (V_{DD}) equal to 11 $[V]$. Table 6.2 shows the measured data.

Code	**Voltage**	**Lift**	**Lift**	**Speed**
$[-]$	v $[V]$	T_G $[g]$	T_N $[N]$	ω_M $[rad\ s^{-1}]$
96	4.14	60.2	0.59	628
104	4.49	69.0	0.68	679
112	4.83	82.3	0.81	727
120	5.18	94.5	0.93	771
128	5.52	105	1.03	815
136	5.87	116	1.14	862
144	6.21	129	1.27	904
152	6.56	140	1.37	945
160	6.90	151	1.48	985

Table 6.2: identification test 1

The second test was carried out by varying the code, maintaining the supply voltage (V_{DD}) equal to 10.5 $[V]$. Table 6.3 shows the measured data.

Code	**Voltage**	**Current**	**Lift**	**Lift**	**Speed**
$[-]$	v $[V]$	i $[A]$	T_G $[g]$	T_N $[N]$	ω_M $[rad\ s^{-1}]$
104	4.28	0.77	61.1	0.60	619
112	4.61	0.84	70.3	0.69	651
120	4.94	0.91	82.0	0.80	704
128	5.27	1.00	92.7	0.91	748
136	5.60	1.10	102.7	1.01	792
144	5.93	1.21	114.6	1.12	836
152	6.26	1.32	123.5	1.21	862
160	6.59	1.46	135.2	1.33	903
168	6.92	1.62	145.9	1.43	932
176	7.25	1.83	155.0	1.52	976
184	7.58	2.04	164.4	1.61	992
192	7.91	2.25	180.4	1.80	1035

Table 6.3: identification test 2

The third test was carried out by varying the code, maintaining the supply voltage (V_{DD}) equal to 11 $[V]$. Table 6.4 shows the measured data.

Code	**Voltage**	**Current**	**Lift**	**Lift**	**Speed**
$[-]$	v $[V]$	i $[A]$	T_G $[g]$	T_N $[N]$	ω_M $[rad\ s^{-1}]$
192	8.28	2.38	205	2.01	1082
200	8.63	2.64	220	2.16	1117
208	8.97	2.85	233	2.28	1152
216	9.32	3.15	245	2.40	1184

Table 6.4: identification test 3

The fourth test was carried out by varying the supply voltage (V_{DD}), maintaining the code equal to 80. Table 6.5 shows the measured data.

V_{DD}	**Voltage**	**Current**	**Speed**
V_{DD} $[V]$	v $[V]$	i $[A]$	ω_M $[rad\ s^{-1}]$
9	2.8235	0.2600	409.9150
9.5	2.9804	0.3400	439.8230
10	3.1373	0.4700	467.9716
10.5	3.2941	0.6500	478.5274
11	3.4510	0.8700	510.1946
11.5	3.6078	1.1300	533.0654
12	3.7647	1.4500	559.4548
12.5	3.9216	1.7400	577.0477
13	4.0784	2.0800	606.9557

Table 6.5: identification test 4

The fifth test was carried out by varying the code, maintaining the supply voltage (V_{DD}) equal to 11.1 $[V]$. Table 6.6 shows the measured data.

Code	**Voltage**	**Speed**
$[-]$	v $[V]$	ω_M $[rad\ s^{-1}]$
32	1.39	185
40	1.74	238
48	2.09	311
56	2.44	352
64	2.79	417
72	3.13	464
80	3.48	521
88	3.83	575
96	4.18	628
104	4.53	679
112	4.88	727
120	5.22	771
128	5.57	815
136	5.92	862
144	6.27	904
152	6.62	945
160	6.96	985

Table 6.6: identification test 5

The sixth test was carried out by varying the code, maintaining the supply voltage (V_{DD}) equal to 11 $[V]$. Table 6.7 shows the measured data.

Code	**Voltage**	**Current**	**Speed**
$[-]$	v $[V]$	i $[A]$	ω_M $[rad\ s^{-1}]$
128	5.52	0.72	792
160	6.9	1.22	968
192	8.28	2	1126
224	9.66	2.9	1249

Table 6.7: identification test 6

The seventh test was carried out by varying the code, maintaining the supply voltage (V_{DD}) equal to 11 $[V]$. In addition, the propeller speed was mechanically fixed to zero. Table 6.8 shows the measured data.

Code	**Voltage**	**Current**
$[-]$	v $[V]$	i $[A]$
96	4.14	1.5
128	5.52	2.9
160	6.9	4.5

Table 6.8: identification test 7

It can be noted that almost all the experiments were done by varying the code. Just in the fourth test, a voltage supply (V_{DD} $[V]$) swing was applied. Furthermore, the lift information was collected in the first three tests. The voltage information (v $[V]$) is directly calculated from the code and the voltage supply as shown in equation (6.56).

$$v = V_{DD}\ \frac{code}{255} \tag{6.56}$$

In the last test, the maximum current with zero motor speed was investigated. It can be noted how high the current becomes by increasing the voltage.

With the previous tables, it was possible to create graphs to visualize the trend of some variables respect to other ones.

To estimate the thrust factor b, the plot of the lift T_N over the squared propeller speed ${\omega_P}^2$ can be investigated. Since a graphic method was used, the subscript GRP was added (to distinguish it from the other methods). The dataset used, was collected from all the previous tables containing T_N and ω_M. Figure 6.18 shows the graph.

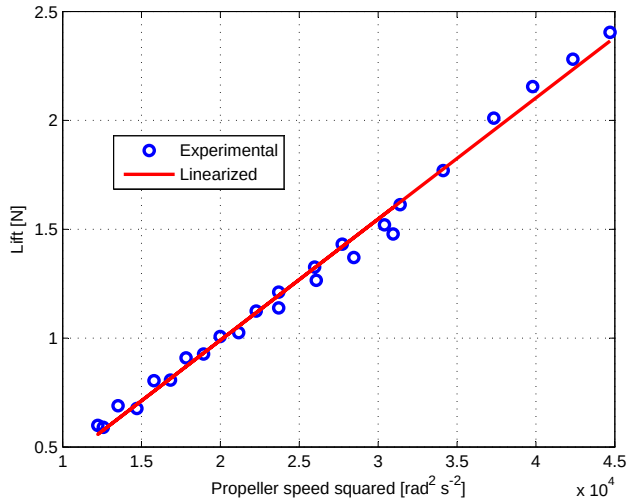


Figure 6.18: *Lift vs speed squared*

It can be seen that figure 6.18 shows both the "raw" experimental data and the linearized ones. This last track was derived from linear regression (see appendix B) of the previous raw data. The method provide a linear approximation between the lift and the propeller speed squared according to equation (6.57).

$$T_N = -0.122 + 55.6 \times 10^{-6}\ {\omega_P}^2 \tag{6.57}$$

From equation (6.57), it follows the derivation of the thrust b_{GRP}.

$$b_{GRP} = \frac{\partial\ T_N}{\partial\ {\omega_P}^2} = 55.6 \times 10^{-6}\ [N\ s^2] \tag{6.58}$$

Another interesting plot is the relation between the propeller speed ω_P and the voltage v. It can be seen from figure 6.19 and equation (6.59) that the propeller speed squared goes linearly with the voltage. Actually, the values of the voltage under 4 [V] follow another trend. However this behavior is far from the working point, hence neglected.

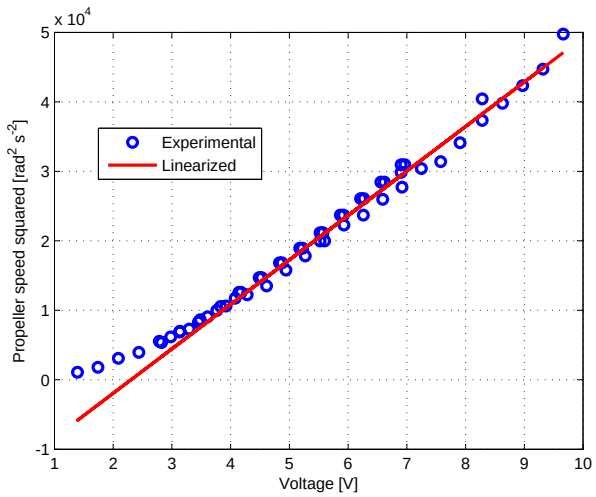


Figure 6.19: *Speed squared vs voltage*

$$\omega_P{}^2 = -14.7 \times 10^3 + 6.4 \times 10^3 \ V \tag{6.59}$$

The plot of the lift respect to the voltage is useful because it is linear (as first approximation) and can be easily implemented in the control loop of the microcontroller. Figure 6.20 and equation (6.60) show this relation.

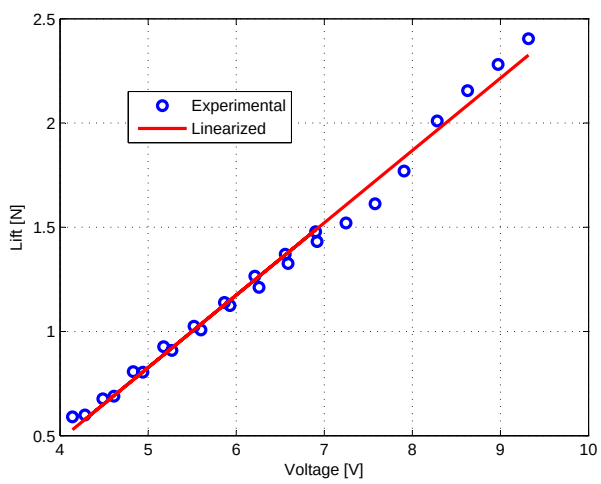


Figure 6.20: *Lift vs voltage*

$$T_N = -0.91 + 0.35 \ V \tag{6.60}$$

To estimate the electric motor resistance R and constant K_E, the linear regression was used one more time. It is a good method because it is easy to evaluate the quality of the approximation.

The motor equation is reported in equation (3.33) and copied in (6.61) for simplicity.

$$v = R\ i + K_E\ \omega_M \tag{6.61}$$

The first method to get a linear trend is to divide both members of equation (6.61) by the motor speed ω_M according to equation (6.62).

$$\frac{v}{\omega_M} = K_E + R\ \frac{i}{\omega_M} \tag{6.62}$$

According to equation (6.62) it is easy to consider K_E as offset and R as linear coefficient. The independent variable becomes the ratio of the current to the motor speed $\frac{i}{\omega_M}$, while the dependent variable is composed of the ratio of the voltage to the motor speed $\frac{v}{\omega_M}$. Figure 6.21 shows the graph.

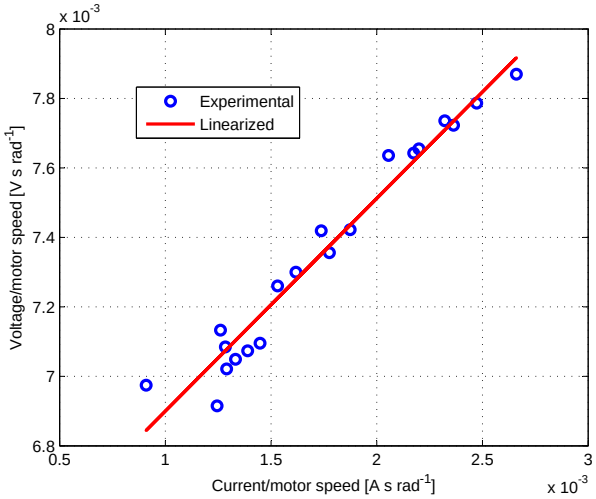


Figure 6.21: $\frac{v}{\omega_M}$ *vs* $\frac{i}{\omega_M}$

Equation (6.63) shows the linear regression approximation.

$$\frac{v}{\omega_M} = 6.3 \times 10^{-3} + 0.61\ \frac{i}{\omega_M} \tag{6.63}$$

From equation (6.63), it follows the derivation of the resistance R_{LR1} [Ω] and electric constant $K_{E_{LR1}}$ [$V\ s\ rad^{-1}$]. The subscript $LR1$ characterize the first

acquisition with the linear regression method.

$$R_{LR1} = 0.61\ [\Omega] \tag{6.64}$$

$$K_{E_{LR1}} = 6.3 \times 10^{-3}\ [V\ s\ rad^{-1}] \tag{6.65}$$

The second method to get a linear trend is to divide both members of equation (6.61) by the current i according to equation (6.66).

$$\frac{v}{i} = R + K_E\ \frac{\omega_M}{i} \tag{6.66}$$

According to equation (6.66) it is easy to consider R as offset and K_E as linear coefficient. The independent variable becomes the ratio of the motor speed to the current $\frac{\omega_M}{i}$, while the dependent variable is composed of the ratio of the voltage to the current $\frac{v}{i}$. Figure 6.22 shows the graph.

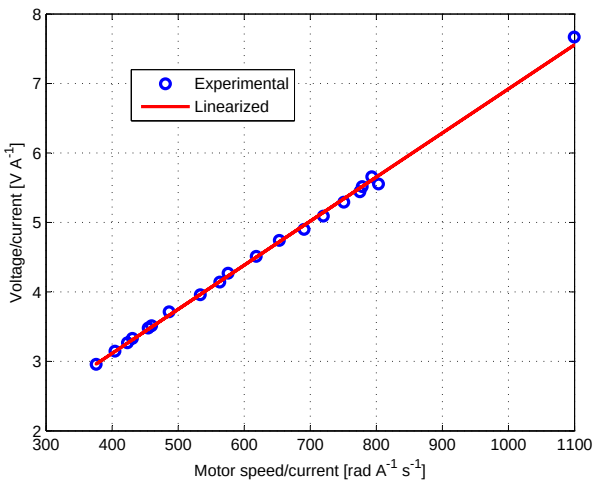


Figure 6.22: $\frac{v}{i}$ *vs* $\frac{\omega_M}{i}$

Equation (6.67) shows the linear regression approximation.

$$\frac{v}{i} = 0.58 + 6.3 \times 10^{-3}\ \frac{\omega_M}{i} \tag{6.67}$$

From equation (6.67), it follows the derivation of the resistance R_{LR2} [Ω] and electric constant $K_{E_{LR2}}$ $[V\ s\ rad^{-1}]$. The subscript $LR2$ characterize the second acquisition with the linear regression method.

$$R_{LR2} = 0.58\ [\Omega] \tag{6.68}$$

$$K_{E_{LR2}} = 6.3 \times 10^{-3}\ [V\ s\ rad^{-1}] \tag{6.69}$$

Another interesting method to estimate the electric motor resistance R and constant K_E is to calculate the RMS relative error of the voltage for several values of R and K_E. The variables which better fit the function are those which provide the lowest RMS relative error.

The motor voltage can be evaluated from the experimental dataset (v_{EXP} $[V]$), or estimated from other values (v_{EST} $[V]$). Equation (6.70) shows how to calculate the estimated voltage.

$$v_{EST} = R\ i + K_E\ \omega_M \tag{6.70}$$

Where both the current i and the motor speed ω_M are taken from the experimental dataset. The definition of the relative error ϵ_k $[-]$ is given according to equation (6.71).

$$\epsilon_k = \frac{v_{EXP} - v_{EST}}{v_{EXP}} \tag{6.71}$$

The subscript k $[-]$ is a variable which goes from 1 to n $[-]$, where n is the number of data acquired. For each couple of values of R and K_E, the RMS relative error ϵ_{RMS} $[-]$ is calculated over all the n experiment points. Equation (6.72) shows the RMS relative error in function of the relative error ϵ_k.

$$\epsilon_{RMS} = \sqrt{\frac{1}{n}\sum_{k=1}^{n} {\epsilon_k}^2} \tag{6.72}$$

By choosing a reasonable range of R and K_E, it is easy to plot the the RMS relative error matrix and verify that it has a minimum. Figure 6.23 shows the 3D plot of the RMS relative error matrix.

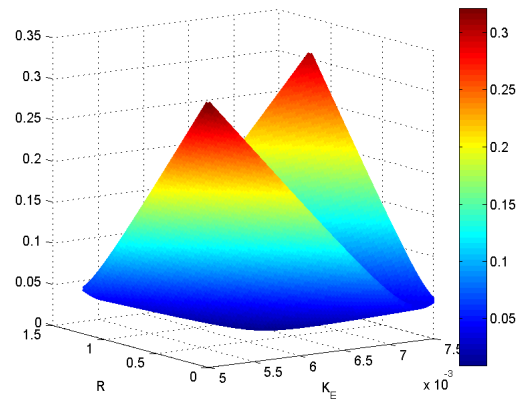


Figure 6.23: *3D plot of the RMS relative error matrix in function of R and Ke*

Another way to represent the information of figure 6.23 is to plot the isohypse in a 2D graph. This means that it is possible to draw a series of curves which represent points with the same RMS relative error value. Figure 6.24 shows the 2D isohypse graph.

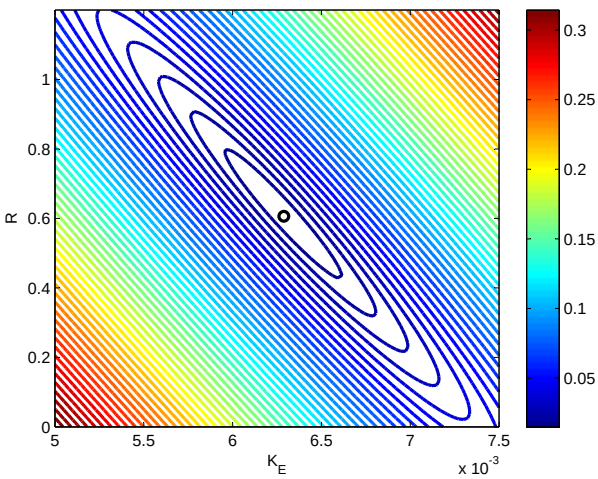


Figure 6.24: *2D plot of the RMS relative error isohypse in function of R and Ke*

From figure 6.24, it follows the derivation of the resistance R_{RMS} [Ω] and electric constant $K_{E_{RMS}}$ [$V\ s\ rad^{-1}$]. The subscript RMS characterize RMS relative error minimization method.

$$R_{RMS} = 0.61\ [\Omega] \tag{6.73}$$

$$K_{E_{RMS}} = 6.3 \times 10^{-3}\ [V\ s\ rad^{-1}] \tag{6.74}$$

6.6 Constants comparison

This section compares the constants found with more than one of the previous methods and defines their final values. Table 6.9 shows the constants summary.

Measure	b	$[N\ s^2]$	K_E	$[V\ s\ rad^{-1}]$	R	$[\Omega]$
Basic					R_{BSC}	1.1
Aerodynamics	b_{BET}	53.8×10^{-6}				
	b_{MT}	53.1×10^{-6}				
Datasheet			$K_{E_{DAT}}$	4.6×10^{-3}	R_{DAT}	1.05
Experimental	b_{GRP}	55.6×10^{-6}				
			$K_{E_{LR1}}$	6.3×10^{-3}	R_{LR1}	0.61
			$K_{E_{LR2}}$	6.3×10^{-3}	R_{LR2}	0.58
			$K_{E_{RMS}}$	6.3×10^{-3}	R_{RMS}	0.61

Table 6.9: Constants comparison

To estimate the value of the thrust factor b, three therms can be considered. Since all these variables take into account some experimental consideration, their values are very similar ($\pm 3\%$). The final thrust factor is calculated as the average of all the three components: $b = 54.2 \times 10^{-6}\ [N\ s^2]$.

The estimation of the motor electric constant K_E takes into account four therms. The datasheet value differ a lot from the other ones (-27%). However, this situation is totally comprehensible because the datasheet data have high tollerances to describe a wide group of coponents. Therefore, the final motor electric constant is calculated just as the average of the three experimental components: $K_E = 6.3 \times 10^{-3}\ [V\ s\ rad^{-1}]$.

Finally, it is possible to estimate the motor resistance R by considering five therms. As in the previous case, the datasheet value is quite far from the other for the same reasons as before. Furthermore, the value taken with the RLC meter is very accurate, but it defines another point of work (where the voltage is almost zero). Therefore, the final motor resistance is calculated, once again, just as the average of the three experimental components: $R = 0.6\ [\Omega]$.

Chapter 7

Sensors: SONAR, IR & IMU

In this chapter, the main quadrotor's sensors are presented. They are fundamental to identify the robot's attitude and its height from the ground. Furthermore, thanks also to a network of distance sensors, it is possible to accomplish obstacle avoidance and trajectory planning on a high level controller.

The first section (7.1: SONAR module) gives an overview of the SONAR module. Particular attention is given to the performance, the interface with the microcontroller (through I^2C) and the communication protocol.

The second section (7.2: IR module) gives an overview of the IR module. It focuses on the performance, the hardware interface with the microcontroller (through ADC) and the interpolation of the module characteristic for the distance estimation.

The third section (7.3: IMU) gives an overview of the IMU. Particular attention is given to its inner sensors and its performance. Furthermore the interface with the microcontroller (through UART) and its communication frame are presented.

7.1 SONAR module

SONAR is the acronym of SOund Navigation And Ranging, which identifies a sensor capable of measuring a distance through ultrasound waves. There are a lot of different types of SONAR systems according to the desired characteristics. In this work the model SRF08 (from Devantech) has been adopted. It is composed of two ultrasound capsules, a light sensor, a PIC microcontroller and other needed circuitry.

The two capsules are transducers which convert ultrasonic waves into electrical signals and vice versa. One capsule is used for the wave transmission while the other one for the wave reception. An on-board light sensor is included in the device to get the additional light information. The microcontroller is the "brain" of the module. It allows the communication with other digital devices (through I^2C) and provides the right signal conditioning to and from the ultrasonic capsules and the light sensor. Furthermore there is other circuitry which is needed to correctly interface the microcontroller and the transducers. A picture of the Sonar module is shown in figure 7.1.

Figure 7.1: *SRF08 SONAR photo*

The SONAR (combined with an IR module) is used in this project to estimate the height of the quadrotor from the ground. Therefore it is mounted on the bottom of the robot, pointing downwards (it has the same direction as the z_B axis in the body-fixed frame). Generally this direction is not the same as z_E (measured in the earth inertial frame) because it depends on the quadrotor attitude. To calculate the earth inertial frame distance (Z), the SONAR data must be multiplied by the cosine of the roll angle (ϕ) and by the cosine of the pitch angle (θ). These two angles are measured with the IMU (presented in section 7.3).

The features of the SRF08 SONAR module are [42]:

- Digital data output through I^2C interface.
  Since the communication between SONAR and MCU is digital, all the raw signal processing is done directly in the module and not accessible from the extern. Hence it is not possible to add a measure error during the data transfer (this is not verified if the output is analogue). The main drawback of this solution is that the MCU must handle a complex communication, therefore it assigns dedicated hardware and adds delays in the process. The I^2C frame communication is explained at the end of this section.

- Range from 0.03 to 6 [m].
  The range of this sensor is quite wide. This characteristic makes it suitable for mobile robotic applications, where an indoor scenario with several obstacles is possible.

- Resolution of about 0.01 [m].
  The resolution of "just" one centimeter is ideal for the purpose of the height control. With a worse one it would have been difficult to stabilize the quadrotor on a certain reference because the sensor would have provided a rough feedback.

- Acquisition period of 70×10^{-3} [s] (a frequency of 14 [Hz]).
  One of the drawbacks of this module is its slowness. To allow a full range detection, the SONAR must wait long enough to acquire an echo from a 6 meters obstacle. Furthermore the module sends eight ultrasound impulses (ping) for each read cycle to achieve a good accuracy and avoid wrong detections. All these features are paid with delays.

- Beam angle of 55 [deg].
  This value is typical for SONAR modules. It comes from the physics of ultrasonic capsules. With this (wide) angle it is easy to detect obstacles even if they don't lie in the perfect direction of the sensor. On the contrary, it is not possible to estimate a certain straight distance if an obstacle lie in the proximity and reflects the wave as it was "in-line" with the sensor direction.

- Weight of 12 [g].
  The light weight makes this module suitable for unmanned aerial vehicle.

- Average power consumption of 50×10^{-3} [W] ($= 10 \times 10^{-3}$ [A] at 5 [V]).
  The power consumption of the module is considerably low. Therefore it does not affect the design of the power supply source, which instead has to deliver a lot of energy to the motors.

The I^2C frame communication is needed to allow the MCU-SONAR interface. The MCU act as a master and imposes the timing. It provides two main commands: "START RANGING", which makes the SONAR start the procedure to measure the distance, and "READ DATA", which reads the SONAR information once it has finished ranging. Figure 7.2 shows a flow chart of the signals in the frame communication. The labels with a red background specify the information flow from the MCU to the SONAR. Those with a green background shows instead a communication in the opposite direction (from SONAR to MCU).

