

# **Politecnico di Milano**

Scuola di Ingegneria Industriale e dell'Informazione

Laurea Magistrale in Ingegneria dell'Automazione



## **Design, identification and control of a tiltrotor quadcopter UAV**

Supervisor: Prof. Marco LOVERA

Co-Supervisor: Ing. Mattia GIURATO

Thesis by:

Claudio MICHELI Matr. 824268

Academic Year 2015 - 2016



*Alla mia famiglia*



# Abstract

Nowadays, Unmanned Aerial Vehicles (UAVs) research has vastly increased due to the various tasks that can be accomplished with the use of such aerial platforms. Among all the UAV categories, multirotor UAVs are getting more and more diffused in both civilian and military fields for short range missions thanks to their simple and reliable structure.

Even if some of the multirotor vehicles present an over actuated structure (see for example hexacopters, octocopters,...), none of them is able to achieve the full control of the 6 DoFs that parametrize the vehicle position/orientation in the space. The kinematic redundancy of such vehicles is only exploited to overcome faulty rotors and increase the overall payload capabilities. An aerial platform that is able to overcome the mobility limitations inherited by traditional multicopter structures is necessary to fully take advantage of the interaction capabilities of a flying vehicle with the environment. This thesis work will deal with the problem of designing an over actuated tiltrotor quadcopter platform able to gain, in a decoupled way, the full control of all the 6 DoFs of the vehicle in the space.

In detail, the mechanical design of an over actuated quadcopter platform in which the arms are able to tilt around their respective axes will be presented. The designed aerial platform will have a total of 8 control inputs provided by the spinning velocities of each rotor group and by the tilting angles of each arm. On the actual prototype, a comprehensive modelling and identification development will be performed in order to obtain a trustworthy simulation environment. The problem of designing a robust control architecture, optimally tuned with  $H_\infty$  synthesis, able to fully control in a decoupled way all the 6 DoFs, is then evaluated.

Finally, the performances of the designed system are proved and illustrated with simulations and real experiments.



# Sommario

Al giorno d'oggi, la ricerca nel campo degli aeromobili a pilotaggio remoto (APR) si è diffusa molto grazie alle svariate possibilità di impiego in cui questo tipo di piattaforme possono essere impiegate.

Fra tutte le categorie di APR, quella che più si sta diffondendo sia in ambiti civili che militari, è quella dei veicoli multirotore. Questo tipo di piattaforme aeree, fornisce una soluzione semplice ed affidabile per la realizzazione di APR da essere impiegati nel corto raggio.

Tuttavia va sottolineato come, sebbene esistano piattaforme multirotore con un numero di attuatori pari o addirittura maggiore rispetto ai 6 Gradi di Libertà (GdL) del veicolo nello spazio (si vedano per esempio esacotteri e octacotteri), nessuno di questi è in grado di muoversi liberamente nello spazio controllando tutti e 6 i GdL. In questo tipo di configurazioni, i rotori ridondanti vengono sfruttati solamente per permettere al veicolo di restare in volo nel caso in cui un motore si dovesse danneggiare e per aumentarne la capacità di carico.

Per poter sfruttare a pieno le capacità di interazione fra una piattaforma aerea e l'ambiente circostante, potrebbe essere utile avere un veicolo in grado di muoversi liberamente in tutti e 6 i GdL, ovvero una piattaforma aerea in grado di controllare le dinamiche di assetto e posizione in modo disaccoppiato.

Il lavoro presentato in questa tesi, consiste proprio nel progettare una piattaforma in stile quadricottero, ma dotata di 4 bracci rotanti su cui sono montati 4 gruppi rotore, per un totale di 8 variabili di controllo, costituite dalla velocità di rotazione dei rotori e dagli angoli di tilting dei bracci, in grado di gestire completamente i 6 GdL del veicolo.

Più in dettaglio, verranno illustrate le fasi di progetto che hanno portato alla realizzazione del prototipo di quadrirotore con bracci tiltanti. Tale prototipo verrà poi modellizzato e i suoi parametri identificati in modo da poter sviluppare un ambiente di simulazione che rappresenti fedelmente il comportamento del veicolo.

Sulla base del modello ottenuto, verrà affrontato il problema di progettazione di un controllo robusto tramite sintesi  $H_\infty$ , in grado di gestire, in modo disaccoppiato, le dinamiche di assetto e posizione del veicolo.

In fine, grazie ad una esaustiva campagna di test eseguiti sia in simulazione, sia in realtà, sono illustrate e valutate le prestazioni dell'architettura di controllo progettata.

# Acknowledgments

Foremost, I would like to express my sincere gratitude to my supervisor Professor Marco Lovera for not only having given me the possibility to work on a project so close to my interests, but also for his patience, all the support and useful advices which always steered me in the right direction.

Besides my supervisor, I would like to thank Mattia for all the generous support from both the technical and moral point of view he gave me during all this thesis work.

I must also express my gratitude to Alessandro and Andrea for the help provided in integrating the tiltrotor code into the Flight Control Unit.

I would also like to thank Davide and all the AIRLab labmates for their insightful comments, the stimulating discussions and having trusted me during the in-flight experiments.

Coming to my friends, thank you Giacomo for all the lunches we have spent together and all your useful advices during the lectures and the thesis period.

My heartfelt thank you, goes to Basi, Fabio, Gigi, Alice and Taty for having sustained and pushed me to work harder to reach my goals, but also to have encouraged me during the tough times i have been through.

Even though our paths separated, I would also thank the person who helped me on being where I am today, thank you Marisa.

Last but not least I must express my very profound gratitude to my family for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis.

This accomplishment would not have been possible without them. Thank you.



# Contents

<b>Abstract</b>	<b>I</b>
<b>Sommario</b>	<b>III</b>
<b>Acknowledgments</b>	<b>V</b>
<b>Contents</b>	<b>IX</b>
<b>List of Figures</b>	<b>XIII</b>
<b>List of Tables</b>	<b>XV</b>
<b>1 Introduction</b>	<b>1</b>
1.1 State of the art . . . . .	3
1.2 Thesis structure . . . . .	5
<b>2 Tiltrotor mechanical design</b>	<b>7</b>
2.1 Preliminary requirements . . . . .	7
2.2 Tilting mechanism design . . . . .	8
2.3 Feasibility evaluation . . . . .	10
2.4 Components . . . . .	13
2.4.1 Servomotors . . . . .	13
2.4.2 Flight Control Unit . . . . .	15
2.4.3 Motors and ESCs . . . . .	15
2.4.4 Propellers . . . . .	17
2.4.5 Battery . . . . .	19
<b>3 Actuators charaterization</b>	<b>21</b>
3.1 Aerodynamic forces and moments . . . . .	21
3.2 Rotor actuators charaterization . . . . .	22
3.3 Tilting actuators charaterization . . . . .	23
3.3.1 Servomotor model definition . . . . .	23

3.3.2	SRIVC system identification method . . . . .	26
3.3.3	Experimental identification . . . . .	29
<b>4</b>	<b>Modelling of the tiltrotor quadcopter</b>	<b>35</b>
4.1	Reference frames . . . . .	35
4.1.1	Coordinate transformations . . . . .	37
4.2	Kinematics . . . . .	39
4.3	Quaternions . . . . .	42
4.4	Coriolis equation . . . . .	45
4.5	Equations of motion . . . . .	46
4.5.1	Translational motion . . . . .	47
4.5.2	Angular motion . . . . .	48
4.6	State space model . . . . .	52
4.7	External forces and mixer matrix . . . . .	53
4.7.1	Linearized mixer matrix . . . . .	55
4.7.2	Direct resolution of the kinematic problem . . . . .	58
<b>5</b>	<b>Identification of attitude dynamics</b>	<b>61</b>
5.1	Dynamic model of roll attitude . . . . .	61
5.1.1	Experimental setup . . . . .	63
5.1.2	Excitation input signal - Random Binary Sequence . .	64
5.1.3	Identification results . . . . .	66
5.1.4	Choice of best identified model . . . . .	67
5.1.5	Time delay influence on grey-box identification . . .	68
5.1.6	Final results and remarks . . . . .	69
<b>6</b>	<b>Simulink implementation of the tiltrotor model</b>	<b>73</b>
6.1	Main blocks definition . . . . .	73
<b>7</b>	<b>Control law design</b>	<b>77</b>
7.1	$H_\infty$ fixed-structure synthesis problem formulation . . . . .	77
7.2	Control law design . . . . .	78
7.2.1	Attitude control . . . . .	78
7.2.2	Translational control . . . . .	84
7.3	Attitude control law robustness analysis . . . . .	87
7.4	Implementation of the controllers in the Flight Controller Unit (FCU) . . . . .	88
<b>8</b>	<b>Experimental results</b>	<b>93</b>
8.1	Testbed experiments . . . . .	93
8.1.1	Reference tracking . . . . .	93

8.1.2	Disturbance rejection . . . . .	95
8.2	In-flight experiments . . . . .	95
8.2.1	Attitude dynamics . . . . .	95
8.2.2	Translational dynamics . . . . .	95
<b>9</b>	<b>Conclusions</b>	<b>101</b>
9.1	Further developments . . . . .	102



# List of Figures

1.1	RUAG ADS 95 Ranger . . . . .	1
1.2	DJI Agras MG-1 . . . . .	2
1.3	Examples of tiltrotor aircrafts . . . . .	3
1.4	Tiltrotor prototypes . . . . .	4
2.1	Tilting mechanism . . . . .	9
2.2	Exploded view of the tiltrotor quadcopter . . . . .	10
2.3	Estimated performance results . . . . .	11
2.4	3D prototype of the tiltrotor quadcopter . . . . .	12
2.5	Real tiltrotor quadcopter . . . . .	12
2.6	Servomotor breakdown . . . . .	13
2.7	Pulse command . . . . .	14
2.8	Servomotor range of motion . . . . .	15
2.9	Flight controller architecture . . . . .	16
2.10	RCTimer HP2814 . . . . .	16
2.11	Three phase modulated signal . . . . .	17
2.12	Propeller's parameters . . . . .	18
2.13	Propellers . . . . .	18
3.1	Servomotor block diagram . . . . .	24
3.2	Electro-mechanical scheme of a DC motor . . . . .	24
3.3	Optitrack system . . . . .	30
3.4	Reference signal and measured output angle . . . . .	31
3.5	Closed loop block diagram . . . . .	32
3.6	Identified model with uncertainty boundaries . . . . .	33
3.7	Identified model output vs real output comparison . . . . .	34
4.1	Reference frames . . . . .	36
4.2	Tiltrotor coordinate systems . . . . .	37
4.3	Axes of rotation . . . . .	38
4.4	3-2-1 Euler rotations . . . . .	39

4.5	Gimbal lock example in airplanes . . . . .	42
4.6	Quaternion components . . . . .	43
4.7	Derivation of the equation of Coriolis . . . . .	45
4.8	<i>i</i> -th tilting arm and relative physical quantities . . . . .	47
4.9	Rotating vector . . . . .	49
4.10	Forces and torques acting on the quadrotor . . . . .	54
5.1	Experimental setup . . . . .	63
5.2	Example of a RBS input signal . . . . .	64
5.3	Error bars diagram . . . . .	67
5.4	Frequency response comparison: grey-box no time-shifted identified model (blue line) vs SRIVC identified model (red line) .	69
5.5	Frequency response comparison: Grey-box time-shifted identified model (blue line) vs SRIVC identified model (red line) .	70
5.6	Time response comparison: Grey-box time-shifted identified model (green line) vs SRIVC identified model (red line) vs real output (blue line) . . . . .	70
5.7	Block diagram of the overall identified system . . . . .	71
6.1	Simulink model . . . . .	74
6.2	Servomotor simulink model . . . . .	75
6.3	ESC+Motor simulink model . . . . .	75
6.4	3D plot of the tiltrotor . . . . .	76
6.5	Tiltrotor block model . . . . .	76
7.1	Standard representation for structured $H_\infty$ synthesis . . . . .	78
7.2	Nested loops attitude control architecture . . . . .	79
7.3	Tracking error requirements . . . . .	81
7.4	Outer-loop disturbance rejection requirement . . . . .	81
7.5	Inner-loop sensitivity functions . . . . .	83
7.6	Outer-loop sensitivity functions . . . . .	83
7.7	Translational dynamics closed-loop block diagram . . . . .	85
7.8	$H_\infty$ requirements for translational dynamics . . . . .	86
7.9	Translational dynamics sensitivity functions . . . . .	87
7.10	System with uncertainty block diagrams . . . . .	88
7.11	Frequency response of the relative gaps between uncertain models and nominal model vs. the shaping filter $W(s)$ . . . . .	89
7.12	$L_{outer}(s)$ stability robustness in front of plant uncertainty . .	90
7.13	Simulink implementation of translational controllers . . . . .	90
7.14	Simulink implementation of attitude controllers . . . . .	91

8.1	Testbed experimental results . . . . .	94
8.2	Disturbance rejection experimental results . . . . .	96
8.3	In-flight attitude experiments . . . . .	97
8.4	Translational simulated model vs real model comparison . . .	98
8.5	Coupled dynamics, translational control disabled . . . . .	98
8.6	Decoupled dynamics, attitude setpoint tracking requiring null translational velocities . . . . .	99
8.7	Decoupled dynamics, translational velocity setpoint tracking while requiring null attitude . . . . .	100
9.1	Tiltrotor quadcopter performing a decoupled dynamics ma- neuver . . . . .	102



# List of Tables

2.1	Parameters of the HS-485HB servomotor . . . . .	14
3.1	Parameters meaning . . . . .	21
3.2	Order of the closed loop system . . . . .	26
3.3	Best estimation performance indexes . . . . .	31
3.4	Servomotor model parameters . . . . .	32
4.1	State variables of the tiltrotor quadcopter model . . . . .	40
5.1	RBS Parameters . . . . .	65
5.2	Weighted average parameters and standard deviation $\sigma$ . . . . .	66
5.3	SRIVC identified time delay . . . . .	66
5.4	Choice of the best model based on VAF . . . . .	68
5.5	Grey-box weighted average parameters considering the time delay . . . . .	68
7.1	Optimal tuning parameters for the inner regulator $R_p(s)$ and the outer regulator $R_\phi(s)$ . . . . .	84



# Chapter 1

## Introduction

Nowadays, Unmanned Aerial Vehicles (UAVs) research has vastly increased due to the various tasks that can be accomplished with the use of such aerial platforms. The term of UAV identifies any kind of flying vehicle that is able to fly without an on-board human pilot. Exploration and surveillance of inaccessible disaster areas, precision agriculture, 3D mapping and data collection are only a few fields in which UAVs are currently being employed in both civilian and military domains. Depending on the purpose, different kinds of vehicle structures can be used: fixed-wing, rotorcraft (or rotary-wing) and Vertical Take Off and Landing (VTOL) vehicles are the most used.



Figure 1.1: RUAG ADS 95 Ranger

Figure 1.1 shows a fixed-wing UAV developed by RUAG Aerospace and adopted by the Swiss Air Army forces to surveil the territory and the borders. It has an autonomy of about 4 hours and is able to fly up to 100

km far from the control station. Figure 1.2 shows a multirotor (octocopter) recently designed by DJI for precision variable rate application of liquid pesticides, fertilizers and herbicides. The drone can carry up to 10 kg of liquid payloads and is able to cover an area of  $4'000\text{-}6'000\ m^2$  in just 10 minutes. This technology allows to drastically reduce the time required for spraying operations: up to 40 to 60 times faster than manual spraying. Those introductory examples are the proof that UAVs not only are fascinating from a scientific point of view, but their use can produce a beneficial impact on our society. This thesis will focus on multirotor UAVs (which are part of the rotorcraft family), more precisely on quadcopters. Effectively, depending on the number of actuators used, a multirotor can be precisely called tricopter, quadcopter, hexacopter or octocopter if it has 3, 4, 6 or 8 rotors respectively. In all the configurations, control of the vehicle is achieved by varying the relative speed of each rotor. Even if some of the presented configurations have an over actuated structure (more independent control inputs than the actual 6 Degrees of Freedom (DoF) that describe the orientation/position of the vehicle in space), none of them is able to arbitrarily follow a trajectory with decoupled attitude and position. In some cases could indeed be useful to have an aerial platform that is able to translate while maintaining its central core leveled. Consider for example the case of the DJI Agras MG-1: the spraying angle is strongly affected by the attitude of the vehicle, since, to translate, the craft must modify its attitude. This problem is also felt in



Figure 1.2: DJI Agras MG-1

the Aerial Photography (AP) field where, to avoid the multirotor attitude to affect the shot, cameras are usually mounted on a gimbal that compensates angular movements of the vehicle. It is thus clear, that an aerial platform that overcomes the flying limits of the classical multirotor structures could increase the exploitation of UAVs in a various fields.

## 1.1 State of the art

In the literature, the problem of designing a fully actuated UAV is still an ongoing challenge. At the state of the art, the most suitable mechanical solutions (see [11] [17] [3] [18]) take inspiration from VTOL aircrafts such as the Bell-Boeing V-22 Osprey (Figure 1.3.a) or the AgustaWestland AW609 (Figure 1.3.b) in which rotors, usually mounted at the end of the wing, can be tilted in order to allow vertical lift capabilities. The same principle



Figure 1.3: Examples of tiltrotor aircrafts

can be applied on a quadrotor frame, but in this case, instead of adding to the vehicle vertical lift capabilities, the tilting action is used to add to the quadrotor translational movement capabilities. Figure 1.4 shows the actual prototypes adopted in the literature until now that take advantage of a tilting mechanism to overactuate a classical quadrotor structure. The tilt-quadrotor ALIV3 platform [3] (Figure 1.4.a) consists of a structure with a central core and four arms each with a motor-propeller set. Two opposing arms have fixed motor-propeller sets, while the other two arms have swivelled motor-propeller sets. In this case the swivelled propeller group is able to roll-tilt and pitch-tilt such that translational forces can be generated. The presented work, [3], aimed to derive an analytical model of the ALIV3 platform (previously designed and realized in [8]), experimentally identify

its parameters and finally derive a trustworthy dynamical model in MATLAB/Simulink that will allow, in future works, the design of a control structure for the vehicle's attitude and the tilting action.

In contrast, the Holocopter [17] (Figure 1.4.b) structure has been designed with a different setup. The tilting action is achieved thanks to a servomotor that is placed at the tip of each quadcopter arm, allowing a tilting movement of the rotor around the arm axis. In their work, Ryll *et al.*, were able to derive a dynamical model of the platform and design a trajectory tracking controller with feedback linearization techniques, which allowed the full decoupled control of the quadrotor position and orientation.

Taking inspiration from these past works, this thesis will be focused in designing a novel tiltrotor quadcopter platform, derive an analytical model of the system, identify its aerodynamical and structural parameters through experimental identification campaigns and finally implement a reliable MATLAB/Simulink dynamical model that will be used to design a robust control structure able to fully control the quadrotor attitude and translational dynamics. The obtained control architecture will be then validated with an experimental campaign conducted on the real system.



Figure 1.4: Tiltrotor prototypes

## 1.2 Thesis structure

The elaborate, will be structured according to the work flow introduced in [14], that in detail consists in:

- Define the preliminary requirements of the tiltrotor quadcopter vehicle.
- Based on the defined requirements, a 3D design of the frame and the tilting actuators will be realised and their feasibility evaluated.
- Once the mechanical structure has been defined and its feasibility has been checked, a mathematical model that describes the kinematics and dynamics of the vehicle will be derived according to classical mechanics principles.
- The analytical model parameters will be then identified in order to obtain a reliable simulation model.
- Based on the identified model, a control architecture that stabilises the vehicle will be then developed with robust  $H_\infty$  control theory.
- The obtained control laws will be finally tested on the real developed platform.



## Chapter 2

# Tiltrotor mechanical design

In this chapter the problem of designing a tiltrotor quadcopter platform is presented and evaluated. Firstly, the main advantages and disadvantages of the existing solutions adopted in the literature (see Section 1.1) are analysed. Based on the performed observations, mechanical requirements will be drafted and a prototype frame solution will be illustrated. Despite the easy-to-design quadrotor structure, when dealing with tilting rotor hubs, a major difficulty is to find a good and reliable actuation mechanism that will be in charge of tilting the arms.

Since the tiltrotor quadcopter is a novel architecture, it can't be found off the shelf, but a custom solution has to be designed. Designing a quadrotor helicopter is basically dealing with numerous design parameters (such as size of the motors and propellers, battery,...) that are closely linked. Taking a decision about all these parameters requires a clear methodology and some design requirements.

### 2.1 Preliminary requirements

To define the hardware requirements of the tiltrotor quadcopter, a visual inspection of the already developed solutions presented in Section 1.1 has been performed. Recalling Figure 1.4, some remarks about the hardware structure can be made:

- The ALIV3 presents a complex mechanism for the tilting actuation. The resulting aerial platform, not only has an asymmetrical frame structure, but has also an higher number of parts that can be easily broken in case of a crash.
- The Holocopter tilting actuation system is placed at the tip of each

arm and thus the tilting actuators result to be the first things to be damaged in case of a crash. Moreover the weight of the servomotors is decentralized on each arm.

For exposed remarks, the main guideline adopted to design the tiltrotor hardware is the following:

- Simple hardware structure, to simplify the maintenance on the vehicle.
- Protected tilting actuation, in order to reduce the probability of damage in case of crash.
- Centralized positioning of the servomotors needed to perform the tilting actuation such that the weight can be assumed to be all concentrated in the center of mass.

Beyond this basic requirement some physical characteristics have also been defined:

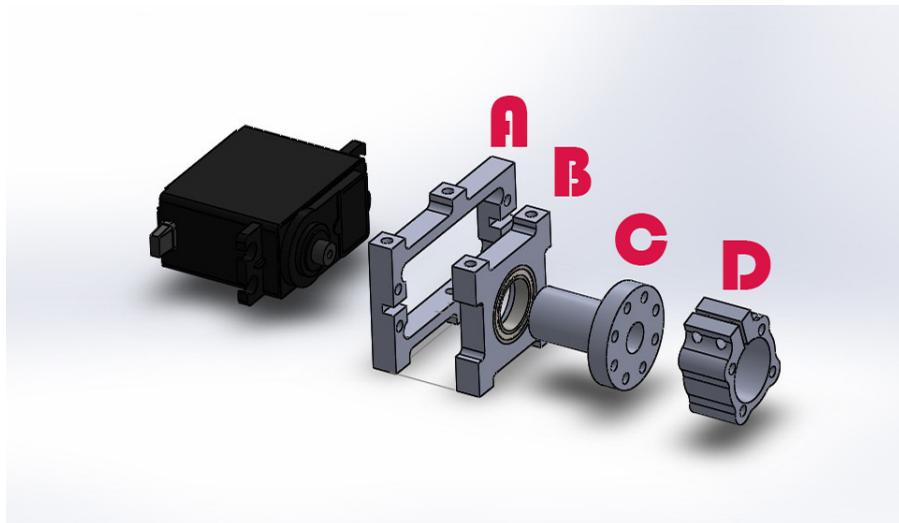
- Frame configuration: the quadrotor consists of four lever arms which are arranged similar to the algebraic sign + (plus). This requirement simplifies the actuation system needed to translate the tiltrotor in both X and Y directions.
- Frame dimensions: medium size, with about 550 mm as distance between opposite motors.
- Overall weight: less than 2 kg.
- Flight time: about 10'.

## 2.2 Tilting mechanism design

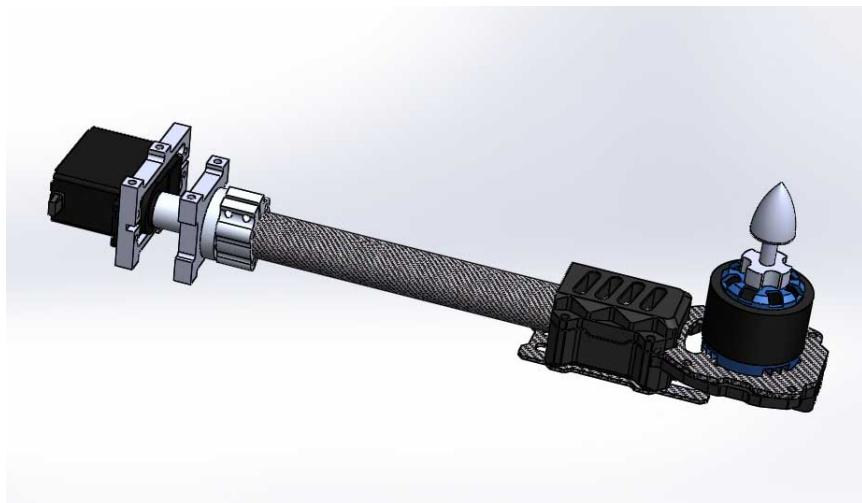
Keeping in mind the defined requirements, the tilting mechanism has been developed with Actobotics<sup>TM</sup> parts: a bearing based precision system that allows to easily prototype robotic structures.

In Figure 2.1.a an exploded view of the mechanism is depicted. The servomotor is connected to the arm via a shaft (Figure 2.1.a.A) that goes through a support equipped with a bearing system in order to reduce friction (2.1.a.B). The top and bottom plates have been custom designed to match the hole pattern of the tilting mechanism.

To reduce manufacturing costs of the prototype, FR-4 glass epoxy material has been chosen. This kind of laminate sheet provides good strength to weight ratios, and it also retains its mechanical properties and electrical insulating qualities in both dry and humid conditions. These attributes, along with good manufacturing characteristics, have been the key factor for the choice of this material. An exploded view of the overall quadrotor structure



(a) Exploded view - Tilting mechanism



(b) Assembled arm

Figure 2.1: Tilting mechanism

is depicted in Figure 2.2.

