

POLITECNICO DI MILANO  
Scuola di Ingegneria Industriale e dell'Informazione  
Laurea Magistrale in Ingegneria dell'Automazione



Design, integration and control of a multirotor UAV  
platform

Relatore: Prof. Marco LOVERA  
Correlatore: Dr. Fabio RICCARDI  
Dr. Martino MIGLIAVACCA

Tesi di laurea di:  
Mattia GIURATO Matr. 816506

Anno Accademico 2014–2015



*Ai miei Nonni,  
che non hanno mai smesso di credere in me.*



# Acknowledgments

Sono molte le persone che voglio personalmente ringraziare per avermi aiutato o sostenuto durante lo svolgimento di questa tesi.

Penso che prima di tutti sia fondamentale ringraziare il Professor Lovera Marco per avermi dato l'opportunità di lavorare ad un progetto così vicino alle mie passioni e di avermi sopportato stoicamente, specialmente durante la stesura della tesi. I suoi suggerimenti e consigli tecnici sono stati fondamentali.

A seguire, due persone nello specifico mi sono state vicino durante tutto il progetto e mi hanno aiutato sia sotto il profilo tecnico che sotto il profilo umano. Grazie mille a Martino Migliavacca e a Fabio Riccardi. Non ce l'avrei mai fatta a concludere la tesi senza di voi.

Grazie inoltre a tutta la gente che ho conosciuto in AIRLab. Grazie mille al Professor Bonarini, a Giulio e a Davide.

Vorrei inoltre ringraziare coloro che sempre ci sono stati e che in un modo o nell'altro hanno fatto in modo che io non abbia mai dato di matto. Coloro che meritano di essere considerati AMICI. Grazie mille a Ludo, Teo, Umbie, Aureliano, Darione, Trevis, Fede, Ste, Tama e Brian. Grazie a tutti gli amici che hanno sempre creduto in me e grazie ai ragazzi di Skyward.

Un ringraziamento speciale va invece alla fantastica persona che giorno dopo giorno, ora dopo ora, errore di codice dopo errore di codice è stata al mio fianco, sopportando i miei sbalzi di umore e aiutandomi nei momenti più difficili. Grazie mille Nina per esserci sempre stata e per aver sempre supportato e sopportato le mie idee.

Infine il ringraziamento più sentito va alla mia famiglia e ai miei Nonni. Grazie per avermi permesso di concludere il mio percorso universitario e per avermi dato così tanto.



# Abstract

Nowadays, multirotor platforms for Unmanned Aerial Vehicles (UAVs), such as quadrotors, are omnipresent in our daily life and in the near future could spread more and more. The research and development for these platforms will become increasingly challenging as covering various fields such as Automation and Control, Electronics, Computer Science and Aerospace Engineering.

Multirotor platforms provide simple and reliable solutions to the problem of designing small scale UAVs for a wide range of civil applications such as, e.g., aerial photography and filming, agriculture, buildings and industrial plants monitoring, security and law enforcement and so on.

In spite of the very large number of projects aimed to the design and development of multirotor platforms currently ongoing, it seems that no systematic approach to the problem has been proposed, in the sense of a consistent process flowing from mission requirements to actual platform implementation. This is quite surprising as the currently available components would enable a very rapid deployment of almost arbitrary configurations, following a dedicated conceptual and preliminary design and optimization stage.

The main goal of this thesis is to present a systematic approach in order to realize a general UAV starting from the mission requirements. In detail, a development of a quadrotor helicopter is presented and the procedure to identify the value of unknown first-principle dynamics model parameters and the implementation of a robust controller are here described.

Moreover the problem of robust control design for the attitude dynamics of a quadrotor helicopter is studied in detail. The proposed approach is based on structured  $H_\infty$  synthesis applied to a quadrotor helicopter which dynamical model has been identified in dedicated experiments.

The goal is to obtain an attitude dynamics controller that guarantees satisfactory performance mainly in near hover conditions. In detail, the thesis presents the performed activities in order to realize a flying quadrotor starting from desired requirements. The operational decisions that have been taken, from the design to the sizing, are described together with the details about model identification and attitude controller synthesis activities.



# Sommario

Al giorno d'oggi, le piattaforme Multirotori utilizzate come Aeromobili a Pilotaggio Remoto (APR), come ad esempio gli elicotteri quadrirotori, sono onnipresenti nella nostra vita quotidiana e nel prossimo futuro potrebbero diffondersi sempre di più. La ricerca e lo sviluppo relativi a queste piattaforme diventeranno una sfida sempre più interessante e che coprirà diversi campi dell'ingegneria come l'Automazione Industriale, Elettronica, Informatica e l'Ingegneria Aeronautica.

Le piattaforme multirotore (in primo luogo la convenzionale configurazione quadrirotore) forniscono una soluzione semplice ed affidabile per la realizzazione di APR in scala ridotta che possono essere utilizzate per un'ampia serie di applicazioni in ambito civile, ad esempio, riprese aeree video e fotografiche, agricoltura, monitoraggio di impianti industriali, sicurezza ecc.

Nonostante il grande numero di progetti volti alla progettazione e lo sviluppo di piattaforme multirotore che attualmente sono in corso (sia in ambito industriale che accademico e hobbistico), sembra che non sia stato ancora proposto nessun approccio sistematico per la progettazione di questo tipo di velivoli. Approccio sistematico inteso come processo a partire dai requisiti di missione fino all'implementazione pratica con successiva fase di ottimizzazione.

La tesi ha come obiettivo quello di creare un ambiente di sviluppo completo per la progettazione e lo sviluppo di piattaforme multirotori che copre l'intero ciclo di progettazione, dalla fase di dimensionamento alla realizzazione vera e propria, comprensiva inoltre dello sviluppo del codice di bordo, fino alla messa in volo.

Il problema di progettazione di un controllo robusto per la dinamica di assetto di un elicottero quadrirotore viene studiata in dettaglio. L'approccio proposto si basa sulla sintesi  $H_\infty$  strutturata, applicata ad un elicottero quadrirotore il cui modello dinamico è stato identificato in esperimenti dedicati. L'obiettivo è quello di ottenere un controllo della dinamica di assetto che garantisca prestazioni soddisfacenti in particolare nel volo a punto fisso.

In dettaglio, la tesi presenta le attività svolte per realizzare un quadrirotore funzionante partendo dai requisiti desiderati. Dalla progettazione al dimensionamento e messa a punto, sono descritte tutte le decisioni operative prese.



# Contents

<b>Introduction</b>	<b>1</b>
<b>1 Multirotor hardware design</b>	<b>5</b>
1.1 Preliminary requirements . . . . .	5
1.2 Flight Control Unit . . . . .	6
1.3 Components description . . . . .	8
1.3.1 Frame . . . . .	8
1.3.2 Motors, ESCs and propellers . . . . .	10
1.3.3 Battery . . . . .	11
1.3.4 Remote controller . . . . .	12
1.3.5 Telemetry module . . . . .	12
1.3.6 Vibration damping . . . . .	14
1.4 Assembled quadrotor . . . . .	14
<b>2 Inertial Measurement Unit calibration</b>	<b>17</b>
2.1 Sensor model . . . . .	17
2.1.1 Accelerometer and magnetometer . . . . .	17
2.1.2 Gyroscope . . . . .	18
2.2 Calibration gimbal . . . . .	19
2.3 Calibration results . . . . .	21
2.3.1 Accelerometer and magnetometer . . . . .	21
2.3.2 Gyroscope . . . . .	23
2.4 Conclusions . . . . .	23
<b>3 Actuators analysis</b>	<b>25</b>
3.1 Momentum theory analysis in hovering flight . . . . .	25
3.2 Testing platform . . . . .	27
3.2.1 Load cell . . . . .	27
3.2.2 Optical tachometer . . . . .	28
3.2.3 The platform . . . . .	29
3.3 Static response estimation . . . . .	29
3.3.1 Least squares method . . . . .	29

3.3.2	Identification problem . . . . .	31
3.4	Dynamic response estimation . . . . .	32
3.5	Experimental results . . . . .	33
3.6	Validation . . . . .	35
<b>4</b>	<b>Dynamics of a quadrotor helicopter</b>	<b>37</b>
4.1	Axis systems . . . . .	37
4.1.1	Earth axes . . . . .	37
4.1.2	Body axes . . . . .	37
4.2	Rotation formalism . . . . .	38
4.2.1	Euler angles . . . . .	38
4.2.2	Quaternions . . . . .	41
4.3	Kinematics and flight dynamics . . . . .	44
4.3.1	Differentiation in a moving axis system . . . . .	44
4.3.2	Equation of motion - Linear motion . . . . .	45
4.3.3	Equation of motion - Angular motion . . . . .	45
4.3.4	Overall states of an aircraft . . . . .	47
4.3.5	External forces and moments . . . . .	48
4.4	Conclusions . . . . .	51
<b>5</b>	<b>Attitude estimation</b>	<b>55</b>
5.1	Madgwick filter . . . . .	55
5.1.1	Filter derivation . . . . .	55
5.1.1.1	Orientation from angular rate . . . . .	55
5.1.1.2	Orientation from vector observation . . . . .	56
5.1.2	Filter fusion derivation . . . . .	60
5.1.3	Magnetic distortion compensation . . . . .	62
5.1.4	Gyroscope bias drift compensation . . . . .	63
5.1.5	Filter gains . . . . .	64
5.1.6	Coefficients tuning . . . . .	65
5.1.6.1	Equipment . . . . .	65
5.1.6.2	Experimental procedure . . . . .	65
5.1.6.3	Tuning results . . . . .	66
5.2	Conclusions . . . . .	66
<b>6</b>	<b>Identification of roll and pitch dynamics</b>	<b>71</b>
6.1	Pitch dynamical model . . . . .	71
6.2	Test-bed . . . . .	73
6.3	Grey-box estimation . . . . .	73
6.4	Identification signal input - RBS . . . . .	75
6.5	Experiments and data analysis . . . . .	77

<b>7 Attitude control law design</b>	<b>83</b>
7.1 Control architecture . . . . .	83
7.2 Fixed-structure $H_\infty$ synthesis . . . . .	85
7.3 Robustness analysis . . . . .	89
7.4 Implementation of the controllers in FCU . . . . .	92
7.5 Experimental results . . . . .	95
7.6 Conclusions . . . . .	95
<b>Conclusions</b>	<b>101</b>
<b>Bibliography</b>	<b>104</b>
<b>A Multicopter safety</b>	<b>105</b>



# List of Figures

1	Quadrotor from 3D Robotics . . . . .	1
2	Aermatica Anteos . . . . .	2
1.1	eCalc logo . . . . .	6
1.2	Input Parameters . . . . .	6
1.3	Output Results . . . . .	6
1.4	R2P modules . . . . .	8
1.5	Quadrotor frame . . . . .	9
1.6	Brushless DC . . . . .	9
1.7	Sswitching signal over the three phases . . . . .	10
1.8	ESC . . . . .	10
1.9	Brushless and ESC mounted on the frame . . . . .	11
1.10	Propellers . . . . .	11
1.11	Propeller's parameters shown . . . . .	12
1.12	LiPo Battery . . . . .	12
1.13	Remote controller . . . . .	13
1.14	Telemetry modules . . . . .	13
1.15	Custom support . . . . .	14
1.16	Damper balls . . . . .	15
1.17	Electronics mounted on the damper . . . . .	15
1.18	Quadrotor assembled . . . . .	15
1.19	FCU detail . . . . .	16
2.1	6 DOF platform . . . . .	19
2.2	3 DOF calibration platforms . . . . .	20
2.3	Calibration platform . . . . .	20
2.4	Calibration platform software sheme in Simulink . . . . .	21
2.5	Typical results for accelerometer calibration . . . . .	22
2.6	Typical results for magnetometer calibration . . . . .	22
2.7	Accelerometer after and before calibration . . . . .	22
2.8	Magnetometer after and before calibration . . . . .	23
2.9	Gyroscope calibration . . . . .	24

3.1	Load cell . . . . .	28
3.2	Optical Tachometer . . . . .	28
3.3	Tachometer's output . . . . .	28
3.4	Testing platform . . . . .	29
3.5	Platform during an identification test . . . . .	30
3.6	Thrust vs Omega . . . . .	34
3.7	Omega vs Throttle . . . . .	34
3.8	Step Response . . . . .	35
3.9	Thrust obtained vs Thrust desired . . . . .	36
4.1	Body axes . . . . .	38
4.2	Euler angles . . . . .	40
4.3	Gimbal lock . . . . .	42
4.4	Quadcopter configuration . . . . .	49
4.5	Mixer and Quadcopter . . . . .	52
4.7	Three dimensional representation . . . . .	52
4.6	Quadcopter simulator . . . . .	53
5.1	Madgwick Filter - IMU Implementation . . . . .	62
5.2	Madgwick Filter - MARG Implementation . . . . .	64
5.3	OptiTrack . . . . .	65
5.4	Marker set . . . . .	65
5.5	RMS of the error with reference to $\beta$ . . . . .	67
5.6	IMU algorithm . . . . .	68
5.7	MARG algorithm . . . . .	69
6.1	Test-bed . . . . .	74
6.2	Quadrotor CAD on SolidWorks . . . . .	75
6.3	Pitch rate Bode diagram - Guess . . . . .	76
6.4	Identification results . . . . .	78
6.5	Pitch rate Bode diagram - Results . . . . .	80
6.6	Input and Output from a test . . . . .	80
7.1	Pitch Control Loops . . . . .	83
7.2	Yaw Control Loop . . . . .	85
7.3	Standard form for structured $H_\infty$ synthesis . . . . .	86
7.4	Tracking requirement . . . . .	88
7.5	Max gain and roll-off requirements . . . . .	89
7.6	Disturbance rejection requirement . . . . .	90
7.8	Additive uncertainty . . . . .	90
7.7	Sensitivity functions . . . . .	91

7.9	Multiplicative uncertainty . . . . .	91
7.10	Loop function . . . . .	92
7.11	Attitude Controllers in Simulink . . . . .	93
7.12	Complete Simulator in Simulink . . . . .	94
7.13	Step test with optimal parameters . . . . .	96
7.14	Step test with guess parameters . . . . .	97
7.15	Load disturbance to the motors . . . . .	98

*List of Figures*

---

# List of Tables

6.1	RBS parameters . . . . .	77
6.2	Values of the identified parameters . . . . .	79
7.1	Tuning parameters . . . . .	87

*List of Tables*

---

# Introduction

## Overview of multi-rotors vehicles

A multicopter is an aerial vehicle the motion of which is controlled by regulating properly the thrust of each rotor.

Multicopters are characterized by a very fast attitude dynamics and require an on-board computer (Flight Controller Unit, FCU) in order to realize a stable flight. As a result, they are “Fly by Wire” systems totally dependent from the FCU. The flight controller combines data from small on-board Inertial Measurement Unit (IMU) to maintain an accurate estimate of its orientation.

There are several configurations of multicopters depending on how many rotors they have: the names tricopter, quadcopter, hexacopter and octocopter are frequently used to refer to 3, 4, 6 and 8-rotor helicopters, respectively. For the purpose of this thesis the quadrotor configuration has been chosen.

The quadcopter shown in Figure 1 is the simplest type of quadrotor multicopter, where each rotor spins in the opposite direction from the two rotors on either side of it (e.g., rotors on opposite corners of the frame spin in the same direction). The quadcopter’s roll and pitch attitude can be controlled by modifying the thrust ob-



Figure 1: Quadrotor from 3D Robotics



Figure 2: Aermatica Anteos

tained by speeding up two rotors on one side and slowing down the other two. The quadcopter rotates around the yaw angle to the left or right by speeding up two motors that are diagonally across from each other, and slowing down the other two. Altitude is controlled by speeding up or slowing down all rotors at the same time. A multicopter becomes a drone, if it is able to complete a totally autonomous flight. Normally, to do so, a navigation system is needed in order to combine the information provided by the IMU with the data acquired by the barometer and the GPS, obtaining then a complete estimation of the attitude and the position. These platforms are becoming more and more interesting in the academic and research fields because they allow the development and testing of new control algorithms, navigation systems and sensor-fusion algorithms.

Nowadays, multirotor platforms provide simple and reliable solutions to several civil applications and different companies are providing many kind of services (e.g., Aermatica, Skyrobotic, 3D Robotics, VirtualRobotix, and so on). In Figure 2 a commercial product from Aermatica is shown. Following, possible areas of use for these aerial vehicles are described.

- **Aerial Inspection** provides invaluable information about an asset's condition that allows making critical operational or maintenance decisions. This kind of inspections are particularly valuable for close visual and thermal inspections of industrial structures and infrastructures (flare stacks, flare tips, bridges, pipelines, electric power lines, etc.) both onshore and offshore in the oil, gas, petrochemical, and utilities industries.
- **Aerial Surveillance** is the area of surveillance methods, usually visual imagery or video, from an airborne vehicle. Technological advances over the past decade have contributed significantly to rapid advances in aerial surveillance

and hardware such as remotely piloted aerial vehicles and high-resolution imagery capable of identifying objects at extremely long distances.

- **Aerial Security** means performing security-aimed monitoring and surveillance operations with the help of airborne vehicles. This kind of activities suggests that human officers (security organizations, law enforcement, police, etc.) would be able to remotely monitor and view video and data acquired from drones while planning and executing their operations.
- **Aerial Mapping** can provide details about structures, vegetation and other features that can be crucial in understanding a property or a landscape. For use in these fields, an image must be georeferenced, e.g., registered to specific places on the earth. In addition, sometimes to create accurate imagery – images that truly represent the surface of the earth – a photograph must be orthorectified, that is adjusted to correct for terrain displacement and camera tilt. Orthophotographs have real-world coordinates and have a uniform scale, so they are good base images to use in mapping and surveying applications.
- **Aerial Search & Rescue** for lost persons has been widely used all over the world in the past decades, even though it is considered to be an extremely costly method when performed with traditional fixed wing aircraft and helicopters. Small, highly maneuverable unmanned aerial vehicles may aid and support search efforts in many situations for a fraction of the cost and resources.
- **Aerial Video and Photography** provides the ability to show viewers remarkably smooth high-definition aerial footage from a perspective that has never been seen before. Companies around the world are starting to take advantage of this, and now aerial video shooting is used in many applications both by video production experts and professionals of other industries.

## Thesis description

Despite the large number of ongoing projects, no systematic approach to the design and realization of these platform has been proposed, in the sense of a consistent process flowing from mission requirements to actual platform implementation.

This is quite surprising as the currently available components would enable a very rapid deployment of almost arbitrary configurations, following a dedicated conceptual and preliminary design and final optimization stage.

In view of the above discussion, this thesis aims at contributing to different areas of the design and integration process for multirotor vehicle.

The main goal of this thesis is to present a systematic approach in order to realize a general UAV, starting from the mission requirements. In detail, a development

of a quadrotor helicopter is presented and the procedure to identify the value of unknown first-principle dynamics model parameters and the implementation of a robust controller are here described. Moreover the problem of robust control design for the attitude dynamics of a quadrotor helicopter is studied in detail. The proposed approach is based on structured  $H_\infty$  synthesis applied to a quadrotor helicopter which dynamical model has been identified in dedicated experiments.

In this thesis, a systematic work flow, starting from the mission requirements to the actual implementation of the control laws into the FCU, is presented.

## Thesis structure

The thesis starts with the first chapter about the formulation of the preliminary requirements and the choice of the components relatively the quadrotor. In Chapter 2, the characterization of the Inertial Measurement Unit (IMU), is described, with a description of a practical method on how to calibrate the sensors in order to obtain more accurate measures. Subsequently, in Chapter 3, the approach and the tools used to characterize the actuators of the quadrotor are described, in particular, the identification of main aerodynamic coefficients of the propellers, useful in case of model-based control. Chapter 4 is dedicated to the description of the reference frames and the motion equations that describe the dynamics of the quadrotor. In order to introduce a closed loop controller, a feedback of the state of the quadrotor is needed, hence in Chapter 5, the attitude estimation algorithm is analyzed and described. Before the introduction of the control algorithm, a description of a practical approach to the identification of the roll/pitch dynamics is needed. Moreover, thanks to the identified model, it will be possible to estimate the value of physical parameters of the quadrotor. Finally, in Chapter 7, the attitude control design problem is considered, and the  $H_\infty$  framework to synthesize a robust controller is studied. The obtained regulator has been then tested in laboratory experiments, in order to evaluate the performance compared with the standard tuning controller. Conclusions and future improvements are presented in the last part.

# Chapter 1

## Multirotor hardware design

In this Chapter the problem of the design of a quadrotor helicopter platform is presented and evaluated. To obtain best performances or particular solutions, one could consider to implement custom and more accurate helicopter mechanism (e.g. variables pitch propellers or tilting rotor hubs), anyhow, for a preliminary stage, the sizing has been done with an online tool considering only the components can be found off the shelf.

### 1.1 Preliminary requirements

The available configurations for multirotor helicopters are several, and each one has different characteristics. For the purpose of this thesis a simple quadrotor configuration has been chosen in order to focus more (at least in a preliminary phase) on the development of the methods and tools instead of the development of the helicopter mechanism and its application.

In order to choose an optimal configuration for the quadrotor helicopter components, one has to satisfy a set of initial requirements. In this project, the following requirements have been chosen:

- Frame configuration: X-quadrotor (this configuration allows future upgrades such as an on board camera);
- Frame dimensions: medium size (450÷550 mm as distance between opposite motors);
- Overall weight: less than 2 kg;
- Flight time: about 10';
- Payload: at least 500 g.

To satisfy these requirements, several iterations with an online tool (eCalc.ch, Figure 1.1) where performed.

## 1. Multirotor hardware design

---

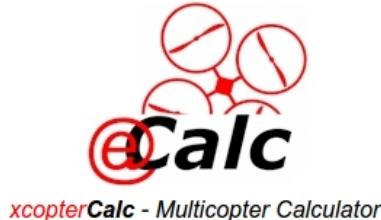


Figure 1.1: eCalc logo

In particular, given the parameters of the components found in manufacturers' data sheets and simulating different configurations, the most suitable solution was found. Figure 1.2-1.3 reports the inputs and outputs of the simulation. According to the obtained results, the requirements have been satisfied.

<b>General</b>	Motor Cooling:	# of Rotors:	Model Weight:	Frame Size:	FCU Tilt Limit:	Field Elevation:	Air Temperature:	Pressure (ONH):
	good	4 flat	1500 g 52.9 oz	500 mm 19.69 inch	30°	500 m ASL 1640 ft ASL	25 °C 77 °F	1013 hPa 29.91 inHg
<b>Battery Cell</b>	Type (Cont. / max. C) - charge state:	Configuration:	Cell Capacity:	Total Capacity:	Resistance:	Voltage:	C-Rate:	Weight:
	custom	3 S 1 P	4000 mAh	4000 mAh	0.0043 Ohm	3.7 V	25 C cont. 50 C max	111 g 3.9 oz
<b>Controller</b>	Type:	cont. Current:	max. Current:	Resistance:				
	custom	30 A	40 A	0.003 Ohm				
<b>Motor</b>	Manufacturer - Type (Kv): RCTimer ▾ HP2814-710 (710) search... Prop-Kv-Wizard	KV (w/o torque): 710 rpm/V	no-load Current: 0.7 A @ 11.1 V	Limit (up to 15s): 500 W	Resistance:	Case Length: 34 mm 1.34 inch	# mag. Poles: 14	Weight: 100 g 3.5 oz
<b>Propeller</b>	Type - yoke twist: custom	Diameter: 12 inch	Pitch: 4.5 inch	# Blades: 2	PConst / TConst: 1.3 / 1.0	Gear Ratio: 1 : 1	<input type="button" value="calculate"/>	

Figure 1.2: Input Parameters

Remarks:									
<b>Battery</b>	<b>Motor @ Optimum Efficiency</b>	<b>Motor @ Maximum</b>	<b>Motor @ Hover</b>	<b>Total Drive</b>		<b>Multicopter</b>			
Load:	12.65 C	Current: 7.17 A	Current: 12.65 A	Current: 4.83 A	Drive Weight: 947 g	All-up Weight: 1500 g			
Voltage:	10.45 V	Voltage: 10.71 V	Voltage: 10.41 V	Voltage: 10.84 V	33.4 oz	52.9 oz			
Rated Voltage:	11.10 V	Revolutions*: 6912 rpm	Revolutions*: 6171 rpm	Revolutions*: 4025 rpm	Thrust-Weight: 1.8 : 1	add. Payload: 855 g			
Capacity:	4000 mAh	electric Power: 76.8 W	electric Power: 131.7 W	Throttle (log): 51 %	Current @ Hover: 19.31 A	30.2 oz			
Energy:	44.4 Wh	mech. Power: 63.4 W	mech. Power: 104.5 W	Throttle (linear): 62 %	P(in) @ Hover: 214.4 W	max Tilt: 30 °			
Flight Time:	4.7 min	Efficiency: 82.6 %	Efficiency: 79.4 %	electric Power: 52.3 W	P(out) @ Hover: 163.7 W	max. Speed: 18 km/h			
Mixed Flight Time:	8.2 min		est. Temperature: 35 °C	mech. Power: 40.9 W	Efficiency @ Hover: 76.4 %	11.2 mph			
Hover Flight Time:	10.6 min			Efficiency: 78.2 %	Current @ max: 50.62 A	est. rate of climb: 3.9 m/s			
Weight:	333 g		est. Temperature: 95 °F	est. Temperature: 29 °C	P(in) @ max: 561.8 W	768 ft/min			
	11.7 oz			84 °F	P(out) @ max: 418.2 W	with Rotor fail:			
				specific Thrust: 7.17 g/W	Efficiency @ max: 74.4 %				
				0.25 oz/W					

Figure 1.3: Output Results

## 1.2 Flight Control Unit

The Flight Control Unit (FCU) is the core of the quadrotor, or better, the brain.

It has been decided to use as electronic boards the R2P (Rapid Robot Prototyping) modules [2]. R2P is an open source HW/SW framework providing components for the rapid development of robotic applications. R2P framework components reuse and easy integration is obtained through modular hardware and an embedded real-time publish/subscribe middleware which allows distributed control loops to be set

up in a flexible way. R2P aims at increasing hardware and software reuse while reducing integration time.

The advantages using these electronic board is the modularity of the architecture instead of a classical monolithic one, because for example one can plug-in additional boards with different functions. At the state of art, these boards provide different functions, in particular, the IMU module provides the main functions of attitude estimation and attitude control, the USB module provides serial communication with a computer or other serial devices and at last the RC module allows the PWM communication with the motors controllers.

Because these boards are not especially designed for quadrotor helicopters or UAVs in general, some modifications have been conducted to the internal firmware. A «control node» in the IMU module's firmware has been introduced. This node is subscribed to the messages published by the «attitude estimation node» (provided by the IMU module) and by the «set-point node» (provided by the USB module), and then, using these information the «control node» can compute the control variable that will be published as a message through the middleware. This last message will be used by the RC module in order to apply the control to the motors.

The code has been implemented in a modular way in order to allow future modifications (for example the same framework could be implemented in a different UAV with different actuator and structure). The main concept is to keep the core of the code unaltered and then change only the parameters related to the particular multirotor platform under development.

Another main feature related to the modular approach is about the possibility to introduce more complex control architectures and separate the calculations between the modules. For example, one module can be dedicated for the low level controllers like PID and use another module for higher level controllers like MPC and then use the output set-points from this controller as input set-point of the lower level controllers.

In detail, the used modules are here presented.

**IMU module (Fig.1.4a)** Provides position and heading measurements from inertial sensors and GPS.

It includes:

- 3-axis accelerometer
- 3-axis gyroscope
- 3-axis magnetometer
- Pressure sensor
- GPS receiver module

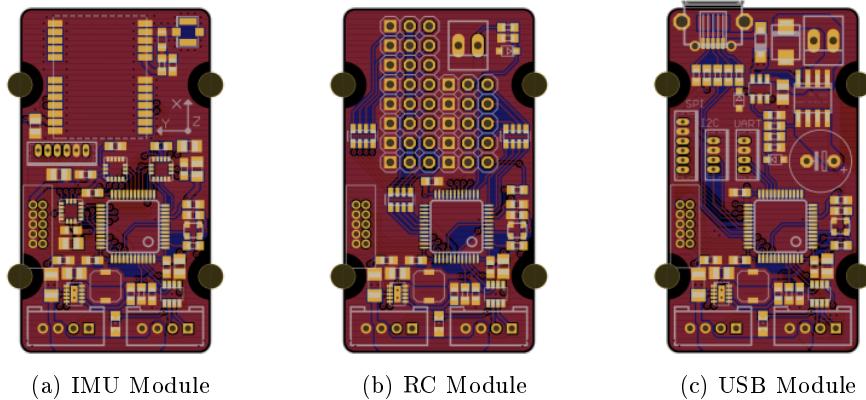


Figure 1.4: R2P modules

**RC module (Fig.1.4b)** Long range remote control and RC servo interface.

- Up to 12 servo outputs
- Up to 4 radio inputs
- PPM input (max 8 channels)
- Supports multiple radio vendors

**USB module (Fig.1.4c)** Interfaces the real-time R2P network with a computer.

- Micro-USB connector
- 1x UART interface
- 1x SPI interface
- 1x I2C interface

## 1.3 Components description

### 1.3.1 Frame

The chosen frame (Figure 1.5) is a Talon V2.0 (HobbyKing).