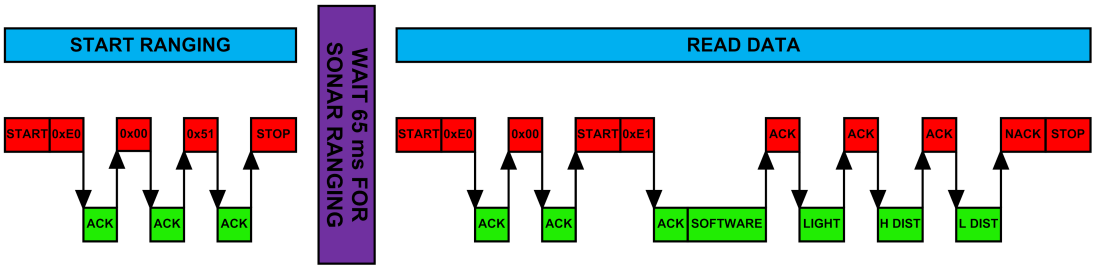


Figure 7.2: *I^2C communication*

"START" and "STOP" are hardware-detected signals which allow synchronization. "ACK" represents the acknowledge and it means that the data have been correctly received. On the contrary "NACK" is the negative acknowledge and it is used when an error occurs or when no more data are requested. The labels preceded by "0x" represent a hexadecimal byte. "0xE0" and "0xE1" are the SONAR address for read and for write operations respectively. "0x00" is the internal address of the command register. "0x51" is the command to start the SRF08 ranging with the output data directly in centimeters. The SONAR data are collected in 4 bytes: "SOFTWARE" represent the software revision, "LIGHT" returns the brightness acquired from the light sensor, "H DIST" and "L DIST" give respectively the high and the low bytes of the measured distance.

7.2 IR module

IR is the acronym of InfraRed, which identifies a light wave within a certain frequency range. These waves can't be seen by people because human eyes are not sensible in this interval (even though IR is quite close to the visible range). The IR module (or system) is a device capable of measuring a distance through IR waves. There are a lot of different types of these systems according to the desired characteristics and the technology used for the ranging (triangulation, phase shift, time of flight, ...). In this work the model GP2Y0A02YK (from Sharp) has been adopted. It is composed of one light emitter, one light detector, a signal processing circuit and other needed circuitry as shown in figure 7.3.

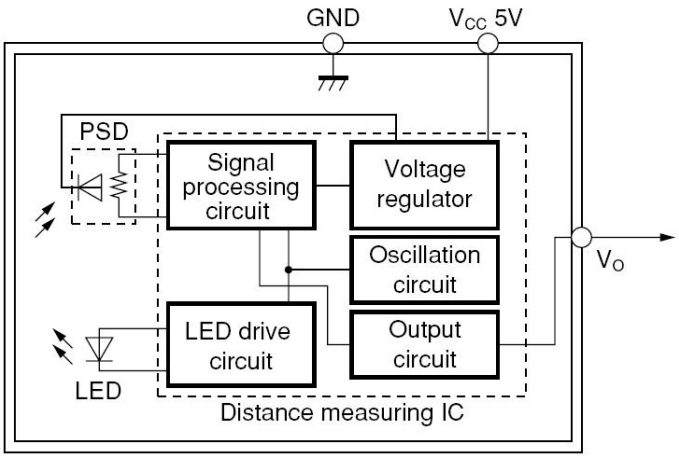


Figure 7.3: *IR Internal Block Diagram*

The IR Light-Emitting Diode (LED) is a transducer which converts electrical signals into IR waves. On the contrary the Position-Sensitive Device (PSD), composed of a photodiodes array, applies a physic conversion in the opposite way: from IR waves to electrical signals. The analyzed IR module adopts the triangulation strategy to compute the distance estimation. The IR LED sends a light beam. When it runs into an obstacle, most of the incident light is reflected according to the object reflectivity. Part of these reflected beam will be collected from a lens and focused in a small area of the PSD. Thanks to geometric consideration it is possible to calculate the distance according to the lighted photodiode (or photodiodes) in the PSD. Figure 7.4 shows the triangulation strategy.

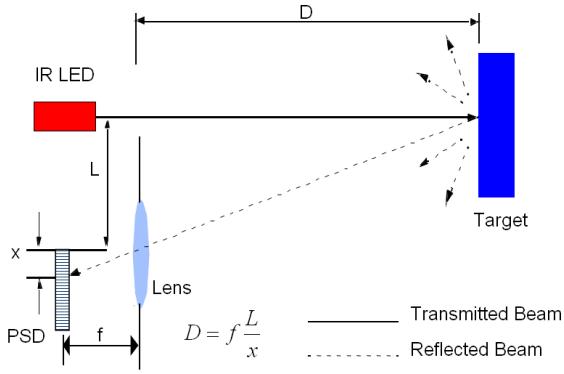


Figure 7.4: *triangulation strategy*

The signal processing circuit is the "brain" of the module. It creates the right signal conditioning to and from the optical devices and provides a certain analog output according to the measured distance. The oscillation circuit modulates on-off the IR waves. It is needed to reduce interferences between the ambient light and that one from the LED. Furthermore, modulated signals are much easier and more reliably detected by a demodulator. The voltage regulator is an additional circuit to stabilize the power supply in the device. Likewise, the output circuit is needed to provide a signal directly readable from a microcontroller (range of a few volts and low output impedance). The LED drive circuit is composed of other circuitry which is needed to correctly interface the signal processing circuit to the LED. A picture of the IR module is shown in figure 7.5.

Figure 7.5: *IR module photo*

The IR module (as well as the SONAR) is used in this project to estimate the height of the quadrotor from the ground. Therefore it is mounted close to the SONAR on the bottom of the robot, pointing downwards. To calculate the earth inertial frame distance (Z), the IR data must be multiplied by the cosine of the roll angle (ϕ) and by the cosine of the pitch angle (θ). These two angles are measured with the IMU (presented in section 7.3).

The features of the GP2Y0A02YK IR module are [43]:

- Analogue data output.
  The communication between the IR module and the MCU is analogue, therefore just an internal ADC block is used and no considerable delays are added in the process. The main drawback of this solution is that there is a measure error during the data conversion (this is not verified if the output is digital). Furthermore, as Sharp recommends, an RC low pass filter was added on the output to reduce the noise on the line. The values of the resistance and the capacitance are 22×10^3 [Ω] and 150×10^{-9} [F] respectively.

- Range from 0.2 to 1.5 [m].
  The range of this sensor is quite wide (even though the SONAR described before has a wider one). This characteristic makes it suitable for mobile robotic applications, where an indoor scenario with several obstacles is possible. A drawback of this module is that the minimum range is 0.2 [m]. Furthermore the distance measure does not saturate for values less than the minimum, but it provides ambiguous data. Therefore the SONAR must be used to determine whether he IR data is valid or not, in low range distances.

- Resolution depending on the distance.
  With a distance of 0.2 [m] the resolution is less than 0.01 [m].
  With a distance of 0.7 [m] the resolution is less than 0.02 [m].
  With a distance of 1.5 [m] the resolution is less than 0.05 [m].
  The resolution is good for short distances, but it get worse by increasing the measured height. However it is still acceptable even because both the IR module and the SONAR contribute in the distance estimation.

- Acquisition period of 40×10^{-3} [s] (a frequency of 25 [Hz]).
  This module is quite slow, but still faster than the SONAR. The reason of this long acquisition period is that the IR module computes a mean of 16 pulses for each distance value to reduce possible errors.

- Beam angle less of 5 [deg].
  This value depends on the LED optics. According to the performance there are IR modules with different angles of detection. Since the beam angle of the GP2Y0A02YK is quite narrow it is possible to estimate a certain straight distance even if an obstacle lie in the proximity (with a wider angle, the obstacle can corrupt the measure).

- Weight of 7 [g].
  The light weight makes this module suitable for unmanned aerial vehicle.

- Average power consumption of 150×10^{-3} [W] ($= 30 \times 10^{-3}$ [A] at 5 [V]).
  The power consumption of the IR module is considerably low (even if it is three times bigger than the SONAR one). Therefore it does not affect the design of the power supply source, which instead has to deliver a lot of energy to the motors.

Since the output voltage to distance characteristic is not linear, a 34 point linearization look-up table has been taken into account to describe in detail the curve. Furthermore, a linear interpolation was added to calculate the distance value on the linearized segment to achieve better accuracy. Figure 7.6 shows the IR module output characteristic.

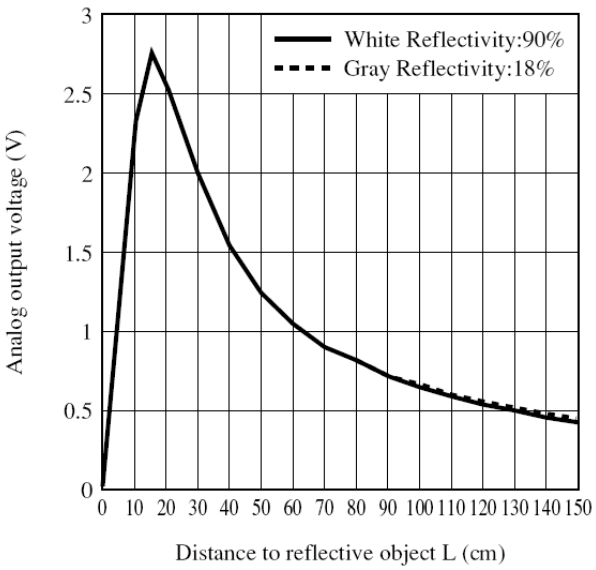


Figure 7.6: *Analog output voltage vs. distance to reflective object*

7.3 IMU

IMU is the acronym of Inertial Measurement Unit, which identifies a sensor capable of measuring the orientation (attitude) of a body through inertial sensors. In this work the device MTi-G (from Xsens) has been adopted. The core of the MTi-G is composed of an integrated GPS, an IMU and an attitude and heading reference system processor. The inertial low-power signal processor runs a real-time Xsens Kalman Filter (XKF) providing inertial enhanced position and velocity estimates. It also provides drift-free GPS enhanced 3D orientation estimates, as well as calibrated 3D acceleration, 3D rate of turn (rate gyro), 3D earth-magnetic field data and static pressure. Furthermore the device is equipped with a flash memory, a power supply circuit and a RS232 transceiver [44]. Figure 7.7 shows the architecture overview.

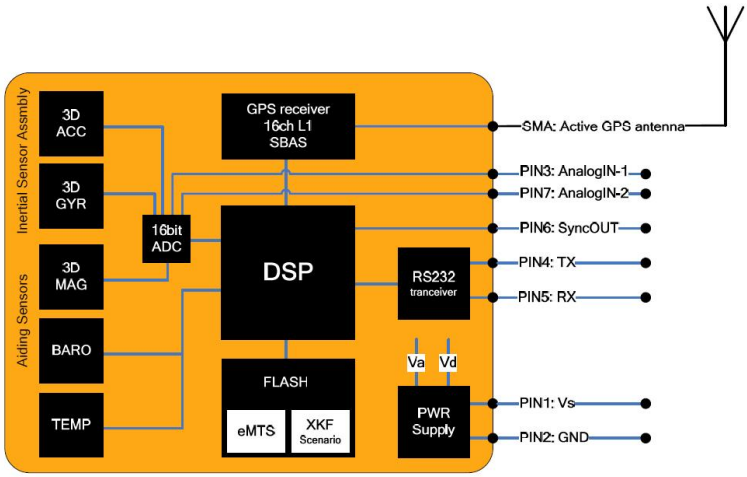


Figure 7.7: *IMU architecture*

The GPS receiver allows to determine the global position and the flight heading with great accuracy. Even though it is a powerful instrument, it was not used in this work for two reasons:

- The stabilization algorithm must work correctly even with no GPS data (this is the case of a normal indoor scenario).
- This project focused on the low level controller, while the GPS is one of the feedbacks of the high level controller.

However the GPS will be surely used in future works with this hardware (this is the reason for interfacing a device with both IMU and GPS).

The sensors mounted on the MTi-G are:

- Three orthogonal Micro Electro-Mechanical System (MEMS) accelerometers (used to measure the accelerations in the three fixed-body axes).
- Three orthogonal MEMS gyroscopes (used to measure the rate of turn with respect to the three body-axes).
- Three orthogonal thin magnetoresistive film magnetometers (used to measure the direction of the magnetic north).
- A MEMS barometer (used to measure the static pressure).
- Several temperature sensors (used to allow compensation for temperature dependency of the various sensors).

The accelerometers, gyroscopes and magnetometers are interfaced with the Digital Signal Processor (DSP) through a 16 bit Analog to Digital Converter (ADC). Furthermore the MTi-G is equipped with a non-volatile flash memory where the configuration options and the filter data are stored.

The DSP is the "brain" of the device. It processes the data from the flash memory, the GPS and the sensors introduced above. After the elaboration of all these inputs, the DSP provides the information about the body movements and position in the format specified in the configuration. These data are externally accessible through an RS232 transceiver.

Furthermore, the MTi-G has also a circuit which filters the power supply and provides the required analogue and digital references. A picture of the IMU is shown in figure 7.8.

Figure 7.8: *MTi-G IMU photo*

The IMU is used in this project to measure the attitude of the helicopter (since the GPS is not used, no further information is available). It is mounted on the top of the robot with its three main axes aligned with the body-fixed ones.

The features of the MTi-G IMU are [44]:

- Digital data output through serial interface.
  The communication between IMU and MCU is digital, therefore it is easy to read the information through a Universal Asynchronous Receiver Transmitter (UART). Since the data flow is one-way (from IMU to MCU), just two bits are used for the communication (in addition to the power supply). The UART frame communication is explained at the end of this section.

- Acceleration full scale of about $\pm$ 50 $[m\ s^{-2}]$.
  Rate of turn full scale of about $\pm$ 300 $[deg\ s^{-1}]$.
  Magnetic field full scale of about $\pm\ 75 \times 10^{-6}$ $[T]$.
  The range of this sensors is quite wide, therefore no saturation is verified.

- GPS position accuracy of about 2 [m].
  Attitude angular resolution of about 0.05 [deg].
  Static accuracy (roll/pitch) less than 0.5 [deg].
  Static accuracy (yaw) less than 1 [deg].
  Even though the angular resolution is quite good, the static accuracy is much more restrictive. The GPS is not used in this work.

- Dynamic accuracy (roll/pitch) of about 1 [deg] RMS.
  Dynamic accuracy (yaw) of about 3 [deg] RMS.
  It can be noted that the dynamic accuracy is quite rough.

- GPS position update rate of 4 [Hz].
  Attitude update rate of 100 [Hz].
  The attitude update rate is quite fast, therefore this device has been chosen as the main control loop frequency.

- Weight of 100 [g].
  The light weight makes this module suitable for unmanned aerial vehicle (the weight measure includes both the device and the connector).

- Average power consumption of 750×10^{-3} [W] ($= 150 \times 10^{-3}$ [A] at 5 [V]).
  The power consumption of the MTi-G is higher than that of SONAR and IR modules. However, as for the other devices, it does not affect the design of the power supply source, which has to deliver much more energy.

The UART frame communication is needed to allow the IMU-MCU interface. The IMU act as a master and imposes the timing. Therefore the main control loop on the MCU is triggered directly by the IMU frame. Even though the bit rate is quite high (115.2×10^3 [$bit\ s^{-1}$]), the delay between the beginning and the end of the frame is about 8×10^{-3} [s]. This interval is so long because the IMU data processing is computed during the communication (and not before). The frame is composed of six field as shown in figure 7.9.

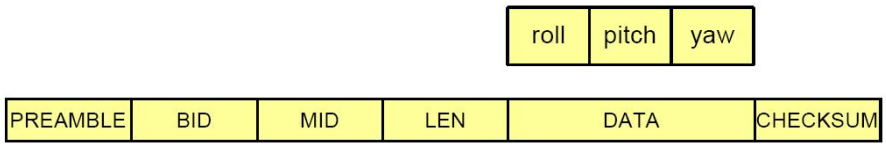


Figure 7.9: *IMU communication*

In the next list, each code preceded by "0x" represents a hexadecimal byte.

- The "PREAMBLE" (1 byte, 0xFA) indicates the start of the packet.
- "BID" (1 byte, 0xFF) specifies the bus identifier or address.
- "MID" (1 byte, 0x32) is the message identifier (MTData in this case).
- "LEN" (1 byte, 0x0C) represents the number of bytes in the "DATA" field.
- "DATA" (12 bytes) specifies the attitude data. It is composed of the three Euler angles (roll, pitch and yaw). Each one is represented (in degrees) with four bytes in the fixed point signed 12.20 format.
- "CHECKSUM" (1 byte) is used for communication error-detection. If all the message bytes (excluding the preamble) are summed and the lower byte value of the result equals zero, the message is valid.

Chapter 8

Actuators: motor drivers

In this chapter, the quadrotor's motor drivers are presented. They are fundamental to interface the Micro Controller Unit (MCU) outputs to the motors (connected with the propellers). Furthermore, thanks to the integrated circuit TD310, it is possible to limit the maximum output current and to process its value providing an analog signal to the MCU.

The first section (8.1: Schematic design) introduces the program for schematic design OrCAD Capture. It provides an overview of the driver circuit needed to interface the motors to the MCU. Furthermore the Choice of the components and their connections is reported.

The second section (8.2: Simulation) shows a few simulations of the circuit presented in the previous part. The program used to analyze the schematic is OrCAD PSpice. Particular attention is given to the switch phase of the motors voltage and to the frequency response of the current filter.

The third section (8.3: PCB design) underlines the importance of a Printed Circuit Board (PCB) design. The program used in this phase is OrCAD Layout. With this process, the circuit acquires robustness and minimizes space and weight.

The fourth section (8.4: Realization & final tests) shows the steps for the final circuit realization. Furthermore a few tests are reported (two oscilloscope screenshots are also attached).

8.1 Schematic design

OrCAD is a proprietary software tool suite used primarily for Electronic Design Automation (EDA). It is adopted mainly to design electronic schematics, process their simulations and realize Printed Circuit Boards [39]. The OrCAD suite has been used in this work because of its powerful tools and its versatility. OrCAD Capture is the software used for circuit schematic capture.

A motor driver must be connected between the MCU and the motor because it amplifies the electric power. Without a driver, it is not possible to supply one of the quadrotor's motor (more than 30 [W]) with a digital MCU output (less than 0.05 [W]). Furthermore an analog filter is designed to provide the average motor current signal to the MCU.

The core of the driver is the integrated circuit TD310. It is a triple IGBT/MOS driver with current sense. Even though it has three different inputs, it has been decided to drive them with the same signal to obtain an output with three times the current capability (obtained by short-circuiting all the outputs). The TD310 includes a current sense comparator which inhibits the output drivers in case of overcurrent. An alarm output signals this event. An operational amplifier is also included for current measurement (as an amplifier before the ADC's input of the MCU). The block diagram of the TD310 is shown in figure 8.1.

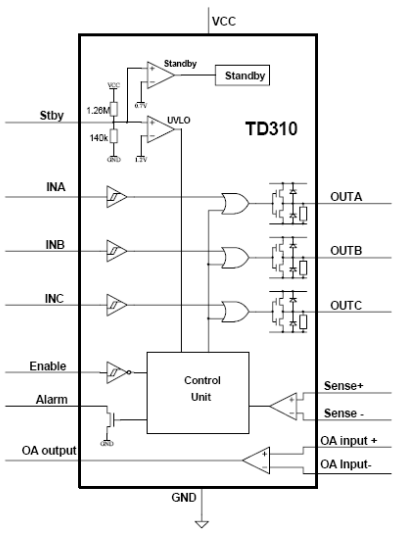


Figure 8.1: *TD310 block diagram*

Since each output can sink and source a current of 0.2 [A] maximum, the global output can supply 0.6 [A]. To deliver the right power to the motor, a power Metal-Oxide-Semiconductor Field-Effect Transistor (MOSFET) has been adopted. Its gate is connected (with a 22 [Ω] series resistor) to the TD310 output. The features of the RFP25N05 MOSFET are:

- Maximum drain current = 25 [A].
- Maximum drain-source voltage = 50 [V].
- Drain-source on resistance = 0.047 [Ω].
- Gate threshold voltage = $2 \div 4$ [V].
- Maximum turn on time = 60×10^{-9} [s].
- Maximum turn off time = 100×10^{-9} [s].
- maximum power dissipation 72 [W].

To measure the motor current (equal to the MOSFET one) a 0.1 [Ω] power resistor has been connected between the transistor source and the ground. The motor is instead placed between the drain and the power supply. A STTH2R06 fast recovery rectifier diode is connected in parallel to the motor (with its anode toward the MOSFET drain) to avoid transistor damages. It has an average forward current of 2 [A] and a typical reverse recovery time of 35×10^{-9} [s].

The current information on the 0.1 [Ω] resistor is low-pass filtered and amplified through a Sallen-Key architecture using the operational amplifier in the TD310. Furthermore, to avoid MCU damages, the series of a 100 [Ω] resistor and 5.1 [V] zener diode is connected to the amplifier output. Thank to this configuration, the zener voltage (connected to the MCU) saturates at 5.1 [V] if the amplifier one is higher.

Another series of resistor and zener diode is used to provide a stable voltage reference to the TD310's enable and the TD310's maximum current reference. The resistor is actually composed of two 1000 [Ω] parallel resistors. Furthermore a 100×10^{-9} [F] capacitor is added in parallel to the zener diode to reduce its voltage oscillations. Figure 8.2 shows the electronic schematic of the motor driver.

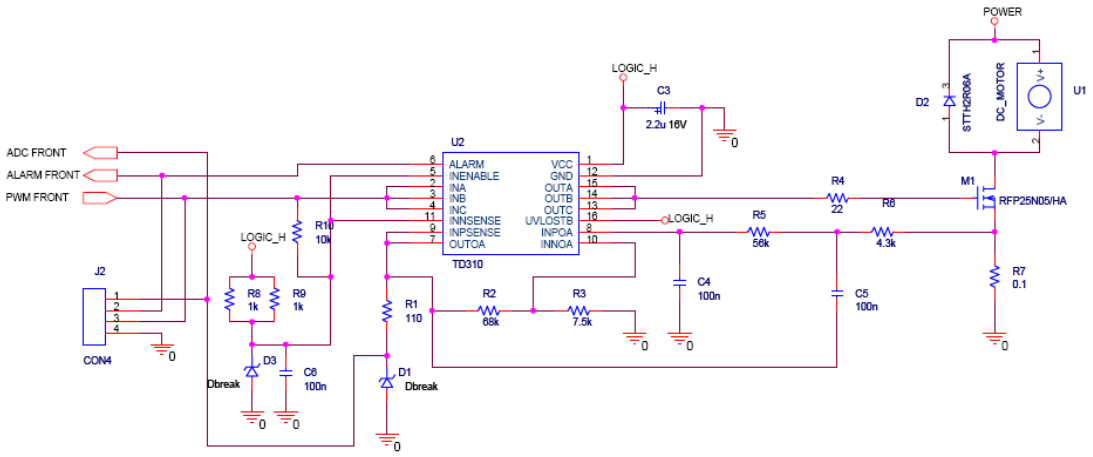


Figure 8.2: *Motor driver schematic*

In addition, a filter on the motor power supply is added to prevent spikes propagation on the logic power supply. The filter is composed of a 1×10^{-3} [H] inductor and a 470×10^{-6} [F] capacitor. Figure 8.3 shows the electronic schematic of the motor driver supply.

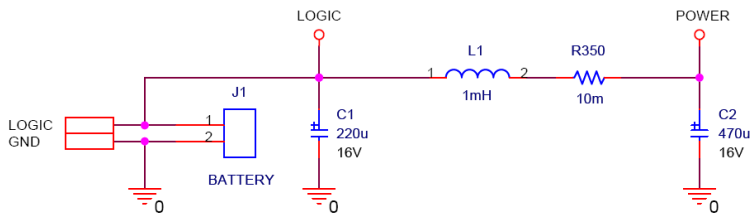


Figure 8.3: *Motor driver supply schematic*

To filter and amplify the motor current signal, it is a good choice to use the operational amplifier integrated in the TD310. A common architecture for this purpose is the low-pass Sallen-Key one. It is composed of an operational amplifier, four resistors and two capacitors. For simplicity they will be named R1, R2, R3, R4, C1 and C2. Figure 8.4 shows the schematic of this configuration [45].