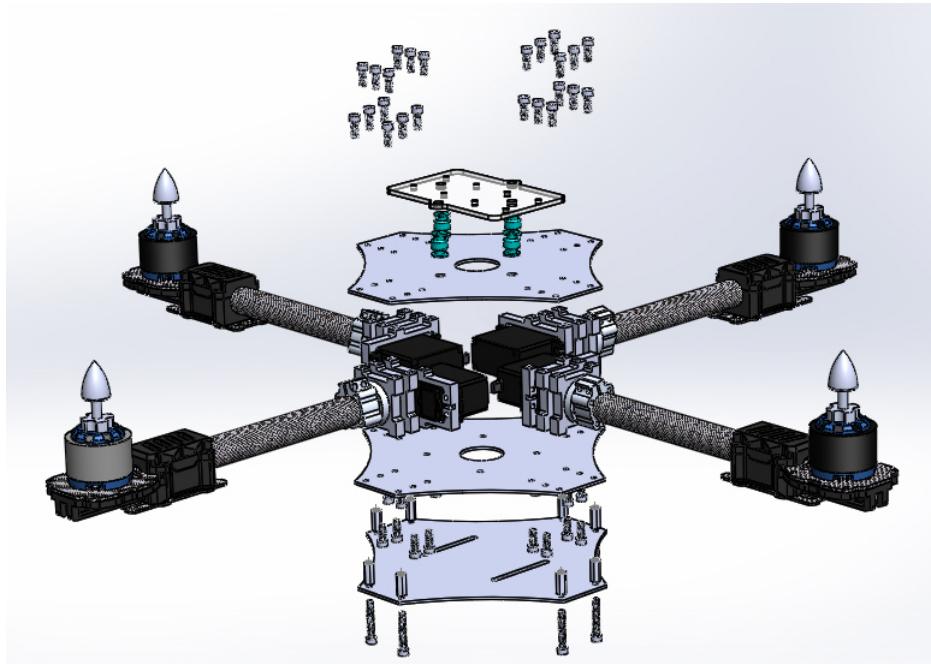


Figure 2.2: Exploded view of the tiltrotor quadcopter

### 2.3 Feasibility evaluation

In order to evaluate the feasibility of the described requirements, an online tool (eCalc.ch) that provides an estimation of the flight performances of the vehicle have been performed. Given all the parameters of each component of the quadcopter, the tool evaluates basic flight parameters such as flight time, power consumption and efficiency. In order to evaluate the best set of components, several simulations with different set of components have been performed. To this aim, a priori estimation of the All-Up-Weight (AUW) has been also performed considering the weight of each part needed to assemble the tiltrotor quadcopter; the estimated AUW has been found to have an average weight of 1800 grams.

General	Motor Cooling: good	# of Rotors: 4	Model Weight: 1800 g incl. Drive	Frame Size: 550 mm no limit	FCU Tilt Limit: no limit	Field Elevation: 500 m ASL 1640 ft ASL	Air Temperature: 25 °C 77 °F	Pressure (QNH): 1013 hPa 29.91 inHg		
Battery Cell	Type (Cont. / max. C) - charge state: custom	- normal	Configuration: 3 S 1 P	Cell Capacity: 4000 mAh 4000 mAh total	max. discharge: 85%	Resistance: 0.0056 Ohm	Voltage: 3.7 V	C-Rate: 20 C cont. 50 C max	Weight: 111 g 3.9 oz	
Controller	Type: max 30A	Current: 30 A cont. 30 A max	Resistance: 0.008 Ohm	Weight: 40 g 1.4 oz	Accessories				Current drain: 0 A	Weight: 0 g 0 oz
Motor	Manufacturer - Type (Kv): RCTimer ▾   HP2814-710 (710)	KV (w/o torque): 710 rpm/V	no-load Current: 0.7 A @ 11.1 V	Limit (up to 15s): 500 W	Resistance: 0.125 Ohm	Case Length: 34 mm 1.34 inch	# mag. Poles: 14	Weight: 100 g 3.5 oz		
Propeller	Type - yoke twist: GemFan	Diameter: 12 inch 304.8 mm	Pitch: 4.5 inch 114.3 mm	# Blades: 2	PConst / TConst: 1.13 / 0.88	Gear Ratio: 1 : 1			calculate	

(a) Input data

Battery	Motor @ Optimum Efficiency	Motor @ Maximum	Motor @ Hover	Total Drive	Multicopter
Load: 11.19 C	Current: 7.09 A	Current: 11.19 A	Current: 6.78 A	Drive Weight: 982 g	All-up Weight: 1800 g
Voltage: 10.35 V	Voltage: 10.57 V	Voltage: 10.26 V	Voltage: 10.59 V	34.6 oz	63.5 oz
Rated Voltage: 11.10 V	Revolutions*: 6818 rpm	Revolutions*: 6204 rpm	Revolutions*: 4700 rpm	Thrust-Weight: 1.3 : 1	add. Payload: 259 g
Energy: 44.4 Wh	electric Power: 75.0 W	electric Power: 114.8 W	Throttle (log): 70 %	Current @ Hover: 27.11 A	9.1 oz
Total Capacity: 4000 mAh	mech. Power: 61.9 W	mech. Power: 92.4 W	Throttle (linear): 74 %	P(in) @ Hover: 301.0 W	max Tilt: 29 °
Used Capacity: 3400 mAh	Efficiency: 82.5 %	Power-Weight: 255.2 W/kg	electric Power: 71.8 W	P(out) @ Hover: 226.6 W	max. Speed: 16 km/h
min. Flight Time: 4.6 min		Power-Weight: 115.8 W/lb	mech. Power: 56.7 W	Efficiency @ Hover: 75.3 %	9.9 mph
Mixed Flight Time: 6.5 min		Efficiency: 80.4 %	Power-Weight: 167.2 W/kg	Current @ max: 44.78 A	est. rate of climb: 1.7 m/s
Hover Flight Time: 7.5 min		est. Temperature: 34 °C	Power-Weight: 75.8 W/lb	P(in) @ max: 497.0 W	335 f/min
Weight: 333 g		93 °F	Efficiency: 78.9 %	P(out) @ max: 369.4 W	with Rotor fail:
11.7 oz			est. Temperature: 31 °C	Efficiency @ max: 74.3 %	
			88 °F		
			specific Thrust: 6.27 g/W		
			0.22 oz/W		

(b) Output data

Figure 2.3: Estimated performance results

Figure 2.3 resumes the results of a flight simulation performed with the best obtained set of components in various critical flight conditions (such as hovering condition and motors at maximum thrust). According to the obtained results, requirements have been satisfied.

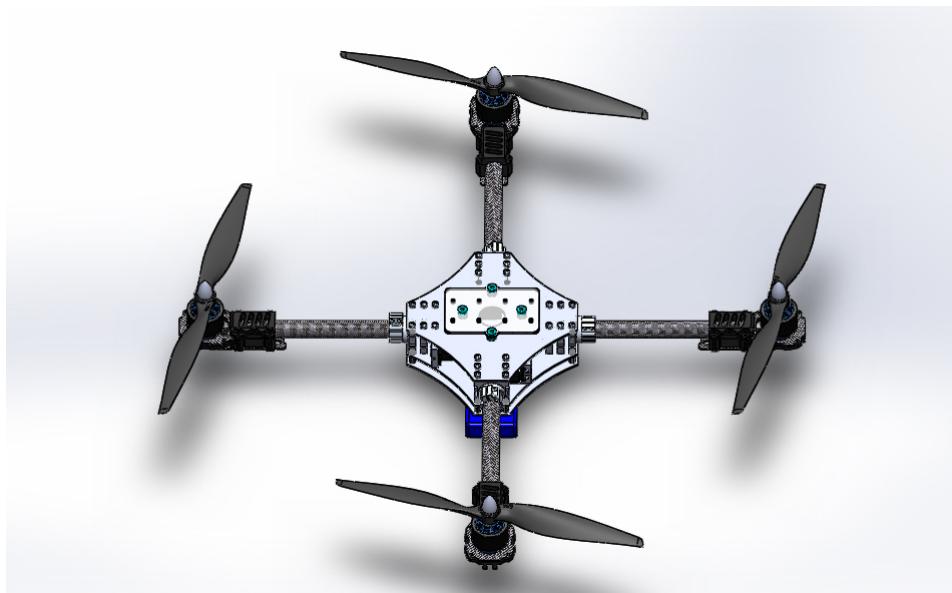


Figure 2.4: 3D prototype of the tiltrotor quadcopter



Figure 2.5: Real tiltrotor quadcopter

## 2.4 Components

### 2.4.1 Servomotors

The tilting mechanism is actuated by a servomotor. Hobby servos are a popular and inexpensive method of motion control, indeed they come off-the-shelf with an integrated control system that allows to directly control the angular position instead of the velocity of the motor. A servomotor

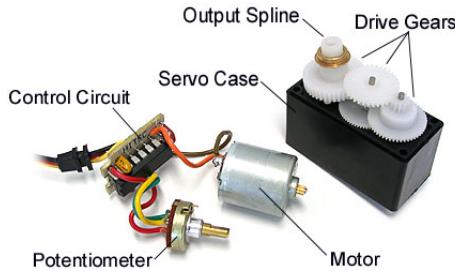


Figure 2.6: Servomotor breakdown

contains the following components (Figure 2.6):

- Control circuit: It receives input/output signals and elaborates a proper control action that is feed to the DC motor.
- Motor: is a simple DC motor that receives the control action from the controller.
- Potentiometer: is used to sense the motor position.
- Drive gears: are in charge to provide a certain amount of torque to the output spline.

Servos are controlled by sending them a pulse of variable width (PWM). The device has three input wires: two of them are for power supply and the last one is used to send the control signal. The pulse has a minimum pulse width, a maximum pulse width, and a repetition rate (Figure 2.7). Each servo has its own specifications related to the minimum and maximum pulse width needed to control the angular position of the output shaft. Given the rotation constraints of the servo, neutral is defined to be the position where the servo has exactly the same amount of potential rotation in the clockwise direction as it does in the counter-clockwise direction. Usually the neutral position is always reached with a pulse of 1.5 milliseconds. The reference angle is determined by the duration of the pulse that is applied to

Parameter	Value
Control System	+Pulse Width Control 1500usec Neutral
Operating Voltage	4.8-6.0 Volts
Operating temperature range	-20 to +60 ° C
Operating Speed (4.8V)	0.22sec/60 deg at no load
Operating Speed (6.0V)	0.18sec/60 deg at no load
Current Drain (4.8V)	8mA/idle and 150mA no load operating
Current Drain (6.0V)	8.8mA/idle and 180mA no load operating
Motor Type	3 Pole Ferrite Motor
Bearing Type	Top Ball Bearing, Lower Bushing
Gear Type	Karbonite Gears
Weight	45g

Table 2.1: Parameters of the HS-485HB servomotor

the control wire, *i.e.*, the duration of the pulse will determine how far the motor will turn. When a pulse that is less than 1.5ms is sent to a servo, the servo rotates its output shaft for an equivalent number of degrees counter-clockwise and holds the given position. When the pulse is wider than 1.5 ms the opposite occurs. For this project the HS-485HB servomotor model

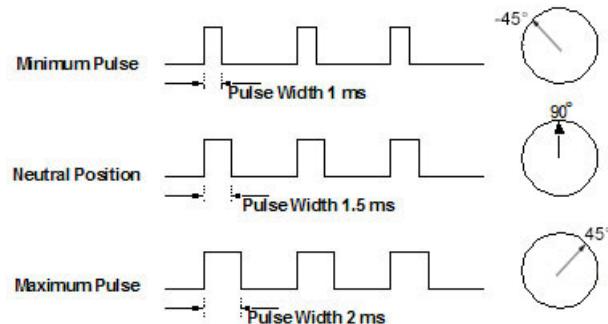


Figure 2.7: Pulse command

has been chosen, in Table 2.1 are reported the main parameters. The operational range of this servo is limited between  $\pm 45$  deg, but for the aim of the actuation needed by the tilting arms this range is sufficient (Figure 2.8). Since the datasheets do not provide any information about the control structure adopted in the control unit, to obtain an accurate model of the servomotor, the identification process will also include an estimation of the controller structure as it will be seen in Chapter 3.3.

#### 2.4.2 Flight Control Unit

The tiltrotor quadcopter is controlled and stabilized through a Flight Control Unit (FCU). The hardware architecture adopted in this thesis relies on R2P (Rapid Robot Prototyping) [2] boards, as previously done in [14] and [10]. R2P is a framework that allows to implement real-time, high-quality architectures for robotic systems with off-the-shelf basic modules (*e.g.*, sensors, actuators, and controllers), integrating hardware and software, which can be assembled in a plug-and-play way. The middleware provided within this framework integrates also a realtime publish/subscribe communication protocol that allows distributed control loops to be set up in a flexible way. The advantages of using these electronic boards is the modularity of the architecture instead of a classical monolithic one, because for example one can plug-in additional modules 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 proximity module provides altitude estimation and control, the USB module provides serial communication with a computer or other serial devices and a SD slot to expand the memory, and at last the RC module allows the PWM communication with the motors controllers. The architecture of the whole system can be seen in the diagram depicted in Figure 2.9.

#### 2.4.3 Motors and ESCs

The chosen motors are BrushLess DC ones (BLDC), from the RCTimer High Performance Series (HP2814). These motors are considered outrunners since the rotating part is on the outside and not in the inside. This setup allows the motors to generate higher torque. This high torque to weight ratio results in a high efficient factor which requires less battery weight and more flight time. The main parameter that describes their behavior is the Kv parameter. This coefficient 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

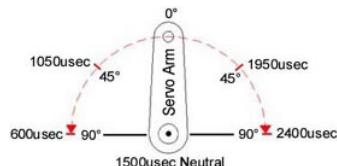


Figure 2.8: Servomotor range of motion

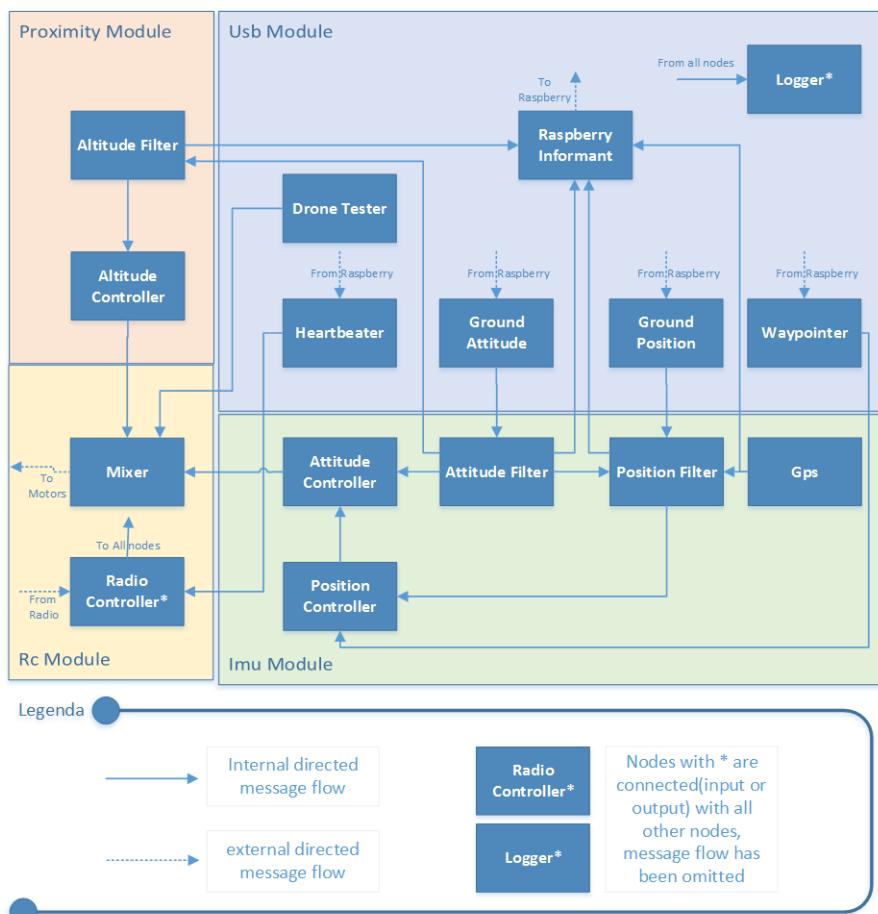


Figure 2.9: Flight controller architecture



Figure 2.10: RCTimer HP2814

7881 rpm). Since this kind of motors is controlled through a three phase modulated power supply, the use of an Electronic Speed Controller (ESC) is required. Brushless ESC systems basically create a tri-phase AC power output of limited voltage from a DC power input: the speed of the motor is controlled by sending a sequence of AC signals generated by varying the switching frequency of the Field Effect Transistors (FET) network (Figure 2.11). Standard ESCs generally accept a nominal 50 Hz PWM input sig-

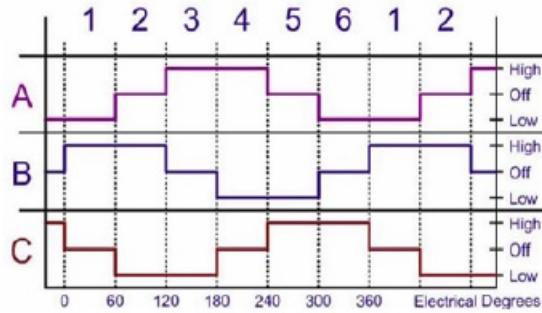


Figure 2.11: Three phase modulated signal

nal whose pulse width varies from 1 ms to 2 ms, but when dealing with multirotors, due to their fast dynamics, a faster firmware that enables the ESC to run up to 400 Hz is usually adopted, it is called SimonK. For this reason the adopted ESCs for the tiltrotor quadcopter are the RCTimer NFS ESC 30A, that come already flashed with SimonK firmware. This choice will allow in future works to raise the looptime of the FCU without doing big modifications to the hardware setup.

#### 2.4.4 Propellers

Propellers are in charge to generate the aerodynamic thrust. When dealing with the choice of the motor/propeller combination, a basic guideline is that the overall generated thrust should be twice the flying weight of the craft. Following this basic rule of thumb will allow the motors to have enough extra thrust to control the vehicle in wind and during aggressive flight maneuvers. For a gentle and smooth flying UAV as in the case of this thesis, the weight to thrust ratio can also be increased such that the craft hovers at around 70% of thrust. Propellers are classified by length, number of blades and pitch. This last parameter represents the travelling distance a propeller would be pulled forward through a solid in a single full

revolution, as depicted in Figure 2.12. Propellers choice is strongly related to the motor power and to the frame size that act as a constraint for the propeller's diameter. Flight efficiency is closely related to the amount of

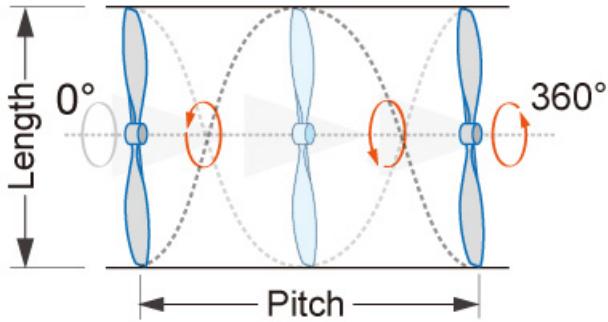


Figure 2.12: Propeller's parameters

air that is in contact with the surface of the propeller. For this reason, a larger propeller diameter will increase the contact surface between air and the propeller and thus also efficiency will increase. In our case, thanks to the simulations developed with eCalc.ch, a good trade-off has been found to be with 12" propellers with 4.5" pitch as the one in Figure 2.13. It



Figure 2.13: Propellers

is important to remark that since on a quadrotor helicopter motors are spinning in opposite directions to balance the generated momentum, two counterwise and two counter-clockwise propellers are needed.

### 2.4.5 Battery

The choice of the battery is related to two main factors: weight and current drawing. Quadcopters typically use Lithium Polymer (LiPo) batteries since compared to other kinds of batteries they have a good weight to power ratio. LiPo batteries have two main parameters: capacity and C rating. The capacity express the amount of energy (in mAh) that can be stored inside battery. The C rating describes the rate at which power can be drawn from the battery (*e.g.*, 1000mAh of capacity with 25C constant and 50C burst means the battery can supply 25A constant and 50A 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 and, according to eCalc, the maximum requested current from motors at full throttle should be around 50A. The chosen LiPo is a Turnigy nano-tech 4000mAh 3S 25-50C (three cells in series). Each cell is 3.7 Volts, so this battery is rated at 11.1 Volts. As can be seen from its specifications, this battery is able to fully satisfy the 50A required by the motors at full throttle without any effort.



# Chapter 3

## Actuators characterization

### 3.1 Aerodynamic forces and moments

The aerodynamic forces and moments generated by the rotors are derived using a combination of momentum and blade element theory, here reported in pills. For an easier reading of the equations below, Table 3.1 summarises all the symbols that will be used for the definition of the aerodynamic forces.

A propeller creates a thrust force out of the rotor supplied power. The

Parameters	
$\rho$	air density
$R$	propeller radius
$A$	disk area
$v_{tip}$	blade tip speed
$\Omega$	rotational speed

Table 3.1: Parameters meaning

magnitude of this force is not constant for a given propeller, but depends on the velocity of the incoming air and the rotational velocity of the propeller itself. In order to make a comparison of propellers of different size easier, aerodynamicists try to get rid of the units. With this simplification, it is possible to use the results of a small scale wind tunnel model to predict the performance of a full scale propeller. For this reason the performance of the rotor/propeller group is usually described by dimensionless (normalized) coefficients that are, the thrust coefficient, the torque coefficient and the power coefficient. Those coefficients are defined as follows:

- Thrust coefficient  $C_T$ :

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

- Rotor shaft torque coefficient  $C_P$ :

$$C_Q = \frac{Q}{\rho A v_{tip}^2 R} = \frac{Q}{\rho A (\Omega R)^2 R}.$$

- Power coefficient  $C_P$ :

$$C_P = \frac{P}{A(\Omega R)^3},$$

where T represents the thrust force generated by the rotor, Q is the torque and P is the power. In the rotorcraft literature those coefficients can be found referred to the solidity ratio

$$\sigma = \frac{A_b}{A},$$

where  $A_b$  represents the effective area of the propeller and not the one of the rotor disk. The numerical values of the presented coefficients will be presented in the following section.

### 3.2 Rotor actuators characterization

In this section the experimental results obtained in [14] to identify the aerodynamic parameters of the same propeller/motor combination used in this thesis will be presented. The identification of the  $C_T$ ,  $C_Q$  and  $C_P$  parameters are useful to obtain an analytical expression of the thrust force and torque generated by the rotor. In a simplified way, those relations can be expressed as

$$\begin{aligned} T &= K_T \Omega^2 \\ Q &= K_Q \Omega^2, \end{aligned}$$

where

$$\begin{aligned} K_T &= C_T \rho A R^2 \\ K_Q &= C_Q \rho A R^3. \end{aligned}$$

As can be seen, from a direct measurement of the thrust force, it is possible to evaluate the aerodynamic parameters through a proper minimization problem procedure (all the details are reported in [14]). The study also identified the dynamic response of the rotor. Here are reported the obtained results:

$$\begin{aligned}\hat{C}_T &= 1.186 \times 10^{-2} \\ \hat{C}_Q &= 9.132 \times 10^{-4} \\ \hat{C}_P &= 9.132 \times 10^{-4}.\end{aligned}$$

The static relation between throttle and rotational speed can be expressed as

$$\begin{cases} \Omega = \hat{m}TH\% + \hat{q} \\ \hat{m} = 6.031 \quad \hat{q} = 80.49. \end{cases}$$

Finally the dynamical model that describes the motor can be modellized as

$$G(s) = \frac{\Omega(s)}{Th(s)} = \frac{\hat{\mu}}{1 + s\hat{\tau}} \quad (3.1)$$

with

$$\hat{\mu} = 5.2 \quad \hat{\tau} = 55 \times 10^{-3}.$$

### 3.3 Tilting actuators characterization

In this section the mathematical modelling of the servomotor will be derived as in [26].