It guarantees strength and lightness due to carbon fiber and aluminum main parts. It also has dimensions that perfectly fit with the second constraint (500mm of distance between opposite motors).



Figure 1.5: Quadrotor frame



Figure 1.6: Brushless DC

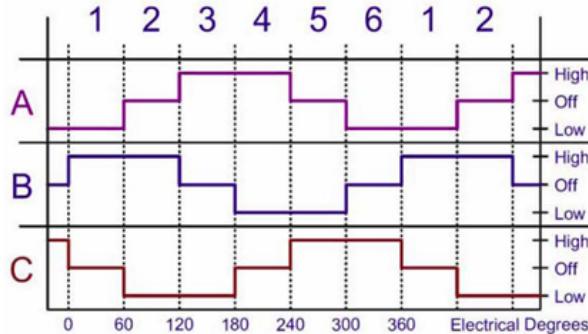


Figure 1.7: Swhitching signal over the three phases



Figure 1.8: ESC

### 1.3.2 Motors, ESCs and propellers

The motors (Figure 1.6) are BrushLess DC (BLDC) from the RCTimer High Performance Series (HP2814).

The main parameter that describes their behavior is the Kv parameter. This is a coefficient that represents the number of revolutions per minute (rpm) for each volt applied to the motor without the load connected (e.g., 710KV with a battery of 11.1V equals a maximum speed of 7881 rpm).

Obviously, they cannot be controlled directly by the constant voltage supplied by a battery because brushless motors are synchronous motors. BLDC motors are powered by an inverter that switches the DC voltage to a DC switching electric signal (Figure (1.7)). These inverters are also known in rc-models field as ESCs (Figure 1.8) that stands for Electronic Speed Controllers.

The ESCs chosen for the quadrotor are the RCTimer NFS ESC 30A.

ESCs control the rotational speed of the motors proportionally to a signal they receive as a set-point from the Flighth Control Unit (FCU). This signal given by the control unit is a Pulse Width Modulation signal (PWM).



Figure 1.9: Brushless and ESC mounted on the frame



Figure 1.10: Propellers

In Figure 1.9 it is possible to see the BLDC with its ESC mounted on the frame.

To generate the aerodynamic thrust force a propeller is needed (Figure 1.10). The propellers on a quadrotor have to rotate in opposite directions in order to balance the generated moments (two propellers clockwise and two propellers counter-clockwise).

The main characteristics for a commercial propeller are three parameters: number of blades, diameter and pitch. This last parameter is the distance a propeller would move if it turned one time through a solid (as shown in Figure 1.11).

The chosen propellers are the ones shown in Figure 1.10: two blades, 12" of diameter and 4.5" of pitch.

### 1.3.3 Battery

Quadcopters typically use Lithium Polymer (LiPo) batteries which come in a variety of sizes and configurations. The chosen LiPo is a Turnigy nano-tech 4000mAh 3S (three cells in series). Each cell is 3.7 Volts, so this battery is rated at 11.1 Volts.

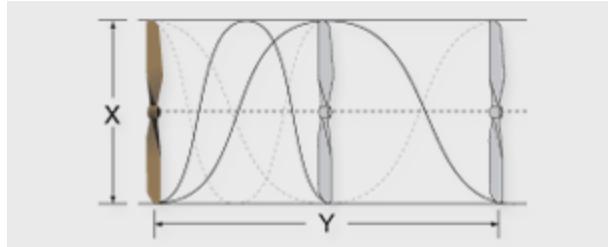


Figure 1.11: Propeller's parameters shown



Figure 1.12: LiPo Battery

LiPo batteries also have a C rating. The C rating describes the rate at which power can be drawn from the battery (e.g., 4000mAh of capacity with 25C constant and 50C burst means the battery can supply 100A constant and 200A of burst, that typically is about 10 seconds). Obviously, the maximum current supplied by the battery has to satisfy the maximum requirement from all the actuators (according to eCalc the maximum supplied current should be around 50A). In this case, the quadrotor mounts four ESCs with maximum 40A each of consumption and the overall request can be supplied by the battery.

### 1.3.4 Remote controller

The main method to control the quadrotor is to send the attitude set point and the thrust request through a remote controller (Figure 1.13).

This manual controller allows principally the operator to control the attitude of the quadrotor but can also «ARM» and «DISARM» the motors for security reasons (ground operations or mission abortion).

### 1.3.5 Telemetry module

To allow the communication from the ground station to the FCU (e.g.: send the attitude set point to the quadrotor, read the instant value of the sensors or the attitude estimated) a telemetry module has been installed (Figure 1.14).



Figure 1.13: Remote controller



Figure 1.14: Telemetry modules

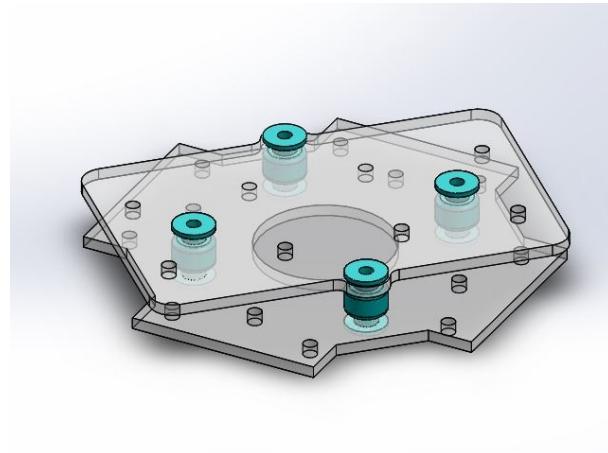


Figure 1.15: Custom support

These modules are a 2-way, half-duplex wireless communication system with a standard TTL UART interface, in addition they are small, light and inexpensive, and typically they allow ranges of more than 300m “out of the box” (the range can be extended to several kilometers with the use of a patch antenna on the ground). These radios operate at 433Mhz.

### 1.3.6 Vibration damping

«A helicopter is a collection of vibrations held together by differential equations.» [16]

In order to reduce drastically the effect of the mechanical vibration transmitted from the motors through the frame, a vibration damper is needed. To solve this problem a custom electronic support (Figure 1.15) which can be mounted on the frame has been designed and realized. The two plates are connected between four vibration damping balls (Figure 1.16) and all the electronic boards of the FCU (Figure 1.17) can be connected on the top plate.

## 1.4 Assembled quadrotor

The assembled quadrotor helicopter is shown in the Figures 1.18-1.19.



Figure 1.16: Damper balls

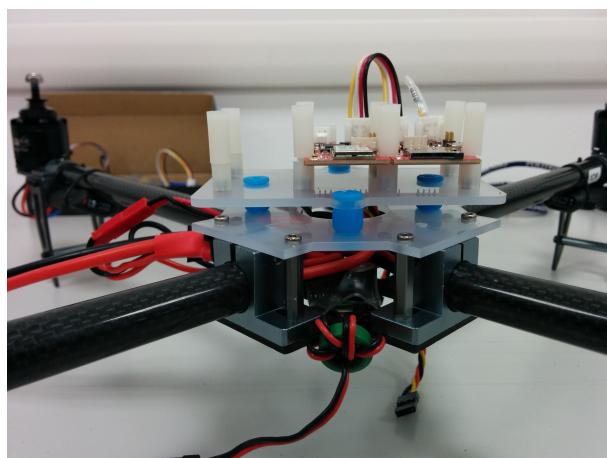


Figure 1.17: Electronics mounted on the damper



Figure 1.18: Quadrotor assembled

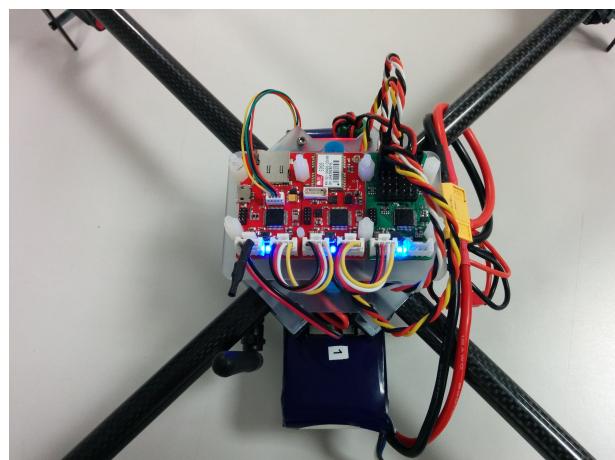


Figure 1.19: FCU detail

## Chapter 2

# Inertial Measurement Unit calibration

Attitude and Heading Reference Systems (AHRS) are self-contained units that provide a measurement of orientation relative to the Earth using an integrated array of gyroscopes, accelerometers and magnetometers. One abbreviation used in technology for this sensor arrays used in AHRS is MARG (Magnetic, Angular Rate, and Gravity). Inertial Measurement Units (IMUs) are similar devices that do not include magnetometers and are limited to measuring an attitude relative to the Earth. The low cost, low power and small dimensions of modern Micro-Electro-Mechanical Systems (MEMS) based sensors mean that AHRS' and IMUs are increasingly accessible tool for many applications like aerial drones.

The main issue using MEMS is about the accuracy of the measurements, and in order to get the sensors more reliable sensor readings a calibration is needed.

Here a low-cost automated calibration method to evaluate the gains and biases of accelerometers, magnetometers and gyroscopes with minimal human intervention (see [6, 10–12]) is presented. The method is implemented with the combination of a motorized calibration gimbal and a post-processing code written in MATLAB.

### 2.1 Sensor model

#### 2.1.1 Accelerometer and magnetometer

A stationary accelerometer will measure the direction and magnitude of the Earth's gravitational field due to the reaction force on the mass within the MEMS structure.

A magnetometer will measure the direction and magnitude of the Earth's magnetic field compounded with local magnetic interference.

For a calibrated sensor, the measured magnitude will be constant for all orientations of the sensor. That is, equation

$$m = \|Ku - b\| = \text{constant} \quad (2.1)$$

will be true for all said measurements where  $m$  is the magnitude of the sensor's respective field,  $K$  is the diagonal matrix of the gains and  $b$  is the vector of biases. Therefore,  $K$  and  $b$  may be found as the solution of the optimization problem in equation (2.2)

$$\min_{b \in \mathbb{R}, K \in \mathbb{R}^3} \left[ \sum_i (m - \|Ku_i - b\|)^2 \right] \quad (2.2)$$

where  $u_i$  represents the un-calibrated raw sensor measurement at the  $i$ -th orientation within a calibration data set.

AHRS and IMU algorithms often only require the direction of the Earth's gravitational or magnetic field to be known. Typically, values of  $9.81 \text{ m/s}^2$  and 1 are used for an accelerometer and magnetometer respectively. For effective calibration, the calibration data set should represent the full range of orientations of the sensor relative to the field.

As an accelerometer will also measure the translational, centripetal, tangential and Coriolis accelerations associated with the motion, it is important that the sensor is stationary during the acquisition of the calibration data set. Magnetometer measurements may be corrupted by magnetic interferences fixed to the sensor frame (hard-iron distortion). Hard-iron distortions result in a constant bias error in each axis and so will therefore be incorporated to the estimated bias,  $b$ , of the sensor. It may be impractical to use a location completely free from magnetic distortions.

### 2.1.2 Gyroscope

MEMS gyroscopes are subject to bias drift so calibration cannot assume a constant bias. Indeed, AHRS and IMU algorithms will typically estimate the gyroscope bias in real-time.

Consequently, gyroscope calibration need only be concerned with estimation of the sensor gains. Exposing the gyroscope to a constant angular velocity for any extended period of time would be impractical due to wires and cables. Alternatively, the gyroscope may be rotated around one axis by a known angle and the gain found as that which ensures the integrated angular velocity measured by the gyroscope is equal to the known angle of rotation. This can then be repeated for all three axes. The correct gain may therefore be found as the solution to equation

$$\min_{k_a \in \mathbb{R}} \left[ r - \Delta t \sum_{t=0}^n (k_a u_{a,t} - b_{a,t}) \right]^2, \begin{cases} t = 0, 1, \dots, n \\ a = x, y, z \end{cases} \quad (2.3)$$

where  $r$  is the known angle of rotation,  $\Delta t$  is the sampling period,  $u_{a,t}$  is the un-



Figure 2.1: 6 DOF platform

calibrated gyroscope measurement at time  $t$ , and  $b_t$  is the bias at time  $t$ .

The bias can be sampled when the gyroscope is stationary at the start and the end of the rotation,  $b_{a,0}$  and  $b_{a,n}$ ; as shown in equation

$$b_{a,t} = b_{a,0} + \frac{t}{n}(b_{a,n} - b_{a,0}), \begin{cases} t = 0, 1, \dots, n \\ a = x, y, z \end{cases} \quad (2.4)$$

The change in angle should be monotonic and occur over a short period of time to minimize integral drift. The angular velocity should be maximized to ensure that almost all the gyroscope output range is accounted for in calibration.

## 2.2 Calibration gimbal

An automated calibration solution requires that the sensory unit is moved through a range of orientations relative to the Earth's gravitational and magnetic fields for the calibration of the accelerometers and magnetometers. It must also be able to rotate the sensory unit around its  $x$ ,  $y$  and  $z$  axes independently for the calibration of the gyroscopes.

Different solutions with several architectures can be found in commerce and a lot of documentation about them can be found in literature [6, 10, 11]. In Figures 2.1-2.2 some examples are shown.

In the context of this thesis a custom calibration platform has been realized.

The solution was the actuated 3 Degrees Of Freedom (DOF) gimbal shown in Figure 2.3. The components are made of plywood in order to minimize the magnetic interference. The platform's brain is an Arduino UNO entirely programmed in Simulink environment (Figure 2.4).

## 2. Inertial Measurement Unit calibration

---

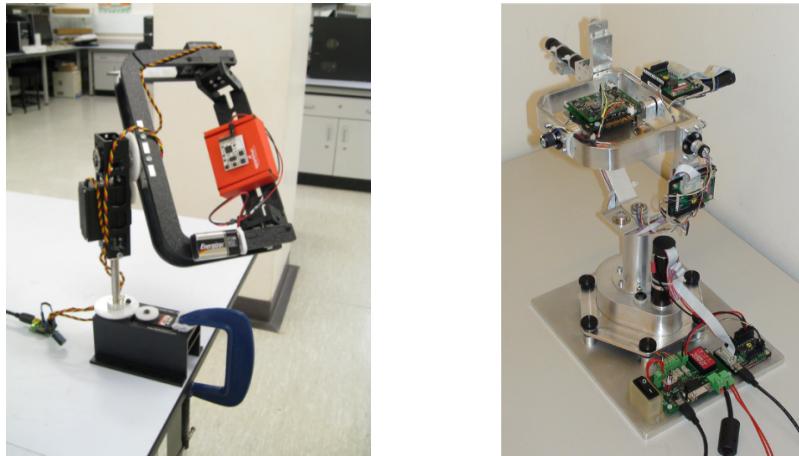


Figure 2.2: 3 DOF calibration platforms

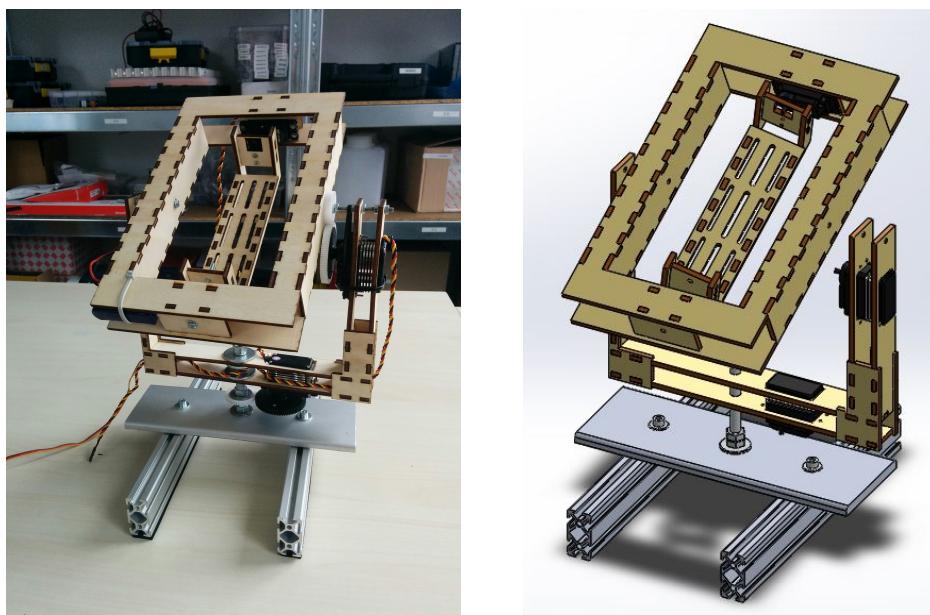


Figure 2.3: Calibration platform

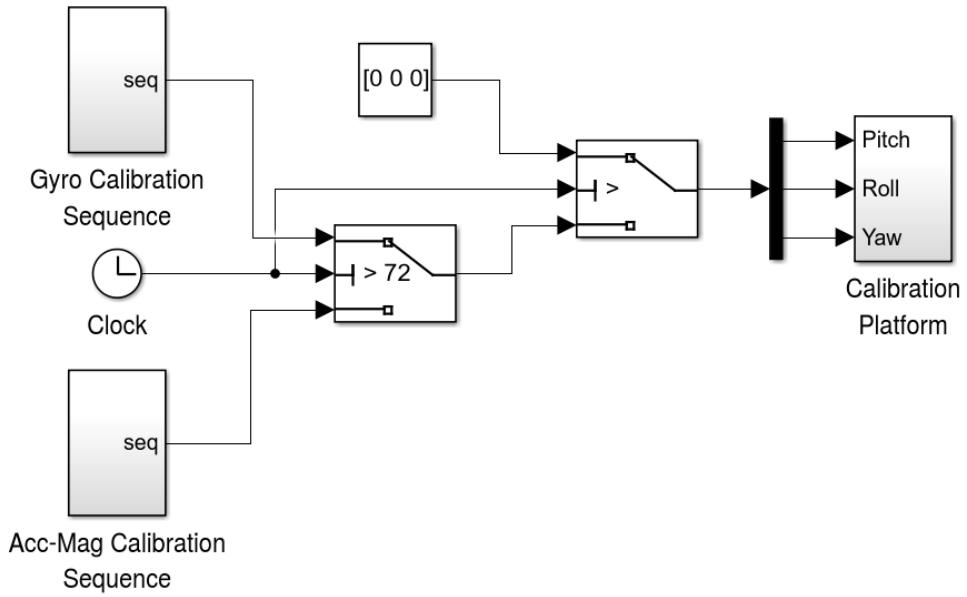


Figure 2.4: Calibration platform software sheme in Simulink

## 2.3 Calibration results

### 2.3.1 Accelerometer and magnetometer

The calibration routine was run several times to verify consistency of the results. The routine requires a couple of minutes to complete. Figures 2.5-2.6 show the calibrated accelerometer and magnetometer measurements for a typical calibration data set respectively. Each figure consists of two plots: the upper plot represents the calibrated individual  $x$ ,  $y$  and  $z$  measurements and the lower verifies the success of the calibration as a comparison between the measured and known magnitudes of the sensor's respective field.

In Figures 2.7-2.8 it is possible to evaluate the reliability of the calibration procedure comparing the calibrated measures with a sphere which radius equals the amplitude of the sensor's field and also comparing the non-calibrated data with the calibrated ones.

## 2. Inertial Measurement Unit calibration

---

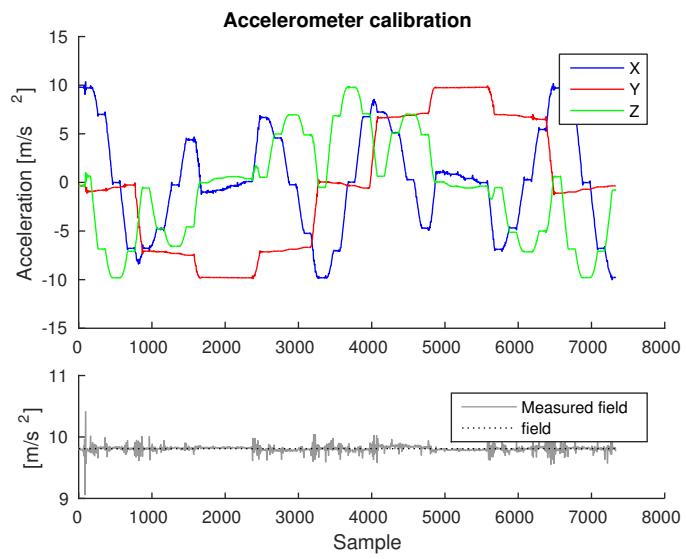


Figure 2.5: Typical results for accelerometer calibration

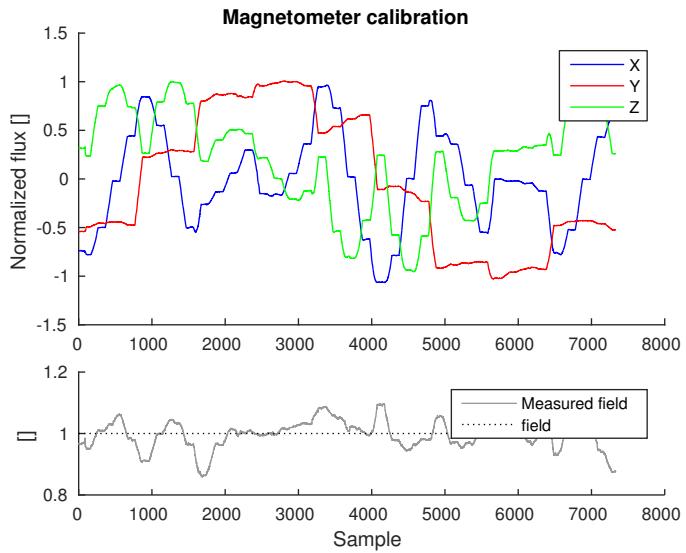


Figure 2.6: Typical results for magnetometer calibration

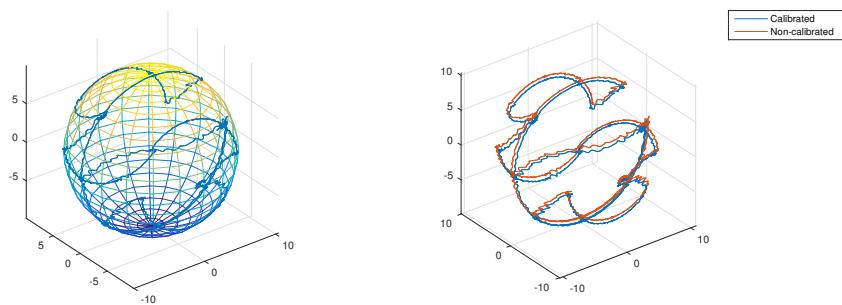


Figure 2.7: Accelerometer after and before calibration

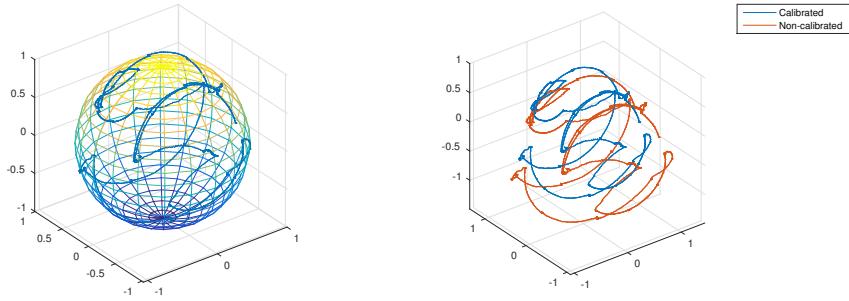


Figure 2.8: Magnetometer after and before calibration

The results obtained by the calibration campaign are reported in Equation 2.5.

$$K_{acc} = \begin{bmatrix} 0.0094 & 0 & 0 \\ 0 & 0.0096 & 0 \\ 0 & 0 & 0.0099 \end{bmatrix}, b_{acc} = \begin{bmatrix} -0.033 \\ -0.027 \\ 0.55 \end{bmatrix},$$

$$K_{mag} = \begin{bmatrix} 0.0023 & 0 & 0 \\ 0 & 0.0024 & 0 \\ 0 & 0 & 0.0026 \end{bmatrix}, b_{mag} = \begin{bmatrix} 0.20 \\ 0.13 \\ -0.41 \end{bmatrix}. \quad (2.5)$$

### 2.3.2 Gyroscope

Figure 2.9 shows the calibrated gyroscope measurements for a typical calibration data set. Each plot represents the calibrated angular velocity, the calculated angle of rotation, the target angle of rotation and the interpolated bias. The calibration may be verified by comparing the final calculated angle with respect to the target angle.

The results obtained by the calibration campaign are reported in Equation 2.6.

$$K_{gyr} = \begin{bmatrix} 0.00032 & 0 & 0 \\ 0 & 0.00032 & 0 \\ 0 & 0 & 0.00031 \end{bmatrix}. \quad (2.6)$$

## 2.4 Conclusions

An automated calibration solution capable of evaluating gains and biases of off-the-shelf IMUs and MARG with minimal human intervention with a custom calibration platform has been presented.

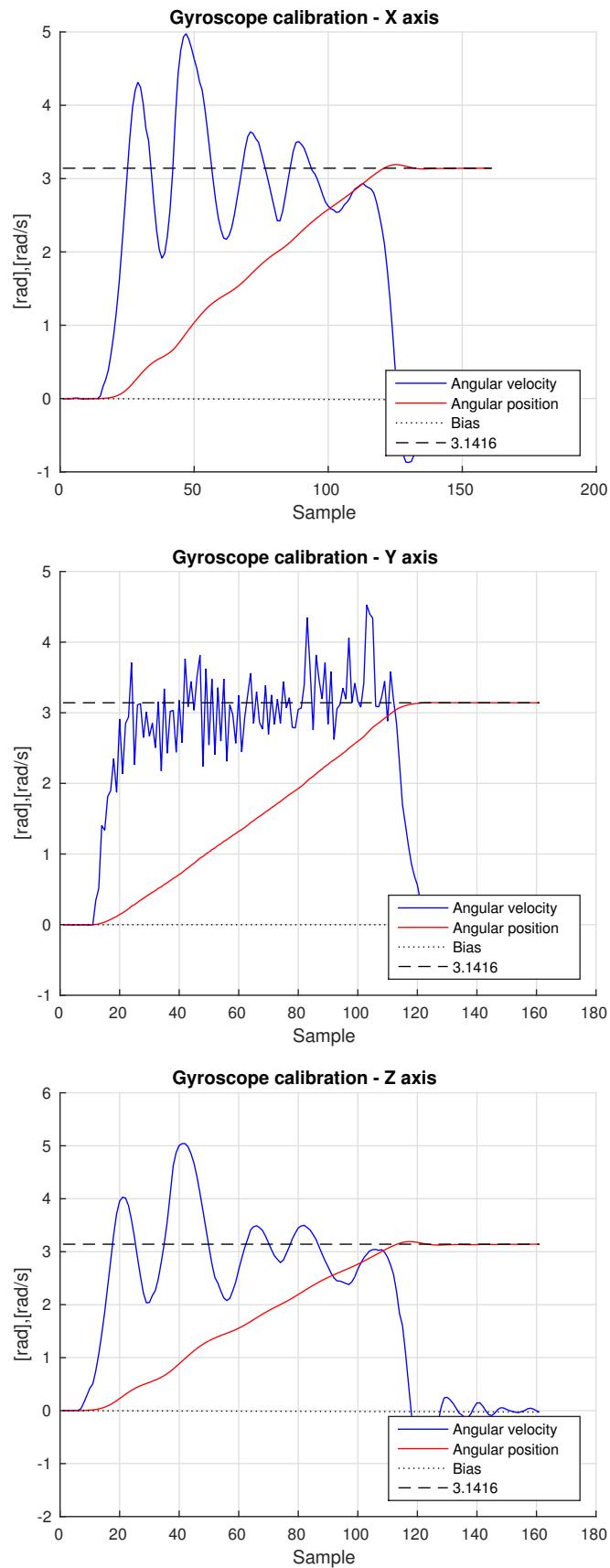


Figure 2.9: Gyroscope calibration

# Chapter 3

## Actuators analysis

The aim of this chapter is to show an approach to identify the thrust, torque and power coefficients ( $C_T$ ,  $C_Q$ ,  $C_P$  respectively) of propellers used for small UAVs. This type of non-dimensional coefficients are useful for a better tune of the numerical simulator in case of model based control design. Furthermore, it is also possible to identify the static and dynamic relationship from percentage of the throttle and rotational speed of the propeller.

### 3.1 Momentum theory analysis in hovering flight

A helicopter, or any other rotating-wing vehicle, must operate in a variety of flight regimes. These include the hover, climb, descent, or forward flight. In addition, the helicopter may undergo maneuvers, which may comprise a combination of these basic flight regimes. In hover or axial flight, the flow is axisymmetric and the flow through the rotor is either upward or downward. This is the easiest flow regime to analyze and, at least in principle, it should be the easiest to predict by means of mathematical models. It has been found, however, that even with modern mathematical models of the rotor flow, accurate prediction of hovering performances is by no means straightforward. Although it must be remembered that the actual physical flow about the rotor will comprise a complicated vortical wake structure. The basic performance of the rotor can be analyzed by a simpler approach that has become known as the Rankine-Froude *momentum theory*. The momentum theory approach allows the derivation of a first-order prediction of the rotor thrust and power, and the principles also form a foundation for more elaborate treatments of the rotor aerodynamics problem.