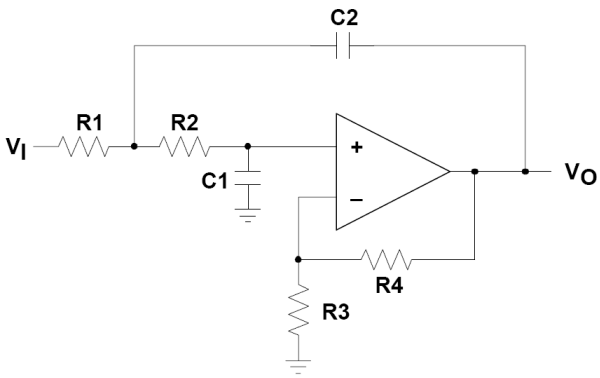


Figure 8.4: *Low-pass Sallen-Key architecture*

The (ideal) low-pass Sallen-Key transfer function is shown in equation (8.1)

$$\frac{V_O}{V_I} = \frac{K}{s^2\ (R1\ R2\ C1\ C2) + s\ (R1\ C1 + R2\ C2 + R1\ C2\ (1-K)) + 1} \tag{8.1}$$

Where K is the filter gain equal to $1 + R4/R3$. In this work its value has been set to ten. Therefore the choice of the resistors R4 and R3 has one degree of freedom. These components values have been fixed to: R3 = 7.5×10^3[Ω] and R4 = 68×10^3 [Ω].

By assuming C1 = C2 = C and R1 = m R2 = m R, the expressions of the cut-off frequency (f_c) and the quality factor (Q) can be rearranged as shown in equations (8.2) and (8.3).

$$Q = \frac{\sqrt{R1\ R2\ C1\ C2}}{R1\ C1 + R2\ C2 + R1\ C2\ (1-K)} = \frac{\sqrt{m}}{1 + 2\ m - m\ K} \tag{8.2}$$

$$f_c = \frac{1}{2\pi\sqrt{R1\ R2\ C1\ C2}} = \frac{1}{2\pi\ R\ C\sqrt{m}} \tag{8.3}$$

By imposing the cut-off frequency equal to 100 [Hz], the quality factor equal to $\sqrt{2}/2$ [-] (Butterworth poles) and the capacitor C equal to 100 $\times 10^{-9}$ [F], the filter components are the following:

- R1 = 4.3 $\times 10^3$ [Ω].
- R2 = 56 $\times 10^3$ [Ω].
- C1 = C2 = C = 100 $\times 10^{-9}$ [F].

8.2 Simulations

The simulation is a fundamental phase of the circuit prototyping. It consists in applying different signals in the circuit and analyzing the components behavior. With the software used in this phase (PSpice) several tests can be applied: bias, time domain (transient), temperature (sweep), AC or DC analysis, ...

The main strengths of this tool are:

- Tests done during simulation are always not destructive while those on the real circuit can be.
- In simulation it is possible to observe several parameters not directly accessible in the real circuit.

For these reasons, a lot of simulations were carried out during this work to verify the circuit behavior before proceeding with the layout process. In this section three tests are presented.

Figure 8.5 shows the behavior of the MOSFET during the transitions on-off and off-on. The red trace represents the input signal of the motor driver provided from the MCU. The green one shows the drain to source MOSFET voltage. It can be noted that both the MOSFET transitions last more than the input ones. The blue trace represents the drain current. A high spike (17 [A]) can be noted during the MOSFET turn on.

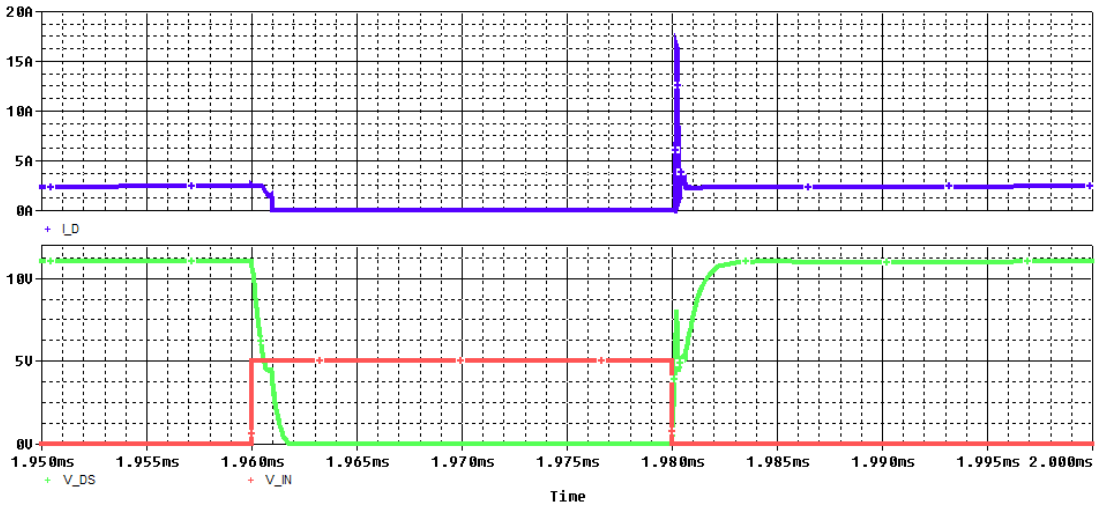


Figure 8.5: *MOSFET transitions with PSpice time domain simulation*

Figure 8.6 shows the frequency response of the current filter. A bilogarithmic representation has been chosen to emphasize the position of the double pole (100 [Hz]) and to verify the frequency trend. It's easy to note that the low frequency amplification is 10 [-], while the high frequency one descend with a 100 factor each decade (40dB/dec).

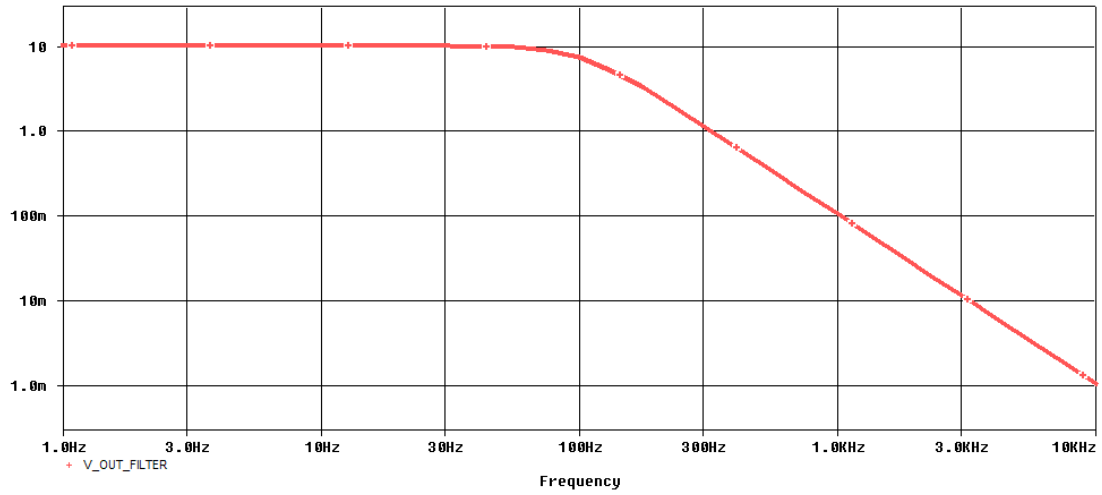


Figure 8.6: *Current filter frequency response PSpice AC sweep simulation*

Figure 8.7 shows the zener diode voltage in function of the logic power supply. For voltages between 9 [V] and 16 [V] the error (respect of the 5.1 [V] nominal value) is kept under 0.05 [V] (1%).

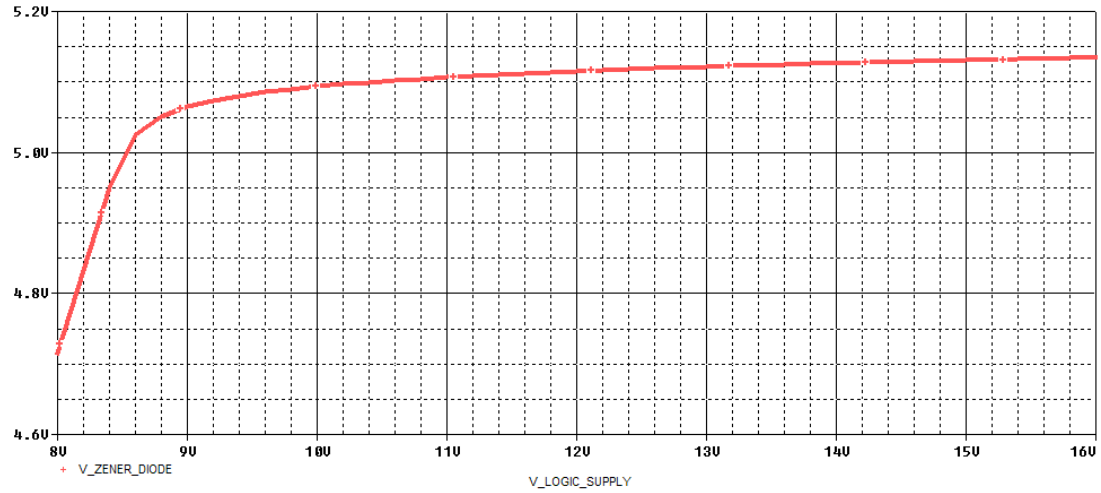


Figure 8.7: *Zener diode stability with PSpice DC sweep simulation*

8.3 PCB design

The Printed Circuit Board (PCB) design is a delicate part of the prototyping. It consists in defining the components topology and routing all the signals avoiding crossings. With the software used in this phase (OrCAD Layout) the placing and routing are aided with tools to simplify the process and reduce errors. With the files generated from OrCAD Layout, it is possible to program a machine for PCB realization and obtain the physic board. There are machines capable also to place components and solder them...

Thanks to the PCB design process, the circuit acquires robustness and minimizes space and weight. In this work, the power drive board has the size of just 4 times 6 centimeters and the weight of about 40 grams. Since the quadrotor motors are four, a equal number of motor drivers will be needed. Figure 8.8 shows the PCB top layer (on the left) and the PCB bottom layer (on the right).

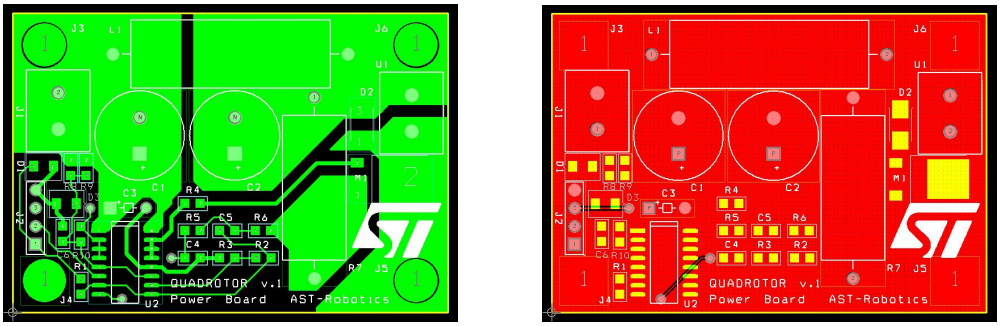


Figure 8.8: *PCB top & bottom layers view*

Figure 8.9 shows the PCB top view with the components footprints and labels.

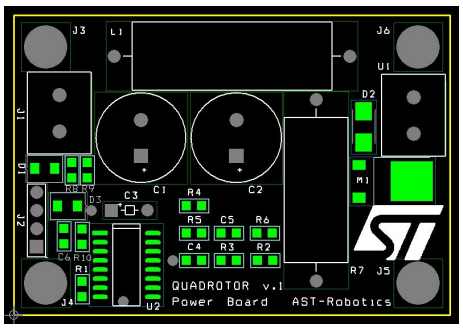


Figure 8.9: *PCB components view*

8.4 Realization & final tests

The Realization & final tests are the last two steps of the circuit prototyping. The realization of the circuit can be done in several ways. In this project, three different boards have been developed:

- In parallel with the design of the circuit schematic, a prototype on a breadboard were built to verify the correctness of both the OrCAD model of the components and the real system behavior.
- Since the power part requires high currents, a realization on stripboard were performed to allow a full circuit testability.
- The last (and definitive) prototype was built on the PCB support created in the previous section. The PCB must work correctly because in this phase is hard to find and correct mistakes.

The final tests step consists in using a multimeter, a waveform generator and an oscilloscope to verify the correctness of each part of the circuit. After the acquisition of these data, it is possible to compare the results with simulations and expectations. Figure 8.10 shows a photo (on a 1:1 scale) of the motor driver circuit in its final version.

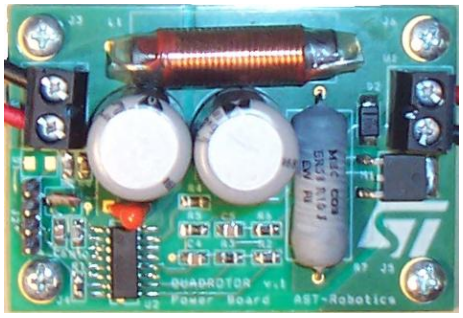


Figure 8.10: *Power board photo*

Figure 8.11 and 8.12 show two oscilloscope snapshots. In yellow the PWM input signal from the MCU is represented while in green is shown the PWM output from the TD310. The signal given to one motor terminal is violet colored (the other terminal is connected to the motor power supply). The last waveform (the pink one) is acquired from the output of the current filter.

In figure 8.11 the PWM output duty cycle is about 95%. Oscillations appear in the motor signal because of instantaneous current limitation in the power supply. The motor current is about 4.2 [A].

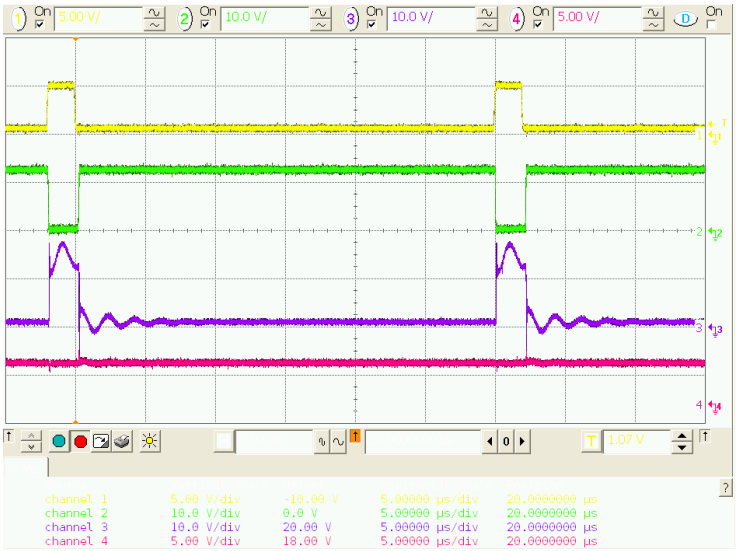


Figure 8.11: *Oscilloscope snapshot with PWM out duty cycle = 95%*

In figure 8.11 the PWM output duty cycle is about 15%. Oscillations appear even in this motor waveform. Furthermore it is easy to see the slow motor turn-off ramp (caused by the the motor indutor). The average current is much less than that one in the previous case: it is just 0.1 [A].

Figure 8.12: *Oscilloscope snapshot with PWM out duty cycle = 15%*

Chapter 9

STR730 Micro Control Unit

In this chapter, the STR730 Micro Control Unit is presented. It can be considered the brain of the quadrotor system since it organizes the communications of all the devices. The peripherals interface and timing is moreover a delicate issue. The software code and architecture allow to keep the whole system under controll and stabilize the helicopter movements.

The first section (9.1: STR730 MCU overview) introduces the Microcontroller features. A lot of peripherals are integrated in the MCU through dedicated hardware to allow a wider range of interfaces and applications. The evaluation board ARMIC30 helped to test them easily.

The second section (9.2: Peripherals organization) shows how the quadrotor system is organized. A table presents the characteristics of the MCU's peripherals and their use. Furthermore all the devices and boards are reported in a block diagram to visualize their interconnections.

The third section (9.3: Software architecture) presents the software structure. A flow chart of the algorithm is presented to better understand the main control cycle and its timing. In addition, it was possible to observe the variables trend and set control parameters through a Matlab interface.

9.1 STR730 MCU overview

The STR7 family of 32-bit microcontrollers combines the industry-standard ARM7TDMI 32-bit Reduced Instruction Set Computer (RISC) core, featuring high performance, very low power and very dense code with a comprehensive set of peripherals and ST's latest embedded flash technology. The ARMIC30 evaluation board used in this work mounts the MCU STR730FZ2T7. Its main features are [46]:

- 32 Million Instruction Per Second (MIPS) at 36 MHz.
  Two clock sources are used to drive the microcontroller, a main clock driven by an external crystal resonator and an internal backup RC oscillator that operates at 2MHz or 32 kHz. The embedded PLL is configured in this work to generate an internal system clock of 36 MHz. The PLL output frequency can be programmed using a wide selection of multipliers and dividers.

- 256 kbytes flash program memory.
  The Flash program memory is organized in 32-bit wide memory cells which can be used for storing both code and data constants. It is accessed by CPU with zero wait states at 36 MHz. It can be programmed using In-Circuit Programming or In-Application programming with a memory endurance of 10K write/erase cycles and data retention of 20 years at 85 C.

- 16 kbytes RAM.
  The 16 kbytes static RAM can be accessed as bytes, half-words (16 bits) or full words (32 bits).

- 64 maskable Interrupt ReQuests (IRQs) with 16 priority levels.
  16 external interrupts lines are available for application use. In addition, up to 32 external wakeup lines can be used as general purpose interrupts.

- 112 I/O ports.
  Up to 112 ports are programmable as general purpose input/output or alternate function.

- 4 Direct Memory Access (DMA) controllers.
  4 DMA controllers, each with 4 data streams, can manage data transfers. Several configuration are possible: memory to memory, peripheral to peripheral, peripheral to memory and memory to peripheral.

- 1 16-bit watchdog timer.
  The 16-bit watchdog timer protects the application against hardware or software failures and ensures recovery by generating a reset.

- 10 16-bit timers.
  Each of the timers have a 16-bit free-running counter with 7-bit prescaler, up to two input capture/output compare functions, a pulse counter function, and a PWM channel with selectable frequency. This provides a total of 16 independent PWMs when added with the PWM modules.

- 6 16-bit Pulse Width Modulation (PWM) modules.
  The six 16-bit PWM modules have independently programmable periods and duty-cycles, with 5+3 bit prescaler factor.

- 3 16-bit timebase timers.
  The three 16-bit Timebase Timers with 8-bit prescaler for general purpose time triggering operations.

- 2 Inter-Integrated Circuit (I^2C) interfaces.
  The two I^2C Interfaces provide multi-master and slave functions, support normal and fast I^2C mode (400 kHz) and 7 or 10-bit addressing modes.

- 4 Universal Asynchronous Receiver Transmitter (UART) interfaces.
  The 4 UARTs allow full duplex, asynchronous, communications with external devices with independently programmable TX and RX baud rates up to 625 kbaud.

- 3 Buffered Serial Peripheral Interfaces (BSPI).
  Each of the three BSPIs allow full duplex, synchronous communications with external devices, master or slave communication at up 6 Mb/s (at 36 MHz System Clock).

- 3 Controller Area Network (CAN) interfaces.
  The three CAN modules are compliant with the CAN specification V2.0 part B (active). The bit rate can be programmed up to 1 MBaud.

- 16 channels ADC.
  The 10-bit Analog to Digital Converter, converts up to 16 channels in single-shot or continuous conversion modes. The minimum conversion time is 3us.

- Joint Test Action Group (JTAG) interface.
  This interface is primarily used for communicating with a PC using a suitable converter circuit. There exists a wide choice of development tools on the market supporting microcontroller Flash memory programming and application debugging.

The block diagram of the STR730 MCU is presented in figure 9.1.

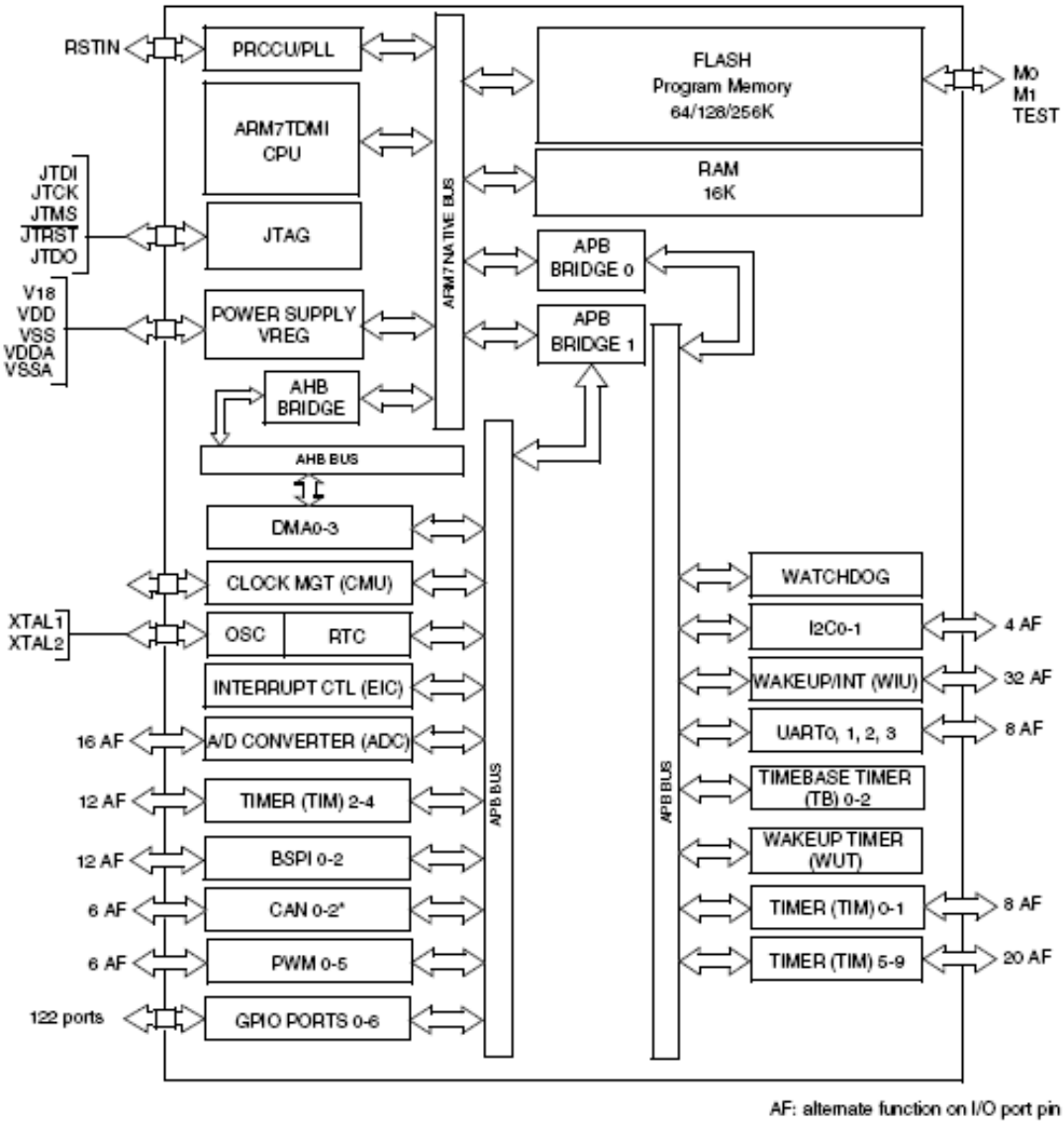


Figure 9.1: *STR730 block diagram*

The STR730 provides competitive features and a lot of built-in peripherals. For this reason, it is possible to expand the project with other devices and functionalities. To make this platform easy to modify, test and expand, the evaluation board ARMIC30 has been used. It is composed of the STR730 with a few circuits (power supplies, clock generation, JTAG interface, ...) to make the CPU work without adding additional hardware. This tool is very useful to reduce the developing phase. Furthermore the ARMIC30 is equipped with several connectors for testing and peripherals interfacing. Figure 9.2 shows a photo of the evaluation board with a few labels to identify the main parts.