#### 3.3.1 Servomotor model definition

Figure 3.1 shows the block diagram of a servomotor under the assumptions that all blocks are linear systems.  $K(s)$  is the regulator transfer function while  $P(s)$  represents the DC motor model, both expressed in the Laplace domain ( $s$  indicates the differential operator  $s = \frac{d}{dt}$ ). The reference signal  $u(t)$  represents the desired angle while the output signal  $\phi(t)$  represents the output of the system, that is, the rotation angle of the DC motor. In order to derive the DC motor transfer function, the electro-mechanical model of the motor is analyzed. In Figure 3.2,  $J$  represents the moment of inertia of the load attached to the motor drive shaft. In the tiltrotor case this load will be the arm with the propeller group. Starting from the Kirchoff's laws we have that

$$Ri(t) + L\dot{i}(t) + e(t) = v(t) \quad (3.2)$$

where  $e(t)$  is the electromotive force, that can be considered proportional to the speed of the motor

$$e(t) = k_e \dot{\phi}(t). \quad (3.3)$$

From the mechanical point of view it is possible to express the torque balance as

$$\tau(t) = J \ddot{\phi}(t) + D_\phi \dot{\phi}(t). \quad (3.4)$$

Recalling that the torque generated by a DC motor is proportional to the current  $i(t)$  we have that

$$\begin{cases} \tau(t) = k_\tau i(t) \\ \tau(t) = J \ddot{\phi}(t) + D_\phi \dot{\phi}(t). \end{cases} \quad (3.5)$$

Therefore, it is possible to derive the transfer function from the input voltage  $v(t)$  to the torque  $\tau(t)$  from equations (3.2), (3.3) and (3.5). The obtained transfer function is

$$\frac{\tau(t)}{v(t)} = \frac{k_\tau (Js + D_\phi)}{(R + Ls)(Js + D_\phi) + k_e k_\tau}, \quad (3.6)$$

where  $R [\Omega]$  is the armature resistance,  $L [H]$  is the inductance,  $k_e [Vs/rad]$  is the electromotive force coefficient,  $k_t [Nm/A]$  is the torque coefficient and finally  $D_\phi [Ns/m]$  is the viscous friction coefficient. The transfer function

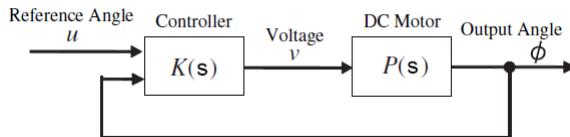


Figure 3.1: Servomotor block diagram

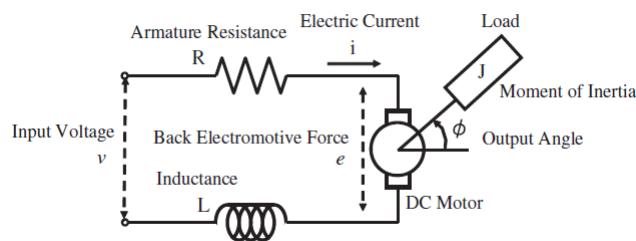


Figure 3.2: Electro-mechanical scheme of a DC motor

from the input torque  $\tau(t)$  to the rotation angle  $\phi(t)$  can be derived from equation (3.5) as follows

$$\frac{\phi(t)}{\tau(t)} = \frac{1}{Js^2 + D_\phi s}. \quad (3.7)$$

Finally the overall transfer function from the input voltage  $v(t)$  to the output rotation angle  $\phi(t)$  can be computed with equations (3.6) and (3.7)

$$P(s) = \frac{k_\tau}{((R + Ls)(Js + D_\phi) + k_e k_\tau)s}. \quad (3.8)$$

This transfer function represents the overall model of the DC motor. Since for most commercial R/C servomotors manufacturers do not provide any information about the control architecture adopted, assumptions on the controller structure must be made, verified and validated later on. Candidates of the possible controller structure are

- PID controller  $K(s) = \frac{K_d s^2 + K_p s + K_i}{s}$ .
- PI controller  $K(s) = \frac{K_i s + K_p}{s}$ .
- PD controller  $K(s) = K_d s + K_p$ .
- DP controller, where the derivative action is placed on the controlled variable  $\phi(t)$  instead of the control error  $e(t)$ , to reduce impulsive outputs that could damage the actuators when the system is subjected to step references. In this case the control variable  $v(t)$  becomes  $v(t) = K_p(u(t) - \phi(t)) - K_d s \phi(t)$ .
- P controller  $K(s) = K_p$ .

In the considered structures  $K_p$ ,  $K_i$  and  $K_d$  are scalar controller gains. As further guess, since the DC motor model in equation (3.8) has an intrinsic integrator, steady state error will be already guaranteed to be zero, and thus, the controller will more probably be a PD, DP or simply P one. Depending on the adopted controller, the order of the whole closed loop system that represents the RC servomotor is affected. Table 3.2 summarises all the possibilities that could arise depending on the chosen controller structure. As will be seen in the following sections, thanks to the Simplified Refined Instrumental Variable method for Continuous-time model identification (SRIVC, [16][13]), a methodology to obtain a proper guess of the relative degree of the servomotor closed loop system with relative parameter estimation will be presented.

Controller	numerator	denominator
PID	2	4
PI	1	4
PD	1	3
DP	0	3
P	0	3

Table 3.2: Order of the closed loop system

### 3.3.2 SRIVC system identification method

The Simple Refined Instrumental Variable method for Continuous-time systems (SRIVC) provided in the CONtinuous-Time System IDentification (CONTSID) toolbox for MATLAB, is a direct time-domain black-box approach that has some notable advantages compared to traditional discrete-time model identification methods, which make it more useful in many practical applications. The one that stand out the most is that, with a direct black-box identification approach, a Continuous Time (CT) model is obtained immediately without the need of transforming Discrete Time (DT) models into CT models as in more classical indirect approaches. Moreover recent studies, [13], have proved that such direct CT methods may perform better than classical DT when dealing with modelling problems, since DT can not perform well without special data pre-treatments. In the next paragraph, the SRIVC method for SISO systems will be briefly explained.

#### Problem formulation

Consider a linear, single-input, single-output, CT system whose input  $u(t)$  and output  $y(t)$  can be described by

$$x(t) = \frac{B(s)}{A(s)} u(t) = G(s)u(t) \quad (3.9)$$

$$y(t) = G(s)u(t) + e(t), \quad (3.10)$$

where the measurement output  $y(t)$  is observed under the influence of the additive measurement noise  $e(t)$ ,  $s$  is the Laplace differentiation operator and

$$B(s) = b_0 s^m + b_1 s^{m-1} + \cdots + b_m \quad (3.11)$$

$$A(s) = s^n + a_1 s^{n-1} + \cdots + a_n. \quad (3.12)$$

Moreover the coefficient of  $s^n$  is set to 1 to determine all the coefficients uniquely. In the following, it is assumed that the algorithm estimates

$\theta = [a_{n-1} \dots a_0 \ b_m \dots b_0]$  from the input and output data  $\{u(t) \ y(t)\}$  observed over  $[0, T]$ . In practice since only the sampled I/O data is available, assumptions about the intersample behaviour of the signals must be done. In this case will be adopted the zero-order hold (ZOH) intersample, so that the signal will be considered constant between two samples and thus the identification problem will rely on the sampled input output data  $Z^N = \{u(t_k); y(t_k)\}_{k=1}^N$ , where  $t_k$  is the sampling interval  $t_k = kT_s$  with  $T_s$  being the sampling time. To explain the SRIVC method, the non-iterative version of the most common IV method will be firstly presented and secondly the method will be extended to the iterative case. The reader is referred, for instance, to [12][13] for a deeper understanding of the traditional state-variable filter (SVF) method on which the SRIVC method is based. In a nutshell, the SRIVC employs an iterative procedure in which, at each iteration, an auxiliary model is used to generate the instrumental variables and prefilter based on the parameters obtained at the previous iteration. The filtering operation is a distinguishing feature of the optimal IV solution. This feature not only guarantees minimum variance estimates, but will also provide a very convenient way of generating the prefiltered time-derivatives of the input and output variables as required for CT model estimation. The SRIVC algorithm is based on the assumption that the additive noise  $e(t)$  in equation (3.10) is a zero mean, normally distributed, white noise process, *i.e.*,  $e(t) = N(0, \sigma^2)$ . In this situation, following the usual Prediction Error Minimization (PEM) approach (which is Maximum Likelihood (ML) estimation in the present situation because of the Gaussian assumption on  $e(t)$ ), a suitable error function  $\varepsilon(t)$  in continuous-time is given by the output error (OE),

$$\varepsilon(t) = y(t) - G(s)u(t), \quad (3.13)$$

which can be rewritten as

$$\varepsilon(t) = \frac{1}{A(s)} [A(s)y(t) - B(s)u(t)]. \quad (3.14)$$

The basis for a stochastic estimation are provided by the minimization of a least squares criterion function in  $\varepsilon(t_k)$ , measured at the sampling instants  $t_k$ . However, since the polynomial operators commute in this linear case, the filter  $F(s) = 1/A(s)$  can be taken inside the brackets,

$$\varepsilon(t) = A(s)y_f(t) - B(s)u_f(t) \quad (3.15)$$

or,

$$\varepsilon(t) = y_f^{(n)}(t) + a_1y_f^{(n-1)}(t) + \dots + a_ny_f^{(0)}(t) - b_0u_f^{(m)}(t) - \dots - b_mu_f^{(0)}(t) \quad (3.16)$$

where

$$\begin{cases} y_f^{(i)}(t) = F_i(s)y(t), & i = 0, \dots, n \\ u_f^{(i)}(t) = F_i(s)u(t), & i = 0, \dots, m. \end{cases} \quad (3.17)$$

The filter  $F(s)$  is defined as

$$F_i(s) = \frac{s^i}{A(s)}. \quad (3.18)$$

The associated estimation model can be finally written for  $t = t_k$  in a linear regression form:

$$y_f^{(n)}(t_k) = \phi_f^T(t_k)\theta + \varepsilon(t_k), \quad (3.19)$$

where  $\phi_f^T(t_k)$  and  $\theta$  are defined as

$$\phi_f^T(t_k) = [-y_f^{(n-1)}(t_k) \dots -y_f^{(0)}(t_k) \ u_f^{(m)}(t_k) \dots u_f^{(0)}(t_k)] \quad (3.20)$$

$$\theta = [a_1 \dots a_n \ b_0 \dots b_n]^T. \quad (3.21)$$

Equation (3.19) forms a basis for the definition of a likelihood function and ML estimation. The parameter estimate  $\hat{\theta}$  can be calculated as

$$\hat{\theta} = \left[ \sum_{k=0}^N \phi_f(t_k) \phi_f^T(t_k) \right]^{-1} \sum_{k=0}^N \phi_f(t_k) y_f^{(n)}(t_k). \quad (3.22)$$

It can be seen that the non-iterative algorithm has two main drawbacks. The first one is related to the need to have some *a priori* informations about  $A(s)$  in order to obtain a good estimation. The second one is that, in practical applications, there is no guarantee that  $e(t)$  is always a white noise. Both of these problems are solved by employing the iterative approach adopted by the SRIVC algorithm. The algorithm employs a similar approach as the one used in the Refined Instrumental Variable (RIV) algorithm for DT systems, that briefly consists in iteratively adjusting the unknown parameters of  $A(s, \hat{\theta}^0)$ , as well as the estimate of the instrumental variable, until they converge. The instrumental variable at each iteration of the algorithm is generated by the following auxiliary model

$$\hat{x}(t) = \frac{B(s, \hat{\theta}_N^{IV})}{A(s, \hat{\theta}_N^{IV})}, \quad (3.23)$$

where  $\hat{\theta}_N^{IV}$  is the estimated parameter vector obtained at the previous iteration of the algorithm according to equation (3.22).

### Model structure identification

Since for the identification of the RC servomotor closed-loop system there is no information related to the controller structure, a method that provides a feedback of the goodness of the structure guess can be useful. A natural way to find the most appropriate model order is to compare the identification results obtained from different model orders. The SRIVC toolbox comes in handy with an associated algorithm that automatically provides statistical measures about the different model orders, based on the number of parameters that must be estimated, so that, statistically speaking, the best choice can be made. The first coefficient adopted is the coefficient of determination  $R_T^2$ , defined as

$$R_T^2 = 1 - \frac{\sigma_e^2}{\sigma_y^2}, \quad (3.24)$$

where  $\sigma_e^2$  is the variance of the estimated noise  $\hat{e}(t_k)$  and  $\sigma_y^2$  is the variance of the measured output  $y(t_k)$ . The  $R_T^2$  provides a normalised measure of how much of the output variance is explained by the deterministic system part of the estimated model, the more  $R_T^2$  is closer to 1 the better the identification. The second coefficient adopted is the Young Information Criterion (YIC), defined as

$$YIC = \ln \frac{\hat{\sigma}^2}{\hat{\sigma}^2} + \ln \left( \frac{1}{n_\theta} \sum_{i=1}^{n_\theta} \frac{\sigma^2 \hat{p}_{ii}}{\hat{\theta}_i^2} \right), \quad (3.25)$$

where  $n_\sigma$  is the number of estimated parameters,  $\hat{p}_{ii}$  is the  $i$ th diagonal element of the block-diagonal SRIVC covariance matrix,  $n_\theta$  is the number of parameters to be estimated,  $\hat{\theta}_i^2$  is the square of the  $i$ th parameter estimate, and the  $\hat{p}_{ii}/\hat{\theta}_i^2$  will represent the normalised measure of uncertainty on the  $i$ th parameter estimate. YIC value provide an information about how well parameters are defined statistically: the more negative is the YIC, the better the definition.

#### 3.3.3 Experimental identification

Thanks to the introduced identification method, in this section the experimental procedure followed to obtain the identification results will be now described.

#### Experimental setup

The experiment consists in applying as input reference to the servomotor a series of step functions with fixed width of 1 [s] and four different amplitudes in order to excite the whole range of motion of the servomotor (at

least the one of interest in this thesis). The reference signal has been chosen as suggested in [26], since other common reference signals such as mixed-frequency sinusoids and M-series have provided unsuccessful results. The measured output needed for the identification campaign consists in the tilting angle  $\alpha_i$ , measured with the rotor group in a hovering condition. In order to reduce load effects of the measuring instrument, a contact-less measure has been performed through the Optitrack system, a 6DoF precision tracking system based on a Infra-Red (IR) array of cameras with incorporate IR flood lights. The cameras are disposed at calibrated positions and orientations such that the measurement subject is always in the Field Of View (FOV) of multiple cameras. Measured data has a nominal sampling rate of 100Hz. Reference commands are generated and sent from a MATLAB

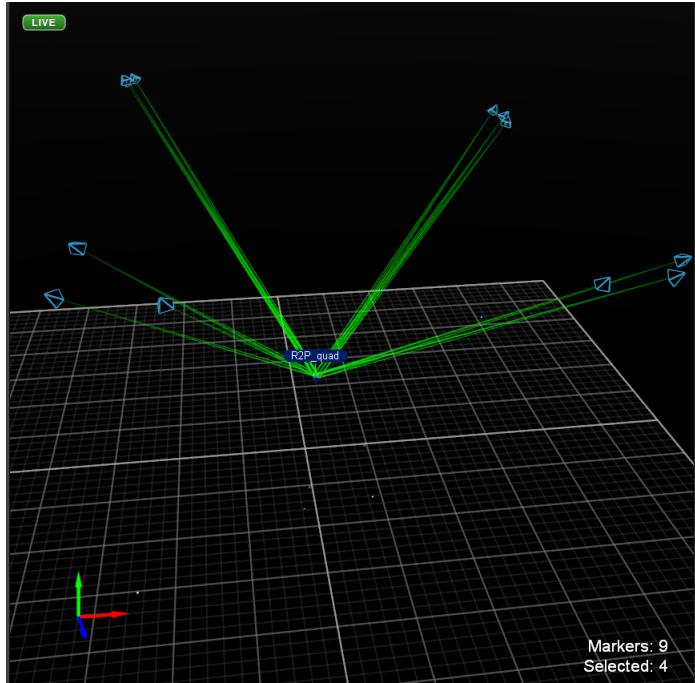


Figure 3.3: Optitrack system

script to the tiltrotor via Bluetooth communication, while measured outputs are automatically logged on the FC. Optitrack data are directly received, elaborated and transmitted to the FC by the RaspberryPI board mounted on the tiltrotor.

Test	$n_b$	$n_f$	$R_T^2$	YIC
1	1	3	0.98081	-10.552
2	1	3	0.98422	-11.199
3	1	3	0.96265	-8.7163
4	1	3	0.97711	-9.7242
5	1	3	0.98457	-11.121
6	1	3	0.97217	-9.886
7	1	3	0.96268	-8.7203

Table 3.3: Best estimation performance indexes

### Identification campaign results

The accomplished identification campaign consisted in 10 experimental tests like the one in Figure 3.4, of which only 7 have been finally used due to data inconsistency of 3 datasets. For each of the 7 dataset a model es-

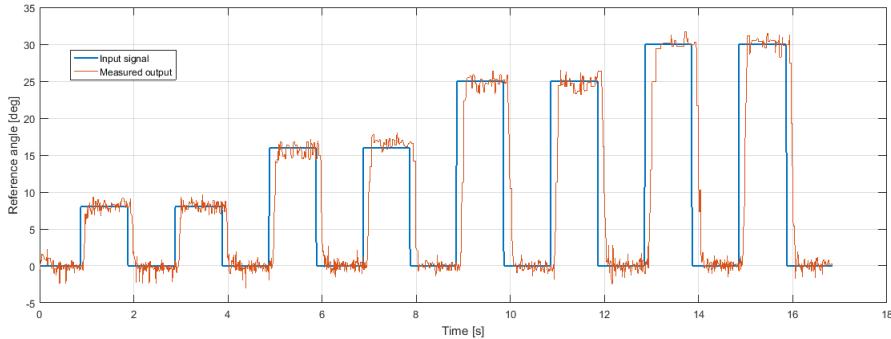


Figure 3.4: Reference signal and measured output angle

timation based on the most negative YIC and highest associated  $R_T^2$  has been performed. Results resumed in Table 3.3 show that the best model choice is the one with a DP controller, since in every experimental campaign the suggested number of parameters is always  $n_b = 1$  and  $n_f = 3$ , where  $n_b$  represents the number of estimated parameters at the numerator  $B(s)$  and  $n_f$  the number of estimated parameters at the denominator  $F(s)$  (see the model structure defined in equation 3.10). As a result of the above considerations, by recalling equation (3.8) and the DP controller structure  $v(t) = K_p(u(t) - \phi(t)) - K_d s\phi(t)$ , the closed-loop servomotor model structure

can be written as:

$$\frac{\phi(s)}{\phi_{ref}(s)} = \frac{K_\tau k_p}{LJs^3 + (RJ + LD_\phi)s^2 + (RD_\phi + k_e k_\tau + K_d k_\tau)s + K_p k_\tau}. \quad (3.26)$$

The overall block diagram of the servomotor closed-loop system is depicted in Figure 3.5, where  $P(s)$  is the transfer function from  $v(t)$  to  $\phi(t)$  previously defined in Equation (3.6). The SRIVC method provides a model with the

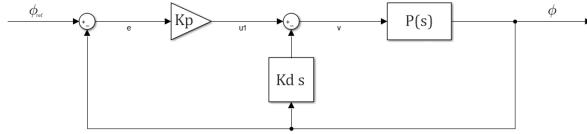


Figure 3.5: Closed loop block diagram

coefficient of  $s^3$  normalized to unity ( $b_1 = 1$ ), so the estimated model can be expressed as

$$\frac{\phi(s)}{\phi_{ref}(s)} = \frac{b_0}{b_1 s^3 + b_2 s^2 + b_3 s + b_4}. \quad (3.27)$$

Since the aim of this identification is to obtain a model that best describes the dynamical behaviour of the servomotor, there is no need to solve the estimated parameters  $b_0, \dots, b_4$  for the electrical values of the servomotor such as the resistance and inductance, so this part will be omitted. To obtain the final parameters, a weighted average and variance has been computed over considering all the 7 tests. The obtained parameters values are reported in Table 3.4. As can be seen from the Bode diagram of the identified model

	Value	$\sigma$
$b_0$	4670.2519	72.11835
$b_2$	28.355992	0.476756
$b_3$	598.45913	7.966064
$b_4$	4650.2325	71.99778

Table 3.4: Servomotor model parameters

depicted in Figure 3.6, the servomotor has a bandwidth of 18 rad/s. To check the validity of the identified model, its response has been evaluated over another set of data that has not been used for the identification procedure. To evaluate the goodness of the identified system the Variance Accounted

For (VAF) has been computed. It is expressed as

$$VAF = \max \left( 1 - \frac{\text{variance}(y - y_{est})}{\text{variance}(y)}, 0 \right) 100. \quad (3.28)$$

Results of the experimental validation procedure are depicted in Figure 3.7: the obtained VAF is 98%.

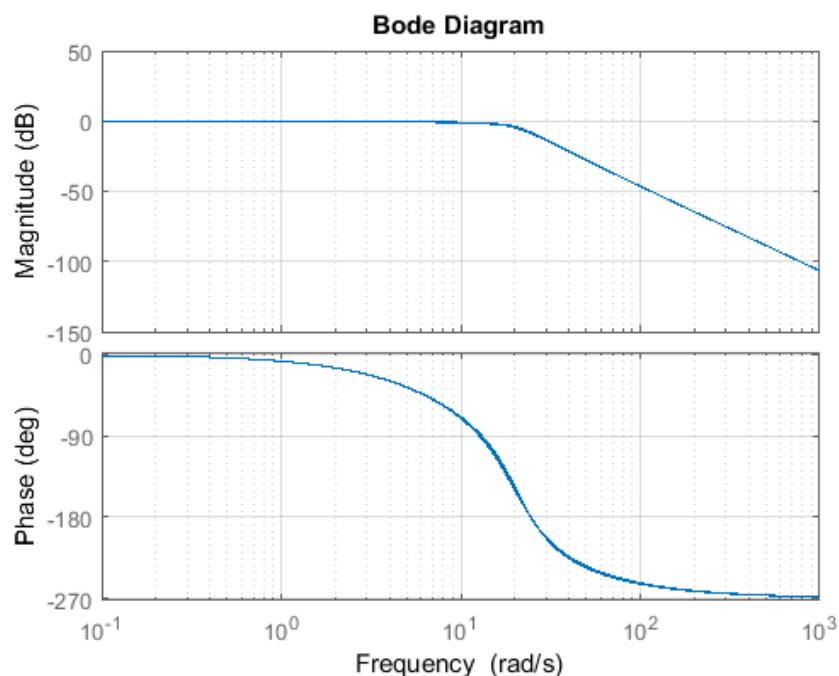


Figure 3.6: Identified model with uncertainty boundaries

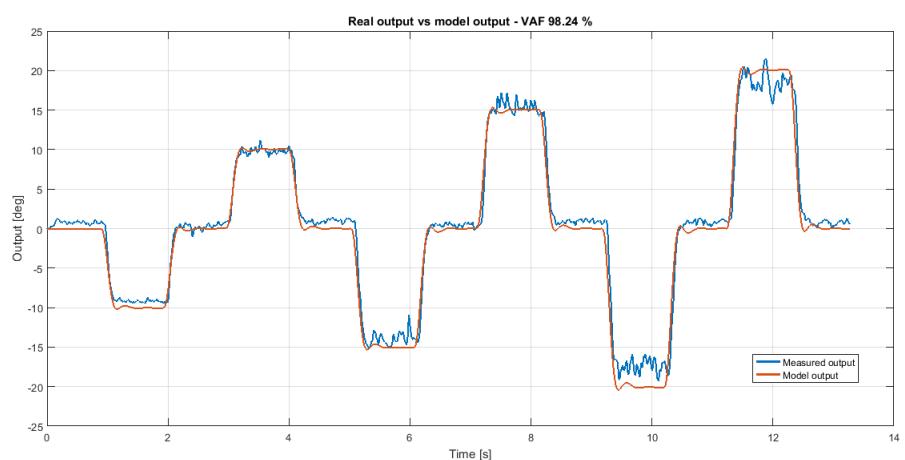


Figure 3.7: Identified model output vs real output comparison

## Chapter 4

# Modelling of the tiltrotor quadcopter

This chapter deals with the problem of characterizing the kinematics and dynamics of the tiltrotor quadcopter. Firstly the reference frames and coordinate systems utilized in this thesis will be presented and a short explanation of the coordinate transformations among different reference frames will be given. Afterward a complete derivation of kinematics and dynamics will be provided. Finally a state space model of the system will be derived based on the obtained equations of motion.

### 4.1 Reference frames

In aviation, guidance, and control of an aircraft or rotorcraft, there are several coordinate systems which are useful to design and analyse dynamics of the vehicle. Usually small scale UAV rotorcraft as the one developed in this thesis are normally utilized at low speeds and in small areas, due to their inherent mechanical design and power limitation. In order to express the kinematic of the vehicle, two main reference frames are defined. The Earth-frame  $\mathcal{F}_E = \{O_E, N, E, D\}$  will denote the fixed inertial frame to which all motion can be referred to, and the body-frame  $\mathcal{F}_B = \{O_B, X_B, Y_B, Z_B\}$  will be the moving frame attached to the center of mass of the vehicle. The Earth-frame is also known as a navigation or ground coordinate system. It is a coordinate frame fixed to the Earth's surface defined with the North-East-Down (NED) convention. According to the convention, the origin of the Earth frame is arbitrarily fixed to a point on the Earth's surface, and for navigation of small scale UAVs, correspond to the take off point since it is also the sensors initialization point. The Earth frame axes are chosen

such that the N-axis points North, E-axis points East and D axis points downward along the ellipsoid normal. With this convention, since D-axis is pointing downward, it is obvious that the actual altitude of the flying vehicle will be defined by a negative number. It is also worth to mention that in autonomous navigation systems there are other references systems that are widely used:

- Geodetic coordinate system: used in GPS-based navigation systems. A coordinate point near the Earth's surface is characterized in terms of longitude, latitude and height (or altitude), which are denoted as  $\lambda$ ,  $\varphi$  and  $h$ .
- ECEF coordinate system: rotates with the Earth around its spin axis. Its origin is located at the center of the Earth, the Z axis points to the north pole, the X axis intersects the sphere of the Earth at  $0^\circ$  latitude and longitude while the Y axis is defined to comply the right-hand rule.