As in all branches of engineering, non-dimensional coefficients are normally employed in helicopter analysis (see [8]).

Considering the induced velocity through the rotor ( $v_i$ ) we can write it as a function of the rotor thrust ( $T$ ), the disk area ( $A$ ), the blade tip speed ( $v_{tip}$ ) and the

density of the air ( $\rho$ )

$$v_i = f_1(T, A, v_{tip}, \rho), \quad (3.1)$$

$$f_2(v_i, T, A, v_{tip}, \rho) = 0. \quad (3.2)$$

According to the Buckingham  $\pi$ -method of dimensional analysis this means there are five variables ( $N = 5$ ), three fundamental dimensions ( $K = 3$ , namely: mass, length and time) and  $N - K = 2$  or two  $\pi$ -groups.

The functional dependence can be written in the form

$$f_3(\pi_1, \pi_2) = 0, \quad (3.3)$$

where  $\pi_1$  and  $\pi_2$  are non-dimensional groupings.

Choosing the variables  $\rho$ ,  $A$ , and  $v_{tip}$  as repeating variables (which are linearly independent) the two non-dimensional  $\pi$ -groups can be each written in terms of these repeating variables plus one other variable.

Making each grouping dimensionally homogeneous, the first  $\pi$ -group can be written as

$$\pi_1 = \rho^{-1} A^{-1} v_{tip}^{-2} T = \frac{T}{\rho A v_{tip}^2} = C_T, \quad (3.4)$$

which is the thrust coefficient.

The second  $\pi$ -group becomes

$$\pi_2 = \rho^0 A^0 v_{tip}^{-1} v_i = \frac{v_i}{v_{tip}} = \lambda_i, \quad (3.5)$$

which is the inflow coefficient.

This means that

$$f_3(\pi_1, \pi_2) = f_3 \left( \frac{T}{\rho A v_{tip}^2}, \frac{v_i}{v_{tip}} \right) = f_3(C_T, \lambda_i), \quad (3.6)$$

$$\lambda_i = f(C_T). \quad (3.7)$$

Therefore, in helicopter analysis the rotor thrust coefficient is formally defined as

$$C_T = \frac{T}{\rho A v_{tip}^2} = \frac{T}{\rho A \Omega^2 R^2}. \quad (3.8)$$

All velocity components are non-dimensionalized by tip speed so the inflow ratio  $\lambda_i$  is related to the thrust coefficient in hover by

$$\lambda_i = \frac{v_i}{v_{tip}} = \frac{v_i}{\Omega R} = \frac{1}{\Omega R} \sqrt{\frac{T}{2\rho A}} = \sqrt{\frac{T}{2\rho A (\Omega R)^2}} = \sqrt{\frac{C_T}{2}}. \quad (3.9)$$

This is based on the 1-D flow assumption where the inflow is uniformly distributed over the disk.

The rotor power coefficient is defined as

$$C_P = \frac{P}{\rho A v_{tip}^3} = \frac{P}{\rho A \Omega^3 R^3}, \quad (3.10)$$

so, based on momentum theory, the power coefficient for the hovering rotor is

$$C_P = \frac{P}{\rho A \Omega^3 R^3} = \left( \frac{T}{\rho A \Omega^2 R^2} \right) \left( \frac{v_i}{\Omega R} \right) = \frac{C_T^{3/2}}{\sqrt{2}}. \quad (3.11)$$

Again, this is calculated on the basis of uniform inflow and no viscous losses, so it is called the ideal power coefficient.

Momentum theory underpredicts the actual power required because viscous effects have been totally neglected. Usually an empirical modification of the momentum results has been introduced, but in order to reduce the possible uncertainties introduced by the identification process (more instruments should be required), has been decided to assume true the ideal formalization.

The corresponding rotor shaft torque coefficient is defined as

$$C_Q = \frac{Q}{\rho A V_{tip}^2 R} = \frac{Q}{\rho A \Omega^2 R^3}. \quad (3.12)$$

Notice that because power is related to torque by  $P = \Omega Q$ , then numerically  $C_P = C_Q$ .

## 3.2 Testing platform

In order to obtain an estimation of the coefficients  $C_T$ ,  $C_Q$  and  $C_P$ , one has to obtain the thrust curve as a function of propeller's speed rotation.

To do so, it is necessary to measure the propeller's thrust against the speed rotation with a load cell and an optical tachometer respectively.

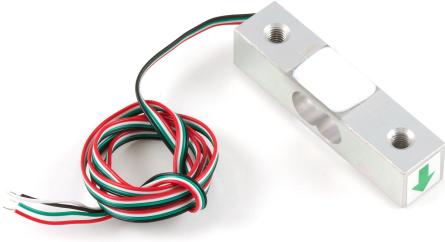
All the data are logged with the low-cost electronic board Arduino.

### 3.2.1 Load cell

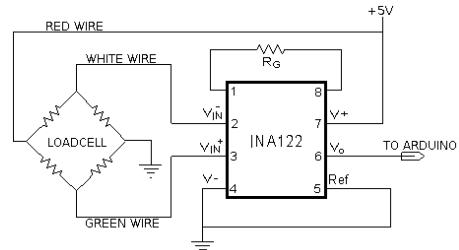
To measure the propeller's thrust, a strain gauge load cell with a differential output is necessary. In order to read with the electronic board the sensor's output an INstrumentation Amplifier (INA) has to be used. The configuration used is shown in the Figure 3.1b.

### 3. Actuators analysis

---



(a) Strain gauge load cell



(b) Schematic

Figure 3.1: Load cell

#### 3.2.2 Optical tachometer

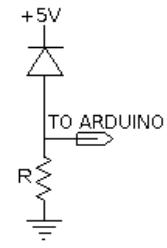
The optical tachometer is composed by a laser source and a photodiode sensor: the laser source produce a beam that collides the photodiode which detect if the beam is interrupted by the rotating propeller.



(a) Laser source



(b) Photodiode



(c) Schematic

Figure 3.2: Optical Tachometer

While the beam is interrupted, the output is LOW (0V) otherwise is HIGH (5V), in this way, when the propeller rotates, the photodiode produce a square wave as output.

Arduino acquires the square wave and measures its frequency, which is proportional to the rotational speed (e.g. if propeller has two blades, the real frequency is a half of the measured one).

In Figure 3.3 it is shown an example of the output where the propeller's period of rotation ( $T$ ) is indicated.

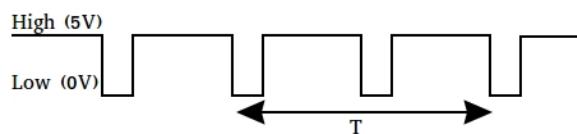


Figure 3.3: Tachometer's output

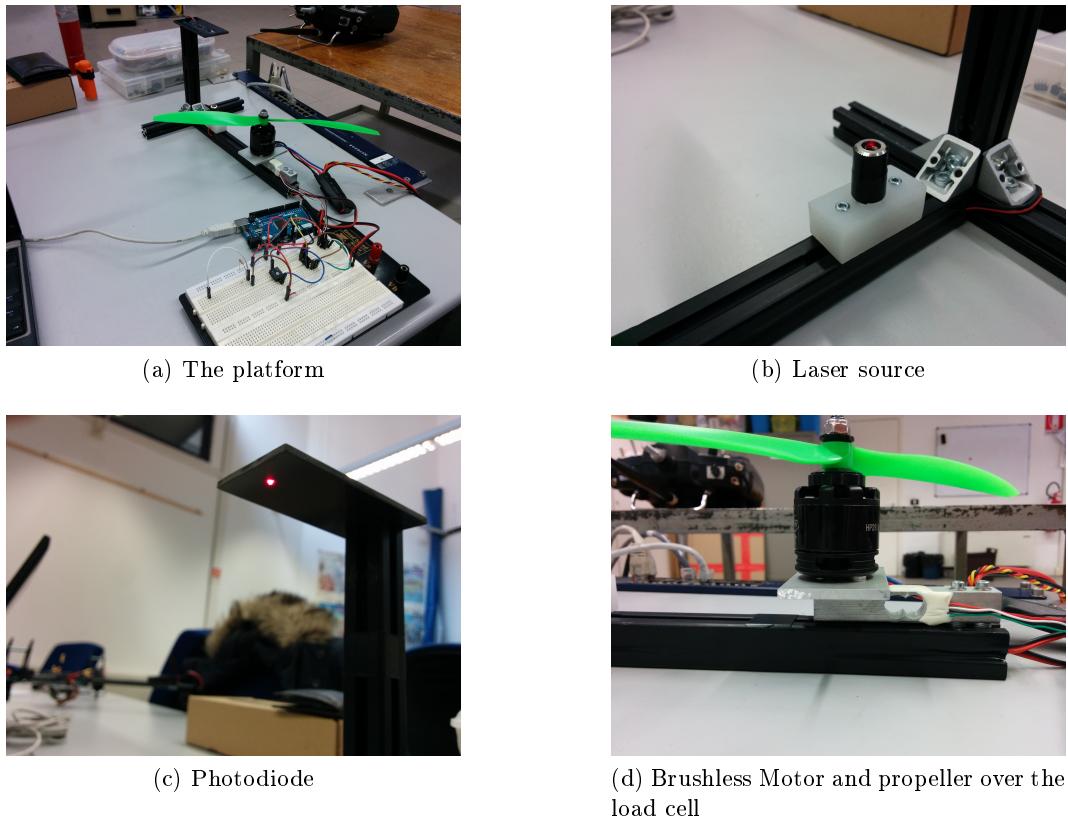


Figure 3.4: Testing platform

### 3.2.3 The platform

In Figure 3.4 the set-up of the platform with testing motor and propeller mounted is shown. Note that, in order to measure the propeller's thrust, the propeller has to be mounted upside down because the thrust must be in the direction of the load cell. One has to notice that, in order not to occur the ground effect, the platform has to be mounted on a workshop-clamp that rises the platform from the table (Figure 2.4).

## 3.3 Static response estimation

### 3.3.1 Least squares method

The method of least squares (see [5]) is a standard approach in regression analysis to find the approximate solution of over-determined systems.

"Least squares" means that the overall solution minimizes the sum of the squares of the errors made in the results of every single equation.

The most important application is in data fitting: The best fit in the least-squares sense minimizes the sum of squared residuals, a residual being the difference between

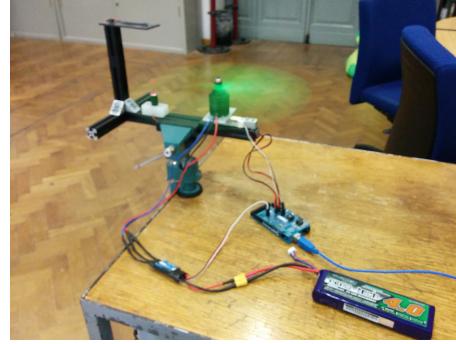


Figure 3.5: Platform during an identification test

an observed value and the fitted value provided by a model.

Least squares problems fall into two categories: linear or ordinary least squares and non-linear least squares, depending on whether or not the residuals are linear in all unknowns.

Here an over-determined system is considered

$$\sum_{j=1}^n X_{ij}\beta_j = y_i, (i = 1, 2, \dots, m), \quad (3.13)$$

of  $m$  linear equations in  $n$  unknown coefficients  $\beta_1, \beta_2, \dots, \beta_n$ , with  $m > n$ .

Such system can be written in matrix form as

$$X\beta = y, \quad (3.14)$$

where

$$X = \begin{bmatrix} X_{11} & X_{12} & \dots & X_{1n} \\ X_{21} & X_{22} & \dots & X_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ X_{m1} & X_{m2} & \dots & X_{mn} \end{bmatrix}, \quad \beta = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_n \end{bmatrix}, \quad y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}. \quad (3.15)$$

The goal is to find the coefficients  $\beta$  which fits the equations “best” in the sense of solving the quadratic minimization problem

$$\hat{\beta} = \min_{\beta} J(\beta), \quad (3.16)$$

where the objective function  $J$  is given by

$$J(\beta) = \sum_{i=1}^m \left| y_i - \sum_{j=1}^n x_{ij}\beta_j \right|^2 = \|y - X\beta\|^2. \quad (3.17)$$

This minimization problem has a unique solution, provided that the  $n$  columns of the matrix  $X$  are linearly independent, given by solving the normal equations

$$(X^T X) \hat{\beta} = X^T y \rightarrow \hat{\beta} = (X^T X)^{-1} X^T y. \quad (3.18)$$

The matrix  $(X^T X)$  is known as the Gramian matrix of  $X$ , which possesses several nice properties such as being a positive semi-definite matrix, while  $(X^T X)^{-1} X^T$  is called left pseudo-inverse.

### 3.3.2 Identification problem

The relationship between propeller's thrust and rotational speed, as shown in Section 3.1, is the following

$$T = K_T \Omega^2, \quad (3.19)$$

where

$$K_T = C_T \rho A R^2. \quad (3.20)$$

During the data acquisition process, the thrust and rotational speed varying percentage of the throttle at steps of 10% (from 10% up to 100%) has been measured.

In this way one can formulate the problem as

$$y = T = \begin{bmatrix} T_{10} \\ T_{20} \\ \vdots \\ T_{100} \end{bmatrix}, \quad X = \Omega^2 = \begin{bmatrix} \Omega_{10}^2 \\ \Omega_{20}^2 \\ \vdots \\ \Omega_{100}^2 \end{bmatrix}, \quad \beta = K_T. \quad (3.21)$$

And find the unique solution as

$$\hat{\beta} = \hat{K}_T = (X^T X)^{-1} X^T y. \quad (3.22)$$

Once found  $\hat{K}_T$  it is trivial to compute the following equations

$$\hat{C}_T = \frac{\hat{K}_T}{\rho A R^2}, \quad (3.23)$$

$$\hat{C}_P = \frac{\hat{C}_T^{3/2}}{\sqrt{2}}, \quad (3.24)$$

$$\hat{C}_Q = \hat{C}_P. \quad (3.25)$$

It is also useful to identify the real static relationship between the percentage of throttle and propeller's rotational speed, which is given by the form

$$\Omega = mTh\% + q. \quad (3.26)$$

The problem can be defined as

$$y = \Omega = \begin{bmatrix} \Omega_{10} \\ \Omega_{20} \\ \vdots \\ \Omega_{100} \end{bmatrix}, \quad X = Th\% = \begin{bmatrix} 10 & 1 \\ 20 & 1 \\ \vdots & \vdots \\ 100 & 1 \end{bmatrix}, \quad \beta = \begin{bmatrix} m \\ q \end{bmatrix}. \quad (3.27)$$

Solving, as before, one can obtain

$$\hat{\beta} = \begin{bmatrix} \hat{m} \\ \hat{q} \end{bmatrix} = (X^T X)^{-1} X^T y. \quad (3.28)$$

### 3.4 Dynamic response estimation

Assuming a first order dynamical relationship between the input throttle ( $Th\%$ ) and the rotational speed of the propeller ( $\Omega$ ) as output

$$G(s) = \frac{\Omega(s)}{Th(s)} = \frac{\mu}{1 + s\tau}, \quad (3.29)$$

the last part of the identification process consists in the application of step-like profiles of throttle in order to estimate the gain ( $\mu$ ) and the time constant ( $\tau$ ) of the model.

To find the desired parameters a model identification approach is used with the matrices  $A, B, C, D$  as described

$$A = -\frac{1}{\tau}, B = \frac{\mu}{\tau}, C = 1, D = 0. \quad (3.30)$$

From the static relationship between the percentage of throttle and propeller's rotational speed it is possible to see that there is a non-linearity in the neighborhood of zero. This is the reason why to estimate the dynamic model, steps that do not start/stop from/at 0% have been generated.

Because of the modelization of the system and the complexity introduced using a non zero initial state, it has been decided to depolarize from the constant offset the input and the output in order to evaluate only the dynamic behavior of the propeller.

Another problem is that it is not possible to use the optical tachometer to measure the rotational speed because the sensor's sampling rate is too low (two sample per

rotation), to solve this problem the load cell has been used to measure the thrust generated and then estimate the actual rotational speed using the following relationship (using the estimated  $\hat{C}_T$ )

$$\Omega = \sqrt{\frac{T}{K_T}}. \quad (3.31)$$

### 3.5 Experimental results

After an identification campaign, an accurate model of the relationship between percentage of throttle and rotational speed and the between rotating speed and thrust have been found. The identified propeller's coefficients are:

$$\begin{cases} \hat{C}_T = 1.186 \times 10^{-2} \\ \hat{C}_Q = 9.132 \times 10^{-4} \\ \hat{C}_P = 9.132 \times 10^{-4}. \end{cases} \quad (3.32)$$

The static relation between throttle and rotational speed is

$$\Omega = \hat{m}Th\% + \hat{q} \rightarrow \begin{cases} \hat{m} = 6.031 \text{ [rad/s]} \\ \hat{q} = 80.49 \text{ [rad/s]} \end{cases} \quad (3.33)$$

And finally the dynamic relation between throttle and rotational speed is

$$G(s) = \frac{\Omega(s)}{Th(s)} = \frac{\hat{\mu}}{1 + s\hat{\tau}} \rightarrow \begin{cases} \hat{\mu} = 5.2 \text{ [rad/s]} \\ \hat{\tau} = 92 \times 10^{-3} \text{ [s]} \end{cases} \quad (3.34)$$

In Figures 3.6-3.8 an example of the data obtained during the identification campaign are shown.

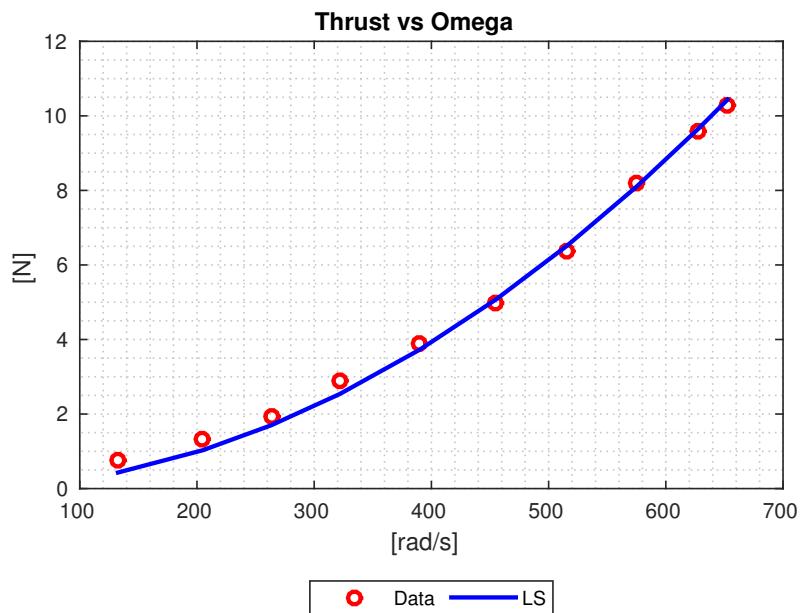


Figure 3.6: Thrust vs Omega

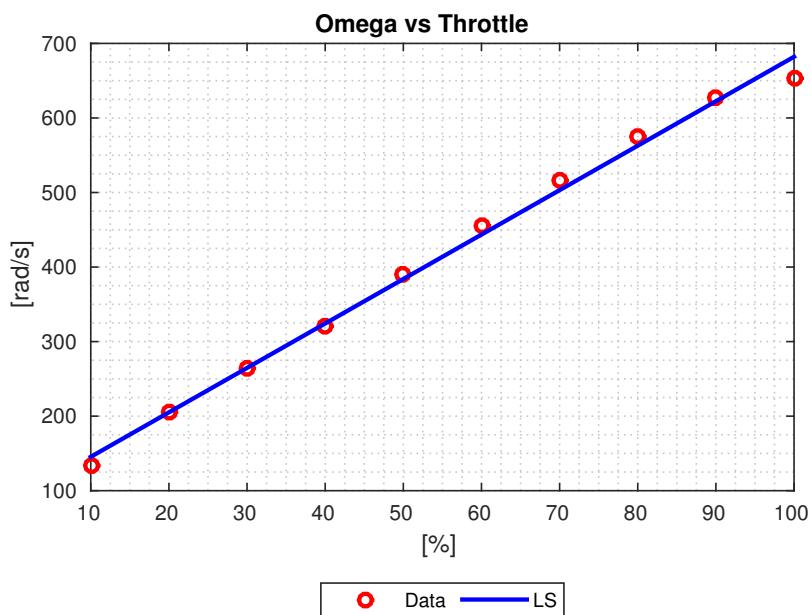


Figure 3.7: Omega vs Throttle

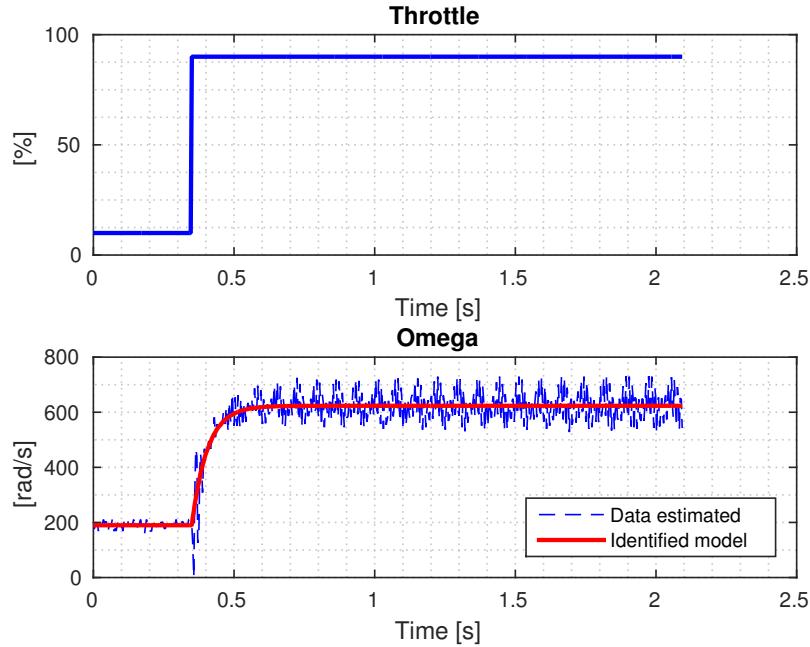


Figure 3.8: Step Response

In Figure 3.8 one can notice a ripple signal superimposed on the data acquired due to the un-balanced plastic propeller's vibration.

### 3.6 Validation

In order to validate the model found before, some experiments have been made in order to generate a desired thrust with the propeller. The desired thrust has been increased from 1N up to 10N with steps of 1N. The linear relationship found is the following

$$y = ax + b \rightarrow \begin{cases} a = 0.989 & [\text{rad/s}] \\ b = 0.212 & [\text{rad/s}] \end{cases} \quad (3.35)$$

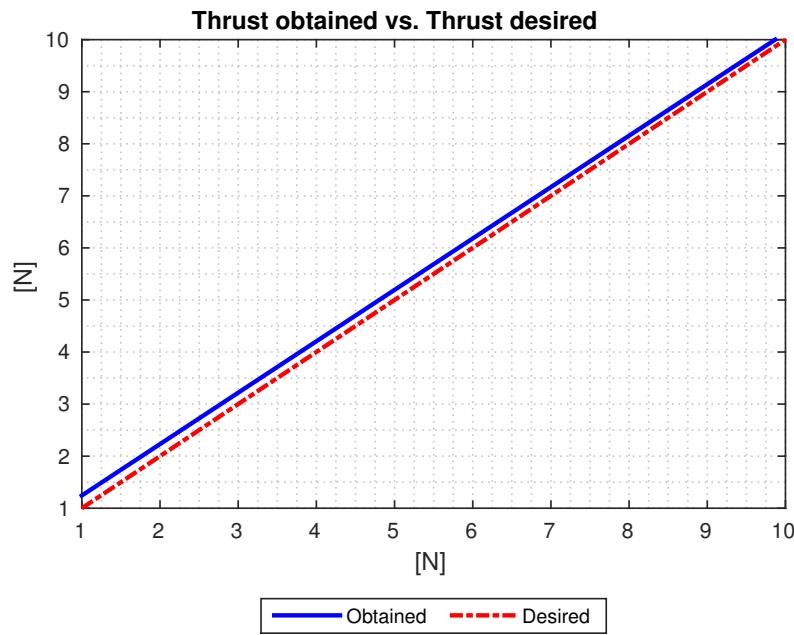


Figure 3.9: Thrust obtained vs Thrust desired

Due to the likelihood obtained between the two lines one can conclude that the parameters identified before ( $\hat{C}_T, \hat{C}_Q, \hat{C}_P, \hat{m}, \hat{q}$ ) are acceptable.

# Chapter 4

## Dynamics of a quadrotor helicopter

In this chapter the conventions, the formalism and the method used to find the dynamical model that describes the behavior of a quadrotor helicopter and the Simulink implementation are described.

Thanks to the implementation of the dynamical model in Simulink a simulator of the quadrotor helicopter with a representation in a three dimensional environment has been realized.

### 4.1 Axis systems

#### 4.1.1 Earth axes

Despite overwhelming evidence to the contrary, the flat and non-rotating Earth is assumed. The origin of the Earth axes is arbitrary. The origin could be the intersection of the equator, the prime meridian and mean sea level. So long as the origin, displacement along one or more of set of orthogonal axes attached rigidly to the origin describes the location of everything else. It is convenient to align these axes with the compass. The three axes are aligned with the axis labeled North, the axis labeled East, and the surface normal that points toward the center of the Earth, Down. These three axes are mutually perpendicular and, when referred to in the order N, E, D, form a right-handed coordinate system, see [3].

#### 4.1.2 Body axes

The body axes are a set of axes with origin at the center of gravity. The  $X$ ,  $Y$ , and  $Z$  axes form a right-handed system. The  $X$  axis lies in the plane of symmetry and generally points forward. The  $Y$  axis points to the right wing, normal to the plane of symmetry, and the  $Z$  axis points down.

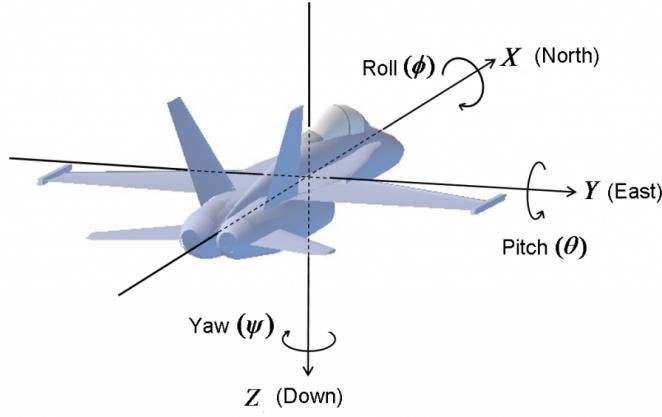


Figure 4.1: Body axes

## 4.2 Rotation formalism

### 4.2.1 Euler angles

Vectors can be rotated about any axes in any order for any number of times until the final orientation is achieved. Adopting the to-from notation, a rotation matrix from system E to system D might be named  $R_{D-E}$ . Thus, vector  $P_E$  in system E can be resolved to system D, that is  $P_D$  through the matrix operation:

$$P_D = R_{D-E} P_E. \quad (4.1)$$

Rotation matrices are written for  $\mathbb{R}^3$  vectors and may rotate a vector around one, two, or all three axes simultaneously. It is easiest to understand how these rotation matrices work by considering rotations about one axis at a time.

Rotation about the  $X$  axis does not change the component of the vector directed along the  $X$  axis, but it does change the  $Y$  and  $Z$  components. The rotation matrix that does this transformation is

$$R_X(\Phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\Phi) & \sin(\Phi) \\ 0 & -\sin(\Phi) & \cos(\Phi) \end{bmatrix}. \quad (4.2)$$

In a similar fashion, the following matrices perform rotations about the  $Y$  axis and the  $Z$  axis:

$$R_Y(\Theta) = \begin{bmatrix} \cos(\Theta) & 0 & -\sin(\Theta) \\ 0 & 1 & 0 \\ \sin(\Theta) & 0 & \cos(\Theta) \end{bmatrix}, \quad (4.3)$$

$$R_Z(\Psi) = \begin{bmatrix} \cos(\Psi) & \sin(\Psi) & 0 \\ -\sin(\Psi) & \cos(\Psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (4.4)$$

One measures a rotation angle about (around) the axis of its rotation. If you align your right thumb with the positive direction of the axis, then your fingers curl in the direction of positive angular displacement. The user fixes the location of 0 deg arbitrarily, but once it is fixed it must not vary.

The rotational matrices are orthonormal, which means the columns in the matrices, if considered as vectors, have unit magnitude (normal), and a dot product of column  $i$  and  $j$  ( $i \neq j$ ) equals zero (orthogonal). This property has a nice consequence. The inverses of matrices  $R_X$ ,  $R_Y$ , and  $R_Z$  are simply their transposes.

$$\begin{aligned} R_X^{-1} &= R_X^T, \\ R_Y^{-1} &= R_Y^T, \\ R_Z^{-1} &= R_Z^T. \end{aligned} \quad (4.5)$$

One can cascade rotations. For example, a rotation about the  $Z$  axis in system 1 can be followed by a rotation about the  $Y$  axis in system 2 and then about the  $X$  axis in system 3. In fact, any number of rotations can be put together in any order. When all is said and done, the resulting cascade can be reduced to a rotation about just three axes.