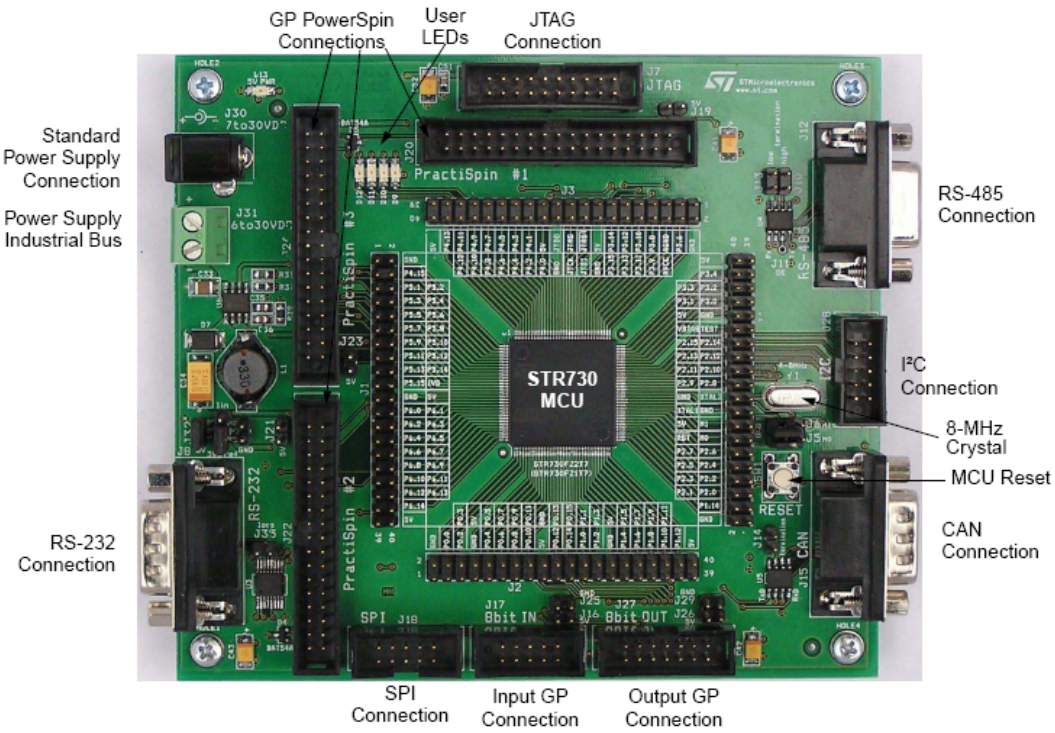


Figure 9.2: *Photo of the evaluation board ARMIC30*

9.2 Peripherals organization

The STR730 has high potentialities as shown in the previous section. All the peripherals are interfaced with it through specific in-circuit devices. Table 9.1 shows a list of the characteristics of the MCU's interconnections.

Periph.	Pins	Mode	Use	Impl.	Timing	Inter.
GPIO4	4.9-4.12	OUT	LEDS	YES	SYNC	NO
GPIO5	5.11-5.14	IN	MOTOR	NO	ASYNC	YES
TIM1-4	0.2,0.12 1.3,1.12	PWM IN	RC	YES	ASYNC	NO
TIM5	NO	TIMER	TEST	YES	ASYNC	YES
TIM7-8	4.0,4.2	PWM IN	RC	NO	ASYNC	NO
PWM1-4	2.3-2.6	PWM OUT	MOTOR	YES	SYNC	NO
TB0	NO	TIMER	TEST	YES	ASYNC	YES
I^2C0	2.14,2.15	I^2C	SONAR	YES	SYNC	YES
UART0	6.8,6.9	UART	IMU	YES	ASYNC	YES
UART3	6.2,6.4	UART	PC	YES	ASYNC	YES
ADC0-3	3.0-3.3	ADC	MOTOR	NO	SYNC	YES
ADC4	3.4	ADC	IR	YES	SYNC	YES
ADC5,6	3.5,3.6	ADC	BATT	YES	SYNC	YES
ADC7-11	3.7-3.11	ADC	IR	NO	SYNC	YES

Table 9.1: multimete

The first column ("Periph.") identifies the name of the MCU in-circuit peripheral, while the second one ("Pins") specifies ports and pins of the communication. In the column "mode", the type of the interface is reported and in "Use" the name of the connected device is specified. "Impl." (abbreviation of implementation) specifies whether the connection MCU-peripheral has been done yet or not. The last two columns show if the timing mode is synchronous or asynchronous respect to the main control cycle ("Timing") and if the communication occurs with a hardware interrupt request ("Inter.").

Even though table 9.1 shows a lot of technical information about the peripherals interfacing, it is still a little bit difficult to have a clear idea of the system organization. Figure 9.3 presents a block diagram of the quadrotor system architecture to visualize devices and boards interconnections.

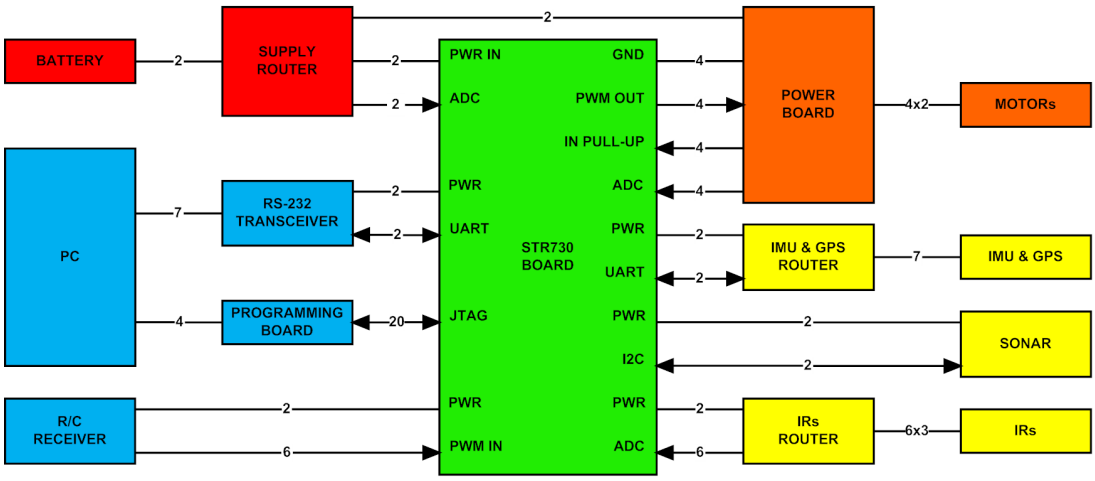


Figure 9.3: *Quadrotor system architecture*

The numbers reported on the lines (or arrows) specify the signals in each connection. Thanks to table 9.1 and figure 9.3 it is possible to better understand the configuration of the devices. It now follows a list which explains the blocks reported in figure 9.3.

- in red the power supply part is represented.
  It is composed by a supply source ("BATTERY") and a board ("SUPPLY ROUTER") which filters the lines, routes them and provide two ADC readable signals. The power supply of the system is divided in two groups: the motor power supply and the logic power supply. The first one must deliver a lot of power to the motors, while the second one drives all the other devices and requires much lower power, but a more stable reference. For this reasons the two lines have been divided and filtered in the "SUPPLY ROUTER" board. Furthermore the information of the lines voltages is acquired from the MCU through two ADC channels (after a level attenuation).

- in blue the PC and the RF receiver communications are presented.
  The PC is interfaced to the STR730 through two different channels. The first one is needed to program the MCU software thanks to a "PROGRAMMING BOARD". This hardware interfaces the USB with the JTAG. The second communication channel is used to send and receive data between PC and MCU while the system is running. Since the PC provides a serial interface and the microcontroller has a UART one, a "RS-232 TRANSCEIVER" is needed to allow hardware compatibility. The "RF RECEIVER" sends the signals imposed by the remote controller to the STR730 PWM inputs.

- in orange the motors part is represented.
  A "POWER BOARD" is needed to correctly drive the motors (actually in the real platform there is one power board for each motor). It receives the power supply directly from the "SUPPLY ROUTER" as stated before. The PWM signals from the STR730 set the motors speed. Two types of feedback are provided to the MCU (in the real platform they are not yet implemented). The first one is composed of four analogue signals proportional to the motors current. They are acquired through ADC inputs. The other type of feedback is represented by the information of the overcurrent alarm. An interrupt is required to handle this situation.

- in yellow the sensors are represented.
  The "IMU & GPS" device communicates with the MCU through a "IMU & GPS ROUTER" which powers the MTi-G and routes the needed signals to the microcontroller's UART. Even the "IRs" needs a circuit ("IRs ROUTER") which correctly routes the IR modules signals. In the real platform just a IR module for height control is mounted, however six modules will be supported in future works. The "SONAR" is directly connected to the STR730's I^2C in-circuit peripheral.

- in green the ARMIC30 evaluation board with the STR730 is represented.
  All the MCU's interfaces have been discussed above. Just one thing must be pointed out: the logic power supply ("PWR IN") connected with the ARMIC30 is an unregulated 9 [V] to 15 [V] supply. The switching regulator in the evaluation board provides instead a stable 5 [V] reference which powers all the devices connected with the STR730 on the "PWR" line.

9.3 Software architecture

The software structure of the algorithm can be simplified in an easy scheme. After the start-up of the system follows the initialization phase. It is needed to set the right values and commands to variables and peripherals. After this procedure, the main control loop begins. Each time, the algorithm waits until the IMU have finished sending the attitude data. The IMU is therefore used as the trigger of the system (100 Hz frequency). Furthermore, during the IMU frame, the other sensors (SONAR and IR) are acquired. In this way it is possible to synchronize all the sensors together.

When the IMU, SONAR and IR data are acquired, the algorithms reads the last data memorized in the PWM inputs (related to the remote controller). The following step is the control algorithm. It interprets and filters both the sensors and the remote controller data. After that it applies a control law to reach the targets. This phase requires a lot of calculus, hence time. In the real platform it lasts more or less four milliseconds.

At the end of the computation, the software sets the PWM outputs data which, through the motor power boards, controls the propellers. To observe the variables and correctness of the algorithm, a PC communication starts right after the PWM setting. A Matlab algorithm has been developed to acquire the data. Thanks to the capability to plot quantities and do computation, this tool has been very useful to test the STR730 code. In addition, it was possible to set control parameters (through a Matlab interface) with a PC communication whenever during the code execution.

Figure 9.4 shows the flow chart of the algorithm to better understand the main control cycle and its timing. On the right part of the figure, a block diagram of the timing, acquisitions and communications is represented. It is clear that a whole cycle lasts 17 milliseconds: 8 for the acquisition, 4 for the control algorithm and 5 for the PC communication. Since the system is triggered every 10 milliseconds, the algorithm must handle more than one phase at the same time (pipeline). Therefore the control algorithm and the communication of a dataset occur while the sensors acquire new quantities.

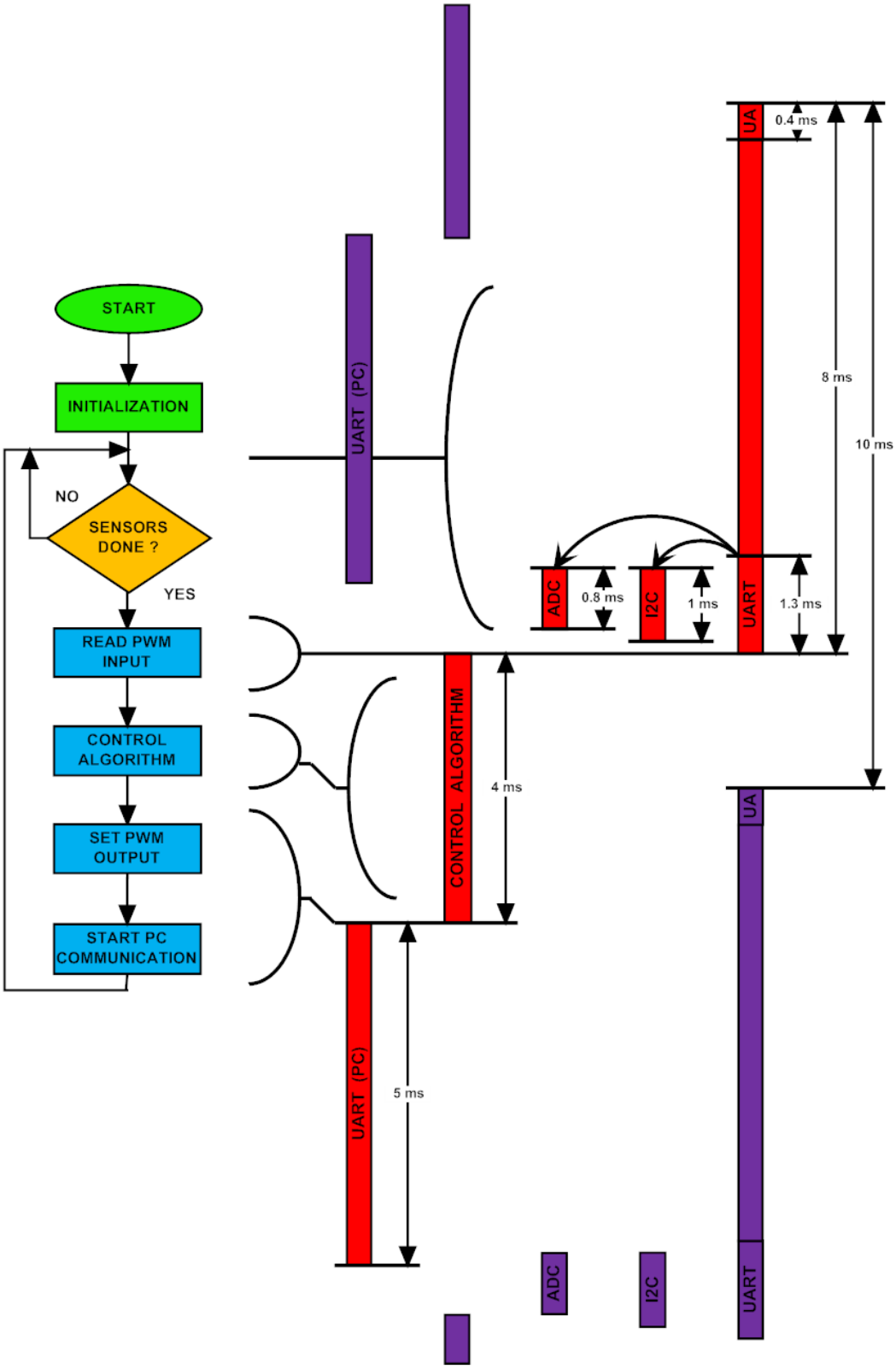


Figure 9.4: *Software flow chart*

Chapter 10

Experimental results

The platform developed during this thesis is a small-scale helicopter with four rotors in cross configuration. Its width and length are 782 mm while the height is 154 mm. The whole system weights about 1 kg without batteries.

The cross-structure, motors and propellers were bought on internet from the RCToys website. It was chosen the Draganflyer V model (without its electronics) for its simple mechanics and reasonable price. The hardware system and the software were instead developed during this thesis work.

The electronics was held together thanks to a plastic box. A lot of holes, with a specific pattern, were drilled for decreasing the weight, avoiding rigidity losses.

Figure 10.1 shows the first layer of the electronic box. In the center-left of the picture it is easy to note two switches connected to the "supply router" board. The MCU demo-board ARMIC30 is located in the middle of the picture. A lot of wires are used to connect the board to external devices. One of them is the RF receiver on the top-right of figure 10.1.

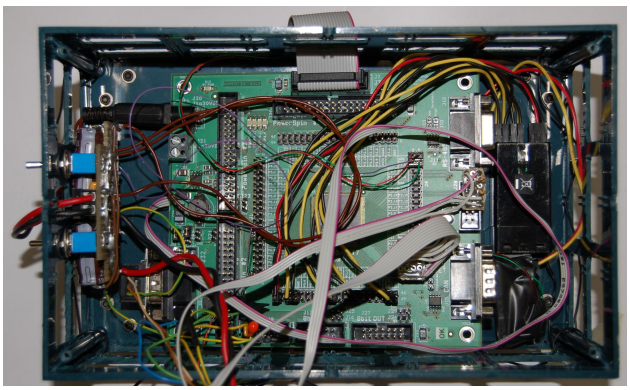


Figure 10.1: *First layer of the electronic box*

Figure 10.2 shows the second layer of the electronic box. In the middle of the picture, the four motor driver boards are shown. They were fixed on a paperboard to avoid electric contacts with the ARMIC30 board (placed below). The wires in these circuits are wider because they have to support higher currents.

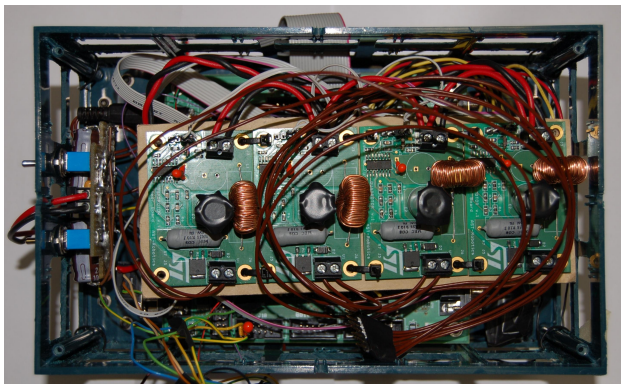


Figure 10.2: *Second layer of the electronic box*

During flight maneuvers, vibrations are transmitted from the propellers to the electronics. To keep the electronics steady in its place, a thin metal cover has been fixed to the box top. This cover pushes a little bit the boards in the box. Thanks to this pressure, the components are not free to move and stay still in their place. The metal cover were drilled for decreasing the weight, as done for the plastic box. Furthermore, The Inertial Measurement Unit was fixed on the metal cover to exploit the available space. Figure 10.3 shows the top view of the electronic box. It is easy to note the holed metal cover and the orange IMU.

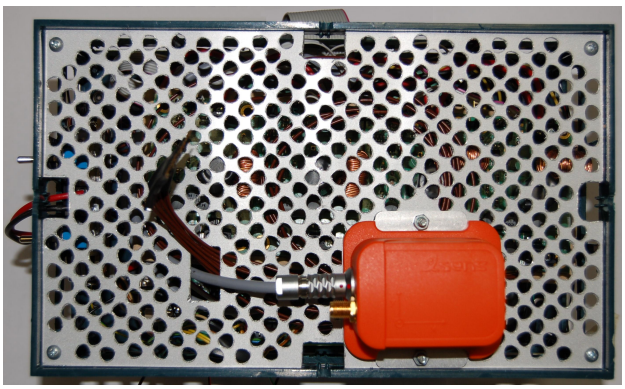


Figure 10.3: *Top view of the electronic box*

The bottom view of the electronic box is shown in figure 10.4. The two devices mounted in the middle are used to measure the height of the quadrotor (from the ground). The SONAR is the left one and has a dark red board. The two ultrasound capsules are easy to identify. The black IR module is placed on the right of the SONAR. For each bottom-corner of the electronic box, a thin leg was fixed to avoid ground-sensors contact. Furthermore, a polyester part was added to the legs as mechanic dumper.

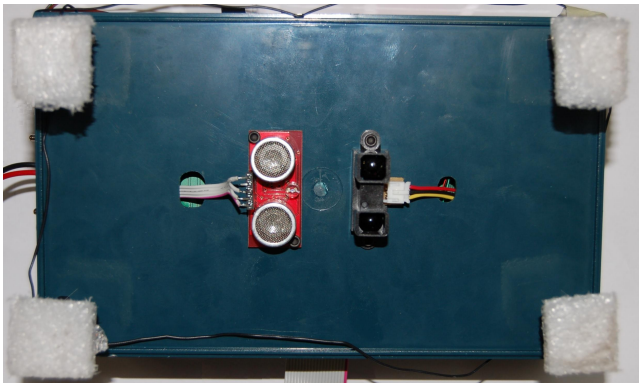


Figure 10.4: *Bottom view of the electronic box*

Figure 10.5 shows a photo of the whole quadrotor platform. The cross structure is fixed to the electronics box in four points. On the center-left of the picture, there is a flat cable used for the MCU programming. On the center-right, instead, a few wires are connected to a RS232 transceiver for the PC interface. The two wires placed at the bottom-center of the picture are used for the power supply.

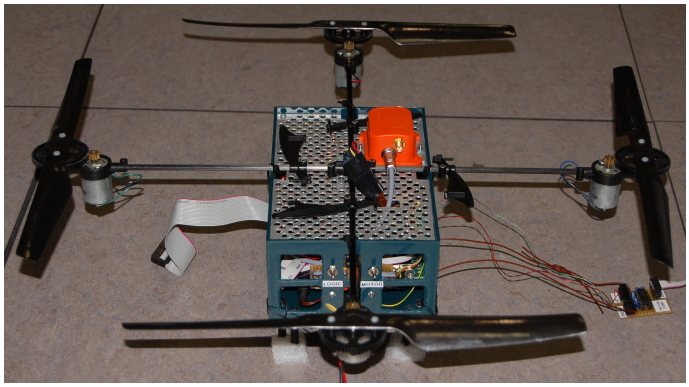


Figure 10.5: *Photo of the quadrotor platform*

Thanks to the MCU-PC interface, it was possible to acquire quadrotor data during flight sessions. Those data helped to tune the control parameters with a quantitative feedback. For all the four stabilization algorithms (height, roll, pitch and yaw), a full PID controller were implemented. The obtained performance were satisfactory even with a rough mechanic structure. In the next figures, a few quadrotor tests are presented. They are divided in two plots to better identify the trend of the signals and the error boundaries.

Figure 10.6 shows the roll stabilization performance. The roll error in static conditions is always lower than one degree.

In figure 10.7 the pitch stabilization performance is shown. Even in this case the pitch error in static conditions is always lower than one degree.

It is very important that both roll and pitch errors are kept low to provide stable flight. In the hovering condition, if one of the two angles are different than zero a longitudinal acceleration occur. This behavior makes difficult to maintain a fixed position without drift.

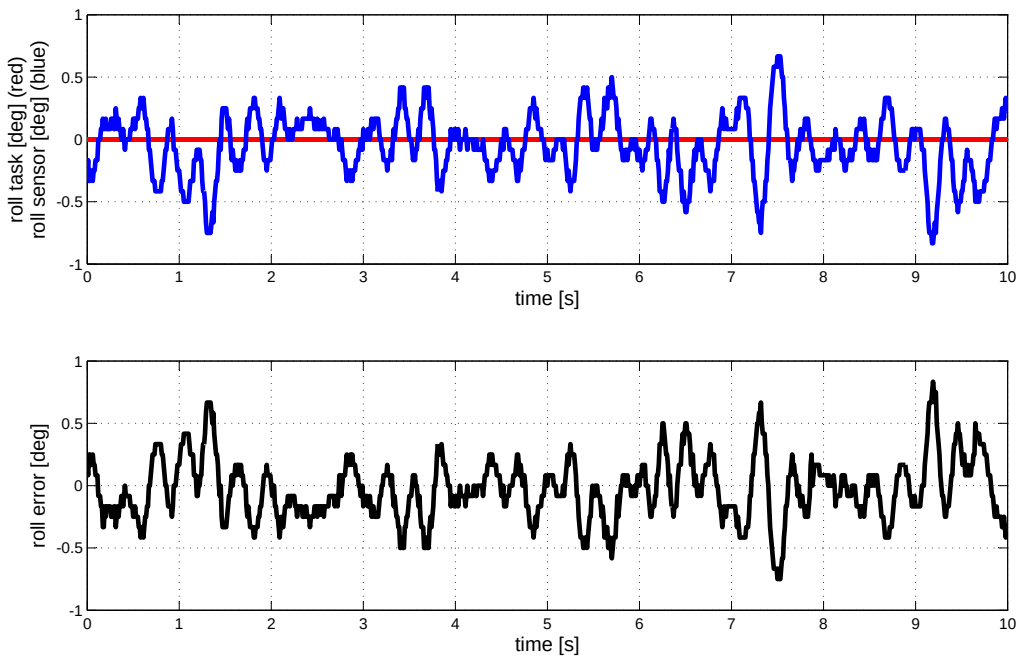


Figure 10.6: *Roll stabilization performance*

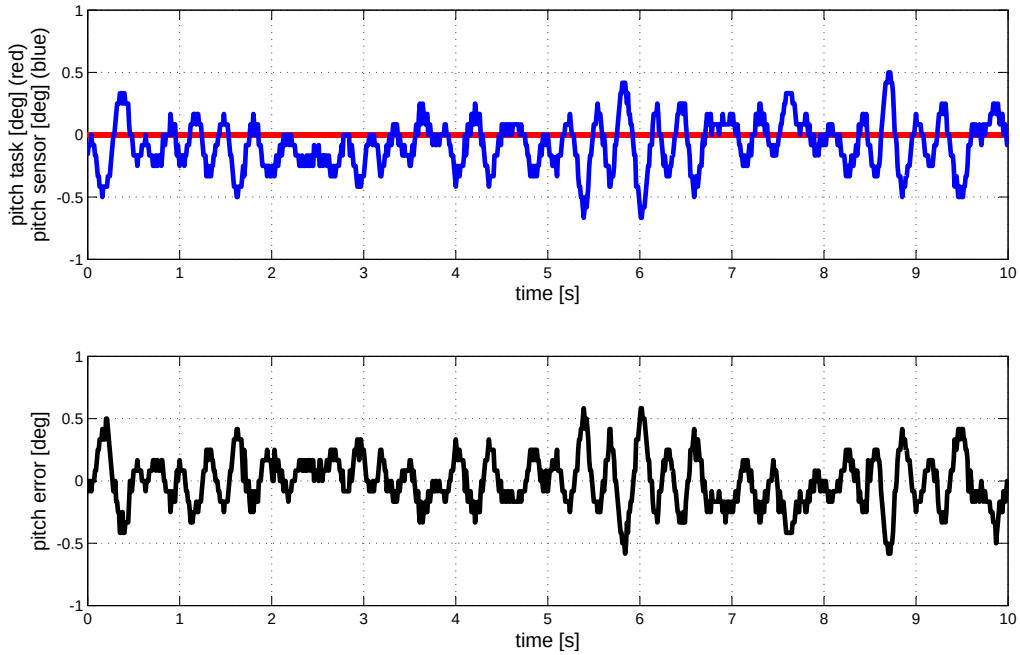


Figure 10.7: *Pitch stabilization performance*

The yaw stabilization has lower requirements: an error in the yaw angle does not cause any longitudinal acceleration in hovering condition. However the dynamic range is much wider than the roll or pitch one. The yaw can be changed between -180 and +180 degrees while the other two (roll and pitch) shows small variation (less than 10 degrees). Therefore, good performance in both dynamic and static tracking is required.