Figure 4.1 shows the introduced coordinate systems. The body frame origin

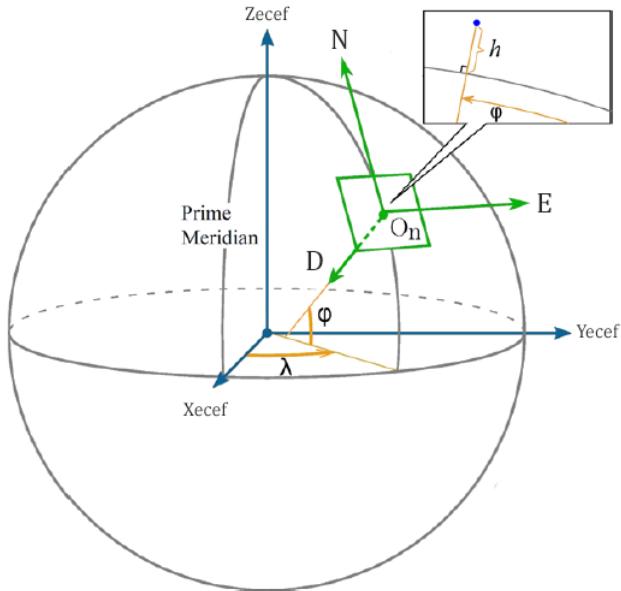


Figure 4.1: Reference frames

and axes are defined as follows:

- The origin is located at the Center of Gravity (CoG) of the flying vehicle.

- The  $X_B$ -axis points forward and usually lies on the plane of symmetry of the flying vehicle.
- The  $Y_B$ -axis points to the right side of the flying vehicle (starboard).
- The  $Z_B$ -axis points downwards to comply the right-hand rule.

To simplify the computation of external forces acting on the quadrotor body, it is also useful to define a propeller group coordinate system  $\mathcal{F}_{P_i} = \{O_{P_i}, X_{P_i}, Y_{P_i}, Z_{P_i}\}$ ,  $i = 1 \dots 4$ , such that the  $X_{P_i}$ -axis of each propeller group points outwards, while the  $Y_{P_i}$ -axis and Z-axis are defined according to the right-hand rule.

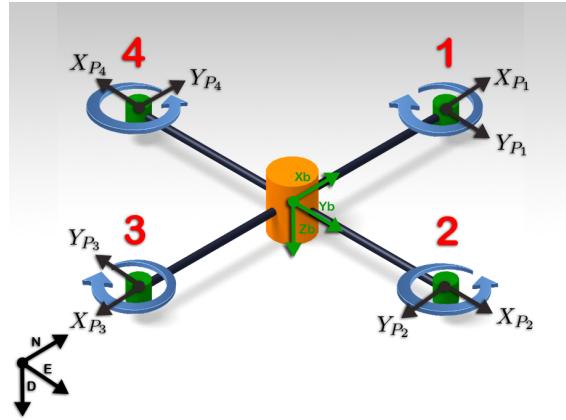


Figure 4.2: Tiltrotor coordinate systems

#### 4.1.1 Coordinate transformations

The transformations among the adopted coordinate frames are introduced in this section.

As it is known, the orientation of one Cartesian coordinate system with respect to another can always be described by three successive Euler rotations. Euler angles provide a way to represent 3D orientation of an object using a combination of three rotations about different axes, *i.e.* Euler rotations. For aerospace applications, the Euler rotations are performed about each of the three Cartesian axes sequentially, following a Z-Y-X (or the so-called 3-2-1) rotation sequence. The angles that arise from those rotations are called Euler angles and in our case will express the orientation of the tiltrotor quadcopter. According to NASA standard notation these three Euler angles are also known as the roll, pitch, and yaw (or heading) angles, which are defined as follows:

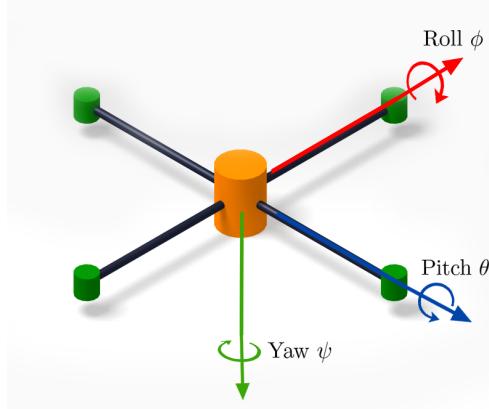


Figure 4.3: Axes of rotation

- Roll angle, denoted with  $\phi$ , is the angle around the  $x$ -axis.
- Pitch angle, denoted with  $\theta$  is the angle around the  $y$ -axis.
- Yaw angle, denoted with  $\psi$  is the angle around the  $z$ -axis.

Thanks to the introduced notation, it is now possible to introduce the concept of rotation matrix.

Starting from a generalized case, a rotation matrix allows to express a vector  $P_E$  in system E in a vector  $P_D$  in system D

$$P_D = R_{DE}P_E. \quad (4.1)$$

In our case, since some state variables of our system are defined in the inertial frame while other quantities are expressed in the local body frame, Euler rotations provide a tool to express the same quantity in different frames. The rotation matrices for each axis are defined as follows

$$R_X(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\phi & -s\phi \\ 0 & s\phi & c\phi \end{bmatrix} \quad R_Y(\theta) = \begin{bmatrix} c\theta & 0 & s\theta \\ 0 & 1 & 0 \\ -s\theta & 0 & c\theta \end{bmatrix} \quad R_Z(\psi) = \begin{bmatrix} c\psi & -s\psi & 0 \\ 0 & 1 & 0 \\ -s\theta & 0 & c\theta \end{bmatrix}, \quad (4.2)$$

where, for the sake of simplicity,  $c\phi = \cos(\phi)$ ,  $s\phi = \sin(\phi)$ .

The overall transformation from the inertial frame to the body frame is given by multiplying each single rotation matrix:

$$R_{BE} = R_X(\phi) R_Y(\theta) R_Z(\psi) \quad (4.3)$$

$$= \begin{bmatrix} c\theta c\psi & c\theta s\psi & -s\theta \\ s\phi s\theta c\psi - c\phi s\psi & s\phi s\theta s\psi + c\phi c\psi & s\phi c\theta \\ c\phi s\theta c\psi + s\phi s\psi & c\phi s\theta s\psi - s\phi c\psi & c\phi c\theta \end{bmatrix}. \quad (4.4)$$

To obtain the inverse transformation we need to recall that, since rotation matrices are orthogonal, their inverses correspond to their transpose:

$$R_X^{-1} = R_X^T$$

$$R_Y^{-1} = R_Y^T$$

$$R_Z^{-1} = R_Z^T$$

and thus the rotation matrix from Body to Earth can be expressed as

$$R_{EB} = R_{BE}^T. \quad (4.5)$$

## 4.2 Kinematics

In this section the kinematics of the tiltrotor quadcopter will be derived based on the previously introduced reference frames. To this aim in Table 4.1 all the state variables used to identify the vehicle's kinematics are reported.

As can be seen, variables related to the position of the tiltrotor quadcopter are given in the Earth frame  $\mathcal{F}_E$  while the angular and linear velocities are defined in the local tiltrotor body frame  $\mathcal{F}_B$ .

This difference is related to the capability of inertial measurement units (IMU), *i.e.*, accelerometers and rate gyros, to produce only measurements in body frame. Those physical quantities can be rotated in different frames

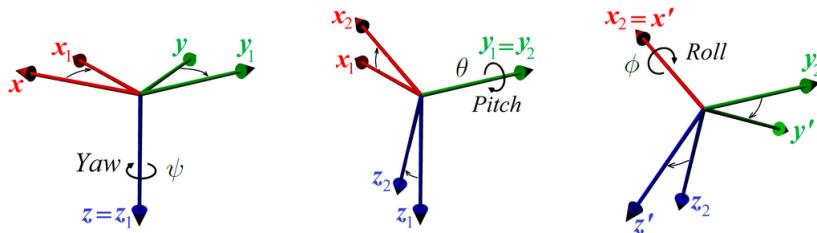


Figure 4.4: 3-2-1 Euler rotations

Symbol	Description	Unit
N	the inertial (north) position of the quadrotor	m
E	the inertial (east) position of the quadrotor	m
D	the altitude of the aircraft	m
p	the roll rate expressed in the body frame	rad/s
q	the pitch rate expressed in body frame	rad/s
r	the yaw rate expressed in body frame	rad/s
u	the body frame velocity measured around local N	m/s
v	the body frame velocity measured around local E	m/s
w	the body frame velocity measured around local D	m/s
$\phi$	Roll angle around $x_b$	rad
$\theta$	Pitch angle around $y_b$	rad
$\psi$	Yaw angle around $z_b$	rad

Table 4.1: State variables of the tiltrotor quadcopter model

throughout Euler rotations as already explained in Section 4.1.

Therefore by defining the following vectors it is possible to express the relationship between position and translational velocities:

$$P_E = \begin{bmatrix} N \\ E \\ D \end{bmatrix} \quad V_E = \frac{d}{dt} P_E = \begin{bmatrix} \dot{N} \\ \dot{E} \\ \dot{D} \end{bmatrix} \quad (4.6)$$

from which

$$V_E = R_{EB} V_b \quad \text{with} \quad V_b = \begin{bmatrix} u \\ v \\ w \end{bmatrix}, \quad (4.7)$$

where  $V_b$  is the linear velocity expressed in body frame.

For rotational kinematics we focus on the angular velocity vector  $\omega_b$ , defined using the standard symbols adopted in the flight dynamics literature

$$\omega_b = \begin{bmatrix} p \\ q \\ r \end{bmatrix}. \quad (4.8)$$

In this case, since the angular rates are defined in the body frame  $\mathcal{F}_b$ , whereas the Euler angles are defined in intermediate frames, the process of deriving a relationship between those quantities becomes more difficult than the one presented for translational kinematics. Following the definition and sequence

of the Euler angles the relationship can be derived as follows.

Let  $\omega_e$  be the vector of Euler's angles time derivatives.

$$\omega_e = \dot{\alpha}_e = \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \quad \text{with} \quad \alpha_e = \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix}. \quad (4.9)$$

Assuming that the Euler rates are small, we have that

$$R_X(\dot{\phi}) = R_Y(\dot{\theta}) = R_Z(\dot{\psi}) = I. \quad (4.10)$$

The vector which expresses the body-axis angular rates is

$$\omega_b = \begin{pmatrix} p \\ q \\ r \end{pmatrix} = \begin{pmatrix} \dot{\phi} \\ 0 \\ 0 \end{pmatrix} + R_X(\phi) \left[ \begin{pmatrix} 0 \\ \dot{\Theta} \\ 0 \end{pmatrix} + R_Y(\theta) \begin{pmatrix} 0 \\ 0 \\ \dot{\psi} \end{pmatrix} \right] \quad (4.11)$$

$$= \begin{pmatrix} 1 & 0 & -\sin(\theta) \\ 0 & \cos(\phi) & \sin(\phi) \cos(\theta) \\ 0 & -\sin(\phi) & \cos(\phi) \cos(\theta) \end{pmatrix} \begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} \quad (4.12)$$

where, according to Euler definitions, the third component of  $\omega_e$  has been rotated by an angle  $\theta$  around the Y axis and about an angle  $\phi$  around the X axis, the second component has been rotated by an angle  $\phi$  around the X axis while the first component has not been rotated since it is already expressed in body frame.

For the sake of notation simplicity (4.12) can be rewritten as

$$\omega_b = S(\phi, \theta) \omega_e.$$

Unlike the  $R_{BE}$  matrix previously presented, the inverse of  $S$  is not its transpose.

It is though simple to verify that

$$S^{-1}(\phi, \theta) = \begin{pmatrix} 1 & \sin(\phi) \tan(\theta) & \cos(\phi) \tan(\theta) \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi)/\cos(\theta) & \cos(\phi)/\cos(\theta) \end{pmatrix} \quad (4.13)$$

and thus

$$\omega_e = S^{-1}(\phi, \theta) \omega_b$$

that is

$$\begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} = \begin{pmatrix} 1 & \sin(\phi) \tan(\theta) & \cos(\phi) \tan(\theta) \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi)/\cos(\theta) & \cos(\phi)/\cos(\theta) \end{pmatrix} \begin{pmatrix} p \\ q \\ r \end{pmatrix}. \quad (4.14)$$

Before going any further, it is useful to make some remarks about the transformation matrix that converts body-frame angular rates to Euler angular rates. Notice that the matrix presents a division by  $\cos(\theta)$  in the last row. When the pitch angle approaches  $+/ - 90$  degrees, the denominator goes to zero and the matrix elements diverge to infinity and thus the orientation can no more be uniquely represented using Euler angles. This particular attitude is called **Gimbal Lock**. The exact orientation at which gimbal lock occurs depends on the order of rotations used. A clear example of a gimbal

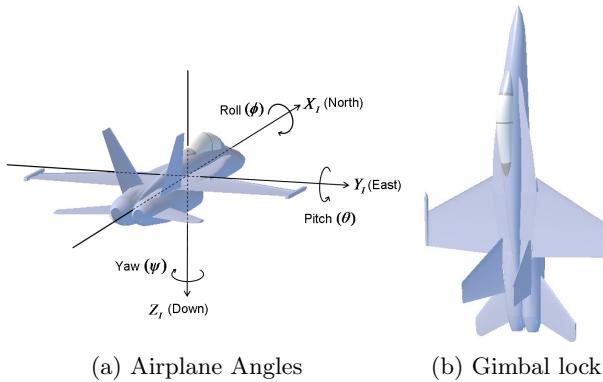


Figure 4.5: Gimbal lock example in airplanes

lock situation is depicted in Figure 4.5. Intuitively, the cause of gimbal lock is that when the pitch angle is 90 degrees, yaw and roll movements become coupled, thus a rotation around the yaw axis or the roll axis will lead to the same airplane movement.

As already mentioned, it is possible to end in this situation by following different rotation sequences: it should be easy to see that the orientation in Figure 4.5.b can be obtained by yawing and then pitching, or by pitching and then rolling.

### 4.3 Quaternions

To avoid gimbal lock, one must choose a different parametrization of the rotation matrix that resolves the mapping between inertial frame and the vehicle body frame that is,  $S(\phi, \theta)$ .

A better way to perform a coordinate transformation is through quaternion representation. Quaternions are widely used as attitude representation parameter of rigid bodies such as spacecraft. In fact quaternions inherently

come along with some advantages such as no singularity and computationally less intense compared to other attitude parameters such as Euler angles or a direction cosine matrix. A quaternion is a four dimensional vector the elements of which consist of a vector part  $\varrho$  and a scalar part  $q_4$

$$q = \begin{bmatrix} \varrho \\ q_4 \end{bmatrix} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix}.$$

Note the scalar part is the last element of the vector and represent the rotation angle, while vector  $\varrho$  represents the  $x-y-z$  components of rotation axis. Moreover the quaternion is also subject to the unit-norm constraint given by  $q^T q = 1$ . The attitude matrix can be written as a homogeneous

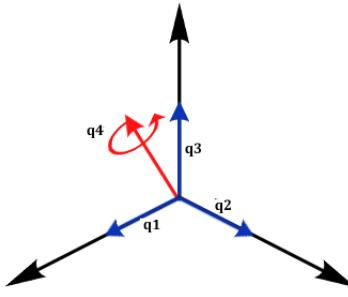


Figure 4.6: Quaternion components

quadratic function of the components of a quaternion

$$A(q) = (q_4^2 - |\varrho|^2) I_3 + 2\varrho\varrho^T - 2q_4 [\varrho \times] \quad (4.15)$$

where  $I_3$  stands for the  $3 \times 3$  identity matrix and  $[\varrho \times]$  correspond to the cross product matrix defined as

$$[\varrho \times] = \begin{bmatrix} 0 & -q_3 & q_2 \\ q_3 & 0 & -q_1 \\ -q_2 & q_1 & 0 \end{bmatrix}. \quad (4.16)$$

For small angles the vector part of the quaternion is approximately equal to half angles so that  $\varrho \approx \alpha_e/2$  and  $q_4 \approx 1$ . Successive rotations can be accomplished using quaternion multiplication (composition) according to

$$A(q')A(q) = A(q' \otimes q) \quad (4.17)$$

where the quaternion composition is defined as

$$q_a \otimes q_b = \begin{bmatrix} q_{b4} q_a + q_{a4} q_b - q_b \times q_a \\ q_{a4} q_{b4} - q_a^T q_b \end{bmatrix}. \quad (4.18)$$

An important remark regarding quaternion operations is that the product is not commutative, thus  $q_a \otimes q_b \neq q_b \otimes q_a$ .

The Euler angles can be obtained from the quaternion form via the relations

$$\begin{aligned} \phi &= \tan^{-1} \left( \frac{A_{2,3}}{A_{3,3}} \right) \\ \theta &= \sin^{-1} (-A_{1,3}) \\ \psi &= \tan^{-1} \left( \frac{A_{1,2}}{A_{1,1}} \right). \end{aligned}$$

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 as

$$q_{AB}^* = q_{BA} = \begin{bmatrix} -q_1 \\ -q_2 \\ -q_3 \\ q_4 \end{bmatrix}.$$

With the quaternion representation the transformation of a vector  $v$  from  $A$  to  $B$  is achieved with the following relationship

$$v_b = q_{AB} \otimes \begin{bmatrix} v_a \\ 0 \end{bmatrix} \otimes q_{AB}^*$$

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 last element to make them 4 element vectors. The attitude kinematics gives the dependency of the time derivative of its relative orientation in space from the angular rate, and in the quaternion representation is expressed as

$$\dot{q} = \frac{1}{2} \Omega(\omega_b) q, \quad (4.19)$$

where  $\omega_b = [p \quad q \quad r]^T$  is the angular velocity vector expressed in body frame and

$$\Omega(\omega) = \begin{bmatrix} -[\omega \times] & \omega \\ -\omega^T & 0 \end{bmatrix}. \quad (4.20)$$

Note that matrix  $\Omega$  depends on the quaternion notation, which may differ in the literature. For a deeper understanding about quaternions see [23] [7].

## 4.4 Coriolis equation

When dealing with rigid bodies in motion, centripetal and Coriolis accelerations must be taken into account. To describe those effects, a short derivation of the equation of Coriolis will be now presented as in [25].

Consider two coordinate frames  $\mathcal{F}_i$  and  $\mathcal{F}_b$ , where  $\mathcal{F}_i$  denotes the inertial frame and  $\mathcal{F}_b$  denotes the body frame of the quadrotor helicopter. Let  $p$  be a vector that is moving in  $\mathcal{F}_b$  while  $\mathcal{F}_b$  is rotating and translating with respect to  $\mathcal{F}_i$ , as in Figure 4.7. Our purpose is to find the time derivative of  $p$  as seen from frame  $\mathcal{F}_i$ . This relationship will be derived through two

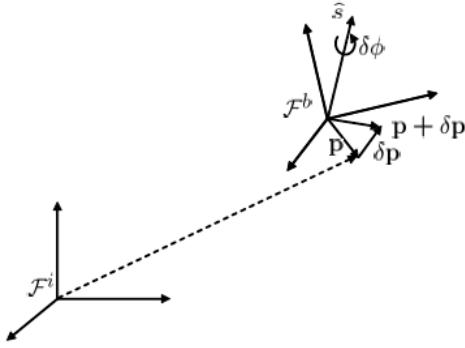


Figure 4.7: Derivation of the equation of Coriolis

steps. Firstly assume that  $\mathcal{F}_b$  is not rotating with respect to  $\mathcal{F}_i$ .

Let  $\frac{d}{dt}p|_i$  and  $\frac{d}{dt}p|_b$  be the time derivatives of vector  $p$  in frame  $\mathcal{F}_i$  and  $\mathcal{F}_b$  respectively.

We thus get

$$\frac{d}{dt}p|_i = \frac{d}{dt}p|_b. \quad (4.21)$$

Secondly, assume that  $p$  is fixed in  $\mathcal{F}_b$  but that  $\mathcal{F}_b$  is rotating with respect to  $\mathcal{F}_i$ , and let  $\hat{s}$  be the instantaneous axis of rotation and  $\delta\phi$  the (right-handed) rotation angle. By using the so called *rotation formula* (the derivation of the formula can be found in [25]) it is possible to redefine the time derivative as follows:

$$p + \delta p = (1 - \cos(-\delta\phi))\hat{s}(\hat{s} \cdot p) + \cos(-\delta\phi)p - \sin(-\delta\phi)\hat{s} \times p. \quad (4.22)$$

Using a small angle approximation and dividing both sides by  $\delta t$  gives

$$\frac{\delta p}{\delta t} \approx \frac{\delta \phi}{\delta t} \times p.$$

Finally, by taking the limit as  $\delta t \rightarrow 0$  and by defining the angular velocity of  $\mathcal{F}_b$  with respect to  $\mathcal{F}_i$  as  $\omega_b \triangleq \hat{s}\dot{\phi}$  we get

$$\frac{d}{dt_i} = \omega_b \times p. \quad (4.23)$$

Since differentiation is a linear operator we can combine equations (4.21) and (4.23) to obtain

$$\frac{d}{dt_i} p = \frac{d}{dt_b} p + \omega_b \times p. \quad (4.24)$$

The obtained relationship (4.24) is usually called *equation of Coriolis* [4] and will be an essential tool in developing equations of motion from Newton's laws.

## 4.5 Equations of motion

In this section the equations of motion of the tiltrotor quadcopter will be derived. To this aim, as already mentioned, the tiltrotor quadcopter is considered as a composition of rigid bodies, *i.e.*, four propellers groups  $P_i$  and the quadrotor body. Since in this flying vehicle a relative motion between the propeller groups and the body due to the tilting action is possible, it is useful to define a rotation matrix  $R_{BP_i}$  that will help in translating physical quantities expressed in the local propeller group frame to the body frame as in [17].

Let

$$R_{BP_i} = R'_Z \left( (i-1) \frac{\pi}{2} \right) R_X(\alpha_i) \quad i = 1, \dots, 4 \quad (4.25)$$

be the rotation matrix from the  $i$ th propeller group to body frame and

$${}^B O_{P_i} = R_Z \left( (i-1) \frac{\pi}{2} \right) \begin{bmatrix} b \\ 0 \\ 0 \end{bmatrix} \quad i = 1, \dots, 4 \quad (4.26)$$

be the origin of the  $i$ th propeller group frame, where  $b$  is the distance between  ${}^B O_{P_i}$  and  $O_B$ . Thanks to the introduced relationships we are now able to express the relative orientation that relates the propeller groups and the body frame.

By exploiting Newton-Euler method, it is possible to derive a complete description of the quadrotor helicopter dynamic model by considering the

forces/momenta generated by the propeller motion, as well as any cross coupling due to gyroscopic and inertial effects arising from the relative motion of the five bodies composing the quadrotor helicopter.

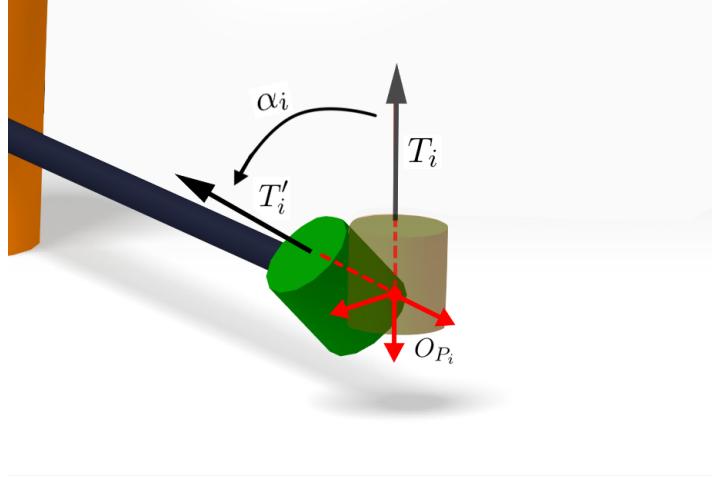


Figure 4.8: *i*-th tilting arm and relative physical quantities

#### 4.5.1 Translational motion

Let  $V_b$  be the velocity vector of the quadrotor helicopter. Newton's laws only hold in inertial frames, therefore Newton's law applied to the translational motion is

$$\frac{d(mV_b)}{dt} = F_{ext} + F_g \quad (4.27)$$

$$V_b = \begin{bmatrix} u \\ v \\ w \end{bmatrix} \quad (4.28)$$

where  $m$  is the mass of the quadrotor and  $F_{ext}$  takes into account the resultant of all external forces applied to the quadrotor, *i.e.*, aerodynamic forces and  $F_g$  is the gravity force expressed in body frame:

$$F_{ext} = \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} \quad F_g = R_{BE} \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} m.$$

Recalling the equation of Coriolis derived in Section 4.4 and by applying it to equation (4.28) we get

$$\frac{d(mV_b)}{dt} = m \left( \frac{dV_b}{dt} + \omega_b \times V_b \right) = F_{ext} + F_g \quad (4.29)$$

$$\omega_b = \begin{bmatrix} p \\ q \\ r \end{bmatrix}.$$

As already stated, we will assume that every quantity is expressed in its local frame, for example,  $\omega_b = {}^B\omega_b$  is the angular velocity of the body and thus will be expressed in the body frame.

Expressed concisely, equation (4.29) becomes

$$m\dot{V}_b + \omega_b \times (mV_b) = F_{ext} + F_g. \quad (4.30)$$

Let now

$$T_{P_i} = \begin{bmatrix} 0 & 0 & -K_t \omega_i^2 \end{bmatrix}^T \quad (4.31)$$

represent the  $i$ -th propeller force (thrust) acting at the origin of the  $i$ -th propeller group  ${}^B O_{P_i}$  (see Section 3.1).