In the preceding discussion, the names  $\Phi$ ,  $\Theta$ , and  $\Psi$  have not been chosen arbitrarily.

In flight dynamics, one encounters a specific order of rotation using angles named  $\Psi$ ,  $\Theta$ , and  $\Phi$ , which represent rotation about the  $Z$  axis, the new intermediate  $Y$  axis, and the newer  $X$  axis, respectively. These angles are the so-called Euler angles, and their naming convention follows NASA standard notation. The matrix that performs this specific action is called an Euler rotation matrix and has the following definition:

$$T_{BE}(\Phi, \Theta, \Psi) = R_X(\Phi)R_Y(\Theta)R_Z(\Psi). \quad (4.6)$$

The subscripts B and E stand for “body” and “Earth”, respectively. The matrix  $T_{BE}$  resolves an Earth-based vector to body axes. Expanding the indicated matrix multiplication, the  $T_{BE}$  matrix has these elements:

$$T_{BE}(\Phi, \Theta, \Psi) = \begin{bmatrix} C_\Theta C_\Psi & C_\Theta S_\Psi & -S_\Theta \\ S_\Phi S_\Theta C_\Psi - C_\Phi S_\Psi & S_\Phi S_\Theta S_\Psi + C_\Phi C_\Psi & S_\Phi C_\Theta \\ C_\Phi S_\Theta C_\Psi + S_\Phi S_\Psi & C_\Phi S_\Theta S_\Psi - S_\Phi C_\Psi & C_\Phi C_\Theta \end{bmatrix}. \quad (4.7)$$

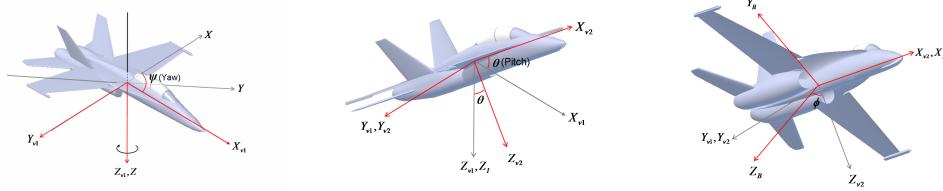


Figure 4.2: Euler angles

A shorthand notation has been adopted, which is  $C_\Theta = \cos(\Theta)$ ,  $S_\Theta = \sin(\Theta)$ ,  $T_\Theta = \tan(\Theta)$ , etc.

The Euler angles are the agreed-upon triple of angles that rotate a velocity or acceleration vector to body axes from Earth axes. The following, is the transformation, mathematically explained:

$$P_e = \begin{bmatrix} N \\ E \\ D \end{bmatrix}, V_e = \begin{bmatrix} \dot{N} \\ \dot{E} \\ \dot{D} \end{bmatrix} = \dot{P}_e, \quad (4.8)$$

$$V_b = T_{BE}(\Phi, \Theta, \Psi)V_e = \begin{bmatrix} u \\ v \\ w \end{bmatrix}, \quad (4.9)$$

where  $P_e$  is the position vector of the aircraft center of gravity in inertial (Earth) axes (the elements are the North, East, and Down position),  $V_e$  is the velocity of the aircraft with respect to the Earth, and  $V_b$  is the inertial linear velocity of the aircraft, resolved to body axes.

We can also define the vector of angular position of the aircraft body axes with respect to the Earth, resolved to the Earth where the elements are roll angle, pitch angle and yaw angle.

$$\alpha_e = \begin{bmatrix} \Phi \\ \Theta \\ \Psi \end{bmatrix}. \quad (4.10)$$

The angular orientation of the aircraft changes with time when an aircraft maneuvers. The Euler rates are a function of the Euler angles and body-axis angular rates.

Euler angles rotate Euler rates into body-axis angular rates. However, the transformation is different from the transformation for linear rates as discussed earlier. The process is complicated by the fact that the Euler angles themselves are involved in the transformation from Euler to body-axis rates. The process proceeds as described below.

Define the Euler rates as

$$\omega_e = \begin{bmatrix} \dot{\Phi} \\ \dot{\Theta} \\ \dot{\Psi} \end{bmatrix} = \dot{\alpha}_e . \quad (4.11)$$

and the body-axis rates as

$$\omega_b = \begin{bmatrix} p \\ q \\ r \end{bmatrix} . \quad (4.12)$$

To get the body axes rates from the Earth-axis rates, consider the Euler rates individually, resolve them individually to intermediate axes, and then finally to body axes. Define the Euler rate elemental vectors:

$$\omega_{\dot{\Phi}} = \begin{bmatrix} \dot{\Phi} \\ 0 \\ 0 \end{bmatrix}, \quad \omega_{\dot{\Theta}} = \begin{bmatrix} 0 \\ \dot{\Theta} \\ 0 \end{bmatrix}, \quad \omega_{\dot{\Psi}} = \begin{bmatrix} 0 \\ 0 \\ \dot{\Psi} \end{bmatrix} . \quad (4.13)$$

Rotate  $\omega_{\dot{\Psi}}$  through the angle  $\Theta$  about the  $Y$  axis, and add the results to the  $\omega_{\dot{\Theta}}$  vector. Rotate that sum about the  $X$  axis through the angle  $\Phi$ , and add the result to the  $\omega_{\dot{\Phi}}$  vector. The result is the vector of body-axis angular rates.

$$\omega_b = \omega_{\dot{\Phi}} + R_X(\Phi) \left[ \omega_{\dot{\Theta}} + R_Y(\Theta) \omega_{\dot{\Psi}} \right] . \quad (4.14)$$

Upon some expansion and rearrangement,

$$\omega_b = E(\Phi, \Theta) \omega_e = \begin{bmatrix} 1 & 0 & -S_\Theta \\ 0 & C_\Phi & S_\Phi C_\Theta \\ 0 & -S_\Phi & C_\Phi C_\Theta \end{bmatrix} \omega_e . \quad (4.15)$$

To get the Earth-axis rate in terms of body-axis rates, one must invert the transformation matrix  $E$ . Unlike the  $T_{BE}$  matrix, the inverse of  $E$  is not the transpose. Worse yet, the inverse is singular at pitch angles of  $\pm 90^\circ$ .

$$E^{-1}(\Phi, \Theta) = \begin{bmatrix} 1 & S_\Phi T_\Theta & C_\Phi T_\Theta \\ 0 & C_\Phi & -S_\Phi \\ 0 & S_\Phi/C_\Theta & C_\Phi/C_\Theta \end{bmatrix} . \quad (4.16)$$

This unfortunate singularity is called gimbal lock.

### 4.2.2 Quaternions

To avoid the gimbal lock, or in other words, lose the unique definition of the direction of the spin axis, one can use the quaternions, see [9].



Figure 4.3: Gimbal lock

A quaternion is a four-dimensional representation of a sphere that can be used to represent the orientation of a rigid body or a coordinate frame in three-dimensional space.

$$q = q_0 + q_1 i + q_2 j + q_3 k,$$

$$q = \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix}, \|q\| = 1, \quad (4.17)$$

$$i^2 = j^2 = k^2 = ijk = -1 .$$

The quaternion elements generate the following coordinate transformation:

$$T_{BE} = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1 q_2 + q_0 q_3) & 2(q_1 q_3 - q_0 q_2) \\ 2(q_1 q_2 - q_0 q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2 q_3 + q_0 q_1) \\ 2(q_1 q_3 + q_0 q_2) & 2(q_2 q_3 - q_0 q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix}. \quad (4.18)$$

The Euler angles representation of  $q_{BE}$  is defined by the equations

$$\Phi = \tan^{-1} \left( \frac{T_{BE}(2,3)}{T_{BE}(3,3)} \right),$$

$$\Theta = \sin^{-1} (-T_{BE}(1,3)), \quad (4.19)$$

$$\Psi = \tan^{-1} \left( \frac{T_{BE}(1,2)}{T_{BE}(1,1)} \right).$$

The quaternion conjugate, denoted by  $*$ , can be used to swap the relative frames described by an orientation. For example, an arbitrary orientation of frame B relative to frame A can be represented by the quaternion  $q_{AB}$ , and its conjugate  $q_{AB}^*$  describes the orientation of frame A relative to frame B ( $q_{BA}$ ). The conjugate of  $q_{AB}$  is defined by the equation

$$q_{AB}^* = q_{BA} = \begin{bmatrix} q_0 \\ -q_1 \\ -q_2 \\ -q_3 \end{bmatrix}. \quad (4.20)$$

The quaternion product, denoted by  $\otimes$ , can be used to define compound orientations. For example, for two orientations described by  $q_{AB}$  and  $q_{BC}$ , the compounded orientation  $q_{AC}$  can be defined by the equation

$$q_{AC} = q_{BC} \otimes q_{AB}. \quad (4.21)$$

For two quaternions,  $a$  and  $b$ , the quaternion product can be determined using the Hamilton rule and defined as equation

$$\begin{aligned} a \otimes b &= \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} \otimes \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix} \\ &= \begin{bmatrix} a_1b_1 - a_2b_2 - a_3b_3 - a_4b_4 \\ a_1b_2 + a_2b_1 + a_3b_4 - a_4b_3 \\ a_1b_3 - a_2b_4 + a_3b_1 + a_4b_2 \\ a_1b_4 + a_2b_3 - a_3b_2 + a_4b_1 \end{bmatrix}. \end{aligned} \quad (4.22)$$

One must take into account that a quaternion product is not commutative, so  $a \otimes b \neq b \otimes a$ .

A three dimensional vector can be rotated by a quaternion using the relationship described by the following equation.

$$v_B = q_{AB} \otimes v_A \otimes q_{AB}^* \quad (4.23)$$

Where  $v_A$  and  $v_B$  are the same vector described in frame A and frame B respectively and each vector contains a 0 inserted as the first element to make them 4 element row vectors.

The quaternion derivative describing the rate of change of orientation of the Earth frame relative to the body frame can be calculated by the equation

$$\begin{aligned} \omega &= \begin{bmatrix} 0 \\ \omega_b \end{bmatrix}, \\ \dot{q}_{BE} &= \frac{1}{2}q_{BE} \otimes \omega. \end{aligned} \quad (4.24)$$

In expanded form:

$$\begin{bmatrix} \dot{q}_0 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 0 & -p & -q & -r \\ p & 0 & r & -q \\ q & -r & 0 & p \\ r & q & -p & 0 \end{bmatrix} \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix}. \quad (4.25)$$

## 4.3 Kinematics and flight dynamics

### 4.3.1 Differentiation in a moving axis system

Elementary calculus teaches that the linear velocity of a particle is the time derivative of position:

$$v = \lim_{\Delta t \rightarrow 0} \left( \frac{\Delta P}{\Delta t} \right). \quad (4.26)$$

A second application gives acceleration:

$$a = \lim_{\Delta t \rightarrow 0} \left( \frac{\Delta v}{\Delta t} \right). \quad (4.27)$$

When the motion is rectilinear, these definitions are easy to apply. When the particle follows a curved path, the position vector changes in length and direction.

Let  $S$  be the position vector of a particle relative to the origin of a system that is moving with angular rate  $\omega_b$  relative to the Earth frame, that is,

$$S = \begin{bmatrix} x \\ y \\ z \end{bmatrix}, \quad \omega_b = \begin{bmatrix} p \\ q \\ r \end{bmatrix}. \quad (4.28)$$

So

$$\frac{\partial S}{\partial t} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} u \\ v \\ w \end{bmatrix}. \quad (4.29)$$

The following expressions give the total velocity and acceleration in the Earth frame:

$$\frac{dS}{dt} = \frac{\partial S}{\partial t} + \omega_b \times S, \quad (4.30)$$

$$\frac{d^2S}{dt^2} = \frac{\partial}{\partial t} \left( \frac{dS}{dt} \right) + \omega_b \times \left( \frac{dS}{dt} \right) = \frac{\partial^2 S}{\partial t^2} + 2\omega_b \times \frac{\partial S}{\partial t} + \frac{\partial \omega_b}{\partial t} \times S + \omega_b \times (\omega_b \times S). \quad (4.31)$$

Dot notation is now employed. Thus,

$$\frac{dS}{dt} = \dot{S} + \omega_b \times S, \quad (4.32)$$

$$\frac{d^2S}{dt^2} = \ddot{S} + \dot{\omega}_b \times S + 2\omega_b \times \dot{S} + \omega_b \times (\omega_b \times S). \quad (4.33)$$

The obtained expression represents the total linear acceleration of a particle of mass with respect to inertial space as it moves in a frame that rotates with respect to inertial space.

### 4.3.2 Equation of motion - Linear motion

To write the linear motion equations one has to take into account the second Newton's Law. Linear momentum is the product of mass and velocity, and so it is a vector. Forces causes momentum to change, and so

$$F = \frac{d(mV_b)}{dt}. \quad (4.34)$$

Applying the equations obtained in the previous paragraph and the chain rule to the preceding expression, and expanding the derivative:

$$F = \left( \frac{dm}{dt} \right) V_b + m \left( \frac{\partial V_b}{\partial t} + \omega_b \times V_b \right) = \dot{m} V_b + m \dot{V}_b + \omega_b \times (mV_b). \quad (4.35)$$

The product of the magnitude of mass changes with time and its current velocity is usually considered small, compared to the product of mass and the total acceleration. The obtained equation represent the rate of change of linear momentum as a result of applied forces. Those forces are aerodynamics, gravity, and others that can be lumped into a category called externally applied loads. Let the vector  $F_{ext}$  represent the external forces:

$$F_{ext} = \begin{bmatrix} F_X \\ F_Y \\ F_Z \end{bmatrix}. \quad (4.36)$$

If the mass is constant, then, expressed concisely, the equation of linear motion is

$$m \dot{V}_b + \omega_b \times (mV_b) = F_{ext}. \quad (4.37)$$

### 4.3.3 Equation of motion - Angular motion

Begin with the equation of linear acceleration, repeated here in vector notation and then rearranged slightly differently in expanded notation:

$$\frac{d^2 S}{dt^2} = \ddot{S} + \dot{\omega}_b \times S + 2\omega_b \times \dot{S} + \omega_b \times (\omega_b \times S). \quad (4.38)$$

After expansion and rearrangement,

$$\begin{aligned} \ddot{x} &= \dot{u} + \dot{q} z - \dot{r} y + q(w + py - qx) - r(v + rx - pz), \\ \ddot{y} &= \dot{v} + \dot{r} x - \dot{p} z + r(u + qz - ry) - p(w + py - qx), \\ \ddot{z} &= \dot{w} + \dot{p} y - \dot{q} x + p(v + rx - pz) - q(u + qz - ry). \end{aligned} \quad (4.39)$$

The left-hand side of the preceding expressions, after multiplying both sides by the mass of the  $j$ -th differential mass element, is an inertial force:

$$\begin{aligned} (dF_X)_j &= dm_j \ddot{x}, \\ (dF_Y)_j &= dm_j \ddot{y}, \\ (dF_Z)_j &= dm_j \ddot{z}. \end{aligned} \quad (4.40)$$

If the differential element of mass is located at point  $S_j$ , the mass resists an angular force (called a moment) about all three axes. The differential inertial moments are

$$dM_j = S_j \times dF_j, \quad (4.41)$$

or

$$\begin{aligned} (dM_X)_j &= (dF_Z)_j y_j - (dF_Y)_j z_j, \\ (dM_Y)_j &= (dF_X)_j z_j - (dF_Z)_j x_j, \\ (dM_Z)_j &= (dF_Y)_j x_j - (dF_X)_j y_j. \end{aligned} \quad (4.42)$$

From this point on, all forces and moments act on the  $j$ -th mass, and so the subscript  $j$  is dropped.

By definition, the origin of the coordinate system is the center of gravity; therefore,

$$\begin{aligned} \sum(x) dm &= 0, \\ \sum(y) dm &= 0, \\ \sum(z) dm &= 0. \end{aligned} \quad (4.43)$$

Combining the obtained equations, what survive are the equations of linear motion:

$$\begin{aligned} m \ddot{x} &= m(\dot{u} + qw - rv), \\ m \ddot{y} &= m(\dot{v} + ru - pw), \\ m \ddot{z} &= m(\dot{w} + pv - qu). \end{aligned} \quad (4.44)$$

The definition of inertia tensor  $I_n$  can be introduced

$$I_n = \begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ I_{xy} & I_{yy} & -I_{yz} \\ -I_{xz} & -I_{yz} & I_{zz} \end{bmatrix}. \quad (4.45)$$

Where the terms are defined as follow

$$\begin{aligned} \int (xy) dm &= I_{xy}, & \int (xz) dm &= I_{xz}, & \int (yz) dm &= I_{yz}, \\ \int (y^2 + z^2) dm &= I_{xx}, & \int (x^2 + z^2) dm &= I_{yy}, & \int (x^2 + y^2) dm &= I_{zz}. \end{aligned} \quad (4.46)$$

If the body frame is coincident with the symmetry axes of the aircraft body, one can assume the inertia tensor as a diagonal matrix.

Then, after tedious expansion, the equations of angular motion appear:

$$\begin{aligned} L &= I_{xx} \dot{p} + (I_{zz} - I_{yy}) qr, \\ M &= I_{yy} \dot{q} + (I_{xx} - I_{zz}) pr, \\ N &= I_{zz} \dot{r} + (I_{yy} - I_{xx}) pq, \end{aligned} \quad (4.47)$$

where  $L$ ,  $M$  and  $N$  are the moments applied on the body-axis  $X$ ,  $Y$  and  $Z$  respectively.

With effort, this set of equations reduces to a much simpler vector and matrix notation. Let the vector  $M_{ext}$  represent the external moments:

$$M_{ext} = \begin{bmatrix} L \\ M \\ N \end{bmatrix}, \quad (4.48)$$

and recall the definition of body-axis angular velocity vector, then, expressed concisely, the equation of angular motion is

$$I_n \dot{\omega}_b + \omega_b \times (I_n \omega_b) = M_{ext}. \quad (4.49)$$

Notice the similarity between the linear equation and angular equation

$$\begin{aligned} m \dot{V}_b + \omega_b \times (m V_b) &= F_{ext}, \\ I_n \dot{\omega}_b + \omega_b \times (I_n \omega_b) &= M_{ext}. \end{aligned} \quad (4.50)$$

These expressions are the generalized equations of motion. The  $3 \times 1$  vectors  $F_{ext}$  and  $M_{ext}$  are the total external forces and moments respectively that are applied to the body at its center of gravity.

#### 4.3.4 Overall states of an aircraft

The usual expressions of the equations of motion use the concept of a state vector. A state vector contains all of the dynamic variables and their time derivatives up to,

but not including, the highest derivative. The states are those necessary to define the highest derivatives.

The ordering of the states in the state vector is not important from a mathematical standpoint, although certain computational economies are realized if three-element sub-vectors remained grouped as just defined. Once the order of the states in the state vector has been selected, the order must be preserved.

The state vector is ordered as following:

$$x = \begin{bmatrix} P_e \\ V_b \\ \omega_b \\ \alpha_e \end{bmatrix} = \begin{bmatrix} N \\ E \\ D \\ u \\ v \\ w \\ p \\ q \\ r \\ \Phi \\ \Theta \\ \Psi \end{bmatrix}. \quad (4.51)$$

Therefore, the derivative of the state vector is defined:

$$\dot{x} = \begin{bmatrix} \dot{N} \\ \dot{E} \\ \dot{D} \\ \dot{u} \\ \dot{v} \\ \dot{w} \\ \dot{p} \\ \dot{q} \\ \dot{r} \\ \dot{\Phi} \\ \dot{\Theta} \\ \dot{\Psi} \end{bmatrix} = \begin{bmatrix} T_{BE}^T V_b \\ -\omega_b \times V_b + F_{ext}/m \\ I_n^{-1} (-\omega_b \times I_n \omega_b + M_{ext}) \\ E^{-1} \omega_b \end{bmatrix}. \quad (4.52)$$

#### 4.3.5 External forces and moments

After the introduction of rigid body dynamics one has to consider the actual geometry of the quadrotor helicopter in order to define the forces and moments that have to be considered and the application point. In Figure 4.4 the chosen configuration (X-configuration), the label of each propeller, and its rotation direction

are shown.

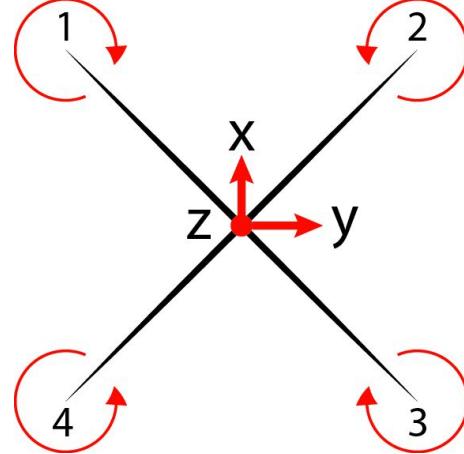


Figure 4.4: Quadcopter configuration

Each propeller produces a thrust and a torque proportional to the square of the rotational speed (Section 3.1)

$$\begin{aligned} T_i &= K_T \Omega_i^2, & K_T &= C_T \rho A R^2, \\ Q_i &= K_Q \Omega_i^2, & K_Q &= C_Q \rho A R^3. \end{aligned} \quad (4.53)$$

where  $C_T$  and  $C_Q$  are the thrust and torque coefficients,  $\rho$  is the air density,  $A$  and  $R$  are the area of the propeller disk and its radius respectively, and  $\Omega$  is the rotational speed. The distance between the center of gravity and the  $j$ -th propeller is  $b$ .

Now it is possible to write the equations of the forces and the moments produced by the four propellers

$$F_{props} = - \begin{bmatrix} 0 \\ 0 \\ K_T (\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \end{bmatrix}, \quad (4.54)$$

$$M_{props} = \begin{bmatrix} K_T \frac{b}{\sqrt{2}} (\Omega_1^2 - \Omega_2^2 - \Omega_3^2 + \Omega_4^2) \\ K_T \frac{b}{\sqrt{2}} (\Omega_1^2 + \Omega_2^2 - \Omega_3^2 - \Omega_4^2) \\ K_Q (-\Omega_1^2 + \Omega_2^2 - \Omega_3^2 + \Omega_4^2) \end{bmatrix}. \quad (4.55)$$

These forces and moments can be rearranged in order to realize the so called “mixer matrix” of the motors ( $\chi$ ).  $\chi$  is a matrix that relates the required thrust and moments around each axis to the rotational speed of the propellers (that are the actual control input of the quadrotor).

$$\begin{aligned} \begin{Bmatrix} T \\ L \\ M \\ N \end{Bmatrix} &= \begin{bmatrix} K_T & K_T & K_T & K_T \\ K_T \frac{b}{\sqrt{2}} & -K_T \frac{b}{\sqrt{2}} & -K_T \frac{b}{\sqrt{2}} & K_T \frac{b}{\sqrt{2}} \\ K_T \frac{b}{\sqrt{2}} & K_T \frac{b}{\sqrt{2}} & -K_T \frac{b}{\sqrt{2}} & -K_T \frac{b}{\sqrt{2}} \\ -K_Q & K_Q & -K_Q & K_Q \end{bmatrix} \begin{Bmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \end{Bmatrix} \quad (4.56) \\ &= \chi \begin{Bmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \end{Bmatrix}, \end{aligned}$$

$$\begin{Bmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \end{Bmatrix} = \chi^{-1} \begin{Bmatrix} T \\ L \\ M \\ N \end{Bmatrix} = \begin{bmatrix} \frac{1}{4K_T} & \frac{\sqrt{2}}{4K_T b} & \frac{\sqrt{2}}{4K_T b} & -\frac{1}{4K_Q} \\ \frac{1}{4K_T} & -\frac{\sqrt{2}}{4K_T b} & \frac{\sqrt{2}}{4K_T b} & \frac{1}{4K_Q} \\ \frac{1}{4K_T} & -\frac{\sqrt{2}}{4K_T b} & -\frac{\sqrt{2}}{4K_T b} & \frac{1}{4K_Q} \\ \frac{1}{4K_T} & \frac{\sqrt{2}}{4K_T b} & -\frac{\sqrt{2}}{4K_T b} & -\frac{1}{4K_Q} \end{bmatrix} \begin{Bmatrix} T \\ L \\ M \\ N \end{Bmatrix}. \quad (4.57)$$

Another force that is actually acting on the center of gravity is the gravitational force that must be rotated into the body frame.

$$F_g = T_{BE}(\Phi, \Theta, \Psi) \begin{Bmatrix} 0 \\ 0 \\ mg \end{Bmatrix} = \begin{Bmatrix} -S_\Theta \\ S_\Phi C_\Theta \\ C_\Phi C_\Theta \end{Bmatrix} mg. \quad (4.58)$$

The last forces one has to consider are the aerodynamic damp caused by the rotating propellers moving through the air (the aerodynamic drag caused by the structure of the quadrotor is neglected, because the quadrotor's body is really thin). Assuming that the quadrotor has to fly in near hover condition, the drag produced by linear translations can be neglected. Only the aerodynamic damp proportional to  $\omega_b = [p \ q \ r]^T$  is considered. Another important assumption is to consider the moments relative to an axis only proportional to the rotational speed around that axis (decoupled moments). As George Box said: "All models are wrong but some are useful".

$$M_{damp} = \begin{bmatrix} \frac{\partial L}{\partial p} & 0 & 0 \\ 0 & \frac{\partial M}{\partial q} & 0 \\ 0 & 0 & \frac{\partial N}{\partial r} \end{bmatrix} \begin{Bmatrix} p \\ q \\ r \end{Bmatrix}. \quad (4.59)$$

The derivatives  $\frac{\partial L}{\partial p}$ ,  $\frac{\partial M}{\partial q}$  and  $\frac{\partial N}{\partial r}$  are called stability derivatives and they have an analytical form, which is taken from the helicopter's dynamics background (see [13–15]). Because of the geometry of the quadrotor  $\frac{\partial L}{\partial p} = \frac{\partial M}{\partial q}$ , the derivatives can be written as the following equations

$$\begin{aligned}\frac{\partial L}{\partial p} &= -4\rho AR^2\Omega^2 \frac{\partial C_T}{\partial p} \frac{b}{\sqrt{2}}, \\ \frac{\partial C_T}{\partial p} &= \frac{C_{L\alpha}}{8} \frac{\sigma}{R\Omega} \frac{b}{\sqrt{2}},\end{aligned}\quad (4.60)$$

where  $\sigma$  is the solidity ratio that can be defined as

$$\sigma = \frac{A_b}{A}. \quad (4.61)$$

with  $A_b$  the total blade area and  $A$  the disk area.

And finally,  $C_{L\alpha}$  is the slope of the thrust coefficient curve ( $C_L$ ). Because the propeller's blade is thin and the actual airfoil is not known, one can assume it as a flat plate, the bi-dimensional analytical solution is equal to  $2\pi$ . In case of known airfoil the guess value can be improved.

Due to the slow rotational speed about the  $Z$  axis with respect to the other two axes, its damping moment can be neglected. Which means  $\frac{\partial N}{\partial r} = 0$ .

Summarizing, the overall forces and moments described before:

$$\begin{aligned}F_{ext} &= F_g + F_{props}, \\ M_{ext} &= M_{damp} + M_{props}.\end{aligned}\quad (4.62)$$

## 4.4 Conclusions

The overall model which describes the behavior of a quadrotor helicopter is the following

$$\begin{aligned}m \dot{V}_b + \omega_b \times (mV_b) &= F_g + F_{props} \\ I_n \dot{\omega}_b + \omega_b \times (I_n \omega_b) &= M_{damp} + M_{props}\end{aligned}\quad (4.63)$$

There are several unknown parameters. In order to find them one has to realize some experimental tests.

$$\Theta = \left[ I_n, C_T, C_Q, \frac{\partial L}{\partial p}, \frac{\partial M}{\partial q} \right]. \quad (4.64)$$

Nevertheless it is possible to realize in Simulink the entire simulator of the quadrotor dynamics (Figures 4.5-4.6). Thanks to this simulator it is possible to test the control architecture and tune to first approximation the coefficients of the regulators. In addition, a graphical and three dimensional environment has been realized (Figure 4.7).