Figure 10.8 shows the yaw stabilization performance. The yaw error in static conditions is always lower than two degrees while that one in dynamic tracking is kept under four degrees.

The height stabilization sensors have low resolution compared to the altitude task. In most of the tests the height was kept lower than twenty centimeters. These values were chosen to test the helicopter performance with ground-effect interferences. Even in these conditions, the quadrtor showed good performance.

In figure 10.9 the height stabilization performance is shown. The height error in static conditions is lower than two centimeters.

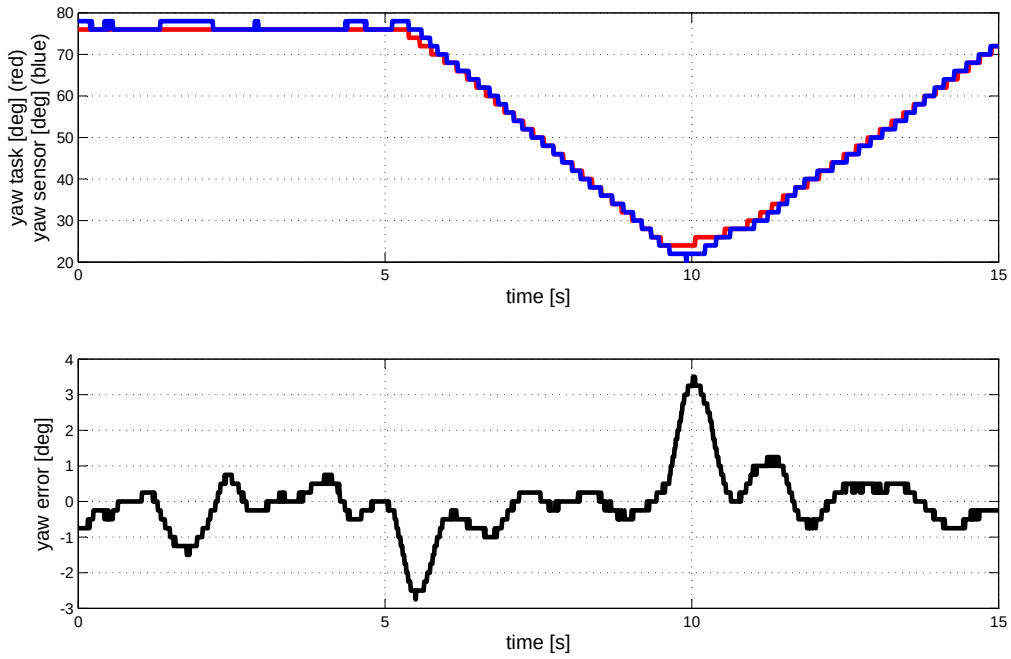


Figure 10.8: *Yaw stabilization performance*

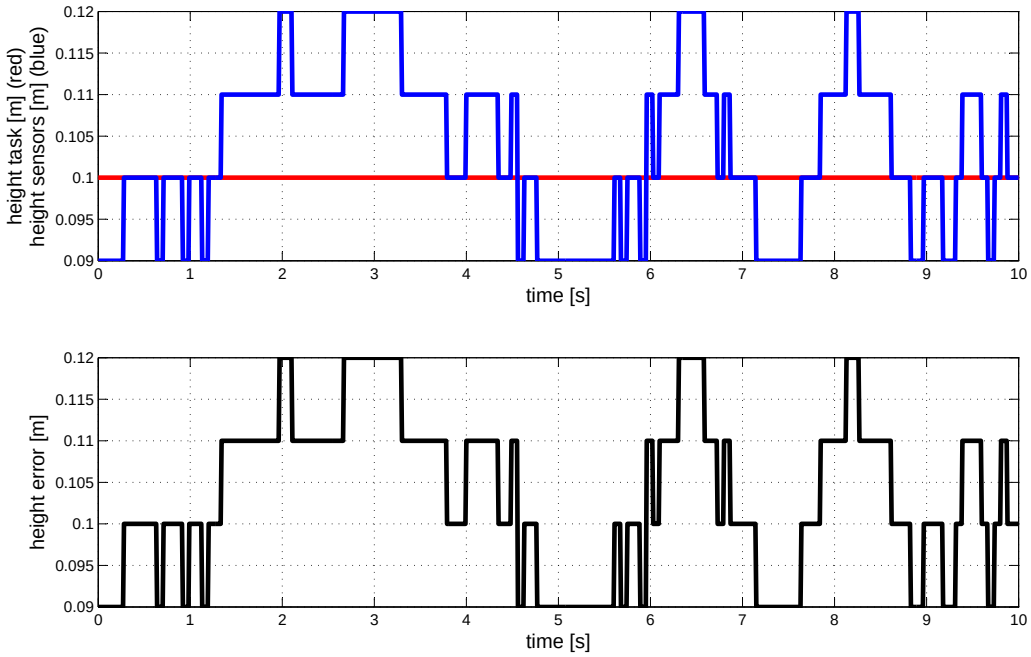


Figure 10.9: *Height stabilization performance*

Chapter 11

Conclusion

The goals of this thesis work were to model the quadrotor helicopter and to test its control algorithm, thanks also to a simulator. Furthermore these theoretical considerations were taken into account to develop a real platform.

The quadrotor model was presented in chapter 3. Two appendices deepened the dynamics and the aerodynamics basics (A and C respectively). In chapter 4 the control algorithm structure was explained. A simulator was adopted to test both dynamics and control, as shown in chapter 5. To develop a real platform, it was necessary to identify the physic constants used in the model. Chapter 6 and appendices B and D showed the identification process. At last, chapters 7 - 9 gave an accurate description of the quadrotor components and their interconnections. In addition, the experimental results were reported in chapter 10.

According to the goals of this project, the research was very detailed in both modelling and simulation. Thanks also to the identification process, the performance of the real platform was satisfactory in spite of the mechanic structure inaccuracies. The quadrotor tests showed roll and pitch errors always less than one degree. Even the yaw error had low values: less than two degrees under static condition and less than four degrees under dynamic tracking. The height stabilization had an error of just two centimeters. This last value was considerably low compared with the one centimeter resolution of the height sensor in that range.

To improve this quadrotor project, a more accurate model of the helicopter can be studied, in particular aerodynamic considerations can help in non hovering operation. Together with this research, the identification of the real platform physics must be much more accurate.

Several control algorithms can be investigated to find the best trade-off between performance and software complexity. A lot of articles which focus on quadrotor stabilization algorithm have been already written. However it would be great to compare them and find better solutions.

Even though the simulator showed already good accuracy and testability, it would be great to be able to simulate the environment too and to use tools which interact with the real platform.

The low level controller, implemented in this thesis, had the goal of height and attitude stabilization. A high level controller can be connected to the previous one to follow position requirements. Other tasks, computed by the high level controller, can be obstacle avoidance and trajectory planning. Of course, to improve the locating performance, several sensors must be connected to the platform. A GPS can be used for knowing the global position in an outdoor scenario while a network of IR modules and/or SONARs can be mounted to have information about objects around the quadrotor as well as its position.

A camera can be used not only to determine the position, but also for a lot of other purpose. For example it can be required for the tracking of mobile targets or for environment mapping. In both cases, a camera capable of pan-tilt rotation could achieve better performance.

The mechanic structure can be developed to carry a higher payload and a dedicate electronic design can lower the size and weight of the needed circuitry. Furthermore the DC-motors can be replaced by the brush-less onces to know the exact propeller position (hence speed) and to have a higher thrust to weight ratio.

A last improvement can be done by replacing the remote controller with a PC to be able to provide more complex tasks. It would be also good to have a feedback from the helicopter and to show internal states, maps, images, ...

Appendix A

Kinematics and Dynamics

This appendix describes the basic equations which identify a 6 DOF rigid body. The Newton-Euler formulation has been adopted in this work.

The quadrotor model can be evaluated according kinematics (first) and dynamics (after) equations:

- Kinematics (section A.1)
- Dynamics (section A.2)

A.1 Kinematics

Kinematics is a branch of mechanics which studies the motion of a body or a system of bodies without consideration of the forces and torques acting on it. To describe the motion of a 6 DOF rigid body it is usual to define two reference frames [33]:

- Earth inertial reference (E-frame)
- Body-fixed reference (B-frame)

The E-frame (o_E, x_E, y_E, z_E) is chosen as the inertial right-hand reference. x_E points toward the North, y_E points toward the West, z_E points upwards respect to the earth and o_E is the axis origin. This frame is used to define the linear position ($\mathbf{\Gamma}^E$ $[m]$) and the angular position ($\mathbf{\Theta}^E$ $[rad]$) of the quadrotor.

The B-frame (o_B, x_B, y_B, z_B) is attached to the body. x_B points toward the quadrotor front, y_B points toward the quadrotor left, z_E points upwards and o_B is the axis origin. o_B is chosen to coincide with the center of the quadrotor cross structure. This reference is right-hand too. The linear velocity ($\boldsymbol{V}^B$ $[m\ s^{-1}]$), the angular velocity ($\boldsymbol{\omega}^B$ $[rad\ s^{-1}]$), the forces ($\boldsymbol{F}^B$ $[N]$) and the torques ($\boldsymbol{\tau}^B$ $[N\ m]$) are defined in this frame.

The linear position $\mathbf{\Gamma}^E$ of the helicopter is determined by the coordinates of the vector between the origin of the B-frame and the origin of the E-frame respect to the E-frame according to equation (A.1).

$$\mathbf{\Gamma}^E = \begin{bmatrix} X & Y & Z \end{bmatrix}^T \tag{A.1}$$

Figure A.1 shows the two frames and their relation.

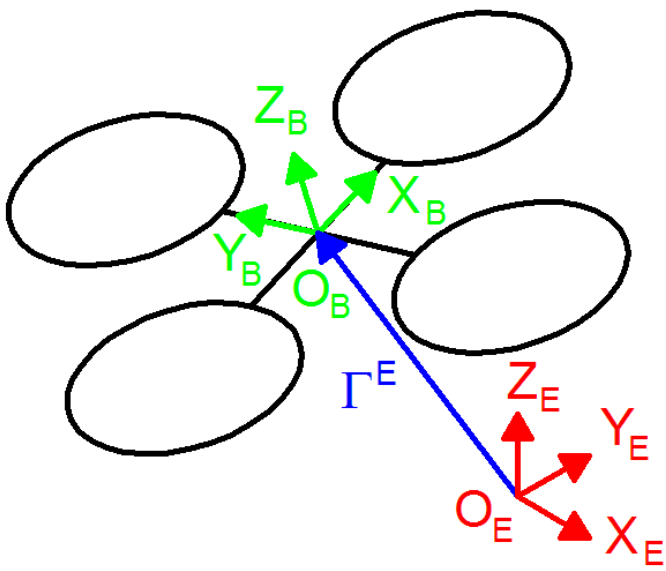


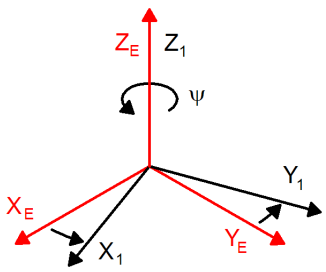
Figure A.1: *Quadrotor frames*

The angular position (or attitude) $\mathbf{\Theta}^E$ of the helicopter is defined by the orientation of the B-frame respect to the E-frame. This is given by three consecutive rotations about the main axes which take the E-frame into the B-frame. In this work the "roll-pitch-yaw" set of Euler angles were used. Equation (A.2) shows the attitude vector.

$$\mathbf{\Theta}^E = \begin{bmatrix} \phi & \theta & \psi \end{bmatrix}^T \tag{A.2}$$

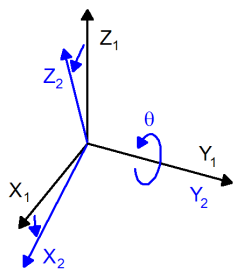
The rotation matrix $\boldsymbol{R}_{\boldsymbol{\Theta}}$ $[-]$ is obtained by post-multiplying the three basic rotation matrices in the following order:

- Rotation about the z_E axis of the angle ψ (yaw) through $\boldsymbol{R}(\psi, z)$ $[-]$.



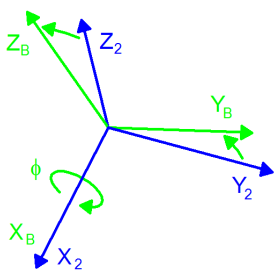
$$\boldsymbol{R}(\psi, z) = \begin{bmatrix} c_\psi & -s_\psi & 0 \\ s_\psi & c_\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{A.3}$$

- Rotation about the y_1 axis of the angle θ (pitch) through $\boldsymbol{R}(\theta, y)$ $[-]$.



$$\boldsymbol{R}(\theta, y) = \begin{bmatrix} c_\theta & 0 & s_\theta \\ 0 & 1 & 0 \\ -s_\theta & 0 & c_\theta \end{bmatrix} \tag{A.4}$$

- Rotation about the x_2 axis of the angle ϕ (roll) through $\boldsymbol{R}(\phi, x)$ $[-]$.



$$\boldsymbol{R}(\phi, x) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_\phi & -s_\phi \\ 0 & s_\phi & c_\phi \end{bmatrix} \tag{A.5}$$

In the previous three equations (and in the following), this notation has been adopted: $c_k = \cos k$, $s_k = \sin k$, $t_k = \tan k$. Equation (A.6) shows the composition of the rotating matrix $\boldsymbol{R}_{\boldsymbol{\Theta}}$.

$$\begin{aligned} \boldsymbol{R}_{\boldsymbol{\Theta}} &= \boldsymbol{R}(\psi, z)\ \boldsymbol{R}(\theta, y)\ \boldsymbol{R}(\phi, x) = \\ &= \begin{bmatrix} c_\psi c_\theta & -s_\psi c_\phi + c_\psi s_\theta s_\phi & s_\psi s_\phi + c_\psi s_\theta c_\phi \\ s_\psi c_\theta & c_\psi c_\phi + s_\psi s_\theta s_\phi & -c_\psi s_\phi + s_\psi s_\theta c_\phi \\ -s_\theta & c_\theta s_\phi & c_\theta c_\phi \end{bmatrix} \end{aligned} \tag{A.6}$$

As stated before, the linear $\boldsymbol{V}^B$ and the angular $\boldsymbol{\omega}^B$ velocities are expressed in the body-fixed frame. Their compositions are defined according to equations (A.7) and (A.8).

$$\boldsymbol{V}^B = \begin{bmatrix} u & v & w \end{bmatrix}^T \tag{A.7}$$

$$\boldsymbol{\omega}^B = \begin{bmatrix} p & q & r \end{bmatrix}^T \tag{A.8}$$

It is possible to combine linear and angular quantities to give a complete representation of the body in the space. Two vectors, can be thus defined: the generalized position $\boldsymbol{\xi}$ [+] and the generalized velocity $\boldsymbol{\nu}$ [+], as reported in equations (A.9) and (A.10).

$$\boldsymbol{\xi} = \begin{bmatrix} \boldsymbol{\Gamma}^E & \boldsymbol{\Theta}^E \end{bmatrix}^T = \begin{bmatrix} X & Y & Z & \phi & \theta & \psi \end{bmatrix}^T \tag{A.9}$$

$$\boldsymbol{\nu} = \begin{bmatrix} \boldsymbol{V}^B & \boldsymbol{\omega}^B \end{bmatrix}^T = \begin{bmatrix} u & v & w & p & q & r \end{bmatrix}^T \tag{A.10}$$

The relation between the linear velocity in the body-fixed frame $\boldsymbol{V}^B$ and that one in the earth frame $\boldsymbol{V}^E$ $[m\ s^{-1}]$ (or $\dot{\boldsymbol{\Gamma}}^E$ $[m\ s^{-1}]$) involves the rotation matrix $\boldsymbol{R}_{\boldsymbol{\Theta}}$ according to equation (A.11).

$$\boldsymbol{V}^E = \dot{\boldsymbol{\Gamma}}^E = \boldsymbol{R}_{\boldsymbol{\Theta}}\ \boldsymbol{V}^B \tag{A.11}$$

As for the linear velocity, it is also possible to relate the angular velocity in the earth frame (or Euler rates) $\dot{\boldsymbol{\Theta}}^E$ $[rad\ s^{-1}]$ to that one in the body-fixed frame $\boldsymbol{\omega}^B$ thanks to the transfer matrix $\boldsymbol{T}_{\boldsymbol{\Theta}}$ $[-]$. Equations (A.12) and (A.13) show the relation specified above.

$$\boldsymbol{\omega}^B = \boldsymbol{T}_{\boldsymbol{\Theta}}{}^{-1}\ \dot{\boldsymbol{\Theta}}^E \tag{A.12}$$

$$\dot{\boldsymbol{\Theta}}^E = \boldsymbol{T}_{\boldsymbol{\Theta}}\ \boldsymbol{\omega}^B \tag{A.13}$$

The transfer matrix $\boldsymbol{T_\Theta}$ can be determined by resolving the Euler rates $\dot{\boldsymbol{\Theta}}^E$ into the body-fixed frame as shown in equations (A.14), (A.15) and (A.16).

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} + \boldsymbol{R}(\phi, x)^{-1} \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + \boldsymbol{R}(\phi, x)^{-1}\boldsymbol{R}(\theta, y)^{-1} \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} = \boldsymbol{T_\Theta}^{-1} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \tag{A.14}$$

$$\boldsymbol{T_\Theta}^{-1} = \begin{bmatrix} 1 & 0 & -s_\theta \\ 0 & c_\phi & c_\theta s_\phi \\ 0 & -s_\phi & c_\theta c_\phi \end{bmatrix} \tag{A.15}$$

$$\boldsymbol{T_\Theta} = \begin{bmatrix} 1 & s_\phi t_\theta & c_\phi t_\theta \\ 0 & c_\phi & -s_\phi \\ 0 & s_\phi / c_\theta & c_\phi / c_\theta \end{bmatrix} \tag{A.16}$$

It is possible to describe equations (A.11) and (A.13) in just one equivalence which relate the derivate of the generalized position in the earth frame $\dot{\boldsymbol{\xi}}$ [+] to the generalized velocity in the body frame $\boldsymbol{\nu}$. The transformation is possible thanks to the generalized matrix $\boldsymbol{J_\Theta}$ [−]. In this matrix, the notation $\mathbf{0}_{3\times3}$ means a sub-matrix with dimension 3 times 3 filled with all zeros. Equations (A.17) and (A.18) show the relation described above.

$$\dot{\boldsymbol{\xi}} = \boldsymbol{J_\Theta}\ \boldsymbol{\nu} \tag{A.17}$$

$$\boldsymbol{J_\Theta} = \begin{bmatrix} \boldsymbol{R_\Theta} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \boldsymbol{T_\Theta} \end{bmatrix} \tag{A.18}$$

A.2 Dynamics

Dynamics is a branch of mechanics which studies the effects of forces and torques on the motion of a body or system of bodies. There are several techniques which can be used to derive the equations of a rigid body with 6 DOF. The Newton-Euler formulation has been adopted in this work.

The equations of motion are more conveniently formulated in a body-fixed because with this notation [33]:

- The inertia matrix is time-invariant.
- Advantage of body symmetry can be taken to simplify the equations.
- Measurements taken on-board are easily converted to body-fixed frame.
- Control forces are almost always given in body-fixed frame.

The decision to describe the equations of motion in the body-fixed frame trades off complexity in the acceleration terms for relative simplicity in the force terms. Two assumptions have been made in this approach:

- The first one states that the origin of the body-fixed frame o_B is coincident with the center of mass (COM) of the body. Otherwise, another point (COM) should be taken into account and thus considerably complicating the body equations.
- The second one specifies that the axes of the B-frame coincide with the body principal axes of inertia. In this case the inertia matrix $\boldsymbol{I}$ is diagonal and, once again, the body equations become easier.

From the Euler's first axiom of the Newton's second law follows the derivation of the linear components of the body motion, according to equation (A.19).

$$\begin{aligned}
m\,\ddot{\boldsymbol{\Gamma}}^E &= \boldsymbol{F}^E \\
m\,\dot{\widehat{\boldsymbol{R}_\Theta \boldsymbol{V}^B}} &= \boldsymbol{R}_\Theta\,\boldsymbol{F}^B \\
m\left(\boldsymbol{R}_\Theta\,\dot{\boldsymbol{V}}^B + \dot{\boldsymbol{R}}_\Theta\,\boldsymbol{V}^B\right) &= \boldsymbol{R}_\Theta\,\boldsymbol{F}^B \\
m\,\boldsymbol{R}_\Theta\left(\dot{\boldsymbol{V}}^B + \boldsymbol{\omega}^B \times \boldsymbol{V}^B\right) &= \boldsymbol{R}_\Theta\,\boldsymbol{F}^B \\
m\left(\dot{\boldsymbol{V}}^B + \boldsymbol{\omega}^B \times \boldsymbol{V}^B\right) &= \boldsymbol{F}^B
\end{aligned} \tag{A.19}$$

Where m $[kg]$ is the quadrotor mass, $\ddot{\boldsymbol{\Gamma}}^E$ $[m\ s^{-2}]$ is the quadrotor linear acceleration vector WRT E-frame, $\boldsymbol{F}^E$ $[N]$ is the quadrotor forces vector WRT E-frame, $\dot{\boldsymbol{V}}^B$ $[m\ s^{-2}]$ is the quadrotor linear acceleration vector WRT B-frame and $\dot{\boldsymbol{R}}_{\boldsymbol{\Theta}}$ $[-]$ is the rotation matrix derivative. Furthermore, the symbol "$\times$" denote the vector product.

Equation (A.20) shows the derivation of the angular components of the body motion from the Euler's second axiom of the Newton's second law.

$$\begin{aligned} \boldsymbol{I}\ \ddot{\boldsymbol{\Theta}}^E &= \boldsymbol{\tau}^E \\ \boldsymbol{I}\ \dot{\widehat{\boldsymbol{T}_{\boldsymbol{\Theta}}\ \boldsymbol{\omega}^B}} &= \boldsymbol{T}_{\boldsymbol{\Theta}}\ \boldsymbol{\tau}^B \\ &\vdots \\ \boldsymbol{I}\ \dot{\boldsymbol{\omega}}^B + \boldsymbol{\omega}^B \times \left(\boldsymbol{I}\ \boldsymbol{\omega}^B\right) &= \boldsymbol{T}_{\boldsymbol{\Theta}}\ \boldsymbol{\tau}^B \end{aligned} \tag{A.20}$$

In equation (A.20) $\boldsymbol{I}$ $[N\ m\ s^2]$ is the body inertia matrix (in the body-fixed frame), $\ddot{\boldsymbol{\Theta}}^E$ $[rad\ s^{-2}]$ is the quadrotor angular acceleration vector WRT E-frame, $\dot{\boldsymbol{\omega}}^B$ $[rad\ s^{-2}]$ is the quadrotor angular acceleration vector WRT B-frame and $\boldsymbol{\tau}^E$ $[N\ m]$ is the quadrotor torques vector WRT E-frame.

By putting together equations (A.19) and (A.20), it is possible to describe the motion of a 6 DOF rigid body. Equation (A.21) shows a matrix formulation of the dynamics.

$$\begin{bmatrix} m\ \boldsymbol{I}_{3\times3} & \boldsymbol{0}_{3\times3} \\ \boldsymbol{0}_{3\times3} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{V}}^B \\ \dot{\boldsymbol{\omega}}^B \end{bmatrix} + \begin{bmatrix} \boldsymbol{\omega}^B \times \left(m\ \boldsymbol{V}^B\right) \\ \boldsymbol{\omega}^B \times \left(\boldsymbol{I}\ \boldsymbol{\omega}^B\right) \end{bmatrix} = \begin{bmatrix} \boldsymbol{F}^B \\ \boldsymbol{\tau}^B \end{bmatrix} \tag{A.21}$$

Where the notation $\boldsymbol{I}_{3\times3}$ means an identity matrix with dimension 3 times 3. In addition, it's easy to see that the first matrix in equation (A.21) is diagonal and constant. This equation is totally generic and is valid for all the rigid body which obey to the hypothesis (or simplifications) previously done. However, it was used in this work to model the quadrotor helicopter, hence the last vector contains specific information about its dynamics. Chapter 3 provides the derivation of the specific dynamic model taking into account the forces and torques in play.