It is now possible to define  $F_{ext}$  as follows

$$F_{ext} = \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} = \sum_{i=1}^4 R_{BP_i} T_{P_i}. \quad (4.32)$$

#### 4.5.2 Angular motion

Due to the tilting propeller groups, things get a little bit more complicated when dealing with rotational motion. To this end let's define some physical quantities that will be used later on. Consider the  $i$ -th propeller group  $P_i$ , its angular velocity is

$$\omega_{P_i} = R_{BP_i}^T \omega_b + [\dot{\alpha}_i \ 0 \ \omega_i]^T,$$

where  $R_{BP_i}^T$  is the rotation matrix from body to  $i$ -th propeller group,  $\dot{\alpha}_i$  is the tilting velocity about  $X_{P_i}$  and  $\omega_i$  is the spinning velocity of the propeller about  $Z_{P_i}$ .

By taking the time derivative of  $\omega_{P_i}$  we get the angular acceleration

$$\dot{\omega}_{P_i} = R_{BP_i}^T \dot{\omega}_b + \dot{R}_{BP_i}^T \omega_b + [\ddot{\alpha}_i \ 0 \ \dot{\omega}_i]^T,$$

where  $\dot{R}_{BP_i}^T$  is the time derivative of a rotation matrix. To compute the time derivative of a rotation matrix some mathematical derivation will be now presented. Let  $R = R(t)$  be a rotation matrix, given as a function of time. Recalling that a rotation matrix is orthogonal (Section 4.1.1) we have that

$$R(t)R^T(t) = I$$

taking the time derivative of both sides yields

$$\dot{R}(t)R^T(t) + R(t)\dot{R}^T(t) = 0 \quad (4.33)$$

thus if we define a new matrix

$$S(t) = \dot{R}(t)R^T(t)$$

it turns out that  $S(t) + S(t)^T = 0$ , which means that  $S(t)$  is a skew-symmetric matrix .

Matrix  $S(t)$  then takes the following form:

$$\omega(t) = \begin{bmatrix} \omega_x(t) \\ \omega_y(t) \\ \omega_z(t) \end{bmatrix} \quad S(\omega(t)) = \begin{bmatrix} 0 & -\omega_z(t) & \omega_y(t) \\ \omega_z(t) & 0 & -\omega_x(t) \\ -\omega_y(t) & \omega_x(t) & 0 \end{bmatrix}. \quad (4.34)$$

Let now  $p(t) = R(t)p'(t)$  be a vector that is rotating over time as in Figure 4.9. Its time derivative is

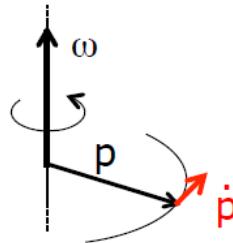


Figure 4.9: Rotating vector

$$\dot{p}(t) = \dot{R}(t)p'(t) = S(t)R(t)p'(t) = S(t)p(t) \quad (4.35)$$

but we can also express the same time derivative as

$$\dot{p}(t) = \omega(t) \times p(t) = S(\omega(t))p(t). \quad (4.36)$$

Thus the skew-symmetric matrix  $S$  can also be interpreted as the operator that computes the cross product.

We can finally conclude that

$$\dot{R}(t) = S(\omega(t))R(t) \iff S(\omega(t)) = \dot{R}(t)R^T(t) \quad (4.37)$$

Coming back to our case, since  $\dot{R}_{BP_i}^T$  represents the rotation matrix from body to  $i$ -th propeller group, the only rotation we must take into account is the tilting action around  $X_{P_i}$  axis. Thus

$$\omega(t) = \begin{bmatrix} \alpha_i(t) \\ 0 \\ 0 \end{bmatrix} \rightarrow S(\omega(t)) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -\dot{\alpha}_i(t) \\ 0 & \dot{\alpha}_i(t) & 0 \end{bmatrix}.$$

To derive the rotational equation of motion, the Euler's rotation equations have been applied. It follows that

$$\tau_{P_i} = I_{P_i} \dot{\omega}_{P_i} + \omega_{P_i} \times I_{P_i} \omega_{P_i} - \tau_{ext_i}, \quad (4.38)$$

where  $\tau_{P_i}$  is a vector that represents the overall torque balance that the propeller group is feeling. This also takes into account the gyroscopic and inertial effects due to the rotation of the propeller group.

$I_{P_i}$  is the (constant) symmetric and positive definite inertia matrix of the  $i$ -th propeller/rotor assembly approximated as an equivalent disc, and  $\tau_{ext_i}$  includes all the external forces applied to the propeller, obtained according to *blade momentum theory* explained in Section 3.1.

$$\tau_{ext_i} = \begin{bmatrix} 0 & 0 & \pm K_q \omega_{P_i Z} \mid \omega_{P_i Z} \mid \end{bmatrix}^T, \quad (4.39)$$

where the sign of the last component is determined by the fact that the torque's direction is always defined opposite to the propeller's rotation, *i.e.*, for a clockwise rotating propeller its torque will be defined counter-clockwise and  $\omega_{P_i Z}$  is the third component of  $\omega_{P_i}$ .

By considering the quadrotor helicopter body  $B$ , the torques generated by the four propellers  $P_i$  and by applying Newton's second law, one then obtains

$$I_n \dot{\omega}_b + \omega_b \times I_n \omega_b = \sum_{i=1}^4 ({}^B O_{P_i} \times R_{BP_i} T_{P_i} - R_{BP_i} \tau_{P_i}), \quad (4.40)$$

where the right-side term is the vector representing the external moments  $L$ ,  $M$  and  $N$  that are applied on the body-axes  $X_b$ ,  $Y_b$  and  $Z_b$  respectively

$$M_{ext} = \begin{bmatrix} L \\ M \\ N \end{bmatrix} = \sum_{i=1}^4 ({}^B O_{P_i} \times R_{BP_i} T_{P_i} - R_{BP_i} \tau_{P_i}) \quad (4.41)$$

and  $I_n$  is the *inertia tensor*

$$I_n = \begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ I_{yx} & I_{yy} & -I_{yx} \\ -I_{zx} & -I_{zy} & I_{zz} \end{bmatrix}, \quad (4.42)$$

where the terms are defined as follows

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

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

$$I_{xy} = I_{yx} = I_{xz} = I_{zx} = I_{zy} = I_{yz} = 0$$

$$I_n = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix}.$$

In equation (4.40) we must also consider the aerodynamic damping  $M_{damp}$  caused by the propellers rotation (the aerodynamic drag caused by the structure of the quadrotor is neglected, because the quadrotor's body is really thin), so equation (4.40) becomes

$$I_n \dot{\omega}_b + \omega_b \times I_n \omega_b = M_{ext} + M_{damp} \quad (4.43)$$

where, assuming that the quadrotor helicopter has to fly in near hover condition, the drag produced by linear translations can be neglected and thus only the aerodynamic damping proportional to  $\omega_b = [p \quad q \quad r]^T$  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) so that  $M_{damp}$  can be expressed as

$$M_{damp} = \begin{bmatrix} \frac{\delta L}{\delta p} & 0 & 0 \\ 0 & \frac{\delta M}{\delta q} & 0 \\ 0 & 0 & \frac{\delta N}{\delta r} \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix}. \quad (4.44)$$

In equation (4.44) the derivatives  $\frac{\delta L}{\delta p}$ ,  $\frac{\delta M}{\delta q}$  and  $\frac{\delta N}{\delta r}$  are called *stability derivatives* and they can be expressed in an analytical form, which is taken from the helicopter's dynamics background as in [19] - [21].

Because of the geometry of the quadrotor helicopter we can assume  $\frac{\delta L}{\delta p} = \frac{\delta M}{\delta q}$ , moreover the stability derivatives can be expressed as

$$\frac{\delta L}{\delta p} = -2\rho A R^2 \Omega^2 \frac{\delta C_T}{\delta p} b \quad (4.45)$$

$$\frac{\delta C_T}{\delta p} = \frac{C_{L\alpha}}{8} \frac{\sigma}{R\Omega} b, \quad (4.46)$$

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

$$\sigma = \frac{A_b}{A}$$

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

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 of which 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, *i.e.*,  $\frac{\delta N}{\delta r} = 0$ .

## 4.6 State space model

The six degrees of freedom model for the quadrotor helicopter kinematics and dynamics can be summarized as follows

$$V_E = R_{EB}V_b \quad (4.47)$$

$$\dot{V}_b = [(F_{ext} + F_g) - \omega_b \times (mV_b)] \frac{1}{m} \quad (4.48)$$

$$\omega_e = S^{-1}(\phi, \theta) \omega_b \quad (4.49)$$

$$\dot{\omega}_b = I_n^{-1} (M_{ext} + M_{damp} - \omega_b \times I_n \omega_b). \quad (4.50)$$

Since the usual expressions of the equations of motion use the concept of a state vector, a state space representation of the derived model will be now presented. 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 remain grouped as defined previously.

In our case the state vector has the following structure

$$x = \begin{bmatrix} P_e \\ V_b \\ \beta_e \\ \omega_b \end{bmatrix} = [N \ E \ D \ u \ v \ w \ \phi \ \theta \ \psi \ p \ q \ r]^T. \quad (4.51)$$

Therefore, the state equations are given by

$$\dot{x} = \begin{bmatrix} \dot{N} \\ \dot{E} \\ \dot{D} \\ \dot{u} \\ \dot{v} \\ \dot{w} \\ \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \\ \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} R_{EB}V_b \\ [(F_{ext} + F_g) - \omega_b \times (mV_b)] \frac{1}{m} \\ S^{-1}(\phi, \theta) \omega_b \\ (M_{ext} + M_{damp} - \omega_b \times I_n \omega_b) I_n^{-1} \end{bmatrix}. \quad (4.52)$$

As already stated in the introduction, unlike standard quadrotor helicopter UAVs, the tiltrotor quadcopter has an augmented mobility thanks to its over-actuated behaviour, that is, the availability of eight independent control inputs given by the spinning velocity of each propeller and the tilting actuation of the arms. With this new mechanical design, the quadrotor helicopter will be able to track any arbitrary trajectory by decoupling its attitude from translational movements. In the following sections the derivation of the relationships between the forces and control inputs will be derived, *i.e.*, the so called *mixer matrix* will be presented.

## 4.7 External forces and mixer matrix

The aim of this section is to describe the forces and torques that act on a conventional quadrotor helicopter, and then extend the discussion to the more general case of the tiltrotor quadcopter. The classical quadrotor helicopter can in fact be seen as a particular case of the tiltrotor one. As already stated, the quadrotor is an underactuated system that is able to move in six degrees of freedom (three translational and three rotational) but has only four control inputs (the speeds of each motor).

This underactuation causes the quadrotor helicopter to have a coupling between translational movements and attitude. Conventionally, its attitude is controlled by changing the rotational speed of each motor. The front motor and back rotor pair rotate in a clockwise direction, while the right and left rotor rotate in a counter-clockwise direction. This configuration is devised in order to balance the angular momentum generated by each of the spinning rotor pairs. There are basically four maneuvers that can be accomplished

by changing the speeds of the four rotors. The roll angle  $\phi$  is controlled by changing the relative speed of the right and left rotors, the pitch angle  $\theta$  is controlled by varying front and back speed of the rotors and the yaw angle  $\psi$  is controlled by varying the speeds of clockwise rotating pair and counter-clockwise rotating ones. Increasing or decreasing the speeds of all four rotors simultaneously controls the collective thrust generated by the quadrotor [5],[27],[6]. In the classic quadcopter case the overall forces acting on the vehicle can be easily computed as

$$F_z = F_f + F_b + F_l + F_r,$$

the rolling torque as

$$L = b(F_l - F_r),$$

the pitching torque as

$$M = b(F_f - F_b),$$

with  $b$  being the length of the quadcopter's arm. Finally the yaw torque balance can be expressed as

$$N = \tau_r + \tau_l - \tau_f - \tau_b.$$

Those equations are usually reorganized in a matrix form, in a way such that a mapping between the desired forces and torques and relative speeds of each motor is obtained:

$$\begin{bmatrix} F_z \\ L \\ M \\ N \end{bmatrix} = \begin{bmatrix} -K_t & -K_t & -K_t & -K_t \\ 0 & -bK_t & 0 & bK_t \\ bK_t & 0 & -bK_t & 0 \\ -K_q & K_q & -K_q & K_q \end{bmatrix} \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix} = \mathcal{X} \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix}. \quad (4.53)$$

The presence of the minus sign in each element of the first row is due to the fact that in the defined reference frame the thrust force  $F_z$  is positive

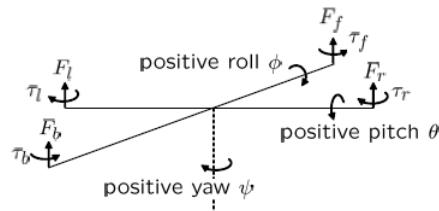


Figure 4.10: Forces and torques acting on the quadrotor

downward.  $\mathcal{X}$  is generally called **mixer matrix**. The inverse relationship that computes the rotational speeds given the forces and moments is simply obtained as

$$\begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix} = \mathcal{X}^{-1} \begin{bmatrix} F_z \\ L \\ M \\ N \end{bmatrix}. \quad (4.54)$$

In the case of the tiltrotor quadcopter, things get more complicated. In fact there is no more a *fully determined* system of four equations and four unknowns, but we have to deal with a non linear *underdetermined system* of eight unknowns and six equations. This situation is well known in the robotics literature as **kinematic redundancy**. In order to derive a *mixer matrix* for the tiltrotor's case, we need to make some simplifying assumptions: in equation (4.38) the gyroscopic effects and inertial ones are neglected since we can assume small accelerations  $\omega_{P_i}$  and velocities  $\omega_{P_i}$  compared to other quantities. Under these assumptions equation (4.38) becomes

$$\tau_{P_i} = -\tau_{ext_i}, \quad (4.55)$$

and then by applying equations (4.41) and (4.32) we get

$$\begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} = \begin{bmatrix} -K_t \sin(\alpha_2)\omega_2^2 + K_t \sin(\alpha_4)\omega_4^2 \\ K_t \sin(\alpha_1)\omega_1^2 - K_t \sin(\alpha_3)\omega_3^2 \\ -K_t \cos(\alpha_1)\omega_1^2 - K_t \cos(\alpha_2)\omega_2^2 - K_t \cos(\alpha_3)\omega_3^2 - K_t \cos(\alpha_4)\omega_4^2 \end{bmatrix} \quad (4.56)$$

$$\begin{bmatrix} L \\ M \\ N \end{bmatrix} = \begin{bmatrix} K_q \omega_4^2 s(\alpha_4) - K_q \omega_2^2 s(\alpha_2) - K_t b \omega_2^2 c(\alpha_2) + K_t b \omega_4^2 c(\alpha_4) \\ K_q \omega_3^2 s(\alpha_3) - K_q \omega_1^2 s(\alpha_1) + K_t b \omega_1^2 c(\alpha_1) - K_t b \omega_3^2 c(\alpha_3) \\ K_q \omega_2^2 c(\alpha_2) - K_q \omega_1^2 c(\alpha_1) - K_q \omega_3^2 c(\alpha_3) + K_q \omega_4^2 c(\alpha_4) + K_t b \omega_1^2 s(\alpha_1) + K_t b \omega_2^2 s(\alpha_2) + K_t b \omega_3^2 s(\alpha_3) + K_t b \omega_4^2 s(\alpha_4) \end{bmatrix}, \quad (4.57)$$

where  $s(\alpha_i)$  and  $c(\alpha_i)$  represent sine and cosine functions of angles  $\alpha_i$ . As further evidence that the classic quadrotor configuration can be seen as a special case of the tiltrotor one, replacing  $\alpha_i = 0$  in equations (4.56) and (4.57) leads to the same result as in (4.53).

#### 4.7.1 Linearized mixer matrix

Since equations (4.56) and (4.57) are nonlinear, in order to determine a direct relationship between forces/torques and rotor spinning velocities, a linearization of equations (4.56) and (4.57) in a hovering condition will be performed.

During a hovering flight, the quadrotor helicopter not only has zero acceleration but also its velocities and tilting angles are assumed to be zero, *i.e.*,  $\phi = \theta = \psi = 0$ ,  $\alpha_i = 0$ . To this end we can also define the speed at which the propellers are rotating during a hovering condition  $\omega_{hover}$ . Considering a single motor we have that

$$\begin{cases} T_i = \frac{1}{4}mg \\ T_i = K_t \omega_{hover}^2 \end{cases} \rightarrow \omega_{hover} = \sqrt{\frac{\frac{1}{4}mg}{K_t}}. \quad (4.58)$$

Let now

$$q = \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \\ \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \end{bmatrix} \quad \text{and} \quad f = \begin{bmatrix} F_{ext} \\ M_{ext} \end{bmatrix}, \quad (4.59)$$

be the vectors that will be used to express the Jacobian as

$$J = \frac{df}{dq} = \begin{bmatrix} \frac{\delta F_x}{\delta \omega_i^2} & \dots & \frac{\delta F_x}{\delta \alpha_i} \\ \vdots & \ddots & \vdots \\ \frac{\delta F_z}{\delta \omega_i^2} & \dots & \frac{\delta F_z}{\delta \alpha_i} \\ \frac{\delta L}{\delta \omega_i^2} & \dots & \frac{\delta L}{\delta \alpha_i} \\ \vdots & \ddots & \vdots \\ \frac{\delta N}{\delta \omega_i^2} & \dots & \frac{\delta N}{\delta \alpha_i} \end{bmatrix} \quad i = 1, \dots, 4. \quad (4.60)$$

Thus, the mapping between forces and spinning velocities and tilting angles can be expressed as

$$\begin{bmatrix} \delta F_x \\ \delta F_y \\ \delta F_z \\ \delta L \\ \delta M \\ \delta N \end{bmatrix} = \mathcal{X}_{tilt} \begin{bmatrix} \delta \omega_1^2 \\ \delta \omega_2^2 \\ \delta \omega_3^2 \\ \delta \omega_4^2 \\ \delta \alpha_1 \\ \delta \alpha_2 \\ \delta \alpha_3 \\ \delta \alpha_4 \end{bmatrix},$$

where

$$\mathcal{X}_{\text{tilt}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & -K_t \omega_{\text{hover}}^2 & 0 & K_t \omega_{\text{hover}}^2 \\ 0 & 0 & 0 & 0 & K_t \omega_{\text{hover}}^2 & 0 & -K_t \omega_{\text{hover}}^2 & 0 \\ -K_t & -K_t & -K_t & -K_t & 0 & 0 & 0 & 0 \\ 0 & -K_t b & 0 & K_t b & 0 & -K_q \omega_{\text{hover}}^2 & 0 & K_q \omega_{\text{hover}}^2 \\ K_t b & 0 & -K_t b & 0 & -K_q \omega_{\text{hover}}^2 & 0 & K_q \omega_{\text{hover}}^2 & 0 \\ -K_q & K_q & -K_q & K_q & K_t b \omega_{\text{hover}}^2 & K_t b \omega_{\text{hover}}^2 & K_t b \omega_{\text{hover}}^2 & K_t b \omega_{\text{hover}}^2 \end{bmatrix}. \quad (4.61)$$

The obtained Jacobian (4.61) will be the *linearized tilt-rotor's mixer matrix*. As can be seen, since the *mixer matrix* is rectangular, it can't be inverted as in the classic quadrotor case but another strategy must be adopted. From the robotics literature we know that to solve the *inverse kinematics* problem, Jacobian-based methods are widely used. Those methods choose, among the infinite system's solutions, the one that minimizes a suitable (possibly weighted) norm. They usually exploit the **pseudoinverse** matrix.

Given a matrix  $A \in M(m, n; \mathbb{R})$ , a pseudoinverse of  $A$  is defined as a matrix  $A^\# \in M(n, m; \mathbb{R})$  satisfying all of the following four criteria:

- $AA^\#A = A$
- $A^\#AA^\# = A^\#$
- $(AA^\#)^T = AA^\#$
- $(A^\#A)^T = A^\#A$

$A^\#$  always exists for any matrix  $A$ . If  $A$  is full (row) rank, its pseudoinverse can be computed as

$$A^\# = A^T (AA^T)^{-1}.$$

If  $A$  is not full rank, it can be computed numerically using the Singular Value Decomposition (SVD) of matrix  $A$  (command *pinv* in MATLAB). Given a system of linear equations, the least squares solution provided by the pseudoinverse will be now presented. For linear systems  $Ax = b$ , with non-unique solutions (such as under-determined systems), the pseudoinverse may be used to construct the solution of minimum Euclidean norm  $\|x\|_2$  among all solutions. If  $Ax = b$ , is satisfiable, the vector  $z = A^\#b$  is a solution, and satisfies  $\|z\|_2 \leq \|x\|_2$  for all solutions. The pseudoinverse can also be expressed in its weighted form: in this way by weighting the variables it is possible to obtain some solutions with a predefined use of the control inputs. The weighted pseudoinverse is defined as

$$A_w^\# = W^{-1} A^T (A^T W^{-1} A^T)^{-1}$$

where  $W$  is a  $n \times n$  symmetric positive definite weighting matrix. Thanks to the introduced mathematical definitions, it is now possible to solve the inverse mixer matrix problem with the weighted pseudoinverse, thus

$$\begin{bmatrix} \delta\omega_1^2 \\ \delta\omega_2^2 \\ \delta\omega_3^2 \\ \delta\omega_4^2 \\ \delta\alpha_1 \\ \delta\alpha_2 \\ \delta\alpha_3 \\ \delta\alpha_4 \end{bmatrix} = \mathcal{X}_{tilt_w}^\# \begin{bmatrix} \delta F_x \\ \delta F_y \\ \delta F_z \\ \delta L \\ \delta M \\ \delta N \end{bmatrix}. \quad (4.62)$$

Thanks to equation (4.62) a relationship between requested forces/torques and the actual control variables of the tiltrotor quadcopter has been realized. It must be stated that this solution is acceptable for the aim that we want to reach in this thesis, that is, make the tiltrotor quadcopter fly and to be able to move on a plane without changing its attitude. In case of more complex trajectories, this solution may be no longer reliable due to the intrinsic limitations that the linearization process introduces, *e.g.*, we shall consider small tilt angles variation (about  $\pm 10^\circ$  from its nominal orientation).

#### 4.7.2 Direct resolution of the kinematic problem

In this thesis, a different approach to solve the mixer matrix inversion has also been developed. Since equations (4.56) and (4.57) are function of the  $\alpha_i$  angles and the  $\omega_i$  spinning velocities, one way to invert the relationship without using the pseudoinverse is to reduce the problem complexity to a fully defined system of equations. This is possible by simply noticing that the translational forward movement is achieved by tilting rotors 2 and 4, while translational side movement is achieved by tilting rotors 1 and 3 such that

$$\alpha_3 = -\alpha_1 \quad (4.63)$$

$$\alpha_4 = -\alpha_2. \quad (4.64)$$

Thanks to equations (4.63) and (4.64), the overall six by eight unknown variables system of equations can be reduced to a fully determined system with only six unknowns, that are the four spinning velocities  $\omega_i$  and the two tilting angles  $\alpha_1$  and  $\alpha_2$ . It is thus possible to solve the translational movement along the X axis for  $\alpha_2$  and the translational movement along the

Y axis for  $\alpha_1$ , assuming the quadrotor to be in an hovering condition, *i.e.*,  $\omega_i = \omega_{hover}$  we get

$$\begin{cases} F_x - (-K_t \sin(\alpha_2)\omega_{hover}^2 + K_t \sin(\alpha_4)\omega_{hover}^2) = 0 \\ F_y - (K_t \sin(\alpha_1)\omega_{hover}^2 - K_t \sin(\alpha_3)\omega_{hover}^2) = 0 \\ \alpha_3 = -\alpha_1 \\ \alpha_4 = -\alpha_2. \end{cases} \quad (4.65)$$

The solutions of (4.65) are

$$\alpha_1 = \arcsin\left(\frac{Fy}{2K_t\omega_{hover}^2}\right) \quad (4.66)$$

$$\alpha_2 = -\arcsin\left(\frac{Fx}{2K_t\omega_{hover}^2}\right). \quad (4.67)$$

It is important to remark that the argument of the arcsine function must be between  $[-1, 1]$ . This requirement is always satisfied since the traslational forces will always be at least one order lower than the spinning velocities in near hovering conditions.



## Chapter 5

# Identification of attitude dynamics

In this chapter the experimental identification campaign performed to identify the unknowns attitude model parameters, *i.e.*, the stability derivative and the model inertia will be presented. In a first attempt, the model parameters will be identified with two different approaches: the grey-box approach, by neglecting the time delay of the system, and the SRIVC approach by taking into account the time delay of the system since this method (Section 3.3.2) is able to both identify the continuous-time model parameters and a time-delay. The results provided by the two identification methods will be then compared and lastly the grey-box identification will be performed again but this time by taking into account the SRIVC identified time-delay. Lastly a comparison of the obtained results will be exposed.