#### 4. Dynamics of a quadrotor helicopter

---

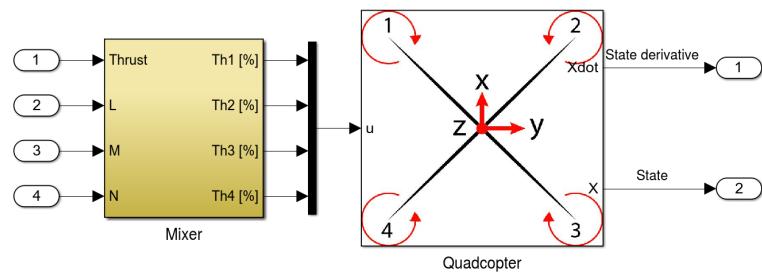


Figure 4.5: Mixer and Quadcopter

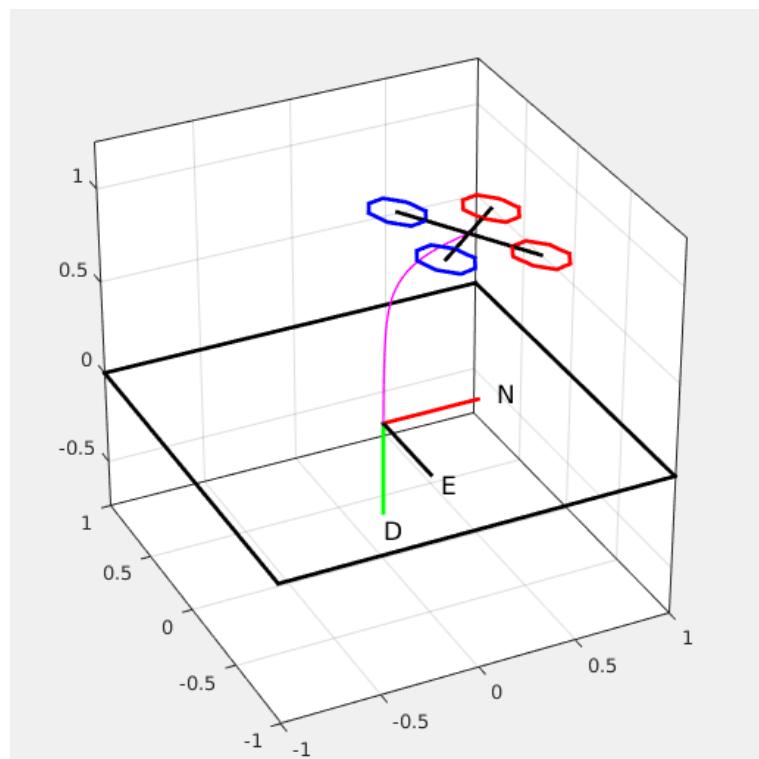


Figure 4.7: Three dimensional representation

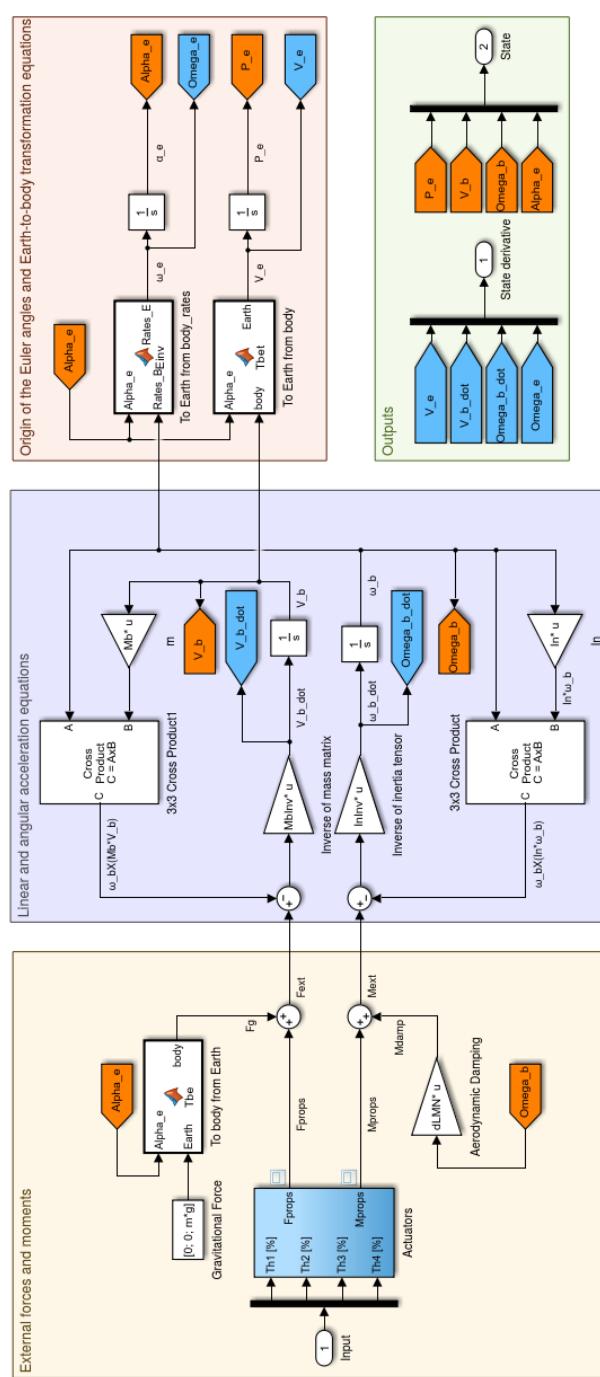


Figure 4.6: Quadcopter simulator



# Chapter 5

## Attitude estimation

Before introduce any kind of control (position, velocity, attitude and so on) one has before to obtain a feedback from the quadrotor that describe its state. Several stase estimator or state observer are actually used such as the Kalman Filter, but the problem with such estimator are usually the complexity from the computational point of view.

In this Chapter is presented a reduced state estimator (can estimate only the attitude and not the position or translational velocity) that is computationally inexpensive and requires few scalar arithmetic operations each filter update.

### 5.1 Madgwick filter

The Madgwick filter (see [9]) is a sensor-fusion filter applicable to IMU and MARG sensors arrays that can be used in order to estimate the attitude of a rigid body with respect to an Earth reference system. The MARG implementation incorporates magnetic distortion and gyroscope bias drift compensation.

The filter uses a quaternion representation for the attitude, allowing accelerometer and magnetometer data to be used in a sensor-fusion technique (an analytically derived and optimized gradient-descent algorithm) to compute the direction of the gyroscope measurement error as a quaternion derivative.

In the following it is assumed that the sensor reference frame is coincident with the reference frame of the rigid body in which the attitude has to be estimated.

#### 5.1.1 Filter derivation

##### 5.1.1.1 Orientation from angular rate

A tri-axis gyroscope will measure the angular rate about the  $x$ ,  $y$  and  $z$  axes of the sensor frame, termed  $\omega_x$ ,  $\omega_y$  and  $\omega_z$  respectively. If these variables (in rad/s) are arranged into the vector  $\omega$  defined as

$$\omega = \begin{bmatrix} 0 \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}, \quad (5.1)$$

the quaternion derivative describing the rate of change of orientation of the Earth frame relative to the body frame  $\dot{q}_{BE}$  can be calculated

$$\dot{q}_{BE} = \frac{1}{2} q_{BE} \otimes \omega. \quad (5.2)$$

In order to keep the notation as clear as possible, it has been decided to avoid the usage of the sub-script *BE* because all the following quaternions represent the orientation of the Earth frame relative to the body frame.

The orientation of the Earth frame relative to the body frame at time  $t$ ,  $q_{\omega,t}$ , can be computed by numerically integrating the quaternion derivative  $\dot{q}_{\omega,t}$  as described by the equations

$$\dot{q}_{\omega,t} = \frac{1}{2} \hat{q}_{(t-1)} \otimes \omega_t, \quad (5.3)$$

$$q_{\omega,t} = \hat{q}_{(t-1)} + \dot{q}_{\omega,t} \Delta t, \quad (5.4)$$

provided that initial conditions are known. In these equations,  $\omega_t$  is the angular rate measured at time  $t$ ,  $\Delta t$  is the sampling period and  $\hat{q}_{(t-1)}$  is the previous estimate of the orientation. The sub-script  $\omega$  indicates that the quaternion is calculated from angular rates.

### 5.1.1.2 Orientation from vector observation

As described before, a tri-axis accelerometer will measure the magnitude and direction of the field of gravity in the sensor frame compounded with linear accelerations due to motion of the sensor, and similarly, a tri-axis magnetometer will measure the magnitude and direction of the earth's magnetic field in the sensor frame compounded with local magnetic flux.

In the context of an orientation filter, it will initially be assumed that an accelerometer will measure only gravity, and that a magnetometer will measure only the Earth's magnetic field.

If the direction of the Earth's field is known in the Earth frame, a measurement of the field's direction within the body frame will allow an orientation of the sensor frame relative to the Earth frame to be calculated. However, for any given measurement there will not be a unique body orientation solution, instead there will be infinite solutions represented by all those orientations achieved by the rotation of the true

orientation around an axis parallel with the field. In some applications it may be acceptable to use an Euler angle representation allowing an incomplete solution to be found as two known Euler angles and one unknown; the unknown angle being the rotation around an axis parallel with direction of the field.

A quaternion representation require a complete solution to be found. This may be achieved through the formulation of an optimisation problem where an orientation of the sensor,  $q$ , is the one which aligns a predefined reference direction of the field in the Earth frame,  $d_E$ , with the measured direction of the field in the body frame,  $s_B$ , using the rotation operation.

Therefore  $q$  may be found as the solution to the optimisation function

$$\min_{q \in \mathbb{R}^4} f(q, d_E, s_B), \quad (5.5)$$

$$f(q, d_E, s_B) = q^* \otimes d_E \otimes q - s_B, \quad (5.6)$$

where equation (5.6) defines the objective function. The components of each vector are defined in equations

$$q = \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix}, \quad (5.7)$$

$$d_E = \begin{bmatrix} 0 \\ d_x \\ d_y \\ d_z \end{bmatrix}, \quad (5.8)$$

$$s_B = \begin{bmatrix} 0 \\ s_x \\ s_y \\ s_z \end{bmatrix}. \quad (5.9)$$

Many optimisation algorithms exist but the gradient descent algorithm is one of the simplest to both implement and compute.

$$q_{k+1} = q_k - \mu \frac{\nabla f(q, d_E, s_B)}{\|\nabla f(q, d_E, s_B)\|}, k = 0, 1, 2, \dots, n, \quad (5.10)$$

$$\nabla f(q, d_E, s_B) = J^T(q, d_E) f(q, d_E, s_B), \quad (5.11)$$

$$f(q, d_E, s_B) = \begin{bmatrix} 2d_x (\frac{1}{2} - q_2^2 - q_3^2) + 2d_y (q_0q_3 + q_1q_2) + 2d_z (q_1q_3 - q_0q_2) - s_x \\ 2d_x (q_1q_2 - q_0q_3) + 2d_y (\frac{1}{2} - q_1^2 - q_3^2) + 2d_z (q_0q_1 + q_2q_3) - s_y \\ 2d_x (q_0q_2 + q_1q_3) + 2d_y (q_2q_3 - q_0q_1) + 2d_z (\frac{1}{2} - q_1^2 - q_2^2) - s_z \end{bmatrix}, \quad (5.12)$$

$$J(q, d_E) = \begin{bmatrix} 2d_y q_3 - 2d_z q_2 & 2d_y q_2 + 2d_z q_3 \\ -2d_x q_3 + 2d_z q_1 & 2d_x q_2 - 4d_y q_1 + 2d_z q_0 \\ 2d_x q_2 - 2d_y q_1 & 2d_x q_3 - 2d_y q_0 - 4d_z q_1 \\ -4d_x q_2 + 2d_y q_1 - 2d_z q_0 & -4d_x q_3 + 2d_y q_0 + 2d_z q_1 \\ 2d_x q_1 + 2d_z q_3 & -2d_x q_0 - 4d_y q_3 + 2d_z q_2 \\ 2d_x q_0 + 2d_y q_3 - 4d_z q_2 & 2d_x q_1 + 2d_y q_2 \end{bmatrix}. \quad (5.13)$$

Equation (5.10) describes the gradient descent algorithm for  $n$  iterations resulting in an orientation estimation of  $q_{n+1}$  based on an 'initial guess' orientation  $q_0$  and a step-size  $\mu$ . Equation (5.11) computes the gradient of the solution surface defined by the objective function and its Jacobian; simplified to the 3 row vectors defined by equations (5.12) and (5.13) respectively.

Equations (5.10)-(5.13) describe the general form of the algorithm applicable to a field predefined in any direction. However, if the direction of the field can be assumed to only have components within 1 or 2 of the principal axis of the global coordinate frame then the equations simplify. An appropriate convention would be to assume that the direction of gravity defines the vertical,  $z$  axis

$$g_E = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \quad (5.14)$$

$$a_B = \begin{bmatrix} 0 \\ a_x \\ a_y \\ a_z \end{bmatrix}. \quad (5.15)$$

Substituting  $g_E$  and normalised accelerometer measurement  $a_B$  for  $d_E$  and  $s_B$  respectively, in equations (5.12) and (5.13) yields equations

$$f_g(q, a_B) = \begin{bmatrix} 2(q_1q_3 - q_0q_2) - a_x \\ 2(q_0q_1 + q_2q_3) - a_y \\ 2(\frac{1}{2} - q_1^2 - q_2^2) - a_z \end{bmatrix}, \quad (5.16)$$

$$J_g(q) = \begin{bmatrix} -2q_2 & 2q_3 & -2q_0 & 2q_1 \\ 2q_1 & 2q_0 & 2q_3 & 2q_2 \\ 0 & -4q_1 & -4q_2 & 0 \end{bmatrix}. \quad (5.17)$$

The Earth's magnetic field can be considered to have components in one horizontal axis and the vertical axis (the vertical is caused by the inclination of the field). This can be represented by equations

$$b_E = \begin{bmatrix} 0 \\ b_x \\ 0 \\ b_z \end{bmatrix}, \quad (5.18)$$

$$m_B = \begin{bmatrix} 0 \\ m_x \\ m_y \\ m_z \end{bmatrix}. \quad (5.19)$$

Substituting  $b_E$  and normalised magnetometer measurement  $m_B$  for  $d_E$  and  $s_B$  respectively, in equations (5.12) and (5.13) yields equations

$$f_b(q, b_E, m_B) = \begin{bmatrix} 2b_x(\frac{1}{2} - q_2^2 - q_3^2) + 2b_z(q_1q_3 - q_0q_2) - m_x \\ 2b_x(q_1q_2 - q_0q_3) + 2b_z(q_0q_1 + q_2q_3) - m_y \\ 2b_x(q_0q_2 + q_1q_3) + 2b_z(\frac{1}{2} - q_1^2 - q_2^2) - m_z \end{bmatrix}, \quad (5.20)$$

$$J_b(q, b_E) = \begin{bmatrix} -2b_zq_2 & 2b_zq_3 \\ -2b_xq_3 + 2b_zq_1 & 2b_xq_2 + 2b_zq_0 \\ 2b_xq_2 & 2b_xq_3 - 4b_zq_1 \\ -4b_xq_2 - 2b_zq_0 & -4b_xq_3 + 2b_zq_1 \\ 2b_xq_1 + 2b_zq_3 & -2b_xq_0 + 2b_zq_2 \\ 2b_xq_0 - 4b_zq_2 & 2b_xq_1 \end{bmatrix}. \quad (5.21)$$

As has already been discussed, the measurement of gravity or the earth's magnetic field alone will not provide a unique orientation of the body. To do so, the measurements and reference directions of both fields may be combined as described by equations

$$f_{g,b}(q, a_B, b_E, m_B) = \begin{bmatrix} f_g(q, a_B) \\ f_b(q, b_E, m_B) \end{bmatrix}, \quad (5.22)$$

$$J_{g,b}(q, b_E) = \begin{bmatrix} J_g(q) \\ J_b(q, b_E) \end{bmatrix}. \quad (5.23)$$

The combination of the solution surface created by the objective functions in equations (5.16) and (5.20) have a minimum defined by a line, the solution surface defined by equation (5.22) has a minimum defined by a single point, provided that  $b_x \neq 0$ .

A conventional approach to optimisation would require multiple iterations of equation (5.10) to be computed for each new iteration and corresponding sensor measurements. Efficient algorithms would also require the step-size  $\mu$  to be adjusted at each iteration to an optimal value, usually obtained based on the second derivative of the objective function, the Hessian. However, these requirements considerably increase the computational load of the algorithm and are not necessary in this application. It is acceptable to compute one iteration per time sample provided that the convergence rate governed by  $\mu_t$  is equal or greater than the physical rate of change of orientation.

The Equation

$$q_{\nabla,t} = \hat{q}_{(t-1)} - \mu_t \frac{\nabla f}{\|\nabla f\|}, \quad (5.24)$$

calculates the estimated orientation  $q_{\nabla,t}$ , computed at time  $t$  based on a previous estimate of orientation  $\hat{q}_{(t-1)}$  and the objective function gradient  $\nabla f$  defined by sensor measurements  $a_{B,t}$  and  $m_{B,t}$  sampled at time  $t$ . The form of  $\nabla f$  is chosen according to the sensors in use, as shown in

$$\nabla f = \begin{cases} J_g^T(\hat{q}_{(t-1)}) f_g(\hat{q}_{(t-1)}, a_{B,t}) \\ J_{g,b}^T(\hat{q}_{(t-1)}, b_E) f_{g,b}(q, a_{B,t}, b_E, m_{B,t}) \end{cases}. \quad (5.25)$$

The symbol  $\nabla$  indicates that the quaternion is calculated from gradient descent algorithm.

An optimal value of  $\mu_t$  can be defined as that which ensures the convergence rate of  $q_{\nabla,t}$  is limited to the physical orientation rate as this avoids overshooting due an unnecessarily large step size. Therefore  $\mu_t$  can be calculated as equation

$$\mu_t = \alpha \left\| \dot{q}_{\omega,t} \right\| \Delta t, \alpha > 1, \quad (5.26)$$

where  $\Delta t$  is the sampling period and  $\dot{q}_{\omega,t}$  is the physical orientation rate measured by gyroscopes and  $\alpha$  is an augmentation of  $\mu$  to account for noise in accelerometer and magnetometer measurements.

### 5.1.2 Filter fusion derivation

An estimated orientation of the body frame relative to the Earth frame,  $\hat{q}_t$ , is obtained through the fusion of the orientation calculations,  $q_{\omega,t}$  and  $q_{\nabla,t}$ , calculated using equations (5.4) and (5.24) respectively. The fusion of  $q_{\omega,t}$  and  $q_{\nabla,t}$ , is described by equation

$$\hat{q}_t = \gamma_t q_{\nabla,t} + (1 - \gamma_t) q_{\omega,t}, \quad 0 \leq \gamma_t \leq 1, \quad (5.27)$$

where  $\gamma_t$  and  $(1 - \gamma_t)$  are weights applied to each orientation calculation.

An optimal value of  $\gamma_t$  can be defined as the one which ensures the weighted divergence of  $q_\omega$  is equal to the weighted convergence of  $q_\nabla$ . This is represented by equation

$$(1 - \gamma_t) \beta = \gamma_t \frac{\mu_t}{\Delta t}, \quad (5.28)$$

where  $\frac{\mu_t}{\Delta t}$  is the convergence rate of  $q_\nabla$  and  $\beta$  is the divergence rate of  $q_\omega$  expressed as the magnitude of a quaternion derivative corresponding to the gyroscope measurement error. Equation (5.28) can be rearranged to define  $\gamma_t$  as equation

$$\gamma_t = \frac{\beta}{\frac{\mu_t}{\Delta t} + \beta}. \quad (5.29)$$

Equations (5.27) and (5.29) ensure the optimal fusion of  $q_{\omega,t}$  and  $q_{\nabla,t}$  assuming that the convergence rate of  $q_\nabla$  governed by  $\alpha$  is equal or greater than the physical rate of change of orientation. Therefore  $\alpha$  has no upper bound, if  $\alpha$  is assumed to be very large then  $\mu_t$ , defined by the equation (5.26), it also becomes very large and the orientation filter equations simplify. A large value of  $\mu_t$  used in equation (5.24) means that  $\hat{q}_{(t-1)}$  becomes negligible and the equation can be re-written as equation

$$q_{\nabla,t} \cong -\mu_t \frac{\nabla f}{\|\nabla f\|}. \quad (5.30)$$

The definition of  $\gamma_t$  in equation (5.29) also simplifies as the  $\beta$  term in the denominator becomes negligible and the equation can be rewritten as equation

$$\gamma_t \cong \frac{\beta \Delta t}{\mu_t}. \quad (5.31)$$

It is possible from equation (5.31) to also assume that  $\gamma_t \cong 0$ .

Substituting equations (5.4), (5.30) and (5.31) into equation (5.27) directly yields equation

$$\hat{q}_t = \frac{\beta \Delta t}{\mu_t} \left( -\mu_t \frac{\nabla f}{\|\nabla f\|} \right) + (1 - 0) \left( \hat{q}_{(t-1)} + \dot{q}_{\omega,t} \Delta t \right). \quad (5.32)$$

Equation (5.32) can be simplified to

$$\hat{q}_t = \hat{q}_{(t-1)} + \dot{\hat{q}}_t \Delta t, \quad (5.33)$$

where  $\dot{\hat{q}}_t$  is the estimated rate of change of orientation defined by

$$\dot{\hat{q}}_t = \dot{q}_{\omega,t} - \beta \dot{q}_{\epsilon,t}, \quad (5.34)$$

and  $\dot{\hat{q}}_{\epsilon,t}$  is the direction of the error of  $\dot{\hat{q}}_t$  defined by

$$\dot{\hat{q}}_{\epsilon,t} = \frac{\nabla f}{\|\nabla f\|}. \quad (5.35)$$

It can be seen from equations (5.33) to (5.35) that the filter calculates the orientation  $\hat{q}$  by numerically integrating the estimated orientation rate  $\dot{\hat{q}}$ . The filter computes  $\dot{\hat{q}}_t$  as the rate of change of orientation measured by the gyroscopes,  $\dot{q}_{\omega,t}$ , with the magnitude of the error,  $\beta$ , removed in the direction of the estimated error,  $\dot{\hat{q}}_{\epsilon,t}$ , computed from the accelerometer and magnetometer measurements. Figure 5.1 shows a block diagram representation of the complete orientation filter implementation for an IMU.

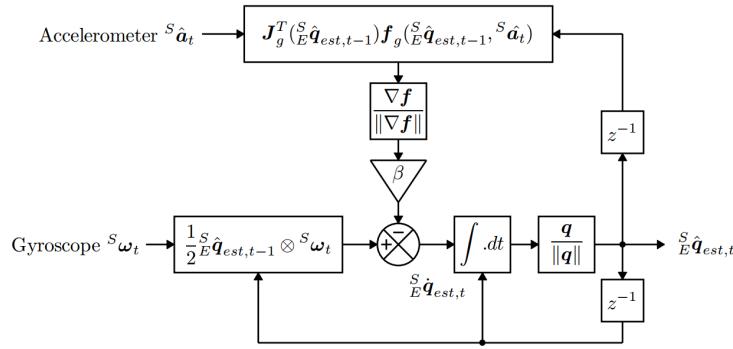


Figure 5.1: Madgwick Filter - IMU Implementation

### 5.1.3 Magnetic distortion compensation

Measurements of the Earth's magnetic field will be distorted by the presence of ferromagnetic elements in the vicinity of the magnetometer. Investigations into the effect of magnetic distortions on an orientation sensor's performance have shown that substantial error may be introduced by sources including electrical appliances, metal furniture and metal structures within a building construction. Sources of interference fixed in the sensor frame, termed soft-iron, cause errors in the measured direction of the earth's magnetic field. Declination errors, those in the horizontal plane relative to the Earth's surface, cannot be corrected without an additional reference of heading. Inclination errors, those in the vertical plane relative to the earth's surface, may be compensated for as the accelerometer provides an additional measurement of the sensor's attitude.

The measured direction of the Earth's magnetic field in the Earth frame at time  $t$ ,  $h_{E,t}$ , can be computed as the normalised magnetometer measurement,  $m_{B,t}$ , rotated by the estimated orientation of the sensor provided by the filter  $\hat{q}_{(t-1)}$ , as described by the equation

$$h_{E,t} = \begin{Bmatrix} 0 & h_x & h_y & h_z \end{Bmatrix}^T = \hat{q}_{(t-1)} \otimes m_{B,t} \otimes \hat{q}_{(t-1)}^*. \quad (5.36)$$

The effect of an erroneous inclination of the measured direction of the earth's magnetic field,  $b_{E,t}$ , is of the same inclination. This is achieved computing  $b_{E,t}$  as  $h_{E,t}$  normalised to have only components in the earth frame  $x$  and  $z$  axes; As described by equation

$$b_{E,t} = \left\{ \begin{array}{cccc} 0 & \sqrt{h_x^2 + h_y^2} & 0 & h_z \end{array} \right\}^T. \quad (5.37)$$

Compensating for magnetic distortions in this way ensures that magnetic disturbances are limited to affect only the estimated heading component of orientation. The approach also eliminates the need for the reference direction of the earth's magnetic field to be predefined.

#### 5.1.4 Gyroscope bias drift compensation

The gyroscope zero bias will drift over time, with temperature and with motion. Any practical implementation of an IMU or MARG sensor array must account for this. An advantage of Kalman-based approaches is that they are able to estimate the gyroscope bias as an additional state within the system model. However, that gyroscope bias drift may also be compensated for by simpler orientation filters through the integral feedback of the error in the rate of change of orientation.

The normalised direction of the estimated error in the rate of change of orientation,  $\dot{\hat{q}}_{\epsilon,t}$ , may be expressed as the angular error in each gyroscope axis using equation

$$\omega_{\epsilon,t} = 2\hat{q}_{(t-1)}^* \otimes \dot{\hat{q}}_{\epsilon,t}, \quad (5.38)$$

derived as the inverse to the relationship defined in equation (5.2). The gyroscope bias,  $\omega_b$ , is represented by the DC component of  $\omega_\epsilon$  and so it may be removed as the integral of  $\omega_\epsilon$  weighted by an appropriate gain,  $\zeta$ . This would yield the compensated gyroscope measurements  $\omega_c$ , as shown in equations

$$\omega_{b,t} = \zeta \sum_t \omega_{\epsilon,t} \Delta t, \quad (5.39)$$

$$\omega_{c,t} = \omega_t - \omega_{b,t}. \quad (5.40)$$

The first element of  $\omega_c$  is always assumed to be zero.

The compensated gyroscope measurements,  $\omega_{c,t}$ , may then be used in place of the gyroscope measurements,  $\omega$ , in equation (5.2). The magnitude of the angular error in each axis,  $\omega_\epsilon$ , is equal to a quaternion derivative of unit length. Therefore the integral gain  $\zeta$  directly defines the rate of convergence of the estimated gyroscope bias,  $\omega_b$ , expressed as the magnitude of a quaternion derivative. As this process requires to use the filter estimate of a complete orientation,  $\hat{q}_{(t-1)}$ , it is only applicable to a MARG

implementation of the filter. Figure 5.2 shows a block diagram representation of the complete filter implementation for a MARG sensor array, including the magnetic distortion (Group 1) and gyroscope bias drift compensation (Group 2).

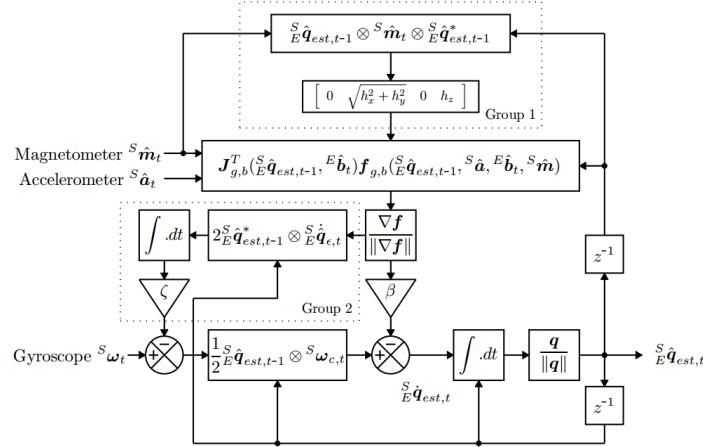


Figure 5.2: Madgwick Filter - MARG Implementation

### 5.1.5 Filter gains

The filter gain  $\beta$  represents all zero mean gyroscope measurement error, expressed as the magnitude of a quaternion derivative. The error sources include: sensor noise, signal aliasing, quantisation errors, calibration error, sensor miss-alignment, sensor axis non-orthogonality and frequency response characteristics. The filter gain  $\zeta$  represents the rate of convergence to remove gyroscope measurement errors which are not mean zero, also expressed as the magnitude of a quaternion derivative. These errors represent the gyroscope bias. It is convenient to define  $\beta$  and  $\zeta$  using the angular quantities  $\tilde{\omega}_\beta$  and  $\dot{\tilde{\omega}}_\zeta$  respectively, where  $\tilde{\omega}_\beta$  represents the estimated mean zero gyroscope measurement error of each axis and  $\dot{\tilde{\omega}}_\zeta$  represents the estimated rate of gyroscope bias drift in each axis. Using the relationship described in equation (5.2),  $\beta$  may be defined by equation

$$\beta = \left\| \frac{1}{2}q \otimes \left\{ \begin{array}{cccc} 0 & \tilde{\omega}_\beta & \tilde{\omega}_\beta & \tilde{\omega}_\beta \end{array} \right\}^T \right\| = \sqrt{\frac{3}{4}}\tilde{\omega}_\beta, \quad (5.41)$$

where  $q$  is any unit quaternion. Similarly,  $\zeta$  may be described by equation

$$\zeta = \sqrt{\frac{3}{4}} \dot{\tilde{\omega}}_{\zeta}. \quad (5.42)$$

### 5.1.6 Coefficients tuning

#### 5.1.6.1 Equipment

In order to find the best value of  $\beta$  that minimises the estimation error, the raw data from the IMU has been logged with a sampling rate of 100 Hz, processed with the Madgwick Filter on MATLAB® and then compared with the orientation given by an OptiTrack.