Appendix B

Linear regression

Linear regression is a method in which data are fitted with a straight line according to the ordinary least squares. The procedure minimizes the sum of the squared residual error obtained from the difference between the experimental data and the linearized function [47]. This linear relation can be defined according to equation (B.1).

$$y = a + b\ x \tag{B.1}$$

Where a and b are two constants identified by the regression. x is the independent variable while y is the dependent variable. Since this appendix explains the theory of linear regression, all its variables and constants are not reported in the list of variables (appendix I) and in the list of constants (appendix H) because they are not directly related to other sections. Therefore they are just initialized and used in this appendix as "stand alone" quantities.

The i-th residual error ϵ_i can be defined according to (B.2).

$$\epsilon_i = y_i - (a + b\ x_i) \tag{B.2}$$

x_i is the i-th element of the independent variable while y_i is the i-th element of the dependent variable.

The constants a and b are those which provide the lowest value of the function $\sum_{i=1}^{N} {\epsilon_i}^2$ (N is the number of data acquired). The sum of the squared residual error Φ is hence a function of a and b as shown in equation (B.3).

$$\Phi = \sum_{i=0}^{N} {\epsilon_i}^2 = \sum_{i=0}^{N} (y_i - a - b\ x_i)^2 \tag{B.3}$$

To find the minimum of equation (B.3), the derivate of Φ respect to a and b must be zero (and its second derivate must be grater than zero). Equations (B.4) and (B.5) show the previous statement.

$$\frac{\partial \Phi}{\partial a} = -2 \sum_{i=0}^{N} (y_i - a - b\, x_i) = 0 \Rightarrow a\, N + b \sum_{i=0}^{N} x_i = \sum_{i=0}^{N} y_i \tag{B.4}$$

$$\frac{\partial \Phi}{\partial b} = -2 \sum_{i=0}^{N} (y_i - a - b\, x_i)\, x_i = 0 \Rightarrow a \sum_{i=0}^{N} x_i + b \sum_{i=0}^{N} {x_i}^2 = \sum_{i=0}^{N} x_i\, y_i \tag{B.5}$$

The linear system of two equations and two variables can be rearranged to find a and b, as shown in equations (B.6) and (B.7).

$$b = \frac{N \sum_i x_i\, y_i - \sum_i x_i \sum_i y_i}{N \sum_i {x_i}^2 - \left(\sum_i x_i\right)^2} \tag{B.6}$$

$$a = \frac{\sum_i y_i \sum_i {x_i}^2 - \sum_i x_i \sum_i x_i\, y_i}{N \sum_i {x_i}^2 - \left(\sum_i x_i\right)^2} \tag{B.7}$$

It can be demonstrated (neglected in this appendix) that this solution really corresponds to a minimum of Φ.

Appendix C

Aerodynamics calculation

The aerodynamics calculation describes the behavior of the rotating propellers in the air. In particular two quantities are important to determine: the thrust and the drag coefficients. They consider the aerodynamic effects and report them in the quadrotor dynamic model.
The computations can be done according the combined momentum and blade element theory which merges the concepts of two different analysis [48]:

- momentum theory (section C.1)
- blade element theory (section C.2)

C.1 Momentum theory

The rotor is modeled as a disk which provides energy to the air, receiving a net force (as counter reaction). The hypotheses of this theory are:

- A flux tube of air crossing the disk is considered without interaction from external air
- The propeller has an infinite number of blades
- the thickness of the disk is an infinitesimal quantity
- the vertical speed of the air is continuous through the disk
- the air is a perfect gas and is incompressible

In this model, T_{MT} $[N]$ is the thrust of the propeller (with MT) and points upwards. The air speeds are defined with respect to the rotor: $v_{-\infty}$ $[m\ s^{-1}]$ is asymptotically over, v_1 $[m\ s^{-1}]$ is directly over, v_2 $[m\ s^{-1}]$ is directly under $v_{+\infty}$ $[m\ s^{-1}]$ is asymptotically under. Similarly, also the air pressures are defined with respect to the rotor: $p_{-\infty}$ $[Pa]$ is asymptotically over, p_1 $[Pa]$ is directly over, p_2 $[Pa]$ is directly under and $p_{+\infty}$ $[Pa]$ is asymptotically under. Figure C.1 shows the sketch of the momentum theory model

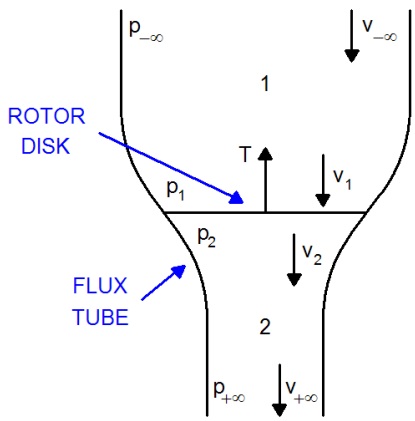


Figure C.1: *Momentum theory*

The thrust supplied by the rotor is proportional to the pressure difference through the disk. It can be also defined as the variation of the air momentum through the disk. Equations (C.1) and (C.2) show the equivalences.

$$T_{MT} = A\ (p_1 - p_2) \tag{C.1}$$

$$T_{MT} = \dot{m}_A\ (v_{-\infty} - v_{+\infty}) = \rho_A\ A\ v_1\ (v_{-\infty} - v_{+\infty}) \tag{C.2}$$

In the previous equations, A $[m^2]$ is the area of the disk ($\pi\ R_P{}^2$), $\dot{m}_A$ $[kg\ s^{-}1]$ is the variation of the air mass through the disk and ρ_A $[kg\ m^{-3}]$ is the air density. According to the hypothesis, the air speed directly over the rotor v_1 is equal to that one directly under the rotor v_2. It is possible to write the Bernoulli equation between the sections $-\infty$ and 1 and between the sections 2 and $+\infty$, as shown in the equations (C.3) and (C.4).

$$p_{-\infty} + \frac{1}{2}\ \rho_A\ v_{-\infty}{}^2 = p_1 + \frac{1}{2}\ \rho_A\ v_1{}^2 \tag{C.3}$$

$$p_2 + \frac{1}{2}\ \rho_A\ v_2{}^2 = p_{+\infty} + \frac{1}{2}\ \rho_A\ v_{-\infty}{}^2 \tag{C.4}$$

Rearranging equations (C.3) and (C.4), considering also that $p_{+\infty} = p_{-\infty}$, the speed v_1 can be rewritten as:

$$v_1 = \frac{v_{+\infty} + v_{-\infty}}{2} \tag{C.5}$$

the inflow speed at the rotor disk v_I $[m\ s^{-1}]$ is

$$v_I = v_1 - v_{-\infty} = \frac{v_{+\infty} - v_{-\infty}}{2} \tag{C.6}$$

It is possible now to define the thrust, considering equations (C.2) and (C.6):

$$T_{MT} = 2\ \rho_A\ A\ v_1\ v_I \tag{C.7}$$

To simplify the calculus, the hover condition was taken into account [6]. In this condition $v_{-\infty} = 0$, thus $v_1 = v_I$. Furthermore the thrust T_{MT} is equal to the weight carried from one propeller W_P $[N]$ ($W_P = \frac{m\ g}{4}$). Equation (C.7) can be reorganized as:

$$W_P = 2\ \rho_A\ A\ {v_I}^2 \quad \Rightarrow \quad v_I = \sqrt{\frac{W_P}{2\ \rho_A\ A}} = 3.68\ [m\ s^{-1}] \tag{C.8}$$

The inflow ratio λ $[-]$ is a quantity used to relate the inflow speed to the rotor tip speed.

$$\lambda = \frac{v_I}{\omega_H\ R_P} = 0.11\ [-] \tag{C.9}$$

Where ω_H $[rad\ s^{-1}]$ is the angular speed of the blade in hovering and R_P $[m]$ is its radius. Since the hovering condition implies zero horizontal speed, the rotor advance ratio $\mu[-]$ is also equal to zero.

C.2 Blade element theory

Momentum theory provides important information on the rotor performance, but its level of detail is quite poor and gives limited analysis. Thus the total aerodynamic forces and torques acting on a rotor are determined by using the blade element theory combined with a few momentum theory concepts. With this approach, the forces and torques are calculated by integrating the individual forces acting on smaller blade elements over the entire rotor. Figure C.2 shows the model of the propeller section.

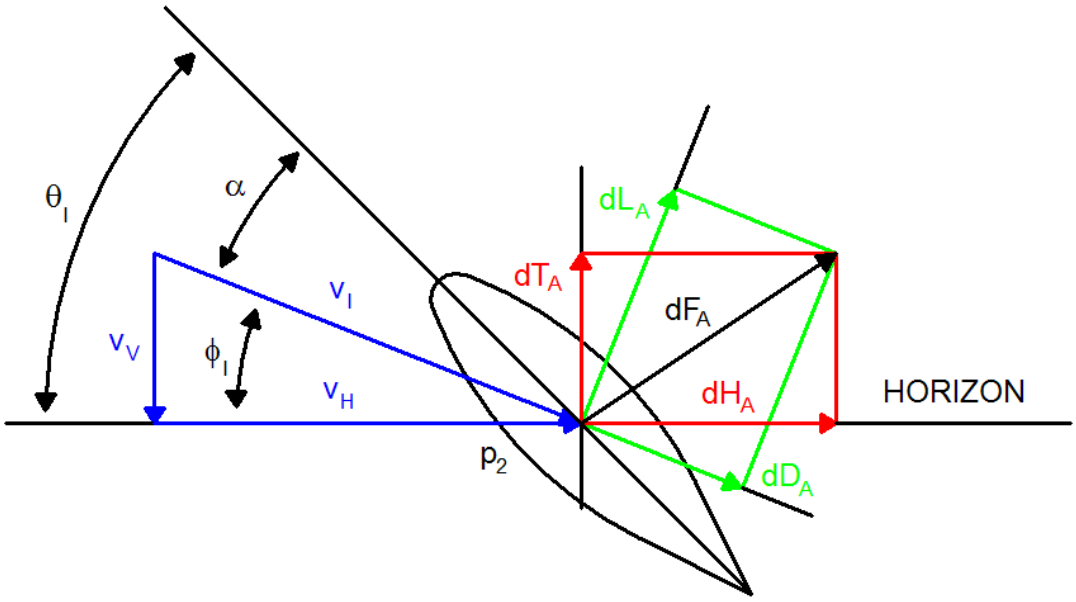


Figure C.2: *Blade element theory*

The "horizon" line is perpendicular to the rotor shaft (hovering condition). θ_I $[rad]$ is the angle of incidence between the horizon line and the blade chord line. α $[rad]$ is the angle of attack between the blade chord line and the local air flow velocity vector v_T $[m\ s^{-1}]$. v_T is the vector sum of the horizontal v_H $[m\ s^{-1}]$ and vertical v_V $[m\ s^{-1}]$ air flow velocity. The angle between the horizon and the local velocity vector is the local inflow angle ϕ_I $[rad]$. dD_{BET} $[N\ m^{-1}]$ is the infinitesimal drag force while dL_{BET} $[N\ m^{-1}]$ is the infinitesimal lift force. The vector sum of dD_{BET} and dL_{BET} is the infinitesimal aerodynamic resultant force dF_{BET} $[N\ m^{-1}]$. dF_{BET} can be also be divided into two infinitesimal aerodynamic vertical dT_{BET} $[N\ m^{-1}]$ and horizontal dH_{BET} $[N\ m^{-1}]$ components.

The v_V velocity vector is due to the inflow motion of the rotor, thus, it is uniform for every section. On the other hand, the v_H velocity vector is due to the angular speed of the blade element, hence, it depends on the radial position of the section r $[m]$. The previous statements are simplified according to the hovering hypothesis. Equations (C.10) and (C.11) shows the velocity components.

$$v_V = v_I = \omega_P \; R_P \; \lambda \tag{C.10}$$

$$v_H = \omega_P \; r = \omega_P \; R_P \; \left(\frac{r}{R_P}\right) \tag{C.11}$$

Where ω_P $[rad \; s^{-1}]$ is the propeller (angular) speed. To continue with the thrust and torque calculus, a few things have be taken into consideration. The infinitesimal lift dL_{BET} and drag dD_{BET} forces, according to aerodynamics theory [48], are determined by equations (C.12) and (C.13).

$$dL_{BET} = \frac{1}{2} \; \rho_A \; {v_H}^2 \; C_L \; c \; dr \tag{C.12}$$

$$dD_{BET} = \frac{1}{2} \; \rho_A \; {v_H}^2 \; C_D \; c \; dr \tag{C.13}$$

Where C_L $[-]$ is the lift coefficient, C_D $[-]$ is the drag coefficient (estimated from [40] as 0.05) and c $[m]$ is the average chord of the blade. The coefficient of lift C_L varies linearly with the angle of attack through the lift slope a $[rad^{-1}]$. For thin airfoils, at small angles of attack, the parameter a is equal to 2π $[rad^{-1}]$ (according [38]).

$$C_L = a \; \alpha = a \; (\theta_I - \phi_I) \tag{C.14}$$

The blade twist is assumed to vary linearly with radial position. Thus the model includes the two constants zero angle of incidence θ_{I_0} $[rad]$ and twist angle of incidence $\theta_{I_{tw}}$ $[rad]$, according to equation (C.15).

$$\theta_I = \theta_{I_0} - \theta_{I_{tw}} \; \frac{r}{R_P} \tag{C.15}$$

Furthermore the angular velocity of the blade section is much larger than the total inflow through the blade. Then, small angle approximations can be used to define the inflow angle ϕ_I.

$$\phi_I = \frac{v_V}{v_H} \tag{C.16}$$

According to equations (C.12), (C.14), (C.15) and (C.16); the infinitesimal lift force dL_{BET} can be rewritten as shown in equation (C.17).

$$dL_{BET} = \frac{1}{2} \; \rho_A \; {v_H}^2 \; a \; \left(\theta_{I_0} - \theta_{I_{tw}} \; \frac{r}{R_P} - \frac{v_V}{v_H}\right) \; c \; dr \tag{C.17}$$

The infinitesimal vertical force dT_{BET} can be simplified considering the previous small angle approximations and that the lift acting on the blade is about an order of magnitude higher than the drag. Equation (C.18) shows the approximation.

$$dT_{BET} = dL_{BET}\ \cos\phi_I - dD_{BET}\ \sin\phi_I \approx dL_{BET} \tag{C.18}$$

The thrust (with BET) T_{BET} $[N]$ is finally found by integrating the vertical forces acting on all the blade element sections. The constant N_B $[-]$ identifies the number of blades per propeller ($N_B = 2$).

$$\begin{aligned}
T_{BET} &= N_B \int_0^{R_P} \frac{dT_{BET}}{dr}\ dr = \\
&= \frac{N_B\ \rho_A\ a\ c}{2} \int_0^{R_P} \left(\theta_{I_0}\ {v_H}^2 - \frac{\theta_{I_{tw}}\ r}{R_P}\ {v_H}^2 - v_V\ v_H \right) dr = \\
&= \frac{N_B\ \rho_A\ a\ c\ {\omega_P}^2\ {R_P}^2}{2} \int_0^{R_P} \left(\frac{\theta_{I_0}}{{R_P}^2}\ r^2 - \frac{\theta_{I_{tw}}}{{R_P}^3}\ r^3 - \frac{\lambda}{R_P}\ r \right) dr = \\
&= N_B\ \rho_A\ a\ c\ {\omega_P}^2\ {R_P}^3\ \left(\frac{\theta_{I_0}}{6} - \frac{\theta_{I_{tw}}}{8} - \frac{\lambda}{4} \right)
\end{aligned} \tag{C.19}$$

The infinitesimal horizontal force dH_{BET} can be simplified considering the previous small angle approximations, but in this case, both terms are significant (same order of magnitude). Equation (C.20) shows the approximation.

$$dH_{BET} = dD_{BET}\ \cos\phi_I + dL_{BET}\ \sin\phi_I \approx dD_{BET} + dL_{BET}\ \left(\frac{v_V}{v_H} \right) \tag{C.20}$$

The propeller torque Q_{BET} $[N\ m]$ is found by integrating the horizontal forces multiplied by the moment arm on all the blade element sections.

$$\begin{aligned}
Q_{BET} &= N_B \int_0^{R_P} \left(\frac{dD_{BET}}{dr} + \frac{dL_{BET}}{dr}\ \frac{v_V}{v_H} \right)\ r\ dr = \\
&= \frac{N_B\ \rho_A\ c}{2} \int_0^{R_P} \big({v_H}^2\ C_D\ r + v_V\ v_H\ a\ \theta_{I_0}\ r + \\
&- v_V\ v_H\ a\ \frac{\theta_{I_{tw}}}{R_P}\ r^2 - {v_V}^2\ a\ r \Big) dr = \\
&= \frac{N_B \rho_A c {\omega_P}^2 {R_P}^2}{2} \int_0^{R_P} \left(\frac{C_D}{{R_P}^2} r^3 + a\lambda \left(\frac{\theta_{I_0}}{R_P} r^2 - \frac{\theta_{I_{tw}}}{{R_P}^2} r^3 - \lambda r \right) \right) dr = \\
&= N_B\ \rho_A\ c\ {\omega_P}^2\ {R_P}^4\ \left(\frac{C_D}{8} + a\ \lambda \left(\frac{\theta_{I_0}}{6} - \frac{\theta_{I_{tw}}}{8} - \frac{\lambda}{4} \right) \right)
\end{aligned} \tag{C.21}$$

Appendix D

Moment of inertia calculation

The moment of inertia describes the dynamic behavior of a body in rotation around a defined axis. It has the same role in rotational dynamics as mass does in basic dynamics. The moment of inertia can be divided in two quantities [39]:

- scalar moment of inertia
- moment of inertia tensor

The scalar moment of inertia is defined as:

$$\boldsymbol{I} = \iiint_V \rho \; r^2 dV \tag{D.1}$$

Where $\boldsymbol{I}$ $[kg\ m^2]$ is the moment of inertia, V $[m^3]$ is the volume occupied by the object, ρ $[kg\ m^{-3}]$ is the spatial density function of the object and r $[m]$ is the (perpendicular) distance of the considered point to the axis of rotation. Since this appendix explain the theory of the moment of inertia, all its variables and constants are not reported in the list of variables (appendix I) and in the list of constants (appendix H) because they are not related to other sections. Therefore they are just initialized and used in this appendix as "stand alone" quantities.

For the same object, different axes of rotation can have different moments of inertia about those axes. The moment of inertia tensor is a convenient way to summarize all moments of inertia of an object with one quantity. The moment of inertia tensor is defined as:

$$\boldsymbol{I} = \begin{bmatrix} I_{XX} & I_{XY} & I_{XZ} \\ I_{YX} & I_{YY} & I_{YZ} \\ I_{ZX} & I_{ZY} & I_{ZZ} \end{bmatrix} \tag{D.2}$$

Here I_{XX} denotes the moment of inertia around the x-axis when the objects are rotated around the x-axis, I_{XY} denotes the moment of inertia around the y-axis when the objects are rotated around the x-axis, and so on.

The following pages show a list of the geometries and their moments of inertia calculus.

- Solid rectangular prism with length L, width W, height H, mass M and a constant density $\rho = \frac{M}{L\ W\ H}$. Figure D.1 shows the geometry.

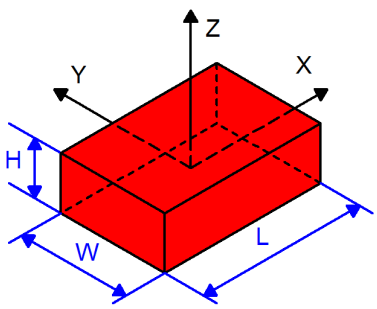


Figure D.1: *Solid rectangular prism*

$$
\begin{aligned}
I_X &= \int_z \int_y \int_x \rho\ \left(y^2 + z^2\right) dx\ dy\ dz = \\
&= \rho \int_{-\frac{H}{2}}^{\frac{H}{2}} \int_{-\frac{W}{2}}^{\frac{W}{2}} \int_{-\frac{L}{2}}^{\frac{L}{2}} \left(y^2 + z^2\right) dx\ dy\ dz = \\
&= \rho \left(L\ H \int_{-\frac{W}{2}}^{\frac{W}{2}} y^2 dy + L\ W \int_{-\frac{H}{2}}^{\frac{H}{2}} z^2 dz \right) = \\
&= \frac{M}{L\ W\ H} \left(L\ H\ \frac{W^3}{12} + L\ W\ \frac{H^3}{12} \right) = \\
&= M \left(\frac{W^2}{12} + \frac{H^2}{12} \right) \qquad \text{(D.3)} \\
I_Y &= \int_z \int_y \int_x \rho\ \left(x^2 + z^2\right) dx\ dy\ dz = \\
&= \rho \int_{-\frac{H}{2}}^{\frac{H}{2}} \int_{-\frac{W}{2}}^{\frac{W}{2}} \int_{-\frac{L}{2}}^{\frac{L}{2}} \left(x^2 + z^2\right) dx\ dy\ dz = \\
&= \ldots = M \left(\frac{L^2}{12} + \frac{H^2}{12} \right) \qquad \text{(D.4)} \\
I_Z &= \int_z \int_y \int_x \rho\ \left(x^2 + y^2\right) dx\ dy\ dz = \\
&= \rho \int_{-\frac{H}{2}}^{\frac{H}{2}} \int_{-\frac{W}{2}}^{\frac{W}{2}} \int_{-\frac{L}{2}}^{\frac{L}{2}} \left(x^2 + y^2\right) dx\ dy\ dz = \\
&= \ldots = M \left(\frac{L^2}{12} + \frac{W^2}{12} \right) \qquad \text{(D.5)}
\end{aligned}
$$

- Solid cylinder with radius R, height H, mass M and a constant density $\rho = \frac{M}{\pi\ R^2\ H}$. Figure D.2 shows the geometry.