### 5.1 Dynamic model of roll attitude

Starting by recalling the equations derived in Chapter 4, the overall attitude dynamical model can be expressed as:

$$\dot{\omega}_b = I_n^{-1} (M_{ext} + M_{damp} - \omega_b \times I_n \omega_b). \quad (5.1)$$

After some expansion and by neglecting the Coriolis terms, the roll dynamics can be expressed as

$$\dot{p} = \frac{1}{I_{xx}} \left[ L + \frac{\delta L}{\delta p} p \right], \quad (5.2)$$

where  $L$  represents the external rolling moment

$$L = K_q \omega_4^2 s(\alpha_4) - K_q \omega_2^2 s(\alpha_2) - K_t b \omega_2^2 c(\alpha_2) + K_t b \omega_4^2 c(\alpha_4),$$

and  $\frac{\delta L}{\delta p}p$  is the stability derivative. Next the simplified dynamical model in equation (5.2) has been linearized in a near-hovering condition where the arms are not tilted and the rotors spinning velocity have a small deviation from the hovering value  $\Omega_h$ ,

$$\begin{aligned}\bar{\omega}_4 &= \bar{\omega}_2 = \Omega_h \\ \delta\Omega_4 &= \Omega_h - \delta\Omega \\ \delta\Omega_2 &= \Omega_h + \delta\Omega.\end{aligned}$$

Equation (5.2) thus becomes

$$\delta\dot{p} = \frac{1}{I_{xx}} \left[ 2K_t b\Omega_h (\delta\Omega_4 - \delta\Omega_2) + \frac{\delta L}{\delta p} \delta p \right] \quad (5.3)$$

$$= \frac{1}{I_{xx}} \left[ 4K_t b\delta\Omega + \frac{\delta L}{\delta p} \delta p \right]. \quad (5.4)$$

As can be seen from equation (5.4), since in this thesis it is assumed that the attitude is only controlled by the spinning velocity of the propellers, no tilting angles appear into the roll dynamical model. Written in a state-space form, the linearized model (5.4) becomes

$$\begin{aligned}\delta\dot{x} &= A\delta x + B\delta u \\ \delta y &= C\delta x + D\delta u\end{aligned}$$

where

$$\delta x = p, \quad \delta u = \delta\Omega$$

$$A = \left[ \frac{1}{I_{xx}} \frac{\delta L}{\delta p} \right], \quad B = \left[ 4K_t b\Omega_h \frac{1}{I_{xx}} \right], \quad C = [1], \quad D = 0.$$

The grey-box identification algorithm used to identify the parameters is the one provided in the MATLAB System Identification Toolbox with the command *greyest*. For the identification problem, the matrices of the system have been parametrized as

$$A = [A1], \quad B = [B1], \quad C = [1], \quad D = 0$$

and the parameters  $I_{xx}$  and  $\frac{\delta L}{\delta p}$  will be derived *a posteriori*. For the SRIVC method, the derived model must be converted into a transfer function:

$$\frac{\delta P(s)}{\delta\Omega(s)} = \frac{\frac{4K_t b\Omega_h}{I_{xx}}}{\left( s - \frac{1}{I_{xx}} \frac{\delta L}{\delta p} \right)}. \quad (5.5)$$

Expressed in a parametric way and by taking into account the time delay of the system, the overall model that the SRIVC method will identify is

$$\frac{\delta P(s)}{\delta \Omega(s)} = \frac{a_s}{s + b_s} e^{-\tau s}. \quad (5.6)$$

The  $K_t$  parameter is assumed already identified from the identification campaign described in Section 3.2, so that the only parameters to be identified will be the stability derivative  $\frac{\delta L}{\delta p}$  and the model inertia  $I_{xx}$ .

### 5.1.1 Experimental setup

The roll attitude identification has been carried out with the tiltrotor fixed on a test-bed that limits the DoF of the system to only the rotation around the X axis as in Figure 5.1. The tiltrotor has been fixed to the test-bed

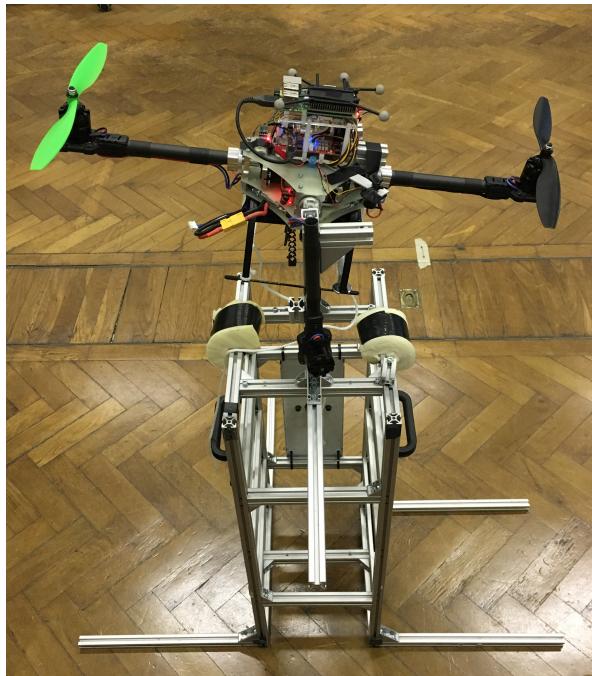


Figure 5.1: Experimental setup

by trying to maintain the axis of rotation as close as possible to the CoG of the system such that the pendulum effect was limited. The test-bed has also been designed high enough from the ground to avoid ground effects disturbances during the identification and also with a smooth fulcrum to reduce friction forces between the tiltrotor and the test-bed. The input

commands are directly generated and sent from MATLAB to the flight controller through a serial bluetooth connection and the output measurements are logged at a rate of 100Hz on an onboard SD card.

### 5.1.2 Excitation input signal - Random Binary Sequence

Past works (see [9] [14] [10]) have proven that a successful way to excite the system during an experimental identification campaign is through a Random Binary Sequence (RBS) signal due to its similar properties to a white noise signal. In order to obtain an excitation spectrum consistent with the expected dominant attitude dynamics, the parameters of the RBS signal, namely the lower and the upper bound of the RBS passband filter expressed in fractions of the Nyquist frequency ( $\omega_{min}$  and  $\omega_{max}$  respectively) and the system guess bandwidth  $BW$ , have been tuned. Practically speaking, the input signal corresponds to the spinning velocity difference between the left and right motor, that will generate an angular displacement on the roll axis. The initial guess of the system bandwidth has been obtained by evaluating

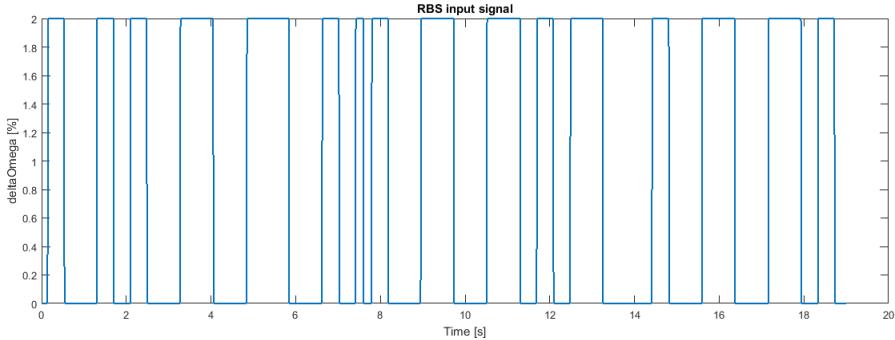


Figure 5.2: Example of a RBS input signal

the dynamical roll model with the values obtained in [14],

$$I_{xx} = I_{yy} = 34.7 \times 10^{-3} \text{ [kgm}^2\text{]}$$

$$\frac{\delta L}{\delta p} = -46.3 \times 10^{-3} \text{ [Nm s]},$$

obtaining a guess bandwidth of 1 rad/s. After several unsuccessful trials to identify the roll dynamics with a RBS signal tuned on the obtained guess bandwidth, the input signal bandwidth has been increased up to about 10 rad/s. This choice is justified by the fact that, since the first guess bandwidth was obtained on a quadcopter with a "X" frame configuration while

the tiltrotor quadcopter has a "+" configuration, the rolling moment of inertia acting on the system can differ a lot due to the different structure. An excitation bandwidth around 8 rad/s has also been used in [21] to identify the attitude dynamics of a bigger "+" shaped quadcopter so the choice of 10 rad/s for a smaller "+" shaped quadcopter seemed reasonable. Indeed, with the new excitation signal meaningful results were obtained. More precisely, the system has been excited with three different RBS signals with an excitation spectrum tuned around the new bandwidth guess (parameters are summarised in Table 5.1).

	$BW$	$\omega_{min}$	$\omega_{max}$
Test A	10	0.1	1
Test B	8	0.1	1
Test C	12	0.1	1

Table 5.1: RBS Parameters

### 5.1.3 Identification results

For each of the three RBS signals introduced in Table 5.1, 10 datasets have been collected and analyzed by computing the weighted average and variance of each identified parameter in both the grey-box and SRIVC identification methods. Results are resumed in Table 5.2. For what it concerns the SRIVC

	A1	$\sigma_{A1}$	B1	$\sigma_{B1}$	$a_s$	$\sigma_{a_s}$	$b_s$	$\sigma_{b_s}$
Test A	-3.88	0.150	5.66	0.131	8.39	0.0064	6.84	0.0065
Test B	-3.83	0.136	5.36	0.122	7.22	0.0061	5.95	0.0061
Test C	-4.05	0.215	6.43	0.217	10.35	0.0082	8.11	0.0083

Table 5.2: Weighted average parameters and standard deviation  $\sigma$

identified time delay, results are resumed in Table 5.3. As can be seen

	Time delay [s]
Test A	0.09006
Test B	0.0752
Test C	0.086967

Table 5.3: SRIVC identified time delay

from Figure 5.3, in both grey-box and SRIVC methods, identification results obtained in test C deviate from the ones obtained in test A and B. Moreover, compared to the SRIVC results, the grey-box method provides identification results with higher parameter uncertainty.

### 5.1.4 Choice of best identified model

To decide which model over the three different identification campaigns best describes the real behaviour of the system, a validation procedure has been followed, and based on the results, the best model group has been chosen. Firstly the best grey-box model and the best SRIVC model over every test A,B and C has been selected based on the VAF. Secondly the average model of each A,B and C test for both grey-box and SRIVC methods has been picked. Lastly each model has been tested with a random dataset, excluded the one used to estimate the best model. Hereafter the model output and measured output have been compared. To evaluate the goodness of the

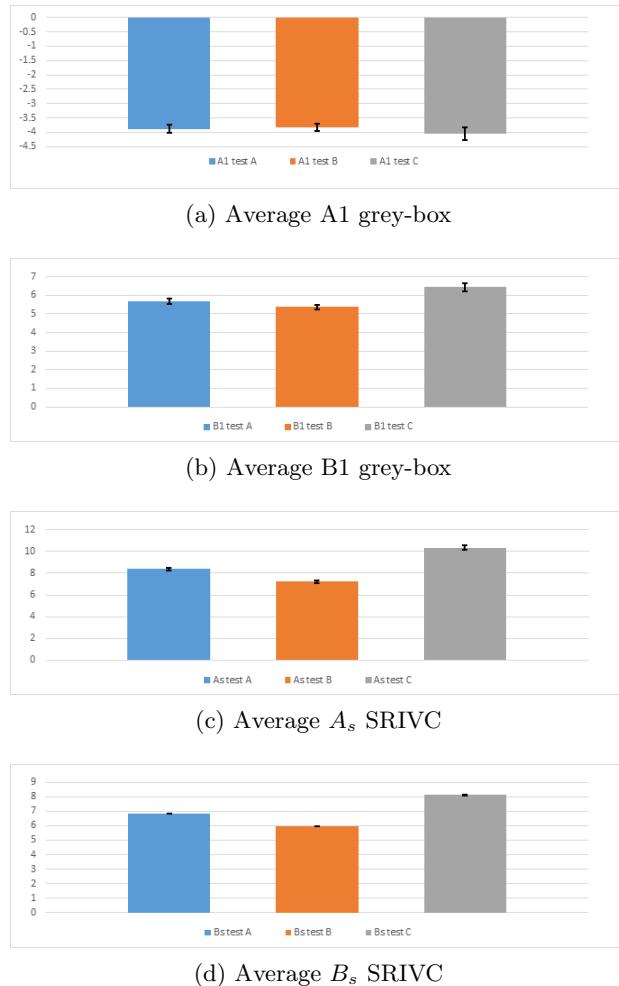


Figure 5.3: Error bars diagram

	VAF %	Best model group		VAF %	Best model group
<b>Random dataset 1</b>			<b>Random dataset 2</b>		
Test best grey	73.18	\	Test best grey	71.43	\
Test best SRIVC	80.43	\	Test best SRIVC	81.14	\
Test avg grey	73.82	A	Test avg grey	70.84	A
Test avg SRIVC	81.08	A	Test avg SRIVC	82.2	A
<b>Random dataset 3</b>			<b>Random dataset 4</b>		
Test best grey	80.59	\	Test best grey	71.61	\
Test best SRIVC	85.83	\	Test best SRIVC	79.69	\
Test avg grey	80.86	B	Test avg grey	72.94	A
Test avg SRIVC	86.38	A	Test avg SRIVC	80.47	A
<b>Random dataset 5</b>			<b>Random dataset 6</b>		
Test best grey	74.83	\	Test best grey	75.44	\
Test best SRIVC	82.15	\	Test best SRIVC	84.3	\
Test avg grey	74.69	A	Test avg grey	74.71	A
Test avg SRIVC	82.33	A	Test avg SRIVC	84.65	A

Table 5.4: Choice of the best model based on VAF

identified model the VAF index has been computed. The results of the 6 validation test are resumed in Table 5.4. From Table 5.4 is clear that the models that better describes the data are the ones obtained with campaign A.

### 5.1.5 Time delay influence on grey-box identification

At this point, to evaluate if the effects of the time delay negatively affected the grey-box identification, the procedure has been repeated over the datasets of campaign A, this time by taking into account the identified SRIVC time delay of 90 ms. The obtained grey-box averaged results are summarized in Table 5.5. By comparing the obtained results in Table

	A1	$\sigma_{A1}$	B1	$\sigma_{B1}$
Test A	-6.47	0.19	8.26	0.15

Table 5.5: Grey-box weighted average parameters considering the time delay

5.2 with the ones in Table 5.5, is evident that the values of the identified parameters are highly affected by the time delay. Infact, thanks to the introduced time shifting to compensate the delay, the grey-box results get closer to the ones obtained with the SRIVC method which always provided better VAF performances (Table 5.4). A clear evidence is also visible through a comparison of the frequency response of the identified systems:

- Figure 5.4, shows that the identified grey-box model obtained neglecting the time delay has provided a model with a slightly higher gain and a bandwidth of 4 rad/s , while the SRIVC method has provided a lower gain and a bandwidth of 7 rad/s. Also the phase has a noticeable difference in the two identified models.
- Figure 5.5 instead shows a clear improvement in terms of similarity with the SRIVC model in the frequency response of the grey-box identified system.

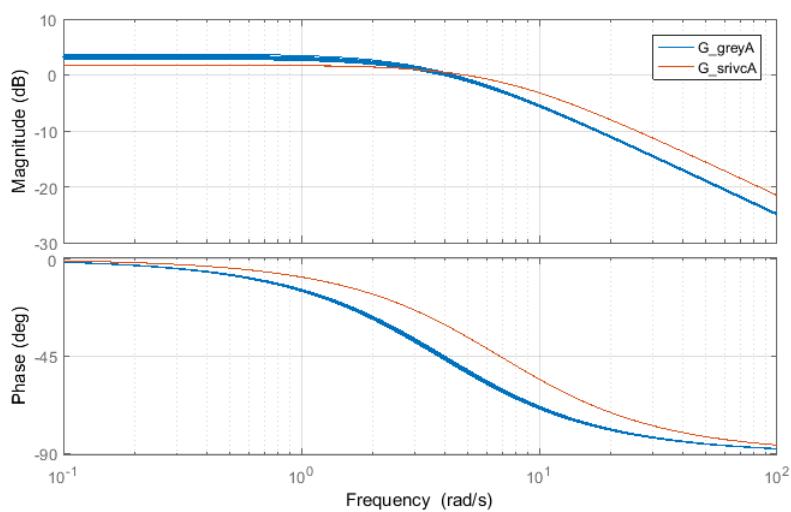


Figure 5.4: Frequency response comparison: grey-box no time-shifted identified model (blue line) vs SRIVC identified model (red line)

As can also be seen from the time response depicted in Figure 5.6, with the compensation of the time delay in the grey-box method, the results achieved are basically the same.

### 5.1.6 Final results and remarks

To sum up, for the identification of the roll attitude dynamics two identification methods have been adopted: the grey-box method and the SRIVC method. In both cases a RBS input signal has been applied to the linearized SISO system and the angular rate output has been measured. With the logged data two identification attempts have been done. In a first attempt the system has been identified with both the grey-box method and the SRIVC method without doing any data preconditioning. For the grey-box identification results achieved at the first attempt were not satisfying

due to the time delay of the system. On the other hand the SRIVC method achieved good results without any data preconditioning, thanks to its ca-

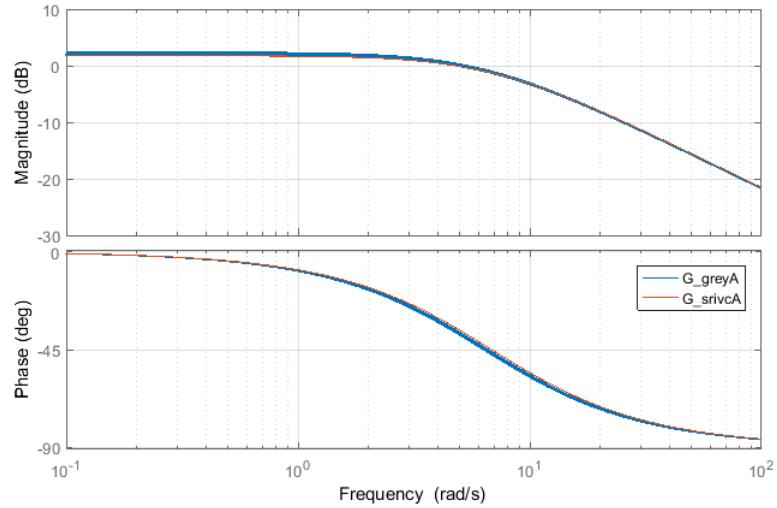


Figure 5.5: Frequency response comparison: Grey-box time-shifted identified model (blue line) vs SRIVC identified model (red line)

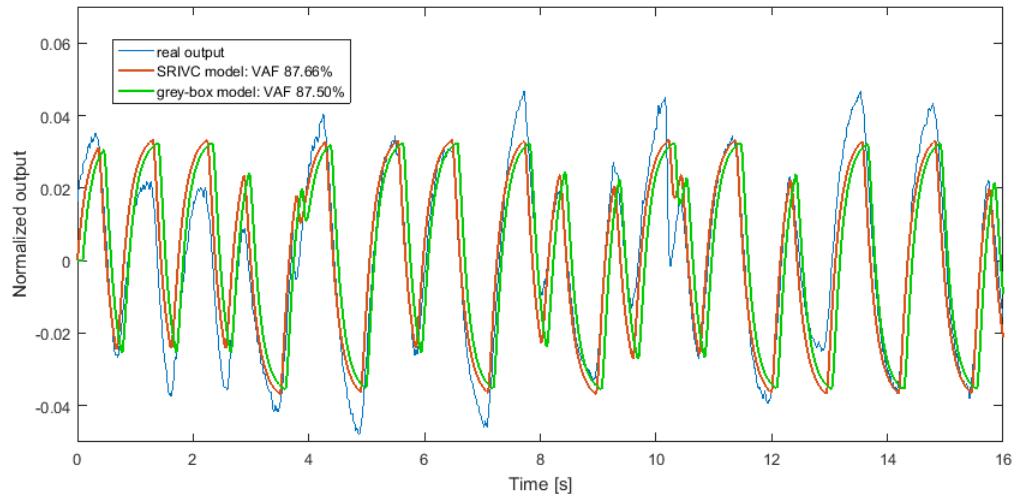


Figure 5.6: Time response comparison: Grey-box time-shifted identified model (green line) vs SRIVC identified model (red line) vs real output (blue line)

pability of handling SISO system with time delays. On a second attempt, datasets have been time-shifted to compensate the time delay of the system, previously identified with SRIVC method. The grey-box procedure has been reapplied on the preconditioned datasets to evaluate the effects of the time delay on the grey-box method. Results have shown that with the second attempt, grey-box identification has increased its performances reaching the same reliability level of the identified SRIVC system. In conclusion, the identified bandwidth of the system ended up to be around 7 rad/s. Since the input RBS signal  $\delta\Omega\%$  has been expressed in a percentage variation of the spinning velocity, in order to obtain a physical values of the stability derivative and the moment of inertia, the static relationship between throttle % and  $\Omega$  [rad/s] introduced in Chapter 3.2 must be taken into account as depicted in Figure 5.7. The values of the stability derivative  $\frac{\delta L}{\delta p}$  and the

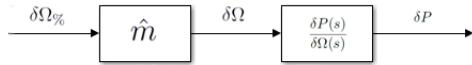


Figure 5.7: Block diagram of the overall identified system

inertial term  $I_{xx}$  can be thus obtained as follows

$$a_s = 4K_t b \Omega_h \hat{m} \frac{1}{I_{xx}} \rightarrow I_{xx} = \frac{4K_t b \Omega_h \hat{m}}{a_s} = 74 \times 10^{-4} \text{ [kgm2]}$$

$$b_s = -\frac{1}{I_{xx}} \frac{\delta L}{\delta p} \rightarrow \frac{\delta L}{\delta p} = -B_s I_{xx} = -51 \times 10^{-3} \text{ [Nm s].}$$



## Chapter 6

# Simulink implementation of the tiltrotor model

In this chapter the tiltrotor simulator will be presented. An overview of the main blocks will be made and after that a deeper presentation of the developed tiltrotor blocks will be performed. The overall system has been developed by implementing in MATLAB Simulink environment the dynamical equations introduced in the previous chapters.

### 6.1 Main blocks definition

The simulink model of the tiltrotor has not been made from scratch, but by starting from a previous work where a dynamical model of a classical quadrotor has been developed. The main blocks of the simulator in Figure 6.1 are the following:

- Tiltcopter block is the main block of the simulator, it is in charge to reply the dynamics of the vehicle.
- Mixer function block implements the relationship that converts physical quantities such as forces and torques into the real control inputs of the motors, that is, PWM signals (see Section 4.7.2 as reference).
- Controllers block implements all the controllers that are on the tiltrotor, such as the attitude controller, the tilting arms controller and so on.
- Sensor models block contains all the models that on the real system are translated into sensors, such as IMU, ultrasonic sensors for altitude, and Kalman filters for translational velocities and position estimation.

- The big input rectangle represents the control values that can be modified to evaluate the behaviour of the tiltrotor.

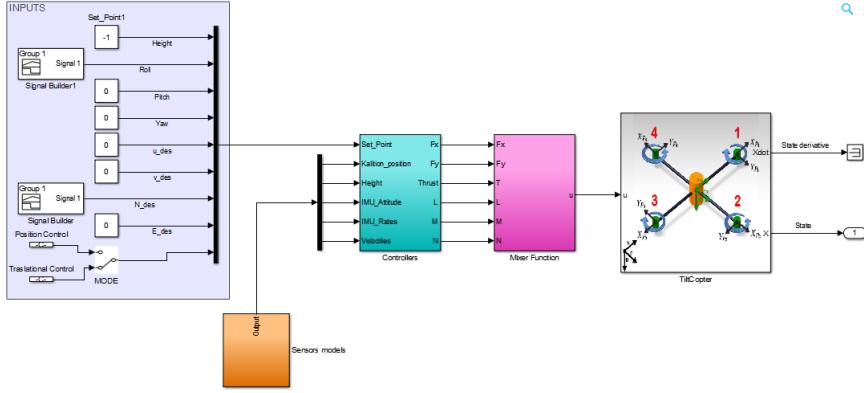


Figure 6.1: Simulink model

In the tiltrotor simulator, the biggest difference lies in the actuators block inside the TiltCopter block (its architecture is depicted in Figure 6.5, but since the Actuators block is so big that can not fit a standard page, the reader is referred to check the original simulink file), infact it has been expanded and modified to simulate the dynamical behavior of the servomotors and to allow the implementation of the new physical quantities introduced by the equations presented in Section 4.5.2. Starting from the identified system of the servomotor obtained in Section 3.3.3 and recalling the transfer function from  $\phi_{ref}(s)$  to  $\phi(s)$  described in equation (3.27), the system has been converted in a state-space representation as follows

$$\begin{cases} \dot{x}_1 = \phi \\ \dot{x}_2 = \dot{\phi} \\ \ddot{x}_3 = \ddot{\phi} \end{cases} \rightarrow \begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = x_3 \\ \dot{x}_3 = -\frac{B_2}{B_1}x_3 - \frac{B_3}{B_1}x_2 - \frac{B_4}{B_1}x_1 + \frac{B_0}{B_1}u \end{cases} \quad (6.1)$$

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \quad A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -\frac{B_4}{B_1} & -\frac{B_3}{B_1} & -\frac{B_2}{B_1} \end{bmatrix} \quad B = \begin{bmatrix} 0 & 0 & \frac{B_0}{B_1} \end{bmatrix} \quad C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (6.2)$$

and then implemented as in Figure 6.2 (in the previous equations  $u$  is the control variable  $\phi_{ref}$ ). A similar procedure has been performed on the ESC+Motor model, in order to obtain the desired physical quantities needed by the tiltrotor dynamical model equations, see Figure 6.3. The simulator

has also been equipped with a 3D animation plot that describes the movement of the tiltrotor in the space, by also characterizing the thrust vector with relative tilting angle, see Figure 6.4. Thanks to the developed simulator, the risk of damage when trying new control architectures on the real system has been drastically reduced since a rough estimation of the tiltrotor behaviour could be already determined during simulations.