The Optitrack (Figure 5.3) is a system based on an array of Infra-Red (IR) sensitive cameras with incorporate IR flood lights. The cameras are fixed at calibrated positions and orientations so that the measurement subject is within the field of view of multiple cameras. The measurements of the orientation given by the OptiTrack (at 100 Hz) is used as reference measure in order to tune the Madgwick Filter.

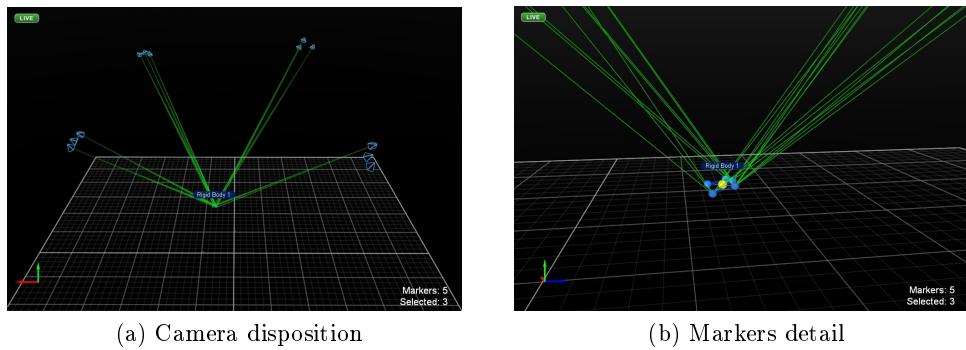


Figure 5.3: OptiTrack

To make the IMU trackable by the IR cameras, a marker set has to be placed over the electronics (Figure 5.4).

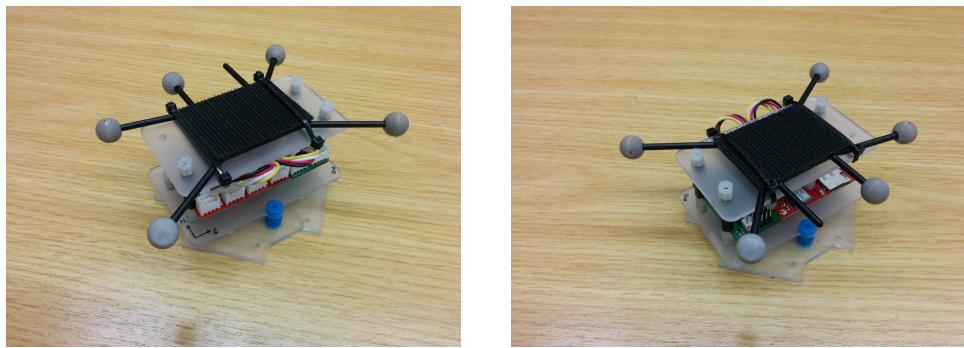


Figure 5.4: Marker set

#### 5.1.6.2 Experimental procedure

The optical measurement data and raw orientation sensor data (obtained by a sequence of rotations performed by hand) were logged simultaneously. The raw

orientation sensor data were then processed on MATLAB® throught both IMU and MARG implementations of the filter using an increasing value of  $\beta$ . For each value of  $\beta$ , the estimation results were compared with the optical measurement and then the Root Mean Squared (RMS) error was calculated. The optimal  $\beta$  has been found as the one which gives the smallest RMS.

In Figure 5.5 two typical results are shown: on the left-hand side the curve obtained with the IMU algorithm and on the right-hand side the curve obtained with the MARG algorithm. One can see that, because of the contribution of both accelerometer and magnetometer, the MARG version needs a smaller value of  $\beta$  to obtain a better results. However, MARG version of the Madgwick filter makes the attitude estimation very sensitive to the magnetic field. One has to take it into account when designing a multicopter because the ESCs can cause several disturbances.

#### 5.1.6.3 Tuning results

In order to better visualise the results of the attitude estimation, quaternions were converted into Euler angles.

Figures 5.6 show typical results of the experiments for the IMU implementation (with optimal  $\beta$  found as described before) compared with the optical measurements. For both figures, on the right-side there are the Euler Angles calculated from the quaternion given by the filter and the OptiTrack and on the left-side there are the errors.

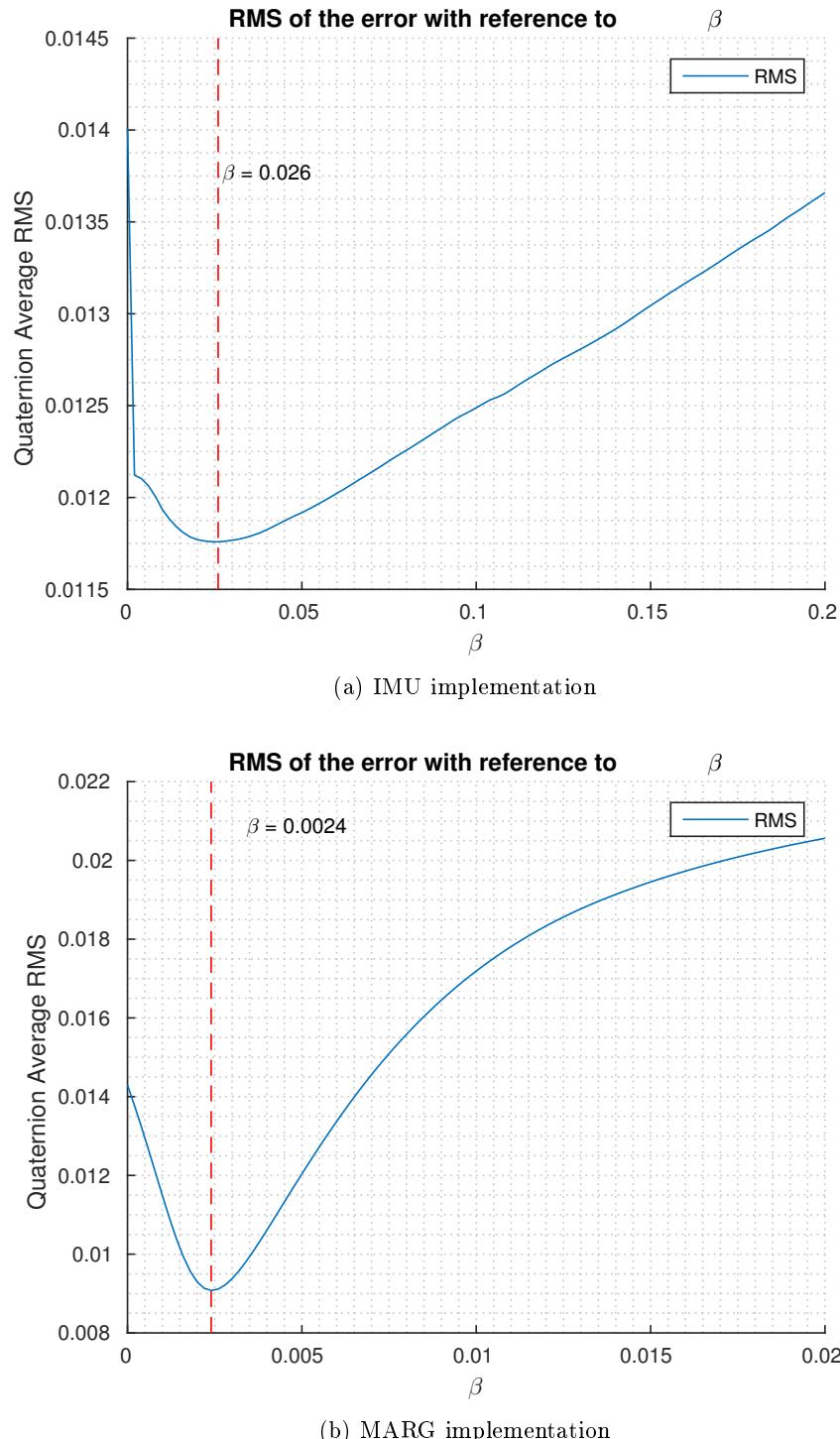
It is also possible to see that in the IMU implementation there is not correction by the magnetometer on the yaw angle,  $\Psi$ , with a resulting drift of the estimated angle.

Figures 5.7, instead, show typical results of the experiments for the MARG implementation (with optimal  $\beta$  found as described before) compared with the optical measurements.

One can see that even though the drift over time is suppressed, the resulting estimated angle is more noisy and the average error is larger.

## 5.2 Conclusions

The application and tuning of the Madgwick Filter in order to estimate the attitude of the quadrotor has been discussed. In particular it is possible to conclude that MARG algorithm, although it guarantees lower drift over time, it is more subjected to disturbance from exogenous magnetic fields.

Figure 5.5: RMS of the error with reference to  $\beta$

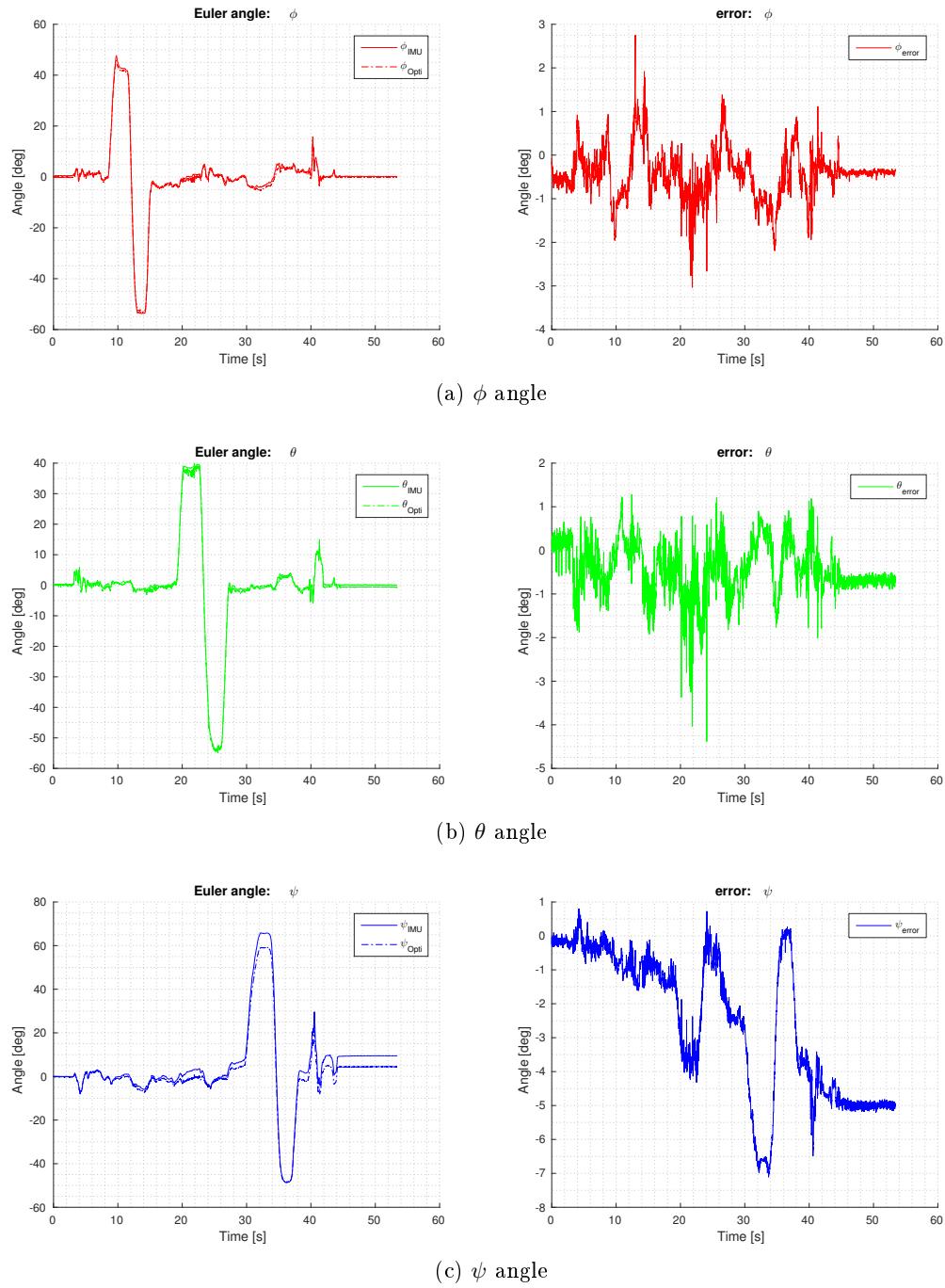


Figure 5.6: IMU algorithm

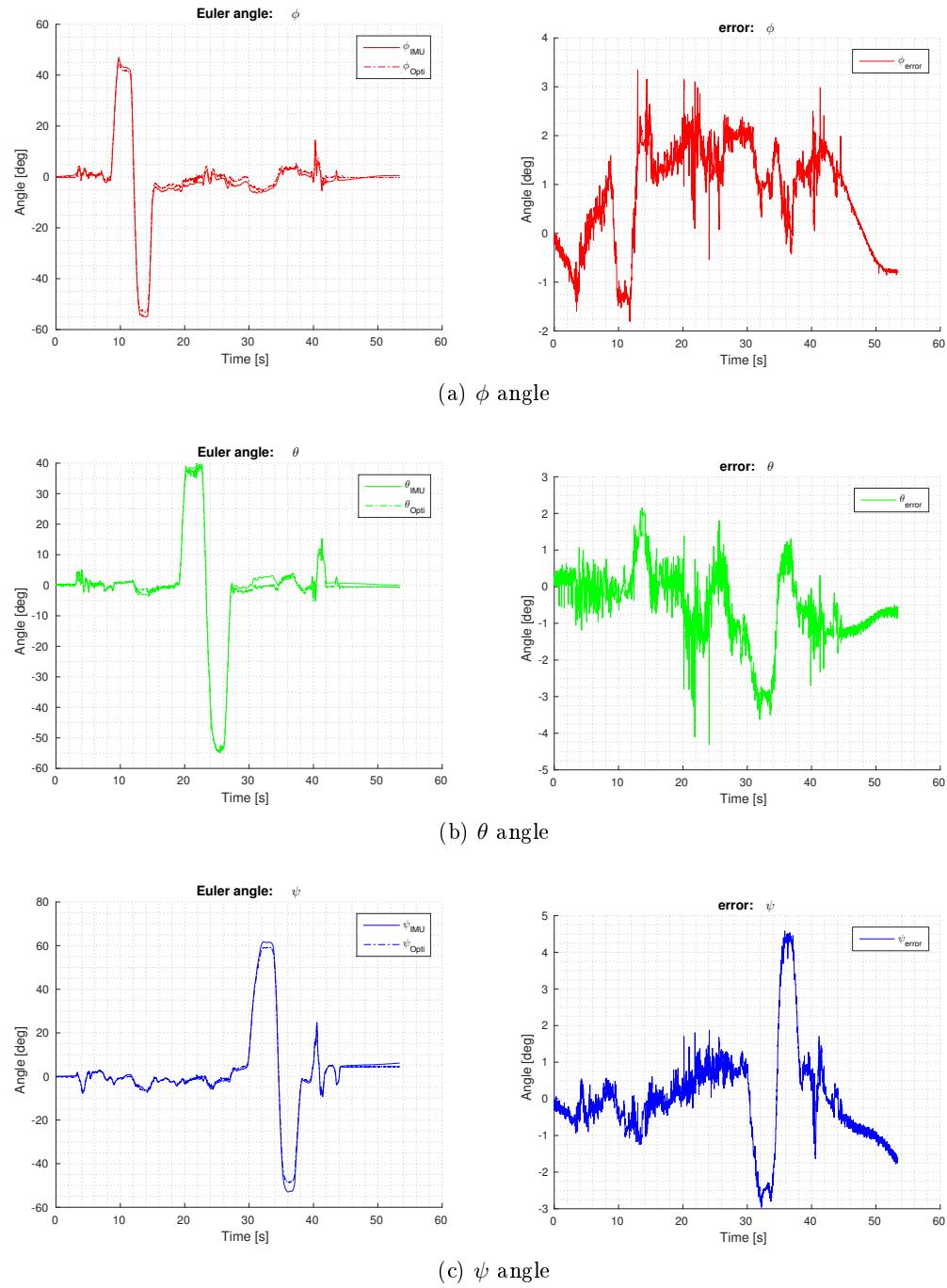


Figure 5.7: MARG algorithm



# Chapter 6

## Identification of roll and pitch dynamics

The aim of this chapter is to show an approach to identify the attitude dynamics of a quadrotor helicopter from data obtained in an experimental identification campaign. More precisely, thanks to the grey-box model identification, it is possible to identify all the unknown parameters that describe the first-principle dynamics model.

### 6.1 Pitch dynamical model

As discussed in the Chapter 4, the mathematical model that describes the attitude of a quadrotor helicopter is the following

$$I_n \dot{\omega}_b + \omega_b \times (I_n \omega_b) = M_{damp} + M_{props}, \quad (6.1)$$

where

$$M_{damp} = \begin{bmatrix} \frac{\partial L}{\partial p} & 0 & 0 \\ 0 & \frac{\partial M}{\partial q} & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix}, \quad M_{props} = \begin{bmatrix} K_T \frac{b}{\sqrt{2}} (\Omega_1^2 - \Omega_2^2 - \Omega_3^2 + \Omega_4^2) \\ K_T \frac{b}{\sqrt{2}} (\Omega_1^2 + \Omega_2^2 - \Omega_3^2 - \Omega_4^2) \\ K_Q (-\Omega_1^2 + \Omega_2^2 - \Omega_3^2 + \Omega_4^2) \end{bmatrix}.$$

Because of the geometric symmetry of the quadrotor body around the  $X$  and the  $Y$  axes, in order to find the unknown parameters, it has been decided to study only the pitch attitude, which unknown parameters are the following

$$\Theta = \left[ I_{yy}, \frac{\partial M}{\partial q} \right], \quad (6.2)$$

and then assume

$$I_{xx} = I_{yy} \quad (6.3)$$

$$\frac{\partial L}{\partial p} = \frac{\partial M}{\partial q}. \quad (6.4)$$

The model under evaluation now becomes

$$I_{yy} \dot{q} + (I_{xx} - I_{zz})pr = \frac{\partial M}{\partial q} q + K_T \frac{b}{\sqrt{2}} (\Omega_1^2 + \Omega_2^2 - \Omega_3^2 - \Omega_4^2). \quad (6.5)$$

One can now rewrite the equation assuming the same rotational speed of the propellers between the two front rotors and the two rear rotors

$$\Omega_1 = \Omega_2 = \Omega_F \quad (6.6)$$

$$\Omega_3 = \Omega_4 = \Omega_R \quad (6.7)$$

$$M = K_T \sqrt{2} b (\Omega_F^2 - \Omega_R^2). \quad (6.8)$$

It is also possible to linearize the dynamical model in near hovering condition assuming a small deviation of the propeller's rotational speed from the hovering value ( $\Omega_H$ )

$$\delta\Omega_F = \Omega_F - \Omega_H, \quad (6.9)$$

$$\delta\Omega_R = \Omega_R - \Omega_H, \quad (6.10)$$

$$\delta M = M \quad , \quad (6.11)$$

and, in order to produce a moment around the  $Y$  axis, a variation of the rotational speed,  $\delta\Omega$ , is introduced as control variable,  $u$

$$u = \delta\Omega = \delta\Omega_F = -\delta\Omega_R. \quad (6.12)$$

The model of the pitch axis then becomes

$$I_{yy} \dot{q} + (I_{xx} - I_{zz})pr = \frac{\partial M}{\partial q} q + \frac{\partial M}{\partial u} \delta\Omega, \quad (6.13)$$

where  $\frac{\partial M}{\partial u}$  is the control derivative of the pitch moment

$$\frac{\partial M}{\partial u} = K_T 4 \sqrt{2} b \Omega_H. \quad (6.14)$$

## 6.2 Test-bed

The pitch attitude identification experiments have been carried out with the quadrotor placed on a test-bed that constraints all translational and rotational DoFs except for pitch rotation, as shown in Figure 6.1. As the test-bed keeps the quadrotor distant enough from the ground (at least two times the radius of the rotors [8]), the ground effect disturbance is avoided.

The excitation sequences for the attitude dynamics of the rotors have been applied in open-loop conditions: the attitude controller was disabled.

## 6.3 Grey-box estimation

Since the pitch attitude identification experiments have been carried out, with the quadrotor placed on a test-bed that allows only the pitch rotation, it follows that

$$\begin{aligned} I_{yy} \dot{q} &= \frac{\partial M}{\partial q} q + \frac{\partial M}{\partial u} \delta \Omega, \\ \dot{\Theta} &= q. \end{aligned} \quad (6.15)$$

The model can be written in state space form as

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du \end{aligned} \quad (6.16)$$

where the state vector it is defined as

$$x = \begin{bmatrix} q \\ \Theta \end{bmatrix}, \quad (6.17)$$

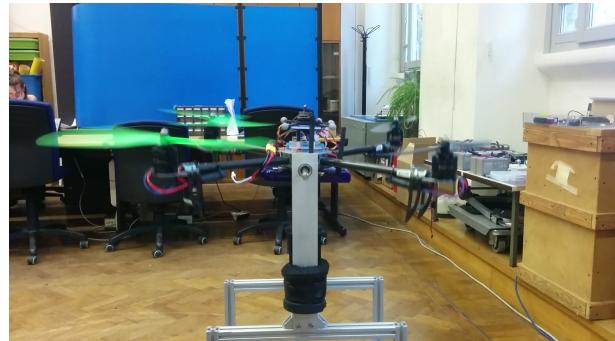
and

$$\begin{aligned} A &= \begin{bmatrix} \frac{1}{I_{yy}} \frac{\partial M}{\partial q} & 0 \\ 1 & 0 \end{bmatrix}, & B &= \begin{bmatrix} \frac{1}{I_{yy}} \frac{\partial M}{\partial u} \\ 0 \end{bmatrix}, \\ C &= \begin{bmatrix} 1 & 0 \end{bmatrix}, & D &= [0]. \end{aligned} \quad (6.18)$$

Given the unknown parameters, system model, input and the related output, using a grey-box model estimation method, an estimated model which best fits the provided input-output behavior can be found. In detail, the MATLAB function `<greyest>` has been used for this purpose.

In the following the settings used for model identification are reported:

- Handling of initial states: the initial state is set to zero.
- Handling of disturbance components: none.
- Estimation focus: estimates the model using the frequency weighting of the transfer function that is given by the input spectrum. Typically, this method



(a) Side view



(b) Front view

Figure 6.1: Test-bed

favors the frequency range where the input spectrum has the most power. This method provides a stable model.

- Search method used for iterative parameter estimation: the descent direction is calculated using the Gauss-Newton search method, an adaptive version of Gauss-Newton, Levenberg-Marquardt and the descent gradient successively at each iteration. The iterations continue until a sufficient reduction in error is achieved.

From each data set it is possible to obtain the vector of the estimated parameters and the covariance matrix.

## 6.4 Identification signal input - RBS

To identify the attitude dynamics, it has been decided to use as a signal input a Random Binary Sequence (RBS) (see [7]), which is a signal with properties similar to white noise.

First of all, it is necessary to know to first approximation what is the bandwidth to be excited in order to find the actual model of the system. To do so, it is necessary to calculate a first guess of the unknown parameters.

To find a first guess of the inertia tensor of the quadrotor, a CAD model of the quadrotor has been realized with correct mass properties.

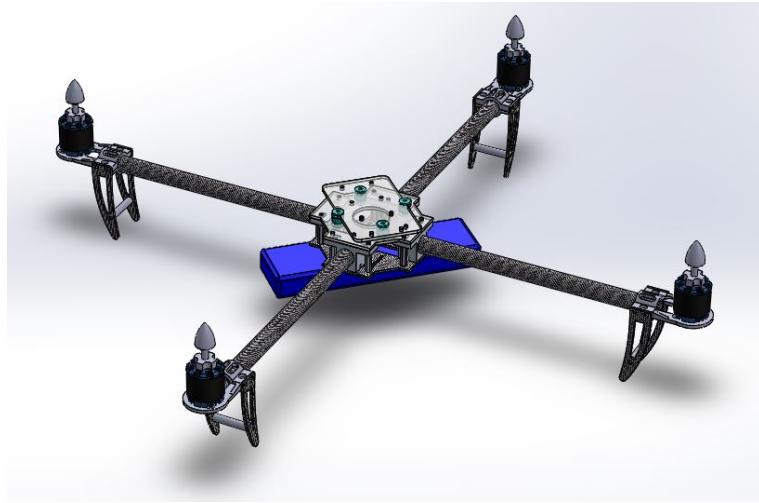


Figure 6.2: Quadrotor CAD on SolidWorks

As introduced before in Chapter 4 ,the stability derivative  $\frac{\partial M}{\partial q}$  can be written as following (see [13–15])

$$\frac{\partial M}{\partial q} = -4\rho AR^2\Omega^2 \frac{\partial C_T}{\partial q} \frac{b}{\sqrt{2}}, \quad (6.19)$$

$$\frac{\partial C_T}{\partial q} = \frac{C_{l\alpha}}{8} \frac{\sigma}{R\Omega} \frac{b}{\sqrt{2}}, \quad (6.20)$$

$$\sigma = \frac{A_b}{A}. \quad (6.21)$$

And, the control derivative of the pitch moment can be found as

$$\frac{\partial M}{\partial u} = K_T 4\sqrt{2}b\Omega_H. \quad (6.22)$$

The guess values for the parameters that have been found are

$$\begin{aligned} I_{yy} &= I_{xx} = 20 \times 10^{-3} \quad [kgm^2] \\ \frac{\partial M}{\partial q} &= \frac{\partial L}{\partial p} = -26 \times 10^{-3} \quad [Nms] \\ \frac{\partial M}{\partial u} &= \frac{\partial L}{\partial u} = 15 \times 10^{-3} \quad [Nms]. \end{aligned} \quad (6.23)$$

Due to an identification campaign, one cannot estimate the three parameters mentioned before, as in the state space model of Equation (6.18), the two derivatives ( $\frac{\partial M}{\partial q}$  and  $\frac{\partial M}{\partial u}$ ) are both divided by the value of  $I_{yy}$ . The grey-box estimation procedure can only estimate the value of  $\frac{1}{I_{yy}} \frac{\partial M}{\partial q}$  and  $\frac{1}{I_{yy}} \frac{\partial M}{\partial u}$ , as with two equations one cannot find the value of three uncertainties. It has been decided to fix the control derivative  $\frac{\partial M}{\partial u}$  at the guess value, or in other hand use the value of  $K_T$  found in the Chapter 3.

It is then possible to calculate the bandwidth of the attitude dynamics writing the model described before with the first guess of the parameters to be estimated. One can see the guessed system has a bandwidth of around 2.6 rad/s.

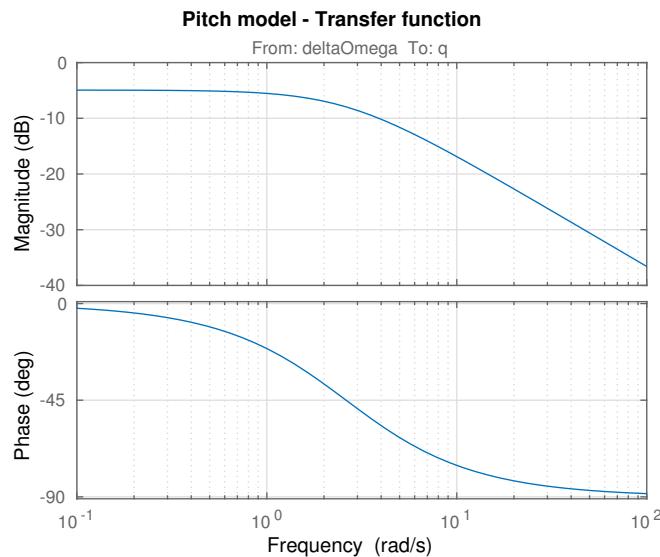


Figure 6.3: Pitch rate Bode diagram - Guess

	$\delta\Omega$ [rad/s]	$\omega_{min}$ [rad/s]	$\omega_{max}$ [rad/s]
Test A	1.9	0.6	6.0
Test B	1.7	0.6	5.4
Test C	1.3	0.6	4.2

Table 6.1: RBS parameters

The parameters of the RBS sequence (signal amplitude and min/max switching interval) are tuned in order to obtain an excitation spectrum consistent with the expected bandwidth, between 0.6 rad/s and 6 rad/s. More precisely, three different identification campaigns were conducted (the campaigns were composed by thirty tests each) where the switching interval and the amplitude were varied. The parameters of the sequence are reported in Table 6.1.

## 6.5 Experiments and data analysis

From each of the three campaigns, the obtained data sets have been elaborated in order to find an unique vector of estimated parameters with related standard deviation. The subscript  $j$  indicates the typology of the test (Test A, ...) while the subscript  $i$  indicates the number of the  $i - th$  out of 30 data set.

$$\hat{\Theta}_j = \frac{1}{30} \sum_{i=1}^{30} \hat{\Theta}_{ji} \quad (6.24)$$

$$\sigma_j = \sqrt{\frac{1}{30} \sum_{i=1}^{30} \sigma_{ji}^2} \quad (6.25)$$

Given the estimated parameters and their standard deviation from each test, and thanks to the Central Limit Theorem, a Normal distribution of the estimated parameters is assumed. In Figure 6.4 these distributions are reported.