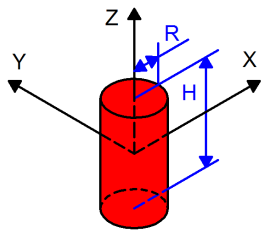


Figure D.2: *Solid cylinder*

$$
\begin{aligned}
I_X &= \int_z \int_y \int_x \rho\ \left(y^2 + z^2\right) dx\ dy\ dz = \\
&= \rho \int_{-\frac{H}{2}}^{\frac{H}{2}} \int_{-\frac{R}{2}}^{\frac{R}{2}} \int_{-\sqrt{\left(\frac{R}{2}\right)^2 - y^2}}^{\sqrt{\left(\frac{R}{2}\right)^2 - y^2}} \left(y^2 + z^2\right) dx\ dy\ dz = \\
&= \rho \int_{-\frac{H}{2}}^{\frac{H}{2}} \int_0^{2\pi} \int_0^R \left(r^2\ \sin^2\theta + z^2\right)\ r\ dr\ d\theta\ dz = \\
&= \rho \left(H \int_0^R r^3 dr \int_0^{2\pi} \sin^2\theta d\theta + 2\pi \int_0^R r dr \int_{-\frac{H}{2}}^{\frac{H}{2}} z^2 dz \right) = \\
&= \frac{M}{\pi\ R^2\ H} \left(H\ \frac{R^4}{4}\ \pi + 2\pi\ \frac{R^2}{2}\ \frac{H^3}{12} \right) = \\
&= M\ \left(\frac{R^2}{4} + \frac{H^2}{12} \right) \qquad \text{(D.6)} \\
I_Y &= \int_z \int_y \int_x \rho\ \left(x^2 + z^2\right) dx\ dy\ dz = \ldots = \\
&= \rho \int_{-\frac{H}{2}}^{\frac{H}{2}} \int_0^{2\pi} \int_0^R \left(r^2\ \cos^2\theta + z^2\right)\ r\ dr\ d\theta\ dz = \\
&= \ldots = M\ \left(\frac{R^2}{4} + \frac{H^2}{12} \right) \qquad \text{(D.7)} \\
I_Z &= \int_z \int_y \int_x \rho\ \left(x^2 + y^2\right) dx\ dy\ dz = \ldots = \\
&= \rho \int_{-\frac{H}{2}}^{\frac{H}{2}} \int_0^{2\pi} \int_0^R \left(r^2\ \cos^2\theta + r^2\ \sin^2\theta\right)\ r\ dr\ d\theta\ dz = \\
&= \ldots = M\ \left(\frac{R^2}{2} \right) \qquad \text{(D.8)}
\end{aligned}
$$

- Solid cylinder with variable density with radius R, height H, mass M and a density $\rho = \frac{M}{2\pi\ R\ H\ r}$. The geometry is the same as in figure D.2.

$$
\begin{aligned}
I_X &= \int_z \int_y \int_x \rho\ \left(y^2 + z^2\right) dx\ dy\ dz = \\
&= \frac{M}{2\pi\ R\ H} \int_{-\frac{H}{2}}^{\frac{H}{2}} \int_{-\frac{R}{2}}^{\frac{R}{2}} \int_{-\sqrt{\left(\frac{R}{2}\right)^2 - y^2}}^{\sqrt{\left(\frac{R}{2}\right)^2 - y^2}} \frac{1}{r}\ \left(y^2 + z^2\right) dx\ dy\ dz = \\
&= \frac{M}{2\pi\ R\ H} \int_{-\frac{H}{2}}^{\frac{H}{2}} \int_0^{2\pi} \int_0^R \frac{1}{r}\ \left(r^2\ \sin^2\theta + z^2\right)\ r\ dr\ d\theta\ dz = \\
&= \frac{M}{2\pi\ R\ H} \left(H \int_0^R r^2 dr \int_0^{2\pi} \sin^2\theta d\theta + 2\pi\ R \int_{-\frac{H}{2}}^{\frac{H}{2}} z^2 dz \right) = \\
&= \frac{M}{2\pi\ R\ H} \left(H\ \frac{R^3}{3}\ \pi + 2\pi\ R\ \frac{H^3}{12} \right) = \\
&= M\ \left(\frac{R^2}{6} + \frac{H^2}{12} \right) \qquad \text{(D.9)} \\
I_Y &= \int_z \int_y \int_x \rho\ \left(x^2 + z^2\right) dx\ dy\ dz = \ldots = \\
&= \frac{M}{2\pi\ R\ H} \int_{-\frac{H}{2}}^{\frac{H}{2}} \int_0^{2\pi} \int_0^R \frac{1}{r} \left(r^2\ \cos^2\theta + z^2\right)\ r\ dr\ d\theta\ dz = \\
&= \ldots = M\ \left(\frac{R^2}{6} + \frac{H^2}{12} \right) \qquad \text{(D.10)} \\
I_Z &= \int_z \int_y \int_x \rho\ \left(x^2 + y^2\right) dx\ dy\ dz = \ldots = \\
&= \frac{M}{2\pi\ R\ H} \int_{-\frac{H}{2}}^{\frac{H}{2}} \int_0^{2\pi} \int_0^R \frac{1}{r} \left(r^2\ \cos^2\theta + r^2\ \sin^2\theta\right)\ r\ dr\ d\theta\ dz = \\
&= \ldots = M\ \left(\frac{R^2}{3} \right) \qquad \text{(D.11)}
\end{aligned}
$$

If the center of mass of the object is not equal to the axes origin (assumed as COM of the quadrotor), there is a theorem (parallel axes theorem) which helps to determine the moments of inertia without recalculating the integrals. The theorem states that if the object COM axis of rotation is displaced by a distance D from the axis of rotation, the displaced and the center moments of inertia are related according to equation (D.12).

$$I_{displaced} = I_{center} + M\ D^2 \tag{D.12}$$

It follows an example with a solid rectangular prism displaced from the axes origin. It has length L, width W, height H, distance from the COM around the x-axis D_X, distance from the COM around the y-axis D_Y, distance from the COM around the z-axis D_Z, mass M and a constant density $\rho = \frac{M}{L\ W\ H}$. Figure D.3 shows the geometry.

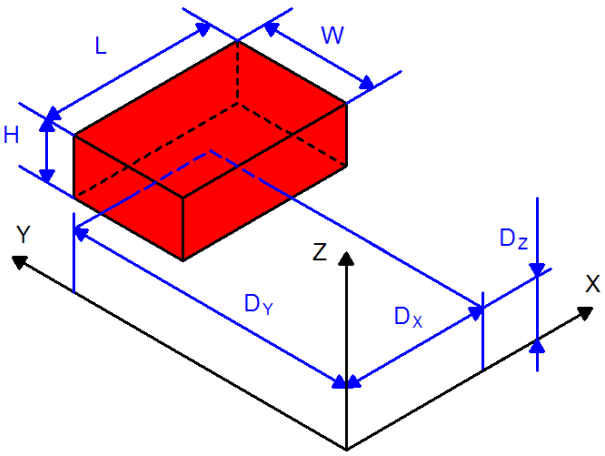


Figure D.3: *Solid rectangular prism displaced*

$$
\begin{aligned}
I_X &= \int_z \int_y \int_x \rho \ \left(y^2 + z^2\right) dx \ dy \ dz = \\
&= \rho \int_{D_Z - \frac{H}{2}}^{D_Z + \frac{H}{2}} \int_{D_Y - \frac{W}{2}}^{D_Y + \frac{W}{2}} \int_{D_X - \frac{L}{2}}^{D_X + \frac{L}{2}} \left(y^2 + z^2\right) dx \ dy \ dz = \\
&= \rho \left(L \ H \int_{D_Y - \frac{W}{2}}^{D_Y + \frac{W}{2}} y^2 dy + L \ W \int_{D_Z - \frac{H}{2}}^{D_Z + \frac{H}{2}} z^2 dz \right) = \\
&= \frac{M}{L \ W \ H} \left(L \ H \left(\frac{W^3}{12} + W \ {D_Y}^2 \right) + L \ W \left(\frac{H^3}{12} + H \ {D_Z}^2 \right) \right) = \\
&= M \left(\frac{W^2}{12} + \frac{H^2}{12} \right) + M \ \left({D_Y}^2 + {D_Z}^2 \right) \qquad \text{(D.13)} \\
I_Y &= \int_z \int_y \int_x \rho \ \left(x^2 + z^2\right) dx \ dy \ dz = \\
&= \rho \int_{D_Z - \frac{H}{2}}^{D_Z + \frac{H}{2}} \int_{D_Y - \frac{W}{2}}^{D_Y + \frac{W}{2}} \int_{D_X - \frac{L}{2}}^{D_X + \frac{L}{2}} \left(x^2 + z^2\right) dx \ dy \ dz = \\
&= \ldots = M \left(\frac{L^2}{12} + \frac{H^2}{12} \right) + M \ \left({D_X}^2 + {D_Z}^2 \right) \qquad \text{(D.14)} \\
I_Z &= \int_z \int_y \int_x \rho \ \left(x^2 + y^2\right) dx \ dy \ dz = \\
&= \rho \int_{D_Z - \frac{H}{2}}^{D_Z + \frac{H}{2}} \int_{D_Y - \frac{W}{2}}^{D_Y + \frac{W}{2}} \int_{D_X - \frac{L}{2}}^{D_X + \frac{L}{2}} \left(x^2 + y^2\right) dx \ dy \ dz = \\
&= \ldots = M \left(\frac{L^2}{12} + \frac{W^2}{12} \right) + M \ \left({D_X}^2 + {D_Y}^2 \right) \qquad \text{(D.15)}
\end{aligned}
$$

By comparing equations (D.13), (D.14) and (D.15) with equations (D.3), (D.4) and (D.5) it's easy to recognize the therm given by the displacement of the object COM from the axis origin.

Appendix E

Acronyms and abbreviations

Acronym	Description
AC	Alternated current
ADC	Analog to Digital Converter
BEMF	Back Electro-Motive Force
BET	Blade Element Theory
BSC	BaSiC
BSPI	Buffered Serial Peripheral Interfaces
CAN	Controller Area Network
COM	Center Of Mass
CPU	Central Processing Unit
DAT	DATasheet
DC	Direct Current
DMA	Direct Memory Access
DOF	Degrees Of Freedom
DSP	Digital Signal Processor

EDA	Electronic Design Automation
Eng.	Engineer
GRP	GRaPhic method
GSM	Global Positioning System
I^2C	Inter-Integrated Circuit
IEEE	Institute of Electrical and Electronic Engineers
IMU	Inertial Measurement Unit
IR	InfraRed
IRQ	Interrupt ReQuest
JTAG	Joint Test Action Group
LED	Light-Emitting Diode
LQR	Linear Quadratic Regulator
LR1	Linear Regression 1
LR2	Linear Regression 2
MCU	Micro Controller Unit
MEMS	Micro Electro-Mechanical System
MIPS	Million Instruction Per Second
MOSFET	Metal-Oxide-Semiconductor Field-Effect Transistor
MT	Momentum Theory
NA	Not Available
Prof.	Professor
PC	Personal Computer
PCB	Printed Circuit Board
PSD	Position-Sensitive Device

PWM	Pulse Width Modulation
RAM	Random Access Memory
RC	Remote Controller
RF	Radio Frequency
RISC	Reduced Instruction Set Computer
RMS	Root Mean Square
ROM	Read Only Memory
UAV	Unmanned Aerial Vehicle
SONAR	SOund Navigation And Ranging
UART	Universal Asynchronous Receiver Transmitter
USB	Universal Serial Bus
VTOL	Vertical Take-Off and Landing
WRT	With Respect To
XKF	Xsens Kalman Filter
μC	MICRO Controller

Appendix F

Units of measurement

Symbol	Name	Quantity	Equivalence
°C	Celsius	Celsius temperature	$K - 273$
bit	binary digit	information's unit	bit
deg	degree	angle	$rad \times 180 \times \pi^{-1}$
g	gram	mass	$kg \times 10^{-3}$
kg	kilogram	mass	kg
$kg\ m^{-3}$	–	spatial density	$kg\ m^{-3}$
m	meter	length	m
m^3	–	volume	m^3
$m\ s^{-1}$	–	velocity	$m\ s^{-1}$
$m\ s^{-2}$	–	acceleration	$m\ s^{-2}$
rad	radian	angle	rad
$rad\ s^{-1}$	–	angular velocity	$rad\ s^{-1}$
$rad\ s^{-2}$	–	angular acceleration	$rad\ s^{-2}$
s	second	time	s

Symbol	Name	Quantity	Equivalence
A	ampere	electric current	A
C	coulomb	electric charge	$A\ s$
F	farad	electric capacitance	$C\ V^{-1} = A^2\ s^4\ kg^{-1}\ m^{-2}$
H	henry	inductance	$V\ s\ A^{-1} = kg\ m^2\ s^{-2}\ A^{-2}$
Hz	hertz	frequency	s^{-1}
J	joule	energy, work	$N\ m = C\ V = kg\ m^2\ s^{-2}$
K	kelvin	temperature	K
N	newton	force	$kg\ m\ s^{-2}$
$N\ m$	–	torque	$kg\ m^2\ s^{-2}$
$N\ m\ s^2$	–	inertia	$kg\ m^2$
Pa	pascal	pressure	$N\ m^{-2} = kg\ m^{-1}\ s^{-2}$
T	tesla	magnetic flux density	$V\ s\ m^{-2} = kg\ A^{-1}\ s^{-2}$
V	volt	voltage	$kg\ m^2\ s^{-3}\ A^{-1}$
W	watt	electric power	$V\ A = J\ s^{-1} = N\ m\ s^{-1}$
			$= kg\ m^2\ s^{-3}$
Ω	ohm	resistance	$V\ A^{-1} = kg\ m^2\ s^{-3}\ A^{-2}$

Appendix G

List of prefixes

Factor	Name	Symbol	Equivalence
10^{24}	yotta-	Y	1 000 000 000 000 000 000 000 000
10^{21}	zetta-	Z	1 000 000 000 000 000 000 000
10^{18}	exa-	E	1 000 000 000 000 000 000
10^{15}	peta-	P	1 000 000 000 000 000
10^{12}	tera-	T	1 000 000 000 000
10^{9}	giga-	G	1 000 000 000
10^{6}	mega-	M	1 000 000
10^{3}	kilo-	k	1 000
10^{2}	hecto-	h	100
10^{1}	deca-	da	10
10^{-1}	deci-	d	0.1
10^{-2}	centi-	c	0.01
10^{-3}	milli-	m	0.001
10^{-6}	micro-	μ	0.000 001
10^{-9}	nano-	n	0.000 000 001
10^{-12}	pico-	p	0.000 000 000 001
10^{-15}	femto-	f	0.000 000 000 000 001
10^{-18}	atto-	a	0.000 000 000 000 000 001
10^{-21}	zepto-	z	0.000 000 000 000 000 000 001
10^{-24}	yocto-	y	0.000 000 000 000 000 000 000 001

Appendix H

List of constants

Symbol	Unit	Value	Description
$\mathbf{0}_{j\times k}$	$-$	see p. 14	zero matrix with j rows and k columns
a	rad^{-1}	2π	lift slope
b	$N\ s^2$	54.2×10^{-6}	thrust factor
b_{BET}	$N\ s^2$	53.8×10^{-6}	thrust factor with BET
b_{GRP}	$N\ s^2$	55.6×10^{-6}	thrust factor with GRP
b_{MT}	$N\ s^2$	53.1×10^{-6}	thrust factor with MT
c	m	0.02	average chord of the blade
d	$N\ m\ s^2$	1.1×10^{-6}	drag factor
g	$m\ s^{-2}$	9.81	acceleration due to gravity
l	m	0.24	center of quadrotor to center of propeller distance
m	kg	1	mass of the quadrotor
v_{RT}	V	9.6	motor rated voltage
A	m^2	75.5×10^{-3}	propeller area
A_P	$rad\ s^{-1}$	-22.7	linearized propeller's speed coefficient
B_P	$rad^2\ s^{-2}\ V^{-1}$	514	linearized input voltage coefficient
C_D	$-$	0.05	drag coefficient
C_P	$rad^2\ s^{-2}$	494	linearized constant coefficient
D_E	m	70×10^{-3}	electronics box to COM distance around the z-axis
D_M	m	12×10^{-3}	motor to COM distance around the z-axis

Symbol	Unit	Value	Description
D_{MG}	m	10.6×10^{-3}	motor gear to COM distance around the z-axis
D_P	m	20×10^{-3}	propeller to COM distance around the z-axis
D_{PG}	m	10.6×10^{-3}	propeller gear to COM distance around the z-axis
$\boldsymbol{E}_B$	$+$	see p. 17	movement matrix WRT E-frame
$\boldsymbol{F}_G^E$	N	see p. 15	gravitational force vector WRT E-frame
$\boldsymbol{G}_H$	$+$	see p. 20	gravitational vector WRT H-frame
H_E	m	20×10^{-3}	electronics box height
H_M	m	30×10^{-3}	motor height
H_{MG}	m	5.2×10^{-3}	motor gear height
H_P	m	2×10^{-3}	propeller height
H_{PG}	m	4.5×10^{-3}	propeller gear height
I_{CX}	$N\ m\ s^2$	460×10^{-6}	cross structure moment of inertia around the x-axis
I_{CY}	$N\ m\ s^2$	460×10^{-6}	cross structure moment of inertia around the y-axis
I_{CZ}	$N\ m\ s^2$	920×10^{-6}	cross structure moment of inertia around the z-axis
I_{EX}	$N\ m\ s^2$	1.7×10^{-3}	electronics box moment of inertia around the x-axis
I_{EY}	$N\ m\ s^2$	1.7×10^{-3}	electronics box moment of inertia around the y-axis
I_{EZ}	$N\ m\ s^2$	1.3×10^{-3}	electronics box moment of inertia around the z-axis
I_{M1X}	$N\ m\ s^2$	11.7×10^{-6}	front motor moment of inertia around the x-axis
I_{M1Y}	$N\ m\ s^2$	2×10^{-3}	front motor moment of inertia around the y-axis
I_{M1Z}	$N\ m\ s^2$	2×10^{-3}	front motor moment of inertia around the z-axis
I_{M2X}	$N\ m\ s^2$	2×10^{-3}	right motor moment of inertia around the x-axis
I_{M2Y}	$N\ m\ s^2$	11.7×10^{-6}	right motor moment of inertia

Symbol	Unit	Value	Description
			around the y-axis
I_{M2Z}	$N\ m\ s^2$	2×10^{-3}	right motor moment of inertia
			around the z-axis
I_{M3X}	$N\ m\ s^2$	11.7×10^{-6}	rear motor moment of inertia
			around the x-axis
I_{M3Y}	$N\ m\ s^2$	2×10^{-3}	rear motor moment of inertia
			around the y-axis
I_{M3Z}	$N\ m\ s^2$	2×10^{-3}	rear motor moment of inertia
			around the z-axis
I_{M4X}	$N\ m\ s^2$	2×10^{-3}	left motor moment of inertia
			around the x-axis
I_{M4Y}	$N\ m\ s^2$	11.7×10^{-6}	left motor moment of inertia
			around the y-axis
I_{M4Z}	$N\ m\ s^2$	2×10^{-3}	left motor moment of inertia
			around the z-axis
I_{MG1X}	$N\ m\ s^2$	0.19×10^{-6}	front motor gear moment of inertia
			around the x-axis
I_{MG1Y}	$N\ m\ s^2$	70.8×10^{-6}	front motor gear moment of inertia
			around the y-axis
I_{MG1Z}	$N\ m\ s^2$	70.6×10^{-6}	front motor gear moment of inertia
			around the z-axis
I_{MG2X}	$N\ m\ s^2$	70.8×10^{-6}	right motor gear moment of inertia
			around the x-axis
I_{MG2Y}	$N\ m\ s^2$	0.19×10^{-6}	right motor gear moment of inertia
			around the y-axis
I_{MG2Z}	$N\ m\ s^2$	70.6×10^{-6}	right motor gear moment of inertia
			around the z-axis
I_{MG3X}	$N\ m\ s^2$	0.19×10^{-6}	rear motor gear moment of inertia
			around the x-axis
I_{MG3Y}	$N\ m\ s^2$	70.8×10^{-6}	rear motor gear moment of inertia
			around the y-axis
I_{MG3Z}	$N\ m\ s^2$	70.6×10^{-6}	rear motor gear moment of inertia

Symbol	Unit	Value	Description
			around the z-axis
I_{MG4X}	$N\ m\ s^2$	70.8×10^{-6}	left motor gear moment of inertia around the x-axis
I_{MG4Y}	$N\ m\ s^2$	0.19×10^{-6}	left motor gear moment of inertia around the y-axis
I_{MG4Z}	$N\ m\ s^2$	70.6×10^{-6}	left motor gear moment of inertia around the z-axis
I_{P1X}	$N\ m\ s^2$	38.8×10^{-6}	front propeller gear moment of inertia around the x-axis
I_{P1Y}	$N\ m\ s^2$	525×10^{-6}	front propeller gear moment of inertia around the y-axis
I_{P1Z}	$N\ m\ s^2$	556×10^{-6}	front propeller gear moment of inertia around the z-axis
I_{P2X}	$N\ m\ s^2$	525×10^{-6}	right propeller gear moment of inertia around the x-axis
I_{P2Y}	$N\ m\ s^2$	38.8×10^{-6}	right propeller gear moment of inertia around the y-axis
I_{P2Z}	$N\ m\ s^2$	556×10^{-6}	right propeller gear moment of inertia around the z-axis
I_{P3X}	$N\ m\ s^2$	38.8×10^{-6}	rear propeller gear moment of inertia around the x-axis
I_{P3Y}	$N\ m\ s^2$	$525 \times 10{-}6$	rear propeller gear moment of inertia around the y-axis
I_{P3Z}	$N\ m\ s^2$	556×10^{-6}	rear propeller gear moment of inertia around the z-axis
I_{P4X}	$N\ m\ s^2$	525×10^{-6}	left propeller gear moment of inertia around the x-axis
I_{P4Y}	$N\ m\ s^2$	38.8×10^{-6}	left propeller gear moment of inertia around the y-axis
I_{P4Z}	$N\ m\ s^2$	556×10^{-6}	left propeller gear moment of inertia around the z-axis
I_{PG1X}	$N\ m\ s^2$	1.3×10^{-6}	front propeller gear moment of inertia

Symbol	Unit	Value	Description
			around the x-axis
I_{PG1Y}	$N\ m\ s^2$	316×10^{-6}	front propeller gear moment of inertia
			around the y-axis
I_{PG1Z}	$N\ m\ s^2$	316×10^{-6}	front propeller gear moment of inertia
			around the z-axis
I_{PG2X}	$N\ m\ s^2$	316×10^{-6}	right propeller gear moment of inertia
			around the x-axis
I_{PG2Y}	$N\ m\ s^2$	1.3×10^{-6}	right propeller gear moment of inertia
			around the y-axis
I_{PG2Z}	$N\ m\ s^2$	316×10^{-6}	right propeller gear moment of inertia
			around the z-axis
I_{PG3X}	$N\ m\ s^2$	1.3×10^{-6}	rear propeller gear moment of inertia
			around the x-axis
I_{PG3Y}	$N\ m\ s^2$	316×10^{-6}	rear propeller gear moment of inertia
			around the y-axis
I_{PG3Z}	$N\ m\ s^2$	316×10^{-6}	rear propeller gear moment of inertia
			around the z-axis
I_{PG4X}	$N\ m\ s^2$	316×10^{-6}	left propeller gear moment of inertia
			around the x-axis
I_{PG4Y}	$N\ m\ s^2$	1.3×10^{-6}	left propeller gear moment of inertia
			around the y-axis
I_{PG4Z}	$N\ m\ s^2$	316×10^{-6}	left propeller gear moment of inertia
			around the z-axis
I_{XX}	$N\ m\ s^2$	8.1×10^{-3}	body moment of inertia around the x-axis
I_{XY}	$N\ m\ s^2$	0	body moment of inertia around the y-axis
			when the object is rotated around the x-axis
I_{XZ}	$N\ m\ s^2$	0	body moment of inertia around the z-axis
			when the object is rotated around the x-axis
I_{YX}	$N\ m\ s^2$	0	body moment of inertia around the x-axis
			when the object is rotated around the y-axis
I_{YY}	$N\ m\ s^2$	8.1×10^{-3}	body moment of inertia around the y-axis
I_{YZ}	$N\ m\ s^2$	0	body moment of inertia around the z-axis

Symbol	Unit	Value	Description
			when the object is rotated around the y-axis
I_{ZX}	$N\ m\ s^2$	0	body moment of inertia around the x-axis
			when the object is rotated around the z-axis
I_{ZY}	$N\ m\ s^2$	0	body moment of inertia around the y-axis
			when the object is rotated around the z-axis
I_{ZZ}	$N\ m\ s^2$	14.2×10^{-3}	body moment of inertia around the z-axis
$\boldsymbol{I}$	$N\ m\ s^2$	see p. 136	body inertia matrix
$\boldsymbol{I}_{j\times j}$	$-$	see p. 14	identity matrix of dimension j
J_B	$N\ m\ s^2$	70.8×10^{-6}	blade rotational moment of inertia
			around the propeller axis
J_M	$N\ m\ s^2$	1.1×10^{-6}	rotational moment of inertia
			around the motor axis
J_{MG}	$N\ m\ s^2$	12.8×10^{-9}	motor gear rotational moment of inertia
			around the motor axis
J_P	$N\ m\ s^2$	72.8×10^{-6}	rotational moment of inertia
			around the propeller axis
J_{PG}	$N\ m\ s^2$	1.38×10^{-6}	propeller gear rotational moment of inertia
			around the propeller axis
J_R	$N\ m\ s^2$	1.08×10^{-6}	rotor rotational moment of inertia
			around the motor axis
J_{TM}	$N\ m\ s^2$	3.68×10^{-6}	total rotational moment of inertia
			around the motor axis
J_{TP}	$N\ m\ s^2$	104×10^{-6}	total rotational moment of inertia
			around the propeller axis
K_E	$V\ s\ rad^{-1}$	6.3×10^{-3}	electric motor constant
$K_{E_{DAT}}$	$V\ s\ rad^{-1}$	4.6×10^{-3}	electric motor constant from DAT
$K_{E_{LR1}}$	$V\ s\ rad^{-1}$	6.3×10^{-3}	electric motor constant from LR1
$K_{E_{LR2}}$	$V\ s\ rad^{-1}$	6.3×10^{-3}	electric motor constant from LR2
$K_{E_{RMS}}$	$V\ s\ rad^{-1}$	6.3×10^{-3}	electric motor constant from RMS
K_M	$N\ m\ A^{-1}$	6.3×10^{-3}	mechanic motor constant
L	H	15×10^{-6}	motor inductance
L_B	m	0.31	blade length