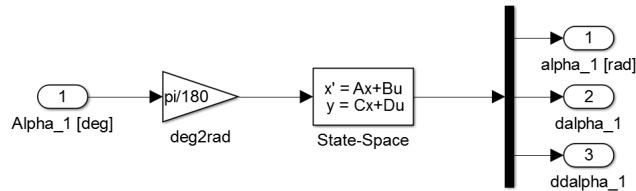


Figure 6.2: Servomotor simulink model

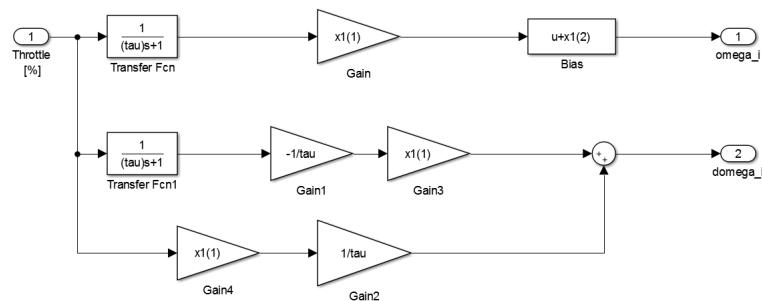


Figure 6.3: ESC+Motor simulink model

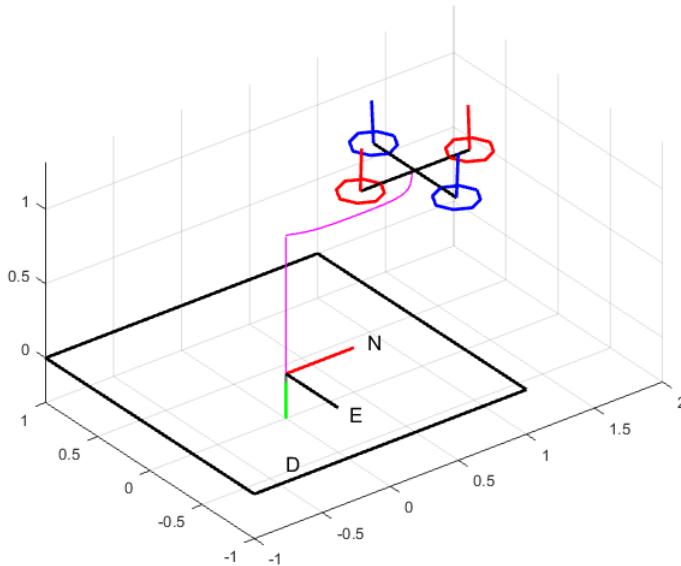


Figure 6.4: 3D plot of the tiltrotor

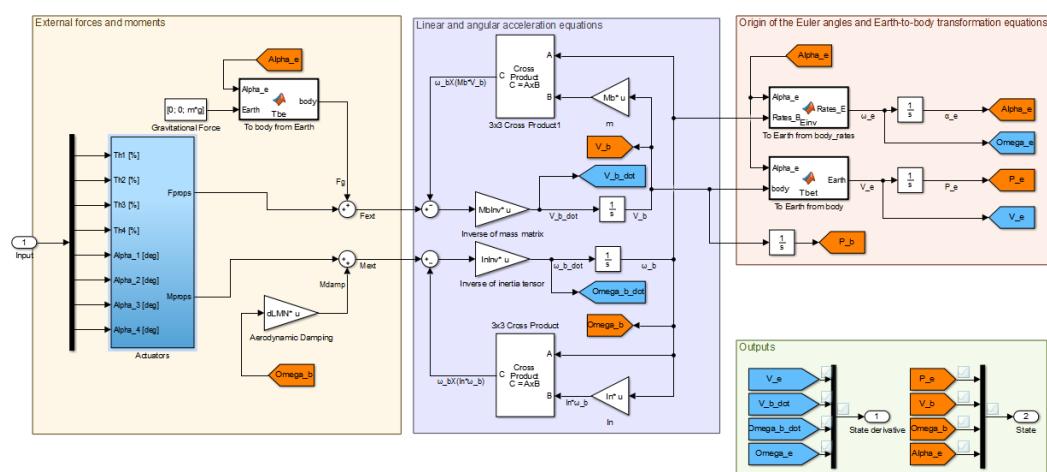


Figure 6.5: Tiltrotor block model

# Chapter 7

## Control law design

In this chapter, the procedure followed to design a robust control architecture to fully control both translational and attitude dynamics will be presented. The proposed approach is based on structured  $H_\infty$  synthesis applied to the linearised model of the tiltrotor quadcopter.

### 7.1 $H_\infty$ fixed-structure synthesis problem formulation

Before introducing the control architecture used to gain full control of the 6 DoFs of the tiltrotor quadcopter, an overview of the  $H_\infty$  control synthesis method will be given. With this approach, it is possible to design robust control systems for plant models with uncertain parameters by specifying stability, performance and robustness tuning requirements such as tracking performance, disturbance rejection and control bandwidth in terms of loop shaping functions. For example, a requirement of good reference tracking and disturbance rejection is equivalent to high ( $> 1$ ) open-loop gain at low frequency while a requirement of insensitivity to measurement noise or modelling error is equivalent to a low ( $< 1$ ) open-loop gain at high frequency. Another advantage of the  $H_\infty$  synthesis is that it is also applicable to MIMO systems. Design requirements are expressed as terms of  $H_\infty$  norm constraints. For SISO systems, the  $H_\infty$  norm express the maximum input/output gain of a stable transfer function and is defined as

$$\|H(s)\|_\infty = \max \bar{\sigma}(H(jw)). \quad (7.1)$$

The  $H_\infty$  synthesis problem can be formulated as follows: given a LTI model  $P(s)$  that contains all the fixed parameters of the control system and a structured controller  $K(s, \theta)$ , where  $\theta$  is the vector containing all the tunable

control parameters, the  $H_\infty$  synthesis searchs for the best parameter vector  $\theta_{opt}$  that minimizes the following norm

$$\|T_{w \rightarrow z}(P(s), K(s, \theta))\|_\infty, \quad (7.2)$$

where  $T_{w \rightarrow z}(P(s), K(s, \theta))$  represents the closed-loop transfer function from a chosen input channel  $w$  to an output channel  $z$ . The standard block diagram for structured  $H_\infty$  synthesis is depicted in Figure 7.1. To solve the

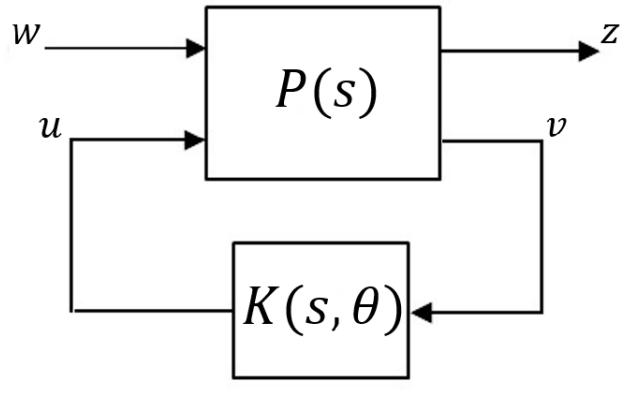


Figure 7.1: Standard representation for structured  $H_\infty$  synthesis

introduced non convex optimization problem, the Robust Control Toolbox of MATLAB & Simulink has been used. In particular the `<systune>` function and `<TuningGoal>` class have been employed. This tool uses specialized nonsmooth optimization techniques to enforce closed-loop stability and minimize the  $H_\infty$  norm as a function of the tunable parameters. Since the result of the optimization process provides a local minimum, the initialization of the tunable parameters will be the key point to obtain a satisfactory result. In the following section, the fixed control architecture defined for the attitude dynamics and the translational dynamics will be presented.

## 7.2 Control law design

### 7.2.1 Attitude control

The attitude control architecture design has been performed assuming a decoupled dynamics between the different DoFs. This assumption can be considered acceptable if the quadrotor is operated around a stable hover with small attitude angles and minimal rotational and translational velocities and

accelerations. This flying condition should not be considered as a limiting factor for the control architecture, since in most of the aerial works in which a multicopter is used, the craft is mainly operated in a working condition close to the introduced hypothesis in order to increase its efficiency in terms of flight time.

The presented control system will address the roll dynamics, but since it has been assumed that the quadrotor has a symmetrical structure, the results are also applicable for the pitch attitude dynamics.

For what concerns the yaw dynamics, this thesis will rely on the control system developed in [10]. The quadrotor attitude control is achieved with cascaded PID loops. As can be seen from Figure 7.2, the inner loop is based

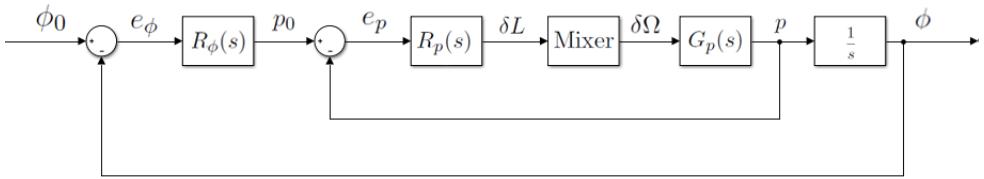


Figure 7.2: Nested loops attitude control architecture

on angular rate feedback (it receives the angular rate reference  $p_0$  and uses the measured angular velocity  $p$  as feedback) while the outer loop is based on the actual angle (it receives the angle reference  $\phi_0$  and uses the estimated angle  $\phi$  as feedback).

In detail, for  $R_\phi(s)$  a PD controller has been chosen (the intrinsic integrator of the system already guarantees null steady state error so there is no need of a PID structure), while for  $R_p(s)$  a PID controller has been chosen. The two regulators, expressed in parallel form, can be written as

$$R_\phi(s) = K_{p\phi} + K_{d\phi} \frac{s}{1 + sT_f}$$

$$R_p(s) = K_{pp} + K_{ip} \frac{1}{s} + K_{dp} \frac{s}{1 + sT_f},$$

where  $T_f$  represents the time constant of the low pass filter of the derivative term. The inner regulator returns as control variable the actual torque needed to achieve the desired angular rate, that is,  $\delta L$  for the roll dynamics and  $\delta M$  for the pitch dynamics. The mixer matrix (see Chapter 4.7.2), represented by the block "Mixer" in Figure 7.2, is in charge of effectively converting the desired torque into the real input of the system, that is, the

difference of spinning velocities of the rotors.

In order to define the initial condition of the tunable parameters for the  $H_\infty$  synthesis, the control architecture has been empirically tuned with a trial and error procedure applied firstly on the developed Simulink model and lastly on the real system (mounted on the test-bed).

With this procedure various requirement sets were tested until a good trade-off in terms of set-point tracking and robustness of the control system were achieved.

The following  $H_\infty$  requirements have been thus defined:

- Tracking requirements
  - Outer loop: the overall closed-loop system (from  $\phi_0$  to  $\phi$ ) must have a maximum relative peak error of 30%, maximum steady-state error equal to 0.0001% and be able to track signals in the bandwidth of 1.6 rad/s. This last requirement defines the rise time of the system, indeed the two parameters are related by the following relationship  $T_r = \frac{2}{\omega_c}$ .
  - Inner loop: for the requested outer loop bandwidth, the trial and error procedure evidenced that the inner-loop (from  $p_0$  to  $p$ ) should have at least 20 rad/s of bandwidth in order to obtain a flyable vehicle. For this reason the following tracking requirements have been defined: maximum relative peak error of 25%, maximum steady-state error equal to 0.0001% and 24 rad/s of bandwidth.

To solve tracking requirements, the Matlab toolbox algorithm adjusts the tunable values such that the following  $H_\infty$  norm is minimized:

$$f(\theta) = \left\| \frac{1}{MaxError} T(s, \theta) \right\|_\infty, \quad (7.3)$$

where  $T(s, \theta)$  is the closed-loop transfer function from a generic input  $u$  to a generic output  $y$  (in our case will be the complementary sensitivity function from  $\phi_0$  to  $\phi$  for the outer loop, and from  $p_0$  to  $p$  for the inner loop) while  $MaxError$  is a SISO LTI transfer function defined as follows

$$MaxError = \frac{(PeakError)s + \omega_c(DCError)}{s + \omega_c}. \quad (7.4)$$

- Disturbance rejection: this requirement specifies the minimum attenuation of a disturbance injected at a specified location in a control

system. In detail the closed-loop sensitivity function  $S(s, \theta)$  has been shaped like a first order high pass filter with 20 dB gain and a cutting frequency at 1.6 rad/s. The algorithm translates this requirement into an optimization problem on the following norm:

$$f(\theta) = \|W_S(s)S(s, \theta)\|_\infty, \quad (7.5)$$

where  $W_S$  is the loop shaping function that approximates the minimum disturbance attenuation requirement.

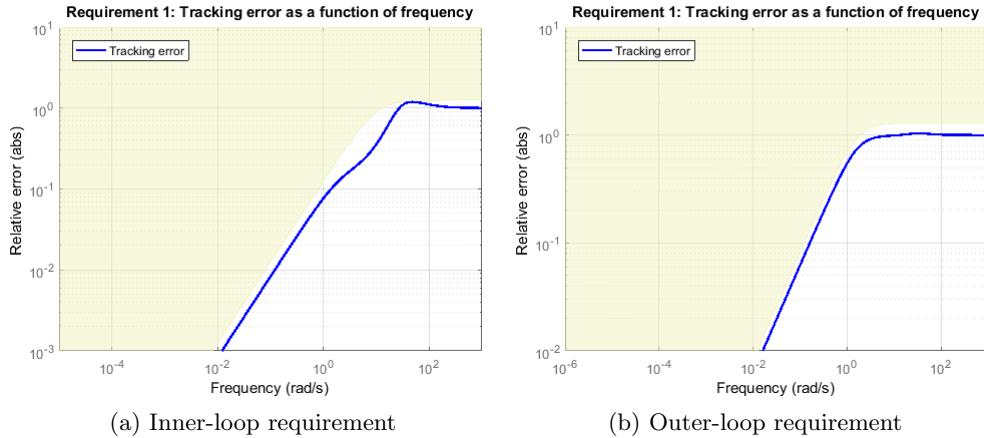


Figure 7.3: Tracking error requirements

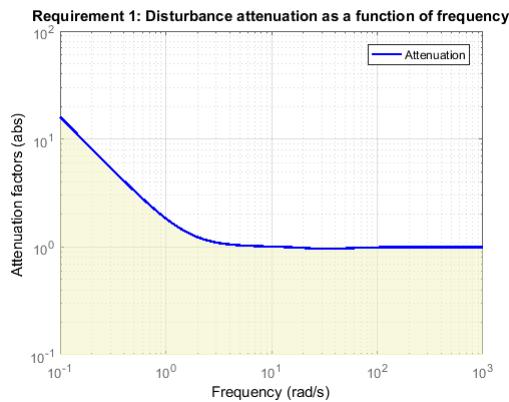


Figure 7.4: Outer-loop disturbance rejection requirement

Figures 7.3-7.4 show the results of the  $H_\infty$  bounds applied on the inner-loop and outer-loop transfer functions respectively.

For the sake of clarity, the definition of the various open-loop transfer functions  $L(s)$ , sensitivity transfer functions  $S(s)$  and complementary sensitivity transfer functions  $T(s)$  used in the  $H_\infty$  synthesis are below reported. Recalling the identified attitude dynamics (Section 5.1.6) defined by

$$G_p(s) = \frac{\delta P(s)}{\delta \Omega(s)} = \frac{\frac{4K_t b \Omega_h}{I_{xx}}}{\left(s - \frac{1}{I_{xx}} \frac{\delta L}{\delta p}\right)} = \frac{1.3908}{s + 6.845}, \quad (7.6)$$

the identified dynamics of the motors (Chapter 3),

$$G_{motor}(s) = \frac{\Omega(s)}{Th(s)} = \frac{1}{1 + s\hat{\tau}} = \frac{1}{1 + 0.055s}, \quad (7.7)$$

and the gain introduced by the mixer matrix which, for a single axis, can be expressed as

$$K_{mixer} = \frac{1}{4K_t \Omega_H}, \quad (7.8)$$

it is possible to define the following transfer functions:

- Inner open-loop transfer function:

$$L_{inner}(s) = R_p(s) K_{mixer} G_p(s) G_{motor}(s) \quad (7.9)$$

- Inner sensitivity function:

$$S_{inner}(s) = \frac{1}{1 + L_{inner}(s)} \quad (7.10)$$

- Inner complementary sensitivity function:

$$T_{inner}(s) = \frac{L_{inner}(s)}{1 + L_{inner}(s)} \quad (7.11)$$

- Outer open-loop transfer function:

$$L_{outer}(s) = R_\phi(s) T_{inner}(s) \frac{1}{s} \quad (7.12)$$

- Outer sensitivity function:

$$S_{outer}(s) = \frac{1}{1 + L_{outer}(s)}. \quad (7.13)$$

- Outer complementary sensitivity function:

$$T_{outer}(s) = \frac{L_{outer}(s)}{1 + L_{outer}(s)}. \quad (7.14)$$

Figure 7.5-7.6 show the frequency responses of the the resulting sensitivity functions obtained after the  $H_\infty$  (local) optimal tuning.

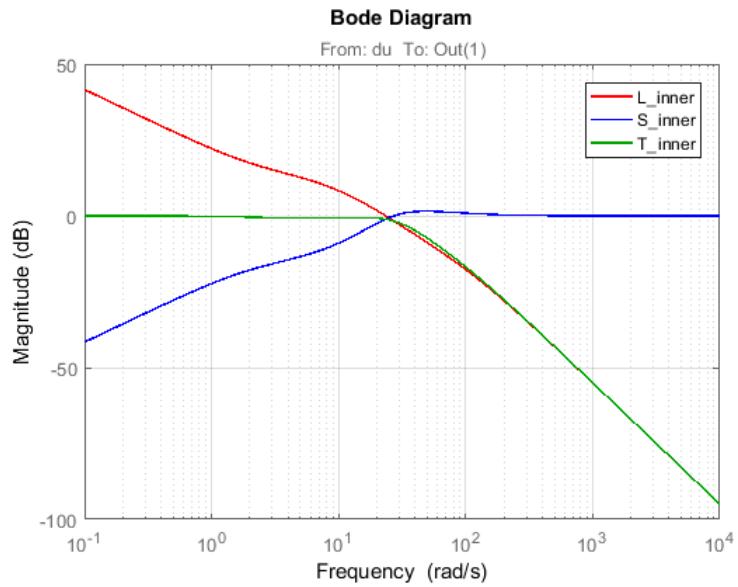


Figure 7.5: Inner-loop sensitivity functions

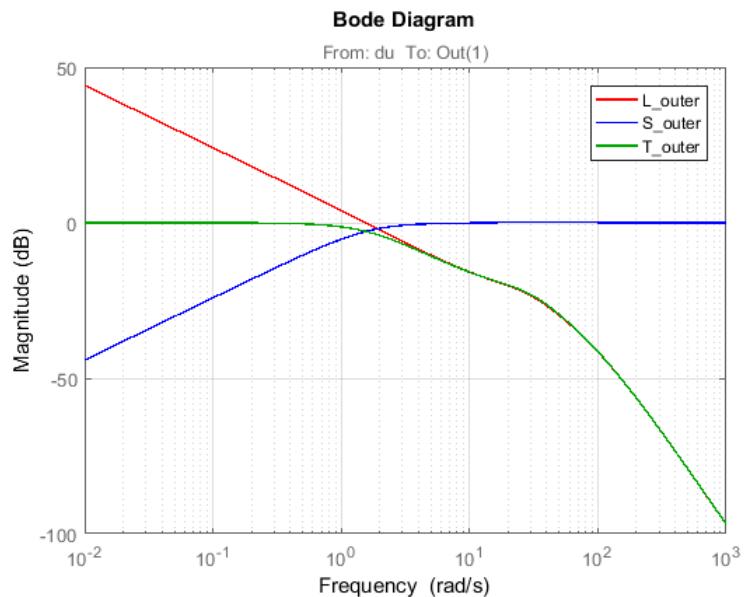


Figure 7.6: Outer-loop sensitivity functions

The resulting optimal parameters for both the inner PID and outer PD regulator are reported in Table 7.1. As reference, the standard tuning parameters obtained through the manual trial and error procedure, are also reported.

Controller parameter	Standard tuning	Optimal tuning
$K_{pp}$	0.58	0.3
$K_{ip}$	0.7	0.72
$K_{dp}$	0.075	0.006
$K_{p\phi}$	1.9	1.61
$K_{d\phi}$	0.005	0.058

Table 7.1: Optimal tuning parameters for the inner regulator  $R_p(s)$  and the outer regulator  $R_\phi(s)$

### 7.2.2 Translational control

In this section the design of the translational control system will be presented. This architecture will be in charge of controlling the translational dynamics in a decoupled way from the attitude of the quadrotor by actuating the tilting arms. The synthesis will be performed on the same assumptions already explained for the attitude control. Recalling the translational dynamics derived in Chapter 4,

$$\frac{d(mV_b)}{dt} = m \left( \frac{dV_b}{dt} + \omega_b \times V_b \right) = F_{ext} + F_g, \quad (7.15)$$

neglecting the gyroscopic effects and by rearranging it for a single axis ( $x_b$  axis of the quadrotor for the case in consideration), equation (7.15) becomes

$$\dot{u} = \frac{1}{m} F_x + F_{gx_b} = \quad (7.16)$$

$$= \frac{1}{m} [-K_t \sin(\alpha_2)\omega_2^2 + K_t \sin(\alpha_4)\omega_4^2]. \quad (7.17)$$

In equation (7.17), the gravitational term  $F_{gx_b}$  does not appear since we are assuming null attitude angles and thus no gravitational force component is acting on the specified axis, while  $F_x$  is the translational force generated by the tilting rotors. Equation (7.17) is then linearised around a near hovering

condition, that is,

$$\omega_i = \Omega_H \quad i = 1, \dots, 4$$

$$\delta\alpha_4 = \alpha_0 + \delta\alpha$$

$$\delta\alpha_2 = \alpha_0 - \delta\alpha.$$

The overall linearised translational dynamics is thus

$$\delta\dot{u} = \frac{1}{m} [-K_t\Omega_H^2\delta\alpha_2 + K_t\Omega_H^2\delta\alpha_4] = \quad (7.18)$$

$$= \frac{1}{m} 2K_t\Omega_H^2\delta\alpha. \quad (7.19)$$

Written in a transfer function form, equation (7.19) becomes:

$$G_u(s) = \frac{U(s)}{\alpha(s)} = \frac{2K_t\Omega_H^2}{m} \frac{1}{s}. \quad (7.20)$$

Figure 7.7 shows the block diagram of the translational control architecture.

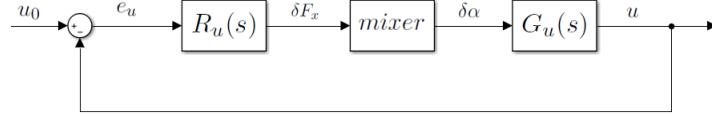


Figure 7.7: Translational dynamics closed-loop block diagram

In detail, for  $R_u(s)$  a PID controller has been chosen, which, expressed in its analytical form, is

$$R_u(s) = K_{pu} + K_{iu} \frac{1}{s} + K_{du} \frac{s}{1 + sT_f}, \quad (7.21)$$

while the block *mixer* express the gain introduced by the mixer matrix for the translational dynamics. It can be expressed as

$$mixer = \frac{1}{2K_t\Omega_H^2}. \quad (7.22)$$

In this case, since the structure of the test-bed does not allow to test translational dynamics, to obtain an initial guess of the tuning parameters for the  $H_\infty$  optimization, the controller has been firstly tuned on the analytical model presented in equation (7.20) and on the Simulink model, and secondly its performance has been improved with an in-flight trial and error procedure. It is important to highlight that the in-flight manual tuning is a risky

operation that could highly damage the quadrotor in case of a bad choice of the tuning parameters, since the analytical model used to obtain a first guess of the tuning parameters, does not take into account any aerodynamic force. To reduce this risk, the use of a more reliable identified translational model that also takes into account such forces could be used in the first stage of the tuning.

The trial and error procedure allowed to define the following  $H_\infty$  requirements:

- Tracking requirements: the closed-loop system (from  $u_0$  to  $u$ ) must be able to track reference signals within the 3 rad/s bandwidth.
- Steady state error: less than 0.0001%.
- Maximum peak error: less than 20%.
- Disturbance rejection: the closed-loop sensitivity function  $S(s, \theta)$  has been shaped like a second order high pass filter with 40 dB gain and a cutting frequency at 3 rad/s.

Results of the  $H_\infty$  synthesis are reported in Figure 7.8 and Figure 7.9.

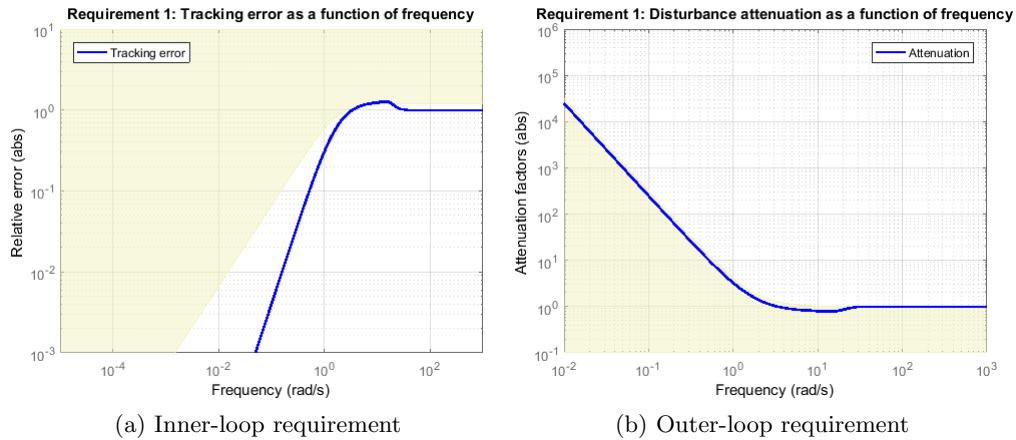


Figure 7.8:  $H_\infty$  requirements for translational dynamics

### 7.3 Attitude control law robustness analysis

The attitude control architecture has been tuned considering the nominal behaviour of the derived dynamical models. A mathematical model of any real system is always just an approximation of the true, physical reality of the system dynamics. Typical sources of the discrepancy include unmodelled (usually high-frequency) dynamics, neglected nonlinearities and system-parameter variations due to environmental changes and torn-and-worn factors. These modelling errors may adversely affect the stability and performances of the closed-loop system. For this reason is advisable to evaluate the robustness of the control architecture in front of system uncertainty. In the case of this thesis, the effects of the parametric uncertainty will be evaluated. A common way to describe unstructured uncertainties of the system is through the concepts of the additive and multiplicative representations.