Before the estimation of an unique value of the parameters, one has to check the compatibility of the obtained data. A compatibility test is then conducted. The compatibility test check if the distance between two results is less or equal to a composition of the related uncertainties scaled by a  $k$  factor. More precisely, if this  $k$  factor is greater than 1 the two results under exam are then considered not compatible. Otherwise, the two results are considered compatible and then their value will be used for the final estimation.

$$|\hat{\Theta}_j - \hat{\Theta}_l| \leq k \sqrt{\sigma_j^2 + \sigma_l^2} \quad j \neq l \quad (6.26)$$

Analyzing the obtained results one can conclude that, to estimate the stability

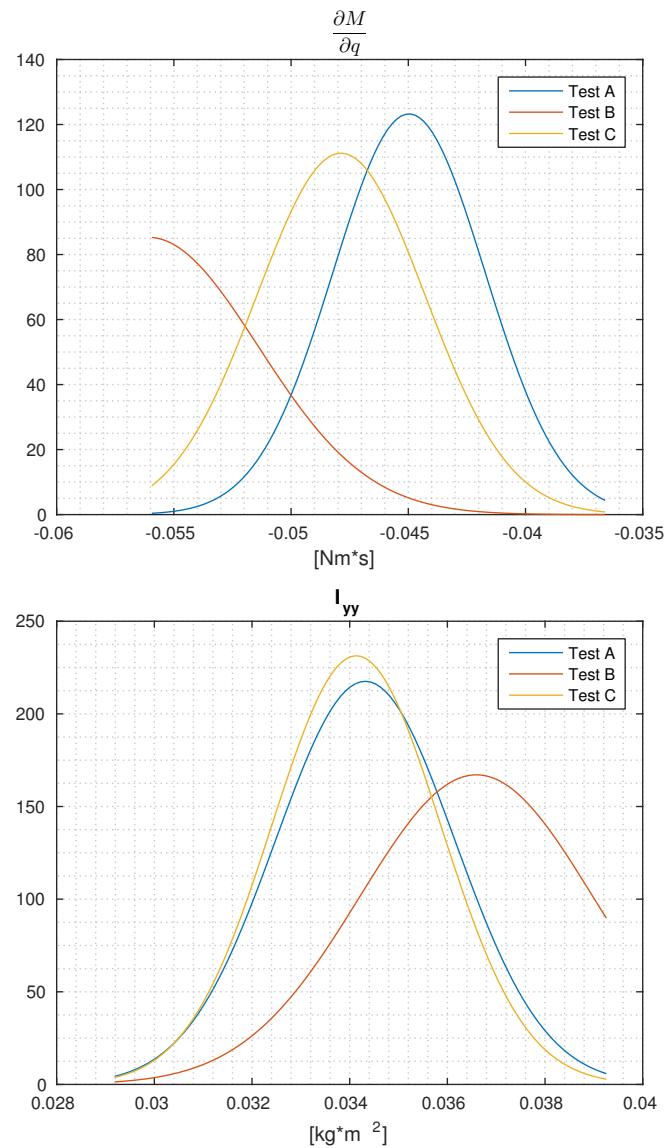


Figure 6.4: Identification results

		Guess	Estimated	$\sigma$
$I_{yy} = I_{xx}$	[kgm <sup>2</sup> ]	$20 \times 10^{-3}$	$34.7 \times 10^{-3}$	$1.16 \times 10^{-3}$
$\frac{\partial M}{\partial q} = \frac{\partial L}{\partial p}$	[Nm s]	$-26 \times 10^{-3}$	$-46.3 \times 10^{-3}$	$2.42 \times 10^{-3}$

Table 6.2: Values of the identified parameters

derivative  $\frac{\partial M}{\partial q}$ , the estimated parameter from the Test B has to be considered incompatible to the ones obtained from Test A and Test C; while the three estimated values of the parameter  $I_{yy}$  can be considered compatible.

The best estimation then achievable is the weighted average of the compatible results, the weight are the reciprocal of the estimated variances.

$$I_{yy} = \frac{\frac{I_{yy}|_A}{\sigma_{I_{yy}}^2|_A} + \frac{I_{yy}|_B}{\sigma_{I_{yy}}^2|_B} + \frac{I_{yy}|_C}{\sigma_{I_{yy}}^2|_C}}{\frac{1}{\sigma_{I_{yy}}^2|_A} + \frac{1}{\sigma_{I_{yy}}^2|_B} + \frac{1}{\sigma_{I_{yy}}^2|_C}} \quad (6.27)$$

$$\frac{\partial M}{\partial q} = \frac{\frac{\frac{\partial M}{\partial q}|_A}{\sigma_{dM}^2|_A} + \frac{\frac{\partial M}{\partial q}|_C}{\sigma_{dM}^2|_C}}{\frac{1}{\sigma_{dM}^2|_A} + \frac{1}{\sigma_{dM}^2|_C}} \quad (6.28)$$

Finally, to obtain the values of standard deviation for each parameter, the average of the variance has been calculated

$$\sigma_{I_{yy}} = \sqrt{\frac{\sigma_{I_{yy}}^2|_A + \sigma_{I_{yy}}^2|_B + \sigma_{I_{yy}}^2|_C}{3}} \quad (6.29)$$

$$\sigma_{dM} = \sqrt{\frac{\sigma_{dM}^2|_A + \sigma_{dM}^2|_C}{2}} \quad (6.30)$$

The problem of characterizing the attitude dynamics of a quadrotor helicopter has been considered and an approach to its identification has been applied to data collected on the real quadrotor, in laboratory conditions.

In the Table 6.2 are reported the estimated data from the identification campaigns  
The nominal transfer function is the following

$$G_n(s) = \frac{0.423}{s + 1.33}. \quad (6.31)$$

It is now possible to plot the estimated Bode diagram of the system considering also the uncertainties obtained on the parameters and evaluate the bandwidth. One can see that the estimated bandwidth is about 1.33 rad/s, slower than the guessed one.

## 6. Identification of roll and pitch dynamics

---

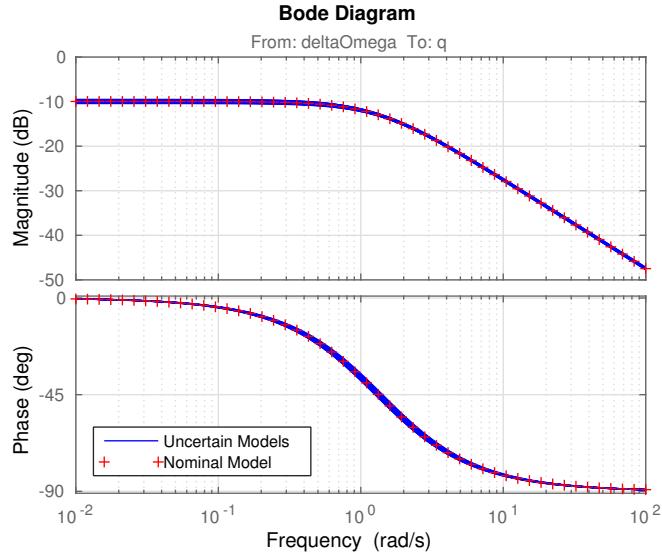


Figure 6.5: Pitch rate Bode diagram - Results

Following, in Figure 6.6, it is shown an example of the RBS signal applied as input and two different outputs: one is the acquired angular rate by the IMU's gyroscope and one is a simulation of the identified system with the same excitation input signal.

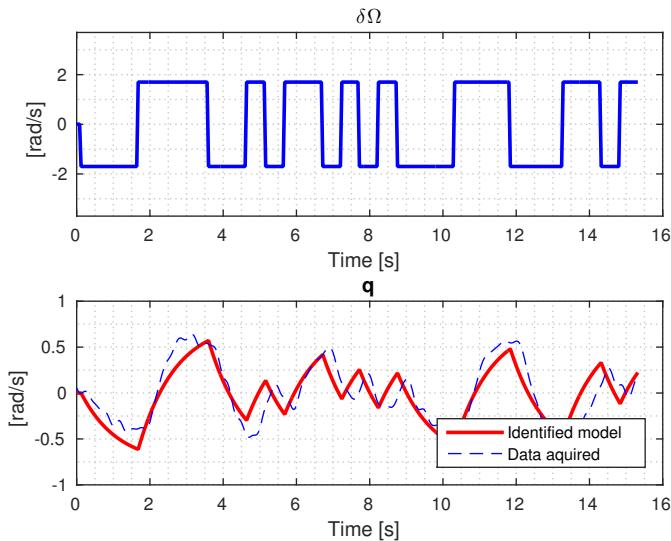


Figure 6.6: Input and Output from a test

To quantify the results of the estimation, with the results obtained, the Variance Accounted For [%] (VAF, Equation (6.32)) has been calculated. In the result shown in Figure 6.6, the obtained VAF is equal to 70.37%.

$$VAF = \max \left( 1 - \frac{\text{variance}(y - y_{est})}{\text{variance}(y)}, 0 \right) * 100 \quad (6.32)$$



# Chapter 7

## Attitude control law design

In this chapter, the scheme adopted to control the attitude of the quadrotor is presented in detail. More precisely, the variables of the attitude that have to be controlled are the roll DoF ( $\Phi$ ), pitch DoF ( $\Theta$ ) and the yaw-rate ( $\dot{\Psi}$ ).

Moreover the formulation of the problem for the Fixed-Structure  $H_\infty$  Synthesis is described and the performance is evaluated in an experimental campaign under laboratory conditions.

### 7.1 Control architecture

Each regulator is designed and tuned assuming decoupled dynamics between the DoFs. This assumption is valid only if the quadrotor is considered in near-hovering conditions, and falls progressively as the forward velocity increases.

The adopted scheme used to control the roll and pitch angles is the same, and it is based on cascaded PID loops. In Figure 7.1 the block diagram relative to the pitch angle is shown.

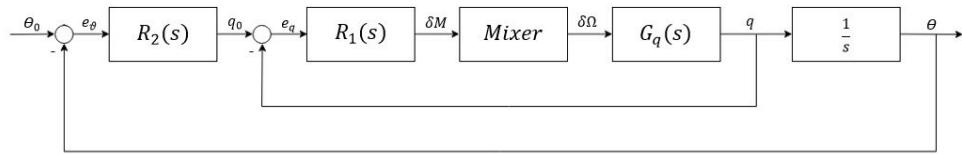


Figure 7.1: Pitch Control Loops

In detail, the outer regulator  $R_2(s)$ , is a PD controller based on attitude feedback (estimated angle  $\Theta$ , set-point  $\Theta_0$ ) and the inner regulator  $R_1(s)$ , is a PID controller based on angular rate feedback (measured angular velocity  $q$ , set-point  $q_0$ ).

$$R_1(s) = K_{p1} + K_{i1} \frac{1}{s} + K_{d1} \frac{s}{1 + sT_f}, \quad (7.1)$$

$$R_2(s) = K_{p2} + K_{d2} \frac{s}{1 + sT_f}. \quad (7.2)$$

These two regulators will be the object of the fixed-structure  $H_\infty$  synthesis described in the next section and all the regulators have been tuned assuming the nominal value of the identified system, equation (7.3).

$$G_n(s) = \frac{0.423}{s + 1.33}. \quad (7.3)$$

The control variable considered as output of the inner regulator is the moment about the considered DoF,  $\delta L$  for the  $\Phi$  DoF and  $\delta M$  for the  $\Theta$  DoF. The real input of the system is the difference between the rotational speed of the opposite rotors,  $\delta\Omega$ , as introduced in Chapter 4, and it is obtained multiplying the moment with the "Mixer Matrix",  $\chi$ .

The controllers have been tuned as a continuous time blocks because the control loops inside the FCU run at 100 Hz, that is two orders of magnitude faster than the dynamics of the quadrotor.

Instead, the formulation of the yaw-rate control loop has been drastically simplified because the yaw dynamics is stable (as described in Chapter 5, the stability derivative  $\partial N/\partial r$  has been neglected). The Equation (7.4) described in Chapter 4, with the estimated propeller parameters described in Chapter 3 is the analytical model that describe the behavior of the yaw DoF.

$$I_{zz} \dot{r} = K_Q (-\Omega_1^2 + \Omega_2^2 - \Omega_3^2 + \Omega_4^2). \quad (7.4)$$

One can now rewrite the equation assuming the same rotational speed of the propellers between the two clockwise rotors and the two counter clockwise rotors

$$\Omega_1 = \Omega_3 = \Omega_{CW} \quad (7.5)$$

$$\Omega_2 = \Omega_4 = \Omega_{CCW} \quad (7.6)$$

$$N = 2K_Q (\Omega_{CCW}^2 - \Omega_{CW}^2). \quad (7.7)$$

It is also possible to linearize the dynamical model in near hovering condition assuming a small deviation of the propeller's rotational speed from the hovering value ( $\Omega_H$ )

$$\delta\Omega_{CW} = \Omega_{CW} - \Omega_H, \quad (7.8)$$

$$\delta\Omega_{CCW} = \Omega_{CCW} - \Omega_H, \quad (7.9)$$

$$\delta N = N \quad , \quad (7.10)$$

and, in order to produce a moment around the  $Z$  axis a variation of the rotational speed,  $\delta\Omega$ , is introduced as control variable,  $u$

$$u = \delta\Omega = \delta\Omega_{CCW} = -\delta\Omega_{CW}. \quad (7.11)$$

The model now becomes

$$I_{zz} \dot{r} = \frac{\partial N}{\partial u} \delta\Omega, \quad (7.12)$$

where  $\frac{\partial N}{\partial u}$  is the control derivative of the yaw moment, given by

$$\frac{\partial N}{\partial u} = 8K_Q\Omega_H. \quad (7.13)$$

The transfer function of the yaw-rate then becomes

$$G_r(s) = \frac{\partial N}{\partial u} \frac{1}{I_{zz}s}. \quad (7.14)$$

The implemented yaw-rate regulator, Equation (7.15), is a simple PI controller. The obtained scheme is shown in Figure 7.2.

$$R(s) = K_p + K_i \frac{1}{s}. \quad (7.15)$$

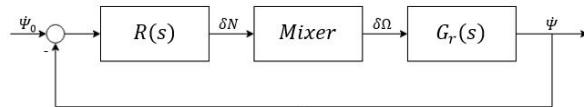


Figure 7.2: Yaw Control Loop

## 7.2 Fixed-structure $H_\infty$ synthesis

$H_\infty$  theory provides a powerful framework for capturing standard control design requirements such as the speed of response, bandwidth, disturbance rejection, and robust stability [4]. This framework extends the classical control techniques such as Bode loop shaping or gain/phase margin analysis to multi-loop and MIMO control architectures. Design requirements are expressed in terms of  $H_\infty$  norm constraints.

Yet traditional,  $H_\infty$  synthesis has practical limitations that have slowed its adoption in industry. In particular,  $H_\infty$  controllers are monolithic whereas most embed-

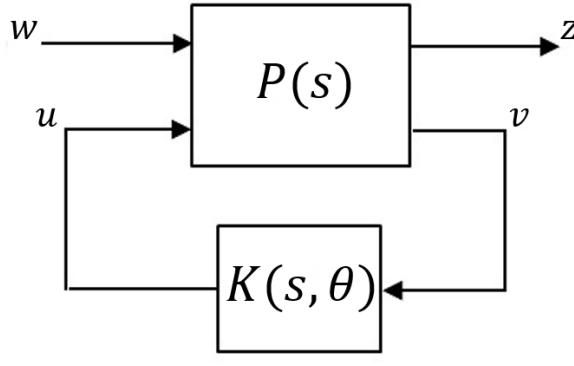


Figure 7.3: Standard form for structured  $H_\infty$  synthesis

ded control architectures are decentralized collections of simple control elements such as gains and PID controllers. In addition, the order (complexity) of  $H_\infty$  controllers tends to be high whereas embedded controllers tend to have low complexity.

The statement of the controller tuning problem, formulated in the framework of structured  $H_\infty$  synthesis, can therefore be given by a LTI model  $P(s)$  which combines all fixed (non tunable) blocks in the control system, and a structured controller  $K(s, \theta)$  combines all tunable control elements in a vector  $\theta$  varying in space  $\mathbb{R}^n$  (Figure 7.3).

External inputs to the system, such as reference signals and disturbances, are gathered in  $w$  and performance-related outputs, such as error signals, are gathered in  $z$ .

At the heart of  $H_\infty$  synthesis there is the  $H_\infty$  norm which measures the peak input/output gain of a given stable transfer function

$$\|H(s)\|_\infty := \max_{\omega} \bar{\sigma}(H(j\omega)). \quad (7.16)$$

In the SISO case, this norm is just the peak gain over frequency. In the MIMO case, it measures the peak 2-norm of the frequency response  $H(j\omega)$  over frequency. Solving the  $H_\infty$  synthesis problem consists in searching the parameter vector  $\theta$  that minimizes

$$\|T_{w \rightarrow z}(P(s), K(s, \theta))\|_\infty \quad (7.17)$$

subject to the constraint that  $K(s, \theta)$  stabilizes  $P(s)$ , where  $T_{w \rightarrow z}(P(s), K(s, \theta))$  is the closed-loop transfer function on the considered input-output channel  $w \rightarrow z$  on which the requirements are defined.

To solve this optimization problem, the Robust Control Toolbox of MATLAB & Simulink has been used. In particular the `<systune>` function and `<TuningGoal>` class.

Controller Parameter	Standard Tuning	Optimal Tuning
$K_{p1}$	0.3	0.3
$K_{d1}$	0.3	0.3
$K_{p2}$	0.05	0.05
$K_{i2}$	1.2	1.61
$K_{d1}$	0.005	0.00512

Table 7.1: Tuning parameters

For the assigned controller structure, the procedure finds the (locally) optimal parameters for the two regulators so as to satisfy the following requirements:

- Tracking requirements: the angle  $\Theta$  should track the reference  $\Theta_0$  with a 1 second response time ( $T_s$ ,  $\omega_c = \frac{2}{T_s}$ ), a maximum steady-state error equal to 0.0001% and a maximum relative peak error of 30%. More precisely, the software has to adjust the parameter values in order to minimize  $J(\theta)$ , where  $F(s, \theta)$  is the closed-loop transfer function from  $\Theta_0$  to  $\Theta$  and  $\theta$  is the vector of the free (tunable) parameters in the control system.

$$J(\theta) = \left\| \frac{1}{MaxError} (F(s, \theta) - 1) \right\|_\infty \quad (7.18)$$

$$MaxError = \frac{(PeakError)s + \omega_c(DCError)}{s + \omega_c} \quad (7.19)$$

- Maximum loop gain and roll-off requirements: the upper bound of the loop gain function  $L(s)$  is the function  $\frac{\omega_c}{s}$ . More precisely, the objective function is the Equation (7.20), where the  $W_T$  is the reciprocal of the maximum loop gain profile.

$$J(\theta) = \|W_T F(s, \theta)\|_\infty \quad (7.20)$$

- Disturbance rejection requirements: as a sensitivity function was adopted a second order high pass filter. The objective function is the Equation (7.21), where  $W(j\omega)$  is the desired shaping function that approximates the minimum disturbance attenuation specified and  $S(j\omega, \theta)$  is the closed-loop sensitivity function.

$$J(\theta) = \max_{\omega \in \Omega} \|W(j\omega)S(j\omega, \theta)\| \quad (7.21)$$

Figures 7.4-7.6 report the requirement bounds and in Table 7.1 the resulting optimal parameters for both the outer loop PD and the inner loop PID controllers are listed, as obtained applying the structured  $H_\infty$  synthesis with the above requirements. The only parameter's value has been fixed from the beginning is  $T_f = 0.01$  s.

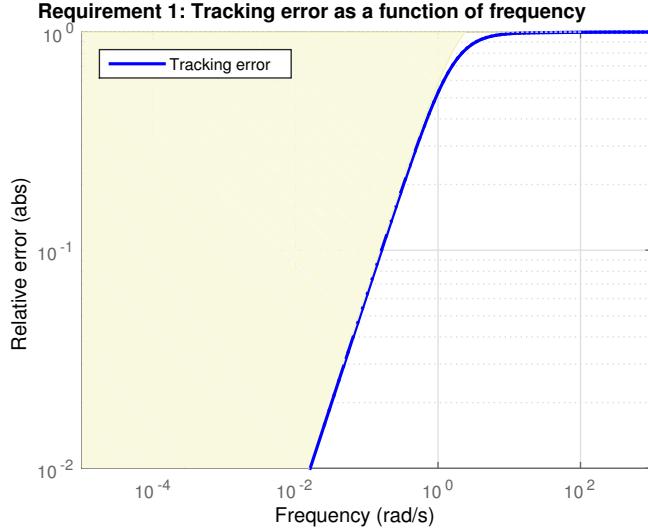


Figure 7.4: Tracking requirement

As reference, the standard tuning already used on the quadrotor, obtained through trial and error empirical procedure done manually, is also reported. The standard tuning was also used as starting guess for the optimization procedure.

The resulting sensitivity functions are here reported (Figure 7.7). In detail, functions (7.22)-(7.23) are the inner loop transfer function and the open-loop transfer function.

$$F_{in}(s) = \frac{G_q(s)\chi R_1(s)}{1 + G_q(s)\chi R_1(s)}, \quad (7.22)$$

$$L(s) = \frac{1}{s} F_{in} R_2(s). \quad (7.23)$$

Then the complementary sensitivity function is defined as

$$F(s) = \frac{L(s)}{1 + L(s)}, \quad (7.24)$$

the sensitivity function as

$$S(s) = \frac{1}{1 + L(s)}, \quad (7.25)$$

and the control sensitivity

$$Q(s) = \frac{R_2(s)}{1 + L(s)}. \quad (7.26)$$

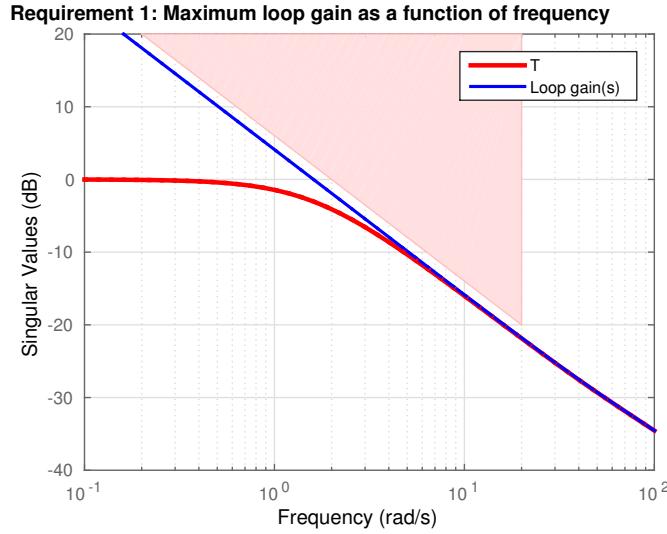


Figure 7.5: Max gain and roll-off requirements

### 7.3 Robustness analysis

The generic problem of synthesis consists in the determination of a controller, able to ensure the achievement of certain specified targets. As a rule, these targets include, primarily, the stability of the control system and benefits obtainment which results have to be satisfactory despite the presence of disturbances and non-accurate knowledge of the process' dynamics, or, in other words the transfer function  $G(s)$ . If one has to deal with the analysis problem of a specified control system and the synthesis of the robust regulator, it is mandatory to take into account the uncertain knowledge of the process, it is then essential to characterize that uncertainty. The additive and multiplicative representation of the uncertainty are here proposed (see [1]).

Additive,  $G_a(s)$ , and multiplicative,  $G_m(s)$ , uncertain models are two types of non structured uncertainties, described as following

$$G_a(s) = G_n(s) + W(s)\Delta(s), \|\Delta\|_\infty < 1. \quad (7.27)$$

$$G_m(s) = G_n(s)(1 + W(s)\Delta(s)), \|\Delta\|_\infty < 1. \quad (7.28)$$

In equations (7.27) and (7.28),  $G_n(s)$  is the nominal model of the system,  $W(s)$  is called “shaping function” and  $\Delta(s)$  represent the actual normalized uncertainty.

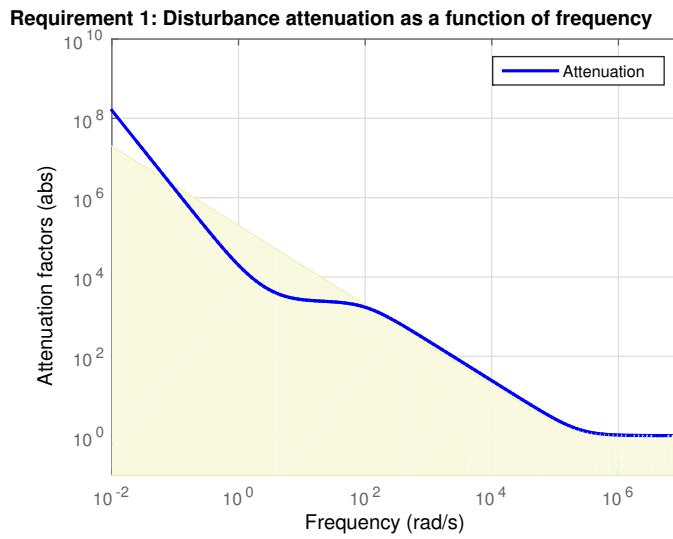


Figure 7.6: Disturbance rejection requirement

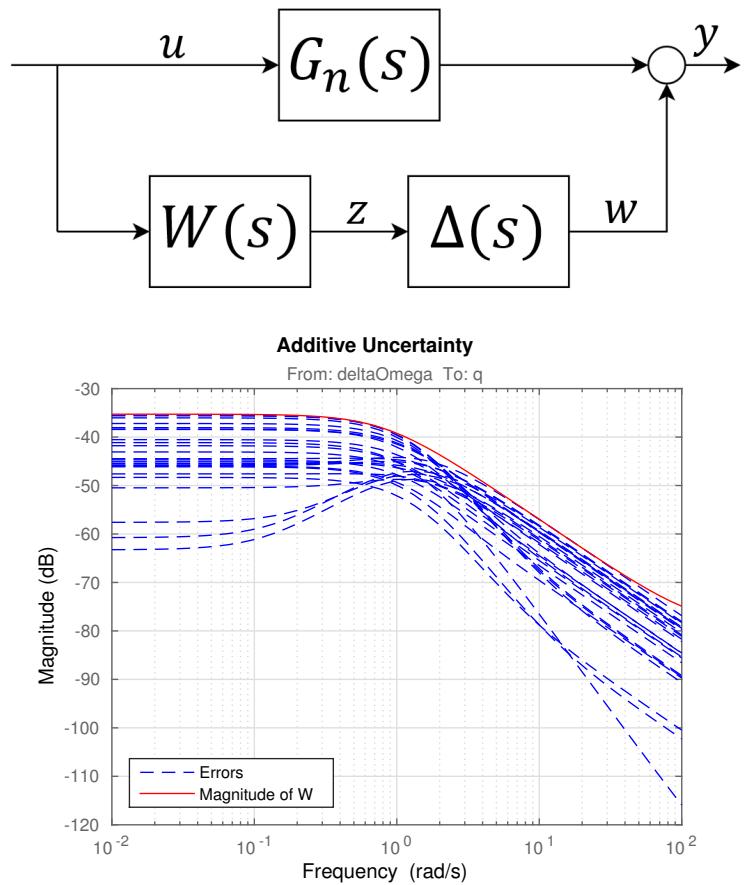


Figure 7.8: Additive uncertainty

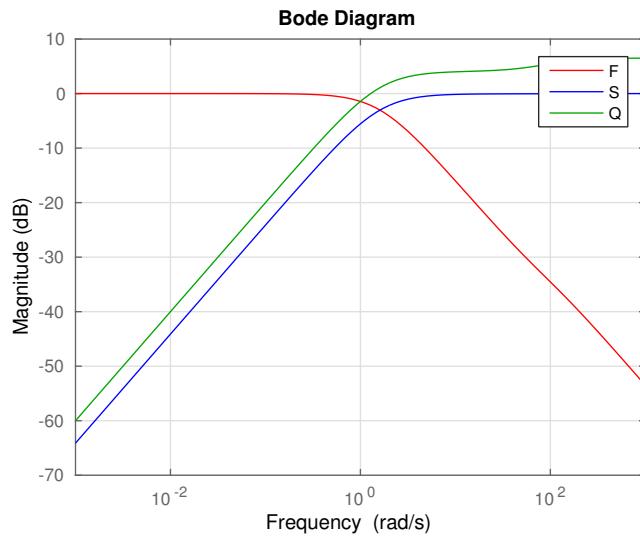


Figure 7.7: Sensitivity functions

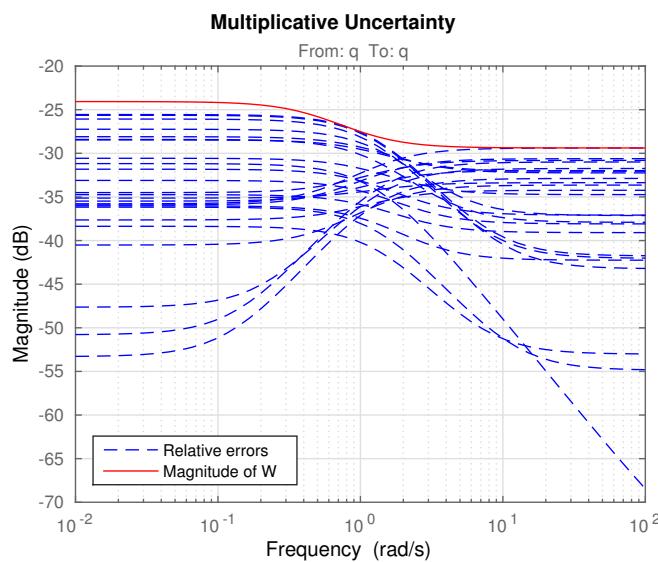
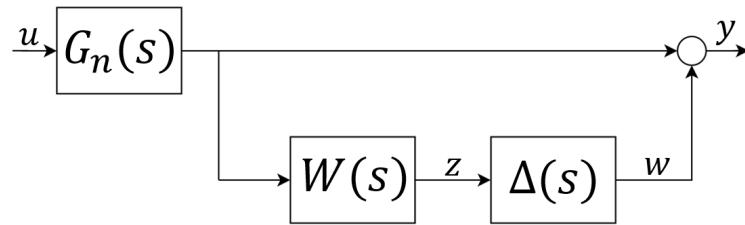


Figure 7.9: Multiplicative uncertainty

Since the regulators have been tuned on the nominal value of the dynamical model, due to the uncertainties defined before, one has to check if the closed control loop is still stable, or in other words, test the parametric robustness of the controllers.