Symbol	Unit	Value	Description
L_C	m	470×10^{-3}	cross structure length
L_E	m	200×10^{-3}	electronics box length
L_M	m	210×10^{-3}	motor to COM distance around the x-axis
L_{MG}	m	210×10^{-3}	motor gear to COM distance around the x-axis
L_P	m	235×10^{-3}	propeller to COM distance around the x-axis
L_{PG}	m	235×10^{-3}	propeller gear to COM distance around the x-axis
M_B	kg	8.8×10^{-3}	blade mass
M_C	kg	25×10^{-3}	half cross structure mass
M_E	kg	200×10^{-3}	electronics box mass
M_M	kg	46×10^{-3}	motor mass
M_{MG}	kg	1.6×10^{-3}	motor gear mass
M_P	kg	8.8×10^{-3}	propeller mass
M_{PG}	kg	5.7×10^{-3}	propeller gear mass
M_R	kg	46×10^{-3}	rotor mass
$\boldsymbol{M}_B$	$+$	see p. 14	system inertia matrix WRT B-frame
$\boldsymbol{M}_H$	$+$	see p. 19	system inertia matrix WRT H-frame
N	$-$	5.6	gear box reduction ratio
N_B	$-$	2	number of blades per propeller
R	Ω	0.6	motor resistance
R_C	m	2.5×10^{-3}	cross structure radius
R_{BSC}	Ω	1.1	motor resistance from BSC
R_{DAT}	Ω	1.05	motor resistance from DAT
R_{LR1}	Ω	0.61	motor resistance from LR1
R_{LR2}	Ω	0.58	motor resistance from LR2
R_M	m	12×10^{-3}	motor radius
R_{MG}	m	4×10^{-3}	motor gear radius
R_P	m	155×10^{-3}	propeller radius
R_{PG}	m	22×10^{-3}	propeller gear radius
R_R	m	8.5×10^{-3}	rotor radius
R_{RMS}	Ω	0.61	motor resistance from RMS
T_{NL}	$N\ m$	0	motor no load torque
T_{RL}	$N\ m$	4.91×10^{-3}	motor rated load torque

Symbol	Unit	Value	Description
W_P	N	2.45	weight carried from one propeller
W_B	m	20×10^{-3}	blade width
W_E	m	200×10^{-3}	electronics bos width
η	$-$	0.9	gear box efficiency
θ_{I_0}	rad	0.67	zero angle of incidence
$\theta_{I_{tw}}$	rad	0.29	twist angle of incidence
λ	$-$	0.11	inflow ratio
μ	$-$	0	rotor advance ratio
ρ_A	$kg\ m^{-3}$	1.2	air density
ω_H	$rad\ s^{-1}$	215	propeller speed in hovering
ω_{M_0}	$rad\ s^{-1}$	2.07×10^3	motor speed constant term
ω_{NL}	$rad\ s^{-1}$	2.07×10^3	motor no load speed
ω_{RL}	$rad\ s^{-1}$	1.83×10^3	motor rated load speed
$\Delta\omega_M$	$rad\ s^{-1}\ N^{-1}\ m^{-1}$	49×10^3	motor speed linear coefficient
Ω_H	$rad\ s^{-1}$	215	propeller speed in hovering

Appendix I

List of variables

Symbol	Unit	Description
dD_{BET}	$N\ m^{-1}$	infinitesimal drag force with BET
dF_{BET}	$N\ m^{-1}$	infinitesimal aerodynamic resultant force with BET
dH_{BET}	$N\ m^{-1}$	infinitesimal aerodynamic horizontal force with BET
dL_{BET}	$N\ m^{-1}$	infinitesimal lift force with BET
dT_{BET}	$N\ m^{-1}$	infinitesimal aerodynamic vertical force with BET
e	V	back electro-motive force of the motor
e	$+$	generic error
e_z	m	height error
e_θ	rad	pitch error
e_ϕ	rad	roll error
e_ψ	rad	yaw error
f_{OLC}	$blade\ s^{-1}$	frequency of the waveform on the oscilloscope
$\boldsymbol{h}$	$-$	PWM code vector
i	A	motor current
j	$-$	generic index
k	$-$	generic index
$\boldsymbol{k}$	$-$	generic vector
$\dot{m}_A$	$kg\ s^{-1}$	variation of the air mass through the disk
n	$-$	number of data acquired
p	$rad\ s^{-1}$	quadrotor angular velocity around x_B WRT B-frame
$\dot{p}$	$rad\ s^{-2}$	quadrotor angular acceleration around x_B WRT B-frame

Symbol	Unit	Description
$p_{-\infty}$	Pa	air pressure asymptotically under the rotor
p_1	Pa	air pressure directly under the rotor
p_2	Pa	air pressure directly over the rotor
$p_{+\infty}$	Pa	air pressure asymptotically over the rotor
q	$rad\ s^{-1}$	quadrotor angular velocity around y_B WRT B-frame
$\dot{q}$	$rad\ s^{-2}$	quadrotor angular acceleration around y_B WRT B-frame
r	$rad\ s^{-1}$	quadrotor angular velocity around z_B WRT B-frame
$\dot{r}$	$rad\ s^{-2}$	quadrotor angular acceleration around z_B WRT B-frame
r	m	radial position of the airfoil section
r	$+$	generic task reference
t	s	time
u	$m\ s^{-1}$	quadrotor linear velocity along x_B WRT B-frame
$\dot{u}$	$m\ s^{-2}$	quadrotor linear acceleration along x_B WRT B-frame
u	$+$	generic controlled variable
v	$m\ s^{-1}$	quadrotor linear velocity along y_B WRT B-frame
$\dot{v}$	$m\ s^{-2}$	quadrotor linear acceleration along y_B WRT B-frame
v	V	input motor voltage
$v_{-\infty}$	$m\ s^{-1}$	air speed asymptotically under the rotor
v_1	$m\ s^{-1}$	air speed directly under the rotor
v_2	$m\ s^{-1}$	air speed directly over the rotor
$v_{+\infty}$	$m\ s^{-1}$	air speed asymptotically over the rotor
v_{EST}	V	estimated dataset voltage
v_{EXP}	V	experimental dataset voltage
v_H	$m\ s^{-1}$	horizontal air flow velocity at the rotor
v_I	$m\ s^{-1}$	air inflow speed at the rotor
v_L	V	motor inductance voltage
v_R	V	motor resistance voltage
v_T	$m\ s^{-1}$	air flow velocity at the rotor
v_V	$m\ s^{-1}$	vertical air flow velocity at the rotor
$\boldsymbol{v}$	V	inputs voltage vector
w	$m\ s^{-1}$	quadrotor linear velocity along z_B WRT B-frame
$\dot{w}$	$m\ s^{-2}$	quadrotor linear acceleration along z_B WRT B-frame

Symbol	Unit	Description
y	$+$	generic process output
z^d	m	desired height
z_{IR}	m	height measured by the IR module
z_{SONAR}	m	height measured by the SONAR
C_L	$-$	lift coefficient
$\boldsymbol{C}_B(\boldsymbol{\nu})$	$+$	Coriolis-centripetal matrix WRT B-frame
$\boldsymbol{C}_H(\boldsymbol{\zeta})$	$+$	Coriolis-centripetal matrix WRT H-frame
$\boldsymbol{E}_H(\boldsymbol{\xi})$	$+$	movement matrix WRT H-frame
F_x	N	quadrotor force along x_B WRT B-frame
F_y	N	quadrotor force along y_B WRT B-frame
F_z	N	quadrotor force along z_B WRT B-frame
$\boldsymbol{F}^B$	N	quadrotor forces vector WRT B-frame
$\boldsymbol{F}^E$	N	quadrotor forces vector WRT E-frame
$\boldsymbol{F}_G^B$	N	gravitational force vector WRT B-frame
$\boldsymbol{G}_B(\boldsymbol{\xi})$	$+$	gravitational vector WRT B-frame
$\boldsymbol{J}_{\boldsymbol{\Theta}}$	$-$	generalized matrix
K_D	$+$	generic derivative coefficient
K_{Dz}	s^{-1}	height derivative coefficient
$K_{D\theta}$	s^{-1}	pitch derivative coefficient
$K_{D\phi}$	s^{-1}	roll derivative coefficient
$K_{D\psi}$	s^{-1}	yaw derivative coefficient
K_I	$+$	generic integral coefficient
K_{Iz}	s^{-3}	height integral coefficient
$K_{I\theta}$	s^{-3}	pitch integral coefficient
$K_{I\phi}$	s^{-3}	roll integral coefficient
$K_{I\psi}$	s^{-3}	yaw integral coefficient
K_P	$+$	generic proportional coefficient
K_{Pz}	s^{-2}	height proportional coefficient
$K_{P\theta}$	s^{-2}	pitch proportional coefficient
$K_{P\phi}$	s^{-2}	roll proportional coefficient
$K_{P\psi}$	s^{-2}	yaw proportional coefficient
$\boldsymbol{O}_B(\boldsymbol{\nu})$	$+$	gyroscopic propeller matrix WRT B-frame

Symbol	Unit	Description
$\boldsymbol{O}_H(\boldsymbol{\zeta})$	$+$	gyroscopic propeller matrix WRT H-frame
P_E	W	electric motor power
P_M	$N\ m\ s^{-1}$	mechanic motor power
P_P	$N\ m\ s^{-1}$	mechanic propeller power
Q_{BET}	$N\ m$	propeller torque with BET
$\boldsymbol{R}(\theta, y)$	$-$	rotation matrix around the y axis
$\boldsymbol{R}(\phi, x)$	$-$	rotation matrix around the x axis
$\boldsymbol{R}(\psi, z)$	$-$	rotation matrix around the z axis
$\boldsymbol{R}_{\boldsymbol{\Theta}}$	$-$	rotation matrix (roll-pitch-yaw)
$\dot{\boldsymbol{R}}_{\boldsymbol{\Theta}}$	$-$	rotation matrix derivative
$\boldsymbol{S}(\cdot)$	$+$	skew-symmetric operator
T_{BET}	N	propeller thrust with BET
T_G	g	propeller lift in grams
T_L	$N\ m$	load torque
T_M	$N\ m$	motor torque
T_{M_P}	$N\ m$	motor torque on propeller axis
T_{MT}	N	propeller thrust with MT
T_N	N	propeller lift in newtons
T_P	$N\ m$	propeller torque
T_{P_M}	$N\ m$	propeller torque on motor axis
$\boldsymbol{T}_{\boldsymbol{\Theta}}$	$-$	transfer matrix
U_1	N	vertical thrust respect to the body frame
U_2	$N\ m$	roll torque respect to the body frame
U_3	$N\ m$	pitch torque respect to the body frame
U_4	$N\ m$	yaw torque respect to the body frame
$\boldsymbol{U}$	$+$	basic movement vector
$\boldsymbol{U}_B(\boldsymbol{\Omega})$	$+$	movement vector WRT B-frame
V	m^3	volume
V_{DD}	V	supply voltage
$\boldsymbol{V}^B$	$m\ s^{-1}$	quadrotor linear velocity vector WRT B-frame
$\boldsymbol{V}^E$	$m\ s^{-1}$	quadrotor linear velocity vector WRT E-frame
$\dot{\boldsymbol{V}}^B$	$m\ s^{-2}$	quadrotor linear acceleration vector WRT B-frame

Symbol	Unit	Description
X	m	quadrotor linear position along x_E WRT E-frame
$\dot{X}$	$m\ s^{-1}$	quadrotor linear velocity along x_E WRT E-frame
$\ddot{X}$	$m\ s^{-2}$	quadrotor linear acceleration along x_E WRT E-frame
Y	m	quadrotor linear position along y_E WRT E-frame
$\dot{Y}$	$m\ s^{-1}$	quadrotor linear velocity along y_E WRT E-frame
$\ddot{Y}$	$m\ s^{-2}$	quadrotor linear acceleration along y_E WRT E-frame
Z	m	quadrotor linear position along z_E WRT E-frame
$\dot{Z}$	$m\ s^{-1}$	quadrotor linear velocity along z_E WRT E-frame
$\ddot{Z}$	$m\ s^{-2}$	quadrotor linear acceleration along z_E WRT E-frame
α	rad	angle of attack
ϵ_k	$-$	relative error
ϵ_{RMS}	$-$	RMS relative error
$\boldsymbol{\zeta}$	$+$	quadrotor generalized velocity vector WRT H-frame
$\dot{\boldsymbol{\zeta}}$	$+$	quadrotor generalized acceleration vector WRT H-frame
θ	rad	quadrotor angular position around y_1 WRT E-frame (pitch)
$\dot{\theta}$	$rad\ s^{-1}$	quadrotor angular velocity around y_1 WRT E-frame (pitch)
$\ddot{\theta}$	$rad\ s^{-2}$	quadrotor angular acceleration around y_1 WRT E-frame (pitch)
θ^d	rad	desired pitch angle
θ_I	rad	angle of incidence
$\boldsymbol{\nu}$	$+$	quadrotor generalized velocity vector WRT B-frame
$\dot{\boldsymbol{\nu}}$	$+$	quadrotor generalized acceleration vector WRT B-frame
$\boldsymbol{\xi}$	$+$	quadrotor generalized position vector WRT E-frame
$\dot{\boldsymbol{\xi}}$	$+$	quadrotor generalized velocity vector WRT E-frame
$\ddot{\boldsymbol{\xi}}$	$+$	quadrotor generalized acceleration vector WRT E-frame
ρ	$kg\ m^{-3}$	spatial density
τ_x	$N\ m$	quadrotor torque around x_B WRT B-frame
τ_y	$N\ m$	quadrotor torque around y_B WRT B-frame
τ_z	$N\ m$	quadrotor torque around z_B WRT B-frame
$\boldsymbol{\tau}^B$	$N\ m$	quadrotor torques vector WRT B-frame
$\boldsymbol{\tau}^E$	$N\ m$	quadrotor torques vector WRT E-frame
ϕ	rad	quadrotor angular position around x_2 WRT E-frame (roll)
$\dot{\phi}$	$rad\ s^{-1}$	quadrotor angular velocity around x_2 WRT E-frame (roll)

Symbol	Unit	Description
$\ddot{\phi}$	$rad\ s^{-2}$	quadrotor angular acceleration around x_2 WRT E-frame (roll)
ϕ^d	rad	desired roll angle
ϕ_I	rad	inflow angle
ψ	rad	quadrotor angular position around z_E WRT E-frame (yaw)
$\dot{\psi}$	$rad\ s^{-1}$	quadrotor angular velocity around z_E WRT E-frame (yaw)
$\ddot{\psi}$	$rad\ s^{-2}$	quadrotor angular acceleration around z_E WRT E-frame (yaw)
ψ^d	rad	desired yaw angle
$\dot{\psi}^d$	$rad\ s^{-1}$	desired yaw angle velocity
ω_M	$rad\ s^{-1}$	motor speed
$\dot{\omega}_M$	$rad\ s^{-2}$	motor acceleration
ω_P	$rad\ s^{-1}$	propeller speed
$\dot{\omega}_P$	$rad\ s^{-2}$	propeller acceleration
$\boldsymbol{\omega}^B$	$rad\ s^{-1}$	quadrotor angular velocity vector WRT B-frame
$\dot{\boldsymbol{\omega}}^B$	$rad\ s^{-2}$	quadrotor angular acceleration vector WRT B-frame
$\boldsymbol{\Gamma}^E$	m	quadrotor linear position vector WRT E-frame
$\dot{\boldsymbol{\Gamma}}^E$	$m\ s^{-1}$	quadrotor linear velocity vector WRT E-frame
$\ddot{\boldsymbol{\Gamma}}^E$	$m\ s^{-2}$	quadrotor linear acceleration vector WRT E-frame
Δ_A	$rad\ s^{-1}$	first propeller speed increment
Δ_B	$rad\ s^{-1}$	second propeller speed increment
$\boldsymbol{\Theta}^E$	rad	quadrotor angular position vector WRT E-frame
$\dot{\boldsymbol{\Theta}}^E$	$rad\ s^{-1}$	quadrotor angular velocity vector WRT E-frame
$\ddot{\boldsymbol{\Theta}}^E$	$rad\ s^{-2}$	quadrotor angular acceleration vector WRT E-frame
$\boldsymbol{\Lambda}$	$+$	quadrotor generalized force vector WRT B-frame
Ω	$rad\ s^{-1}$	overall propellers' speed
Ω_1	$rad\ s^{-1}$	front propeller speed
Ω_2	$rad\ s^{-1}$	right propeller speed
Ω_3	$rad\ s^{-1}$	rear propeller speed
Ω_4	$rad\ s^{-1}$	left propeller speed
$\boldsymbol{\Omega}$	$rad\ s^{-1}$	propellers' speed vector
$\dot{\boldsymbol{\Omega}}$	$rad\ s^{-2}$	propellers' acceleration vector

Bibliography

[1] *RCToys*. http://www.rctoys.com. 2

[2] *Silverlit*. http://www.silverlit.com. 2

[3] *Microdrones GmbH*. http://www.microdrones.com. 2

[4] L. Beji K. M. Zemalache and H. Marref. Control of an under-actuated system: Application to a four rotors rotorcraft. *IEEE International Conference on Robotics and Biomimetics*, pages 404 – 409, 2005. 2

[5] A. Abichou L. Beji and K. M. Zemalache. Smooth control of an x4 bidirectional rotors flying robot. *Fifth International Workshop on Robot Motion and Control*, pages 181 – 186, 2005. 2

[6] G. Fay. Derivation of the aerodynamic forces for the mesicopter simulation. 2001. 2, C.1

[7] P. McKerrow. Modelling the drganflyer four-rotor helicopter. *Proceedings of the IEEE International Conference on Robotics and Automation*, (4):3596 – 3601, 2004. 2, 6.2

[8] M. Achtelik K. M. Doth G. Hirzinger D. Gurdan, J. Stumpf and D. Rus. Energy-efficient autonomous four-rotor flying robot controlled at 1 khz. *IEEE International Conference on Robotics and Automation*, pages 361 – 366, 2007. 2

[9] J. P. How D. Pucci de Farias M. Valenti, B. Bethke and J. Vian. Embadding health management into mission tasking for uav teams. *Proceedings of the 2007 American Control Conference*, pages 5777 – 5783, 2007. 2

[10] P. Murrieri S. Bouabdallah and R. Siegwart. Design and control of an indoor micro quadrotor. 2

[11] A. Dzul P. Castillo and R. Lozano. Real-time stabilization and tracking of a four-rotor mini rotorcraft. *IEEE Transaction on Control System Technology*, 12(4):510 – 516, July 2004. 2

[12] R. Lozano P. Castillo and A. Dzul. Stabilization of a mini rotorcraft having four rotors. *Proceedings of 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2693 – 2698, 2004. 2

[13] A. Palomino S. Salazar-Cruz and R. Lozano. Trajectory tracking for a four rotor mini-aircraft. *Proceedings of the 44th IEEE Conference on Decision and control, and the European Control Conference 2005*, pages 2505 – 2510, 2005. 2

[14] A. Noth S. Bouabdallah and R. Siegwart. Pid vs lq control techniques applied to an indoor micro quadrotor. 2

[15] A. Tayebi and S. McGilvray. Attitude stabilization of a four-rotor aerial robot. *43rd IEEE Conference on Decision and Control*, pages 1216 – 1221, 2004. 2

[16] A. Tayebi and S. McGilvray. Attitude stabilization of a vtol quadrotor aircraft. *IEEE Transaction on Control System Technology*, 14(3):562 – 571, May 2006. 2, 6.1

[17] A. Fradkov B. Andrievsky and D. Peaucelle. Adaptive control experiments for laas "helicopter" benchmark. pages 760 – 765, 2005. 2

[18] Y. Morel and A. Leonessa. Direct adaptive tracking control of quadrotor aerial vehicles. *Florida Conference on Recent Advances in Robotics*, pages 1 – 6, 2006. 2

[19] R. Lozano P. Castillo and A. Dzul. Stabilization of a mini rotorcraft with four rotors. *IEEE Control Systems Magazine*, pages 45 – 55, 2005. 2

[20] T. Madani and A. Benallegue. Backstepping control for a quadrotor helicopter. *Proceedings of 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3255 – 3260, 2006. 2

[21] T. Madani and A. Benallegue. Backstepping sliding mode control applied to a miniature quadrotor flying robot. pages 700 – 705, 2006. 2

[22] T. Madani and A. Benallegue. Control of a quadrotor mini-helicopter via full state backstepping technique. *Proceedings of the 45th IEEE Conference on Decision and Control*, pages 1515 – 1520, 2006. 2

[23] A. Mokhtari and A. Benallegue. Dynamic feedback controller of euler angles and wind parameters estimation for a quadrotor unmanned aerial vehicle. *Proceedings of the 2004 IEEE International Conference on Robotics and Automation*, pages 2359 – 2366, 2004. 2

[24] A. Benallegue V. Mister and N. K. M'Sirdi. Exact linearization and non-interacting control of a 4 rotors helicopter via dynamic feedback. *IEEE International Workshop on Robot and Human Interactive Communication*, pages 586 – 593, 2001. 2

[25] T. Hamel N. Guenard and R. Mahony. A practical visual servo control for a unmanned aerial vehicle. *2007 IEEE International Conference on Robotics and Automation*, pages 1342 – 1348, 2007. 2

[26] M. Valenti G. P. Tournier and J. P. How. Estimation and control of a quadrotor vehicle using monocular vision and moire patterns. *AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2006. 2

[27] T. Hamel N. Metni and F. Derkx. Visual tracking control of aerial robotic systems with adaptive depth estimation. *Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference 2005*, pages 6078 – 6084, 2005. 2

[28] J. P. Ostrowski E. Altug and C. J. Taylor. Quadrotor control using dual camera visual feedback. *Proceedings of the 2003 IEEE International Conference on Robotics and Automation*, pages 4294 – 4299, 2003. 2

[29] M. G. Earl and R. D'Andrea. Real-time attitude estimation techniques applied to a four rotor helicopter. *43rd IEEE Conference on Decision and Control*, pages 3956 – 3961, 2004. 2

[30] C. Coza and C. J. B. Macnab. A new robust adaptive-fuzzy control method applied to quadrotor helicopter stabilization. 2006. 2

[31] M. Tarbouchi J. Dunfied and G. Labonte. Neural network based control of a four rotor helicopter. *2004 IEEE Intrnational Conference on Industrial Technology*, pages 1543 – 1548, 2004. 2

[32] J. S. Jang S. L. Waslander, G. M. Hoffmann and C. J. Tomlin. Multi-agent quadrotor testbed control design: Integral sliding mode vs. reinforcement learning. 2

[33] T. Perez. *Ship Motion Control:Course Keeping and Roll Stabilisation using Rudder and Fins*. Springer-Verlag, 2005. 3.2, 3.2, A.1, A.2

[34] M. Alaküla. *Power Electronic Control*. KFS, 2003. 3.3

[35] R. Scattolini P. Bolzern and N. Schiavoni. *Fondamenti di Controlli Automatici*. Mc Graw-Hill, 1998. 4.2

[36] *MathWorks*. `http://www.mathworks.com`. 5.1

[37] *College of Computation and Information Science*. `http://www.ccs.neu.edu`. 5.1

[38] *Aerospaceweb*. `http://www.aerospaceweb.org`. 6.1, C.2

[39] *Wikipedia*. `http://en.wikipedia.org`. 6.1, 8.1, D

[40] S. Bouabdallah. *Design and control of quadrotors with application to autonomous flying*. PhD thesis, EPFL, 2007. 6.1, C.2

[41] D.J. Auld and K. Srinivas. *Aerodynamics for Students*. University of Sydney, 1995. `http://www.ae.su.oz.au/aero`. 6.3

[42] *Robot Electronics*. `http://www.robot-electronics.co.uk`. 7.1

[43] *Sharp Microelectronics*. `http://www.sharpsma.com`. 7.2

[44] *Xsens*. `http://www.xsens.com`. 7.3, 7.3

[45] *Texas Instrument*. `http://www.ti.com`. 8.1

[46] *STMicroelectronics*. `http://www.st.com`. 9.1

[47] R. Ottoboni E. Bava and C. Svelto. *Fondamenti della misurazione*. Esculapio, 2000. B

[48] M. Arra. *L'elicottero*. Hoepli, 2001. C, C.2

Index

A

Aerodynamics 62, 129

B

Basic movements 8
Blade element theory 132

C

Circuit schematic 92
Circuit simulations 96
Circuit tests 99
Control algorithms 5, 31

D

Data extrapolation 64, 66
DC-motor 22
Dynamics 13, 22, 53, 124

E

Experimental results 111

G

Gear box 25

I

Identification 49
IMU 87
IR module 83

K

Kinematics 12, 119

L

Linear regression 72, 127

M

MCU 101
Modelling 7, 32
Moment of inertia 24, 53, 135
Momentum theory 129
Motor drivers 91

N

Newton-Euler formalism 12

P

PCB design 98
PID techniques 36

R

RMS relative error 76

S

Sallen-Key architecture 95
Sensors 79
Simulator 41
Software structure 109
SONAR module 80
System architecture 28, 106

U

UAV 1, 5