A Monte Carlo simulation on the controlled system with the uncertain values of the parameters has been conducted. In Figure 7.10 it is possible to evaluate the results obtained using the multiplicative uncertainty modelization of 50 possible uncertain models which  $\Delta(s)$  has unitary peak gain . From the analysis of the open loop function  $L(s)$  (Figure 7.10) one can conclude that the stability of the tuned controllers is not affected by the parametric uncertainty.

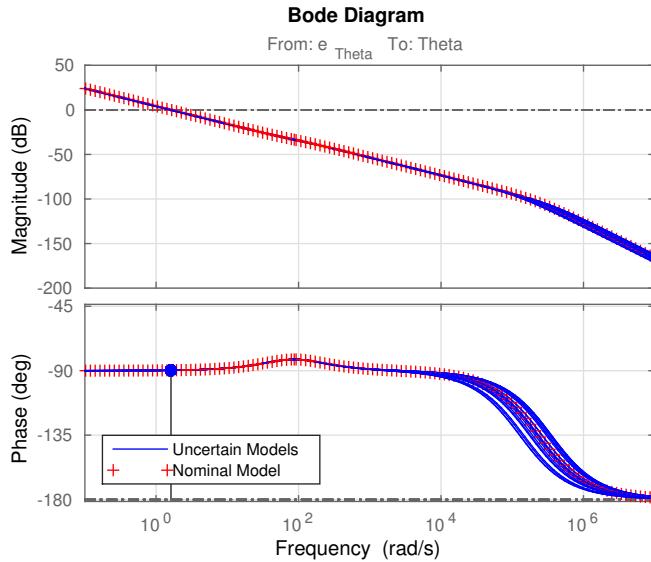


Figure 7.10: Loop function

## 7.4 Implementation of the controllers in FCU

In order to write the designed and tuned regulators inside the FCU, the previously described architecture has been implemented in Simulink (Figure (7.11)).

Then, with the automated code generator, Simulink Coder, it is possible to generate the C and C++ code from the Simulink diagram. It is possible to generate code that can be directly implemented on the target hardware, without the necessity to interact with the C/C++ code. One has only to set the type of the language (C++), the frequency at which the FCU is running (100 Hz) and the target hardware architecture (ARM Cortex).

The advantages are that with the same diagram one can design, test and simulate, in MATLAB and Simulink framework, the desired control loop with the previously designed (Chapter 4) simulator of the quadrotor dynamics, as shown in Figure 7.12.

From Figure 7.11, one can see there is a rotation matrix between the inner regulators and the outer regulators. This is because, in order to consider the relationship between the angular velocity in body reference frame,  $p, q, r$ , input of the inner loop, and the angular velocity in Earth reference frame,  $\dot{\Phi}, \dot{\Theta}, \dot{\Psi}$ , output of the outer regulators, one has to introduce the rotation matrix  $E(\Phi, \Theta)$  as described in Chapter

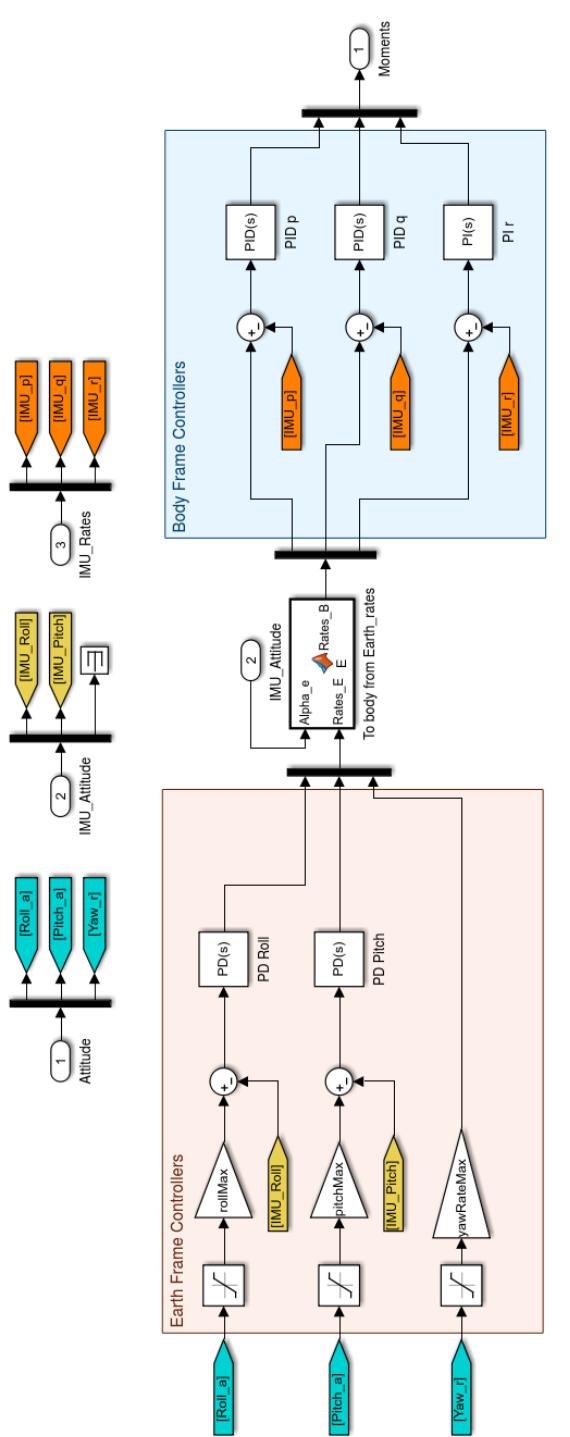


Figure 7.11: Attitude Controllers in Simulink

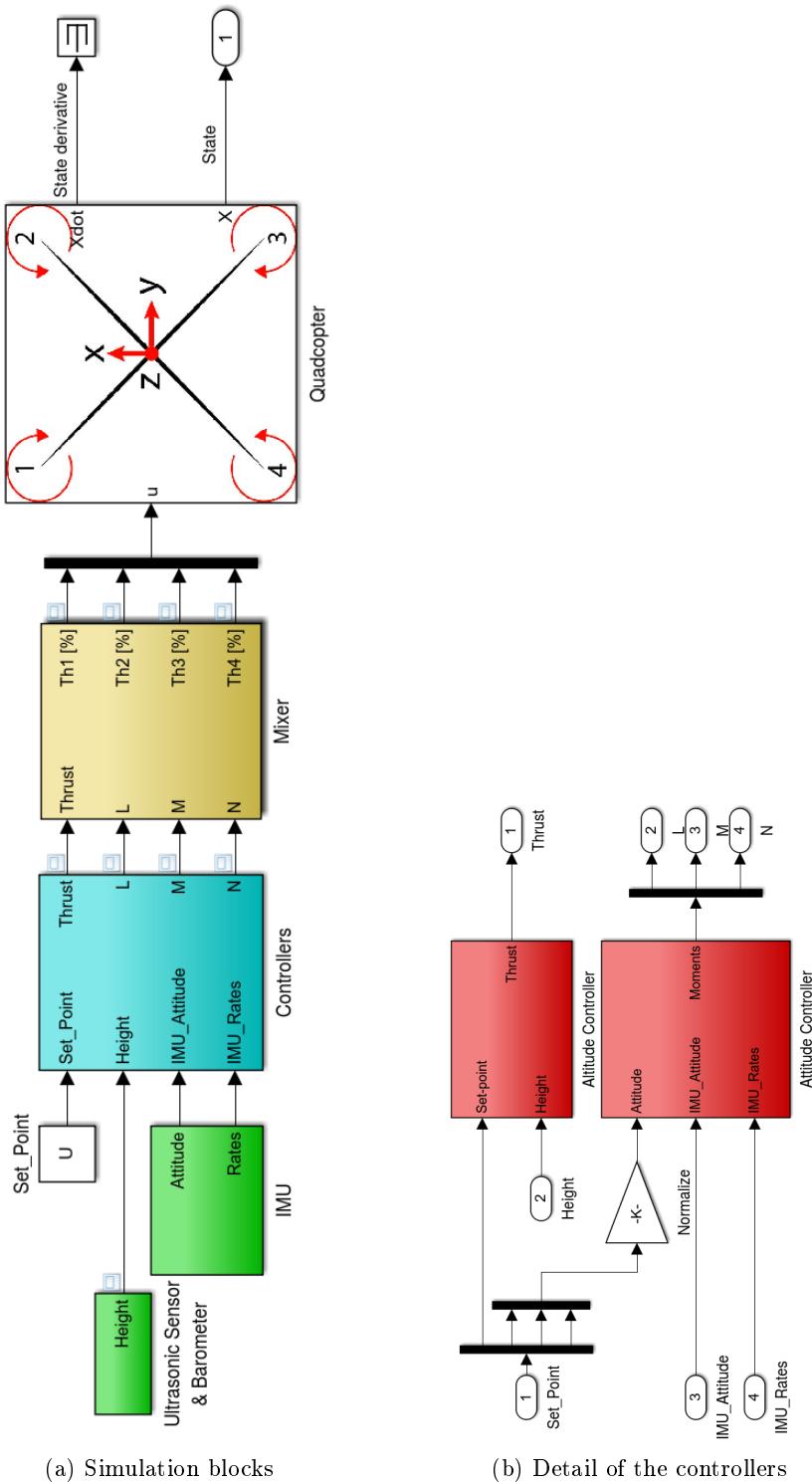


Figure 7.12: Complete Simulator in Simulink

4.

In addition, the input set-point of the overall regulation block has been normalized, in order to have an input limited between  $\pm 1$  and then scaled with a gain factor corresponding to the maximum desired angle/angular-rate.

While, from Figure 7.12 it is possible to notice, in addition to the attitude regulators, object of interest of this thesis, an altitude regulator, that will not be discussed in this thesis.

## 7.5 Experimental results

To validate the obtained tuning, an experimental campaign has been conducted on the test-bed that constraints all the DOFs except for the pitch dynamics. All controllers have been disabled except for the pitch attitude one. A desired pitch angular set-point command is assigned, testing different step amplitudes ranging from  $10^\circ$  to  $30^\circ$  of pitch attitude.

During the experiments, all four rotors are working, with a base throttle that guarantees a total thrust equal to the vehicle weight (hovering). Clearly the experimental data are also affected by IMU measurements noise but, thanks to the test-bed, the vehicle is not affected by the ground effect.

In Figures 7.13-7.14 one can see the obtained results in two different settings conditions: the first one is with the  $H_\infty$  parameters and the second one is with the guess parameters used as a starting point for the  $H_\infty$  synthesis. The results show how the  $H_\infty$  controllers are more precise than the guess regulators because are less subjected to the oscillations of the angle, the rising time is smaller and also the control variable is more conservative.

In addition, to evaluate the disturbance rejection of the controller architecture (with the  $H_\infty$  parameters), a load disturbance has been applied to the motors (Figure 7.15). In detail, the percentage of throttle of the front motors has been subjected to a square-wave disturbance signal. The test has been conducted under the same laboratory conditions as described above.

From the obtained results one can see how the integral action of the inner loop guarantees a static precision even in case of load disturbance to the quadrotor's motors.

## 7.6 Conclusions

The attitude control architecture and its implementation on the main board have been presented in this chapter. In detail, the work-flow from the mission requirements to the optimal tuning of the regulators have been discussed in details.

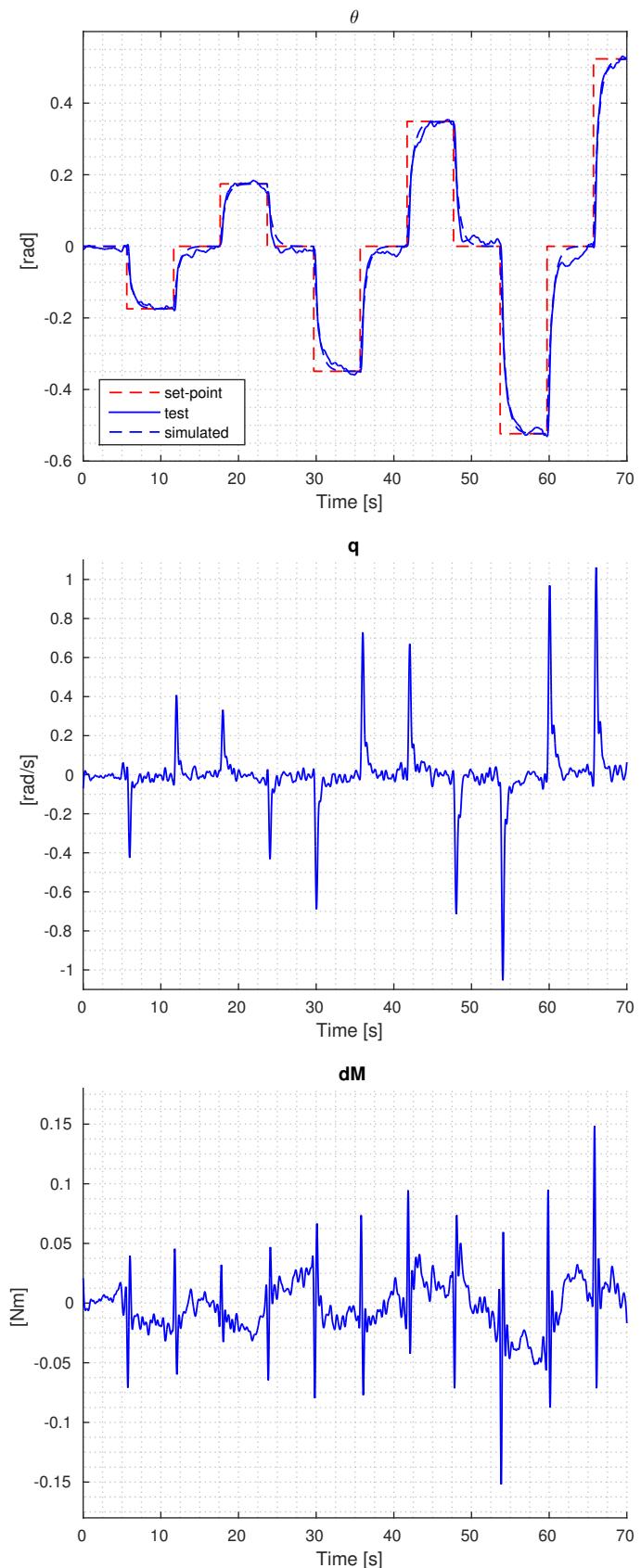


Figure 7.13: Step test with optimal parameters

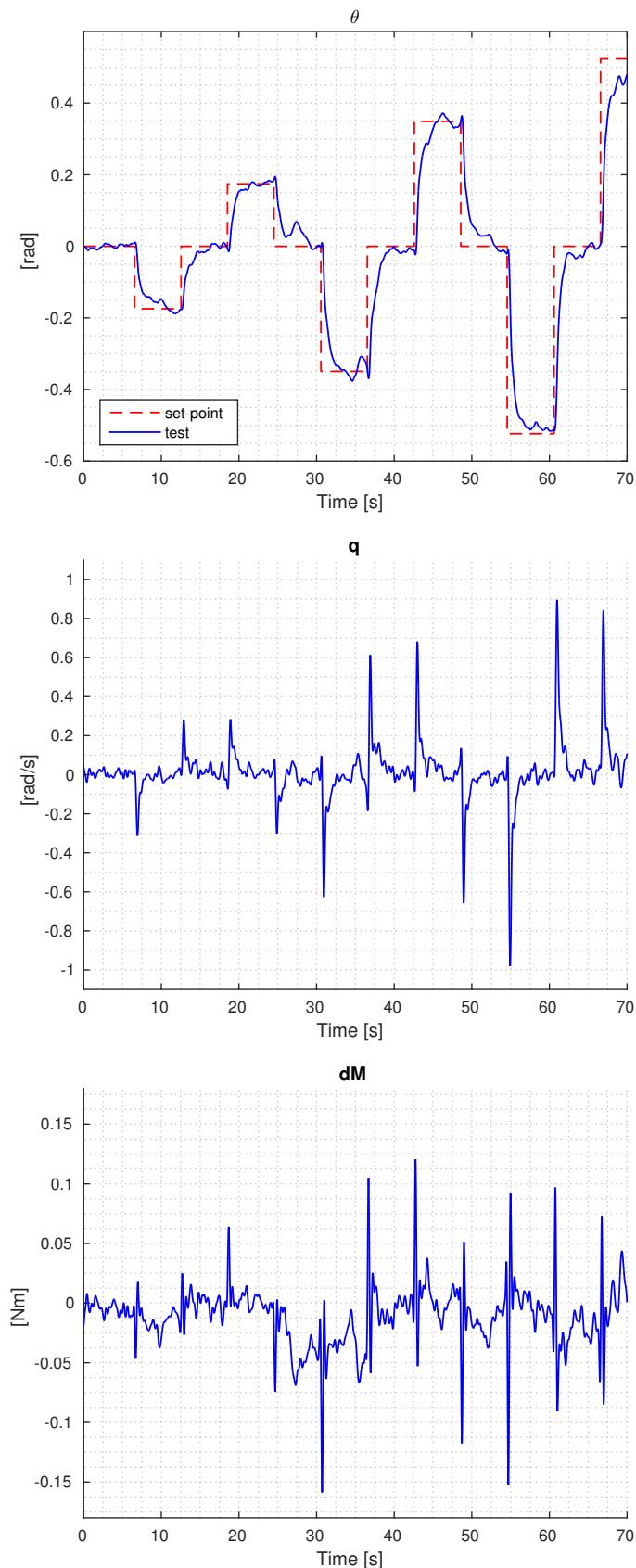


Figure 7.14: Step test with guess parameters

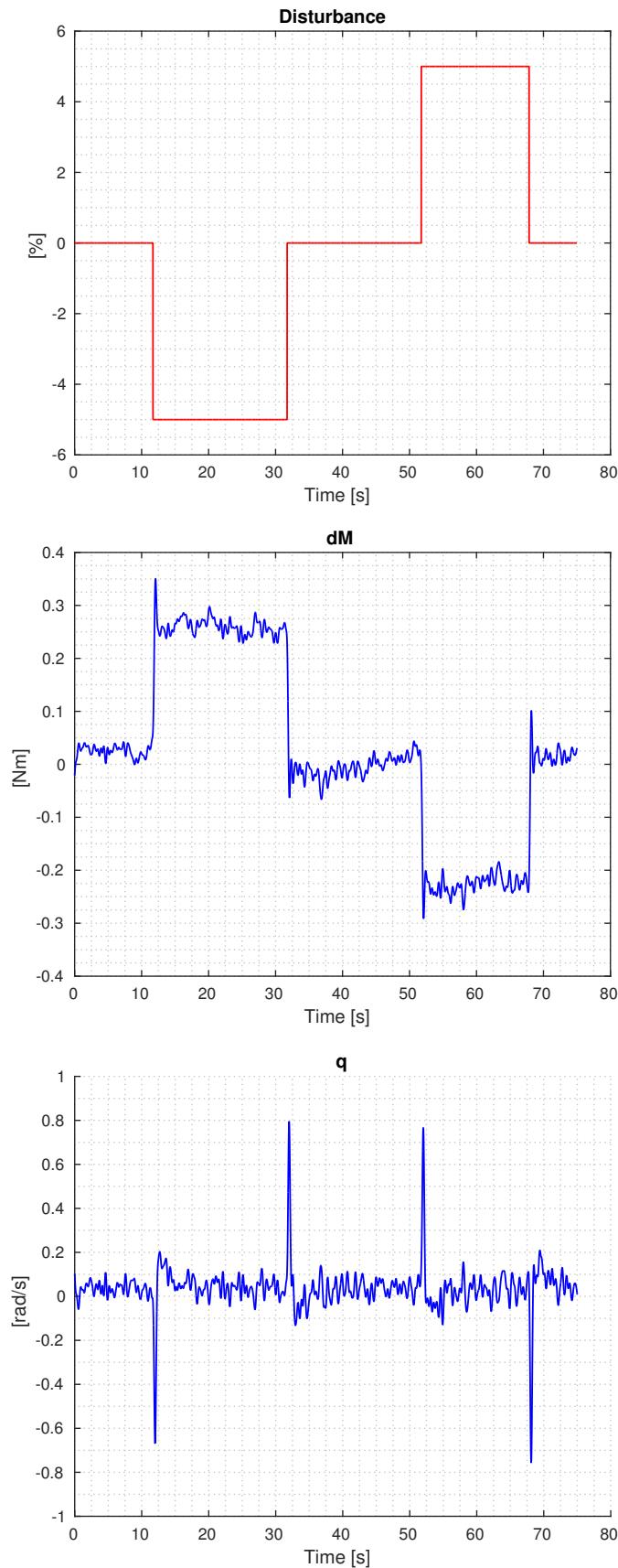


Figure 7.15: Load disturbance to the motors

From the data obtained in experimental campaigns under the same laboratory conditions, one can conclude that the regulators obtained with the  $H_\infty$  structured synthesis have a satisfactory dynamical response and have no affection to the parametric uncertainties of the model under control.



# Conclusions

In this thesis, a systematic approach for the development of a quadrotor helicopter has been presented.

In detail, the description of the component sizing, the calibration of the integrated IMU and the implementation of the attitude estimation algorithm have been described.

Moreover, the methods and the tools used for the characterization of the actuators and the roll/pitch attitude dynamics have been studied. Thanks to these methods and tools used in several identification campaigns, the propeller's parameters and other physical coefficients of the quadrotor has been estimated.

Furthermore, the design of the attitude control architecture, its optimal tuning using the  $H_\infty$  structured synthesis and the actual implementation on the on board Flight Control Unit have been presented.

In addition, in Appendix A is described a utility hand-book about the use of a general multicopter in order to reduce the possibility of injuries.

## Further developments

The possible further developments of this project are several.

- First of all, one could make more automated the identification procedures described in this thesis in order to obtain more reliable estimations with less human intervention.
- One could implement a more complete state estimator of the quadrotor in order to estimate more states values such as the translational velocities and then introduce higher-level control laws for these attitude parameters.
- Could be also possible to introduce another R2P module such as the Proximity module in order to obtain more information from the environment like the distance from the ground or from the objects nearby. In this way one could implement an altitude regulator (only for altitude less than 10m), the automatic take-off, landing and the obstacle avoidance functions.

- Enable the GPS and the barometer inside the installed IMU module to obtain a complete state estimation and introduce then a guidance and navigation algorithm.
- Furthermore, an interesting field of study could be the adoption of the proposed approach for the realization of a different platform with different mission requirements, e.g., adopt this procedural approach for the design of overactuated platforms such as platforms with tilting rotors.

# Bibliography

- [1] Paolo Bolzern, Patrizio Colaneri, and Arturo Locatelli. *Controllo e filtraggio H2/Hinf*. Polipress, November 2010.
- [2] Andrea Bonarini, Matteo Matteucci, Martino Migliavacca, and Davide Rizzi. R2p: An open source hardware and software modular approach to robot prototyping. *Robotics and Autonomous Systems*, 62(7):1073–1084, 2014.
- [3] Mark E. Dreier. *Introduction to Helicopter and Tiltrotor Flight Simulation*. AIAA Education Series.
- [4] Pascal Gahinet and Pierre Apkarian. Structured hinf synthesis in matlab. 2011.
- [5] Bruce P Gibbs. *Advanced Kalman filtering, least-squares and modeling: a practical handbook*. John Wiley & Sons, 2011.
- [6] John J Hall, Robert L Williams II, and F van Graas. Inertial measurement unit calibration platform. *Journal of Robotic Systems*, 17(11):623–632, 2000.
- [7] Ravindra V Jategaonkar. *Flight vehicle system identification(a time domain methodology)*. American Institute of Aeronautics and Astronautics, 2006.
- [8] J Gordon Leishman. *Principles of Helicopter Aerodynamics with CD Extra*. Cambridge university press, 2006.
- [9] S Madgwick, Ravi Vaidyanathan, and A Harrison. An efficient orientation filter for inertial measurement units (imus) and magnetic angular rate and gravity (marg) sensor arrays. Technical report, 2010.
- [10] Sebastian OH Madgwick. Automated calibration of an accelerometers, magnetometers and gyroscopes-a feasibility study. Technical report, Technical report, x-io Technologies Limited, Bristol, UK, 2010.
- [11] Øyvind Magnussen, Morten Ottestad, and Geir Hovland. Calibration procedure for an inertial measurement unit using a 6-degree-of-freedom hexapod. Technical report.

## Bibliography

---

- [12] Benjamin Peter. Development of an automatic imu calibration system. Master's thesis, 2011.
- [13] Paul Pounds, Robert Mahony, and Peter Corke. Modelling and control of a large quadrotor robot. *Control Engineering Practice*, 18(7):691–699, 2010.
- [14] Raymond W Prouty. *Helicopter performance, stability, and control*. 1995.
- [15] Fabio Riccardi, Pietro Panizza, and Marco Lovera. Identification of the attitude dynamics for a variable-pitch quadrotor uav. In *European Rotorcraft Forum (ERF) 2014 proceedings*, 2014.
- [16] John Watkinson. *Art of the Helicopter*. Butterworth-Heinemann, 2003.

## Appendix A

### Multicopter safety

Multicopters are powerful, flying robots and require a conscious awareness for safety concerns. Following is described a decalogue to be followed in order to avoid injuries.

First of all it is necessary to underline that most countries have a prominent aircraft organization. In Italy for example there is the ENAC that has established (and continues to update) rules for UAV's and for FPV flight.

1. WARNING: Your first priority must be the safety of people!
2. Crashes can happen, because of pilot error or hardware or software malfunction.
3. If you are flying anywhere near other people, you are putting them at risk!
  - Be sure to maintain safe distances between yourself, and spectators and your copter.
  - Circumstances will require that you will need to make your own determination of what is a “safe distance” from people and property. At a minimum, consider: at least 3m but not further than 10m from you.
  - Keep all other people, property and obstacles considerably further away from your copter.
  - Ensure that no one gets between you and your copter.
  - Spectators should always be a safe distance behind the pilot.
  - If people intrude beyond what you have determined to be the “safe” area, land immediately and do not take off until they are clear.
4. If you are flying too high or near airports you are putting manned aircraft and the people on them at risk!
  - Get to know where your nearest airports are and do not fly anywhere near them.

## A. Multicopter safety

---

5. Always ensure the battery cable is NOT connected to the power distribution board or harness until you are ready to fly.
  - Always Turn on the transmitter and ensure the throttle stick is all the way down BEFORE connecting the battery.
  - After landing the first thing you should do is disconnect your battery cable.
  - Do not turn off the transmitter until after you have disconnected the battery.
  - Always remove your props while you are testing motors, your hands, arms and face and those of your friends will thank you.
  - When the battery is connected, always assume the motors are armed; You can check with a short throttle pulse.
  - Don't pick up the model and the radio at the same time, you may bump the throttle.
  - Do not attempt to fly longer than your batteries safe capacity, it is very bad for the battery and can cause a crash.
6. It is very important to have excess power available.
  - If you have insufficient power, the automatic controls can require more throttle than is available and destabilization may result.
  - Ideally your copter should be able to hover at about 50% throttle (mid stick).
7. Especially while you are learning, it is recommended that you avoid expensive, stiff, ultra-sharp carbon fiber props.
  - Get cheaper, more flexible and more breakable plastic propellers.
  - Some of the super carbon fiber ones can cut better than a Ginsu and while they are almost indestructible – You are not.
8. Important primary response to a crash, inadequate landing or unknown flight controller state.
  - The first thing to do is throw a towel over your copters propellers (Props may start spinning unexpectedly).
  - Then immediately disconnect the battery.
  - A large towel is your most important piece of safety equipment followed by a fire extinguisher and a first aid kit.

- 
- Generally better to use the first one than the last one.

9. Always follow the law:

- Our personal use of Multicopters (models in general) is continually under attack by those who fear ‘drones’ and invasion of privacy. If you break the law, or invade someone’s privacy, or put them in harm’s way, you threaten the future of our personal use of models. Please, understand the law and the rights of others – and fly accordingly.

10. Most important: Keep a safe distance between your Copter and People!