A system with additive uncertainty is modelled as follows

$$G_a(s) = G_n(s) + W(s)\Delta(s), \|\Delta\|_\infty < 1, \quad (7.23)$$

while a system with multiplicative uncertainty is modelled as

$$G_m(s) = G_n(s)(1 + W(s)\Delta(s)), \|\Delta\|_\infty < 1, \quad (7.24)$$

where, in both cases  $G_n(s)$  represents the nominal system model,  $W(s)$  is a shaping function and  $\Delta(s)$  represents the normalized uncertainty.

In Figure 7.11.a and Figure 7.11.b, the red frequency response represents

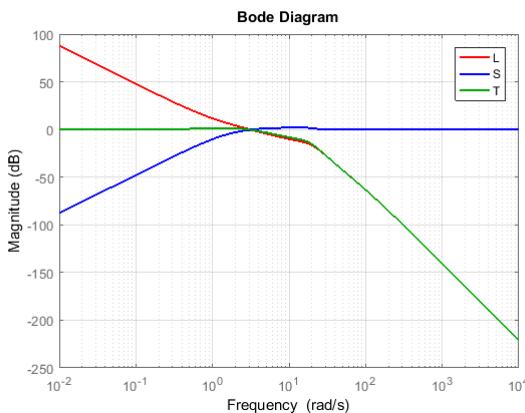


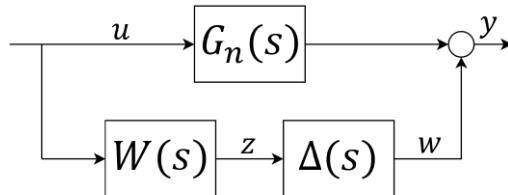
Figure 7.9: Translational dynamics sensitivity functions

the shaping filter that best approximates the maximum relative error between the nominal model and the sampled uncertain model in the case of additive uncertainty and multiplicative uncertainty respectively, while the dashed blue lines represent the relative gap from the nominal model of random sampled uncertain frequency functions.

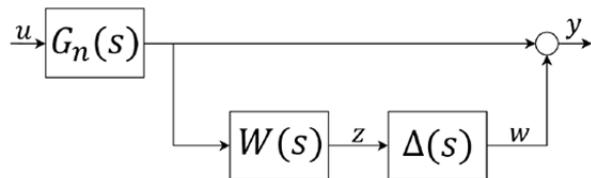
To evaluate the overall robustness of the system, a Monte Carlo simulation with 50 random sampled uncertain models has also been performed on the open-loop system  $L_{outer}(s)$ . As can be seen from Figure 7.12, the closed-loop system is not affected by the parametric uncertainty.

## 7.4 Implementation of the controllers in the Flight Controller Unit (FCU)

The FCU used in this thesis, has the main advantage that has been designed with the capability of integrating C++ Simulink auto generated code. This feature simplifies the actual implementation on the real system of the designed control architectures. Indeed, once the control architecture has been designed and tested in Simulink environment as introduced in Chapter 6, it is possible to automatically generate the relative code by specifying the



(a) Additive uncertainty



(b) Multiplicative uncertainty

Figure 7.10: System with uncertainty block diagrams

target hardware architecture (in our case an ARM Cortex micro controller running at 100Hz) and the preferred language (in our case C++).

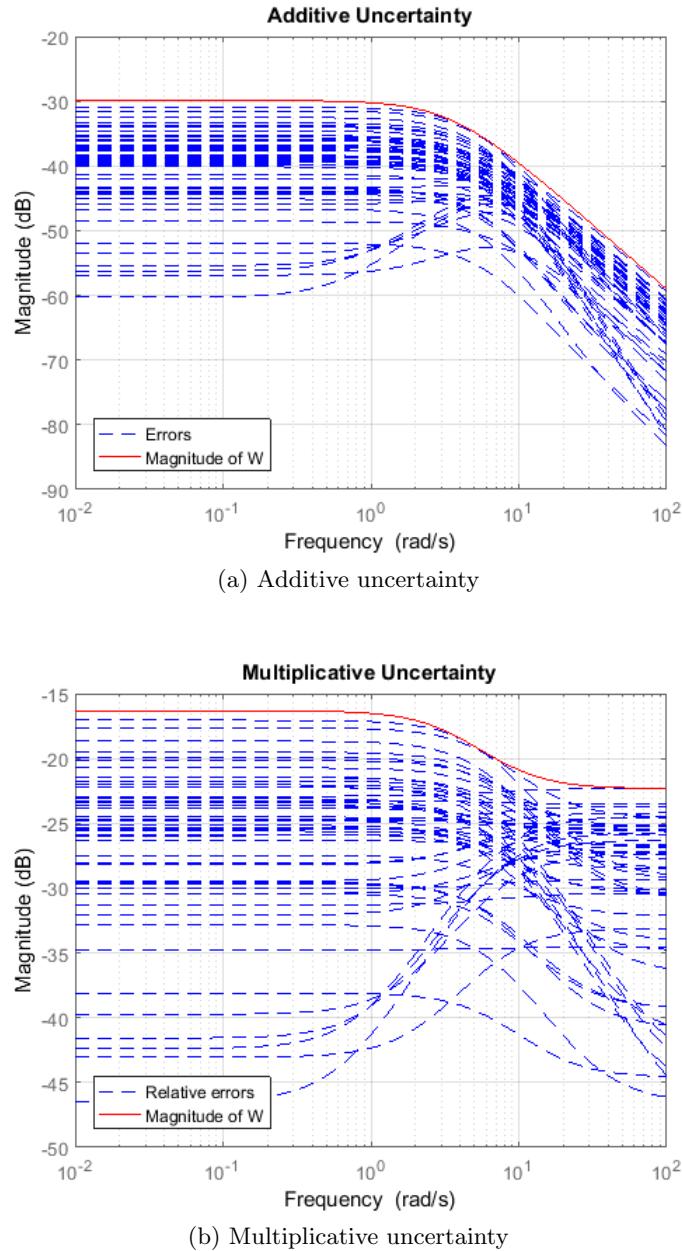


Figure 7.11: Frequency response of the relative gaps between uncertain models and nominal model vs. the shaping filter  $W(s)$

The actual developed Simulink control architectures are depicted in Figure 7.13 and Figure 7.14.

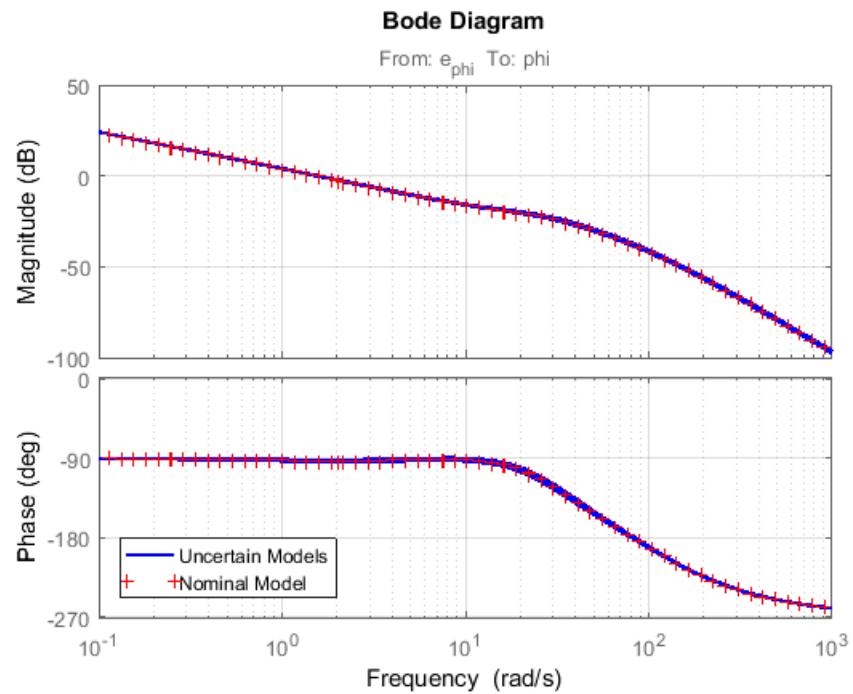


Figure 7.12:  $L_{outer}(s)$  stability robustness in front of plant uncertainty

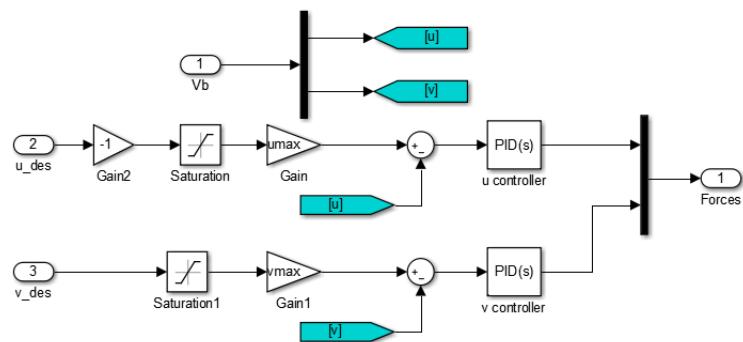


Figure 7.13: Simulink implementation of translational controllers

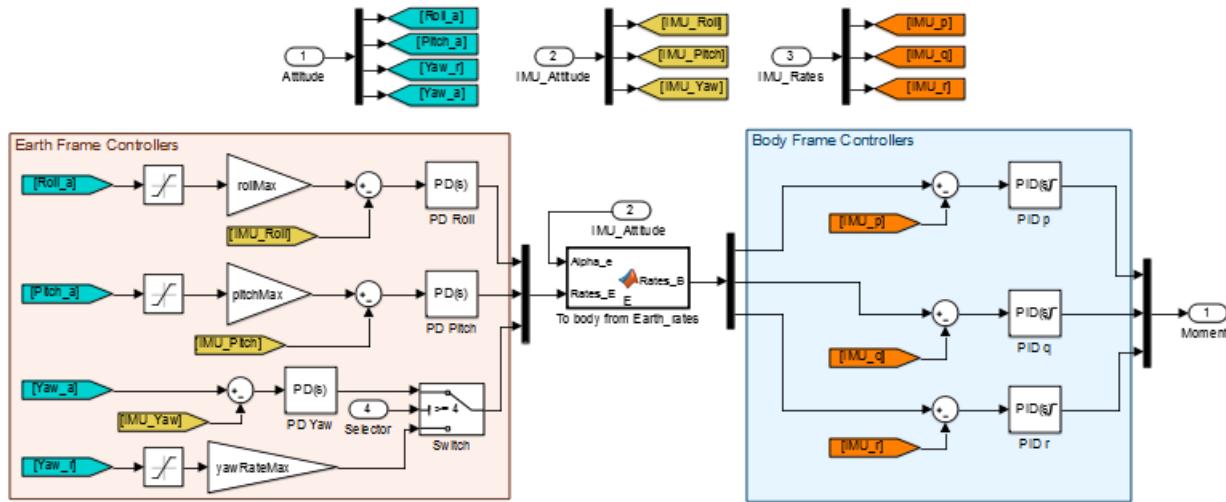


Figure 7.14: Simulink implementation of attitude controllers



## Chapter 8

# Experimental results

In this chapter the experimental campaign aimed at validating the proposed optimally tuned controllers will be presented.

More in detail, results obtained in the developed simulation environment and results obtained during testbed and in-flight experiments will be compared. The results are meant to illustrate the performances in terms of set-point tracking and disturbances rejection of the designed control system.

### 8.1 Testbed experiments

The first sets of conducted experiments have been performed with the quadrotor constrained to a single DoF on the testbed, allowing only roll attitude rotations.

During the experiments, the motors in charge of generating the desired angular displacement have been maintained at 45% throttle. This value has been obtained from in-flight tests and represents the actual thrust percentage that guarantees the rotors to generate a total thrust equal to the vehicle weight (hovering condition).

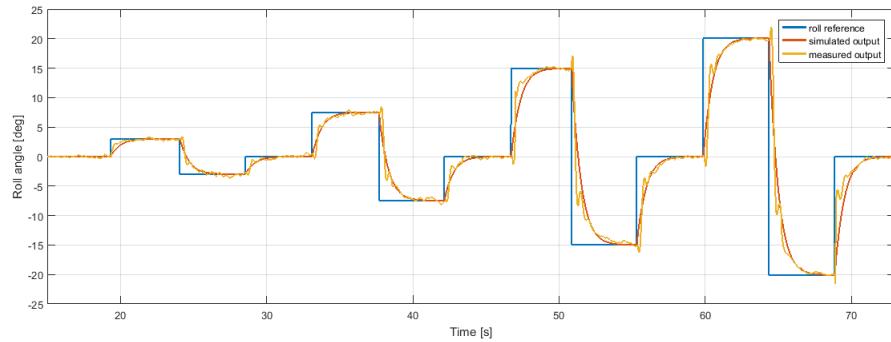
Moreover, all controllers are disabled except for the attitude roll one, that operates on motors 2 and 4 respectively.

#### 8.1.1 Reference tracking

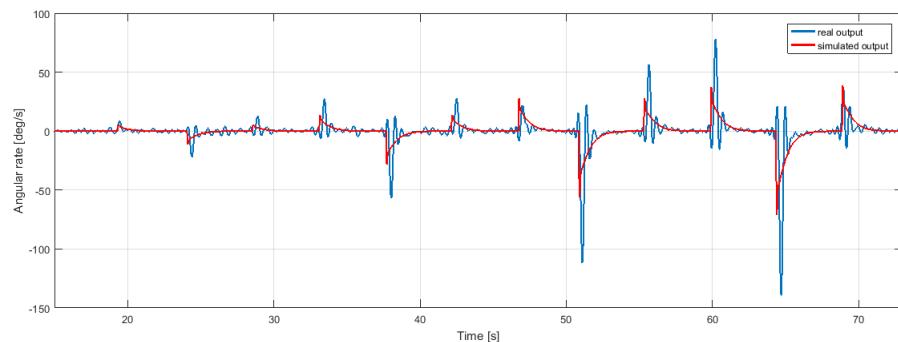
The performed tests consisted in assigning a desired roll angular set-point, testing different step amplitudes ranging from  $3^\circ$  to  $20^\circ$  of roll attitude.

As can be seen from Figure 8.1, the developed simulation environment provides a good estimation of the real quadcopter behaviour. The attitude model can be thus assumed validated. Moreover, from the reference tracking point of view, the experimental results confirm that requirements defined

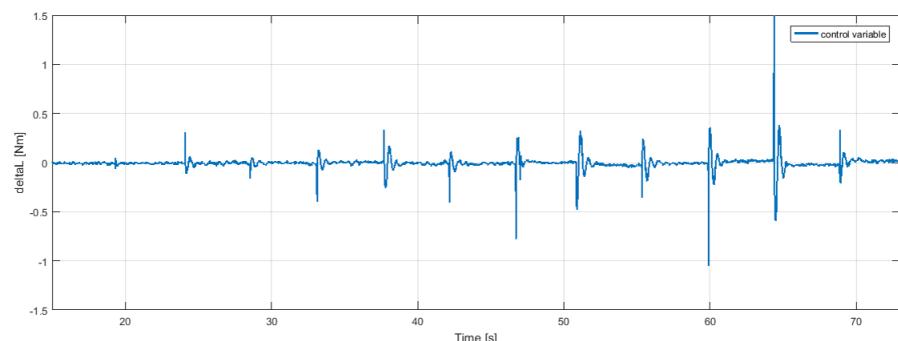
in the  $H_\infty$  synthesis are satisfied with a reasonable use of the control variable (saturation are never reached).



(a) Roll simulated output vs real output



(b) Angular rate simulated output vs in-flight real output



(c) Real control variable

Figure 8.1: Testbed experimental results

### 8.1.2 Disturbance rejection

Some tests have also been performed to evaluate the response of the designed control architecture in front of load disturbances. This kind of experiment is useful to evaluate the reaction of the vehicle in case of wind gusts in real case situations.

The experiments have been conducted requiring null roll attitude and by introducing a throttle percentage variation disturbance on motor 2, to simulate a load disturbance.

As can be seen from Figure 8.2, the robust controller is able to reject load disturbances of  $\pm 5\%$  and  $\pm 7\%$ , by suddenly recovering to the desired quadcopter attitude.

## 8.2 In-flight experiments

### 8.2.1 Attitude dynamics

To evaluate the robustness of the designed control system in a real case scenario (and not in a constrained, single DoF ideal case as in the testbed one) and also to check the validity of the simulated environment in in-flight situations, some free-constrained experiments have been performed.

For attitude control, the experiments consisted in giving an alternating roll setpoint while having the altitude control (to simplify pilot maneuvers) and attitude control activated. Figure 8.3 shows the obtained results.

As can be seen, even though the simulator performance is degraded compared to the testbed experiments, it can still provide good information about the quadcopter dynamics during an in-flight situation.

### 8.2.2 Translational dynamics

For what concerns the translational control, since the realisation of a suitable testbed that could constrain the quadrotor to a single translational DoF was not an easy task, only in-flight experiments have been collected. Also in this case, the tests aimed to validate the simulator model and to check the robustness of the designed control architecture. Figure 8.4, shows the results provided by the simulator compared with the real ones.

As can be seen, the simulated environment is able to provide trusty informations about the translational dynamics of the quadcopter during in-flight experiments.

To illustrate the augmented capabilities of the tilting rotor quadcopter over a traditional one, further experiments have been conducted. In a first test,

the quadrotor has been feed with a roll angle input reference while having the translational control disabled (traditional quadcopter behaviour). As can be seen from Figure 8.5, due to the coupled dynamics of the tradi-

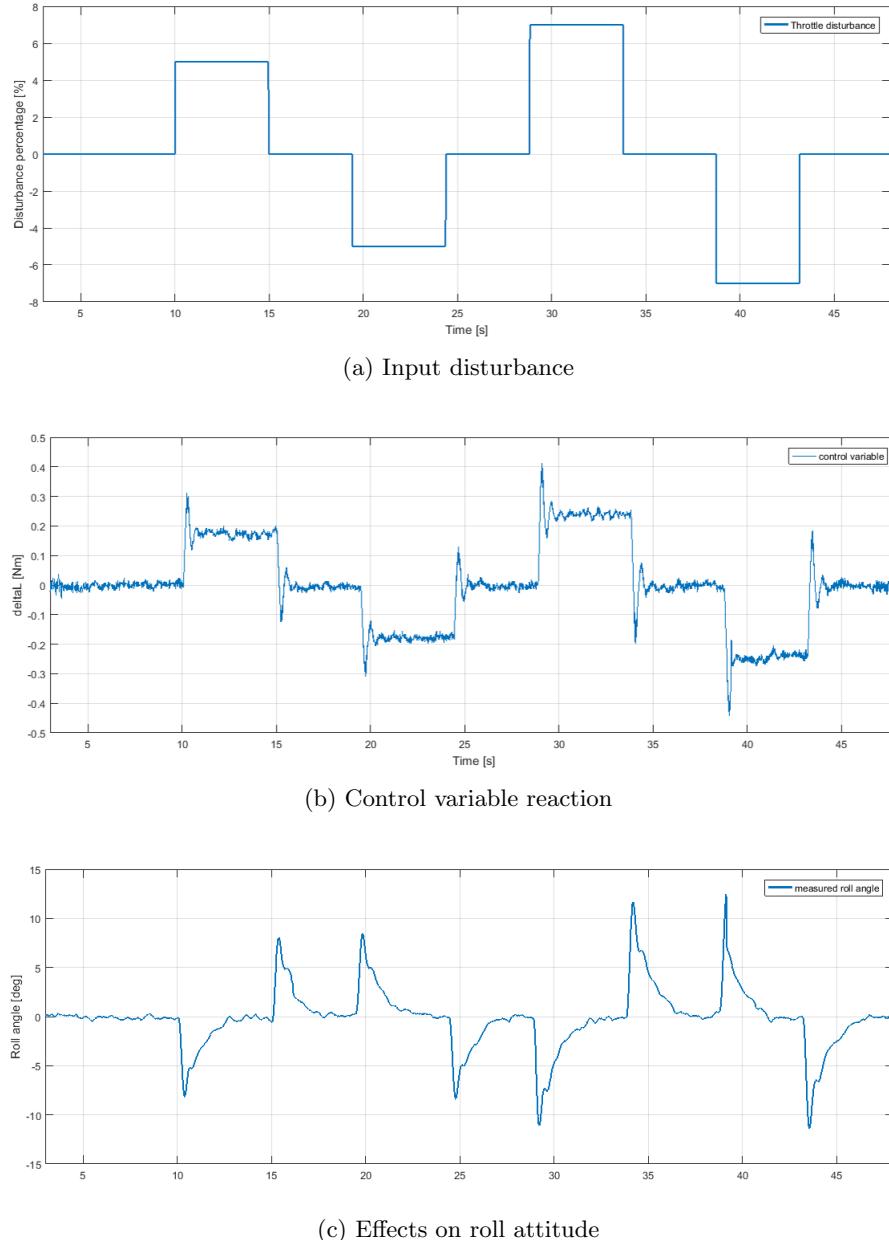


Figure 8.2: Disturbance rejection experimental results

tional quadcopter, a variation in the quadcopter attitude, produces also a displacement in its translational velocity. In a second test, the translational control has been enabled and a roll reference has been given while requiring null translational velocities. As can be seen from Figure 8.6, thanks to the enabled translational control, the quadrotor was able to track the desired

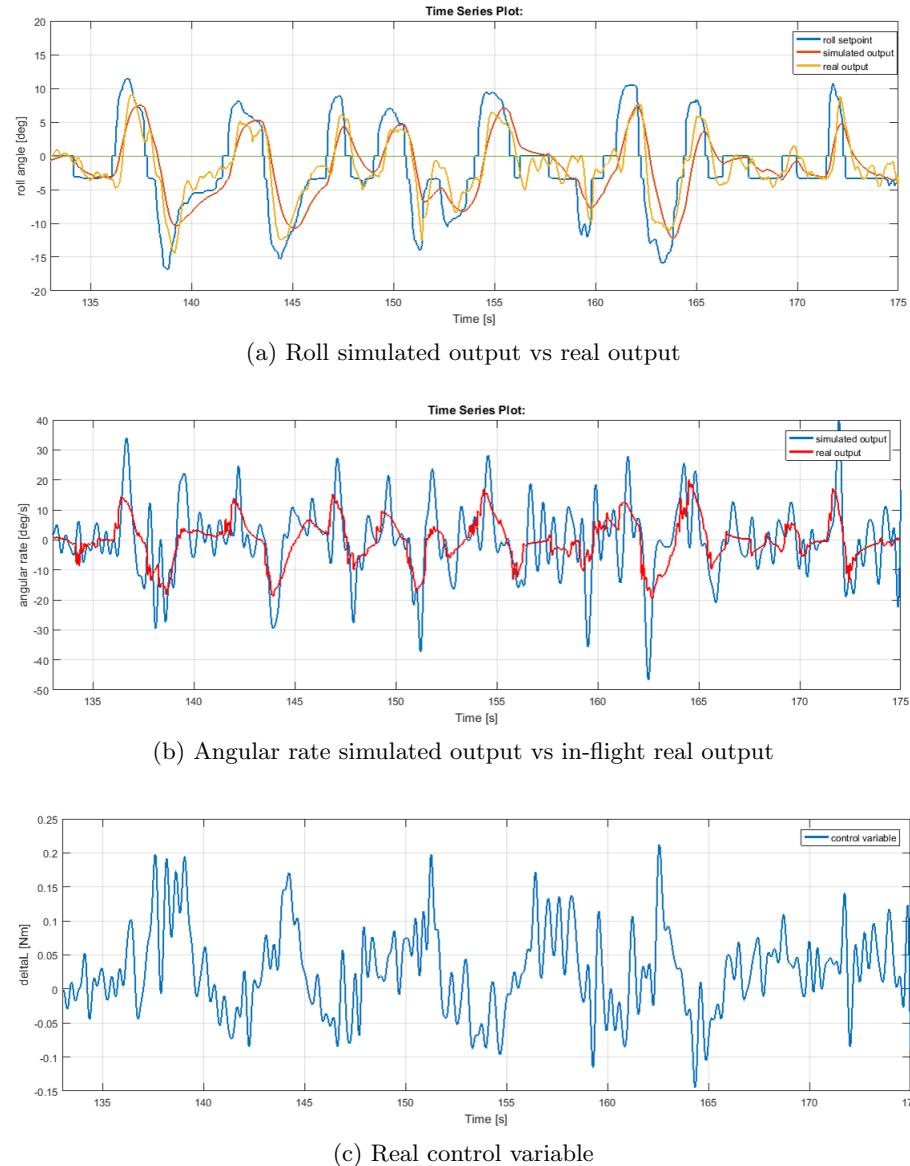


Figure 8.3: In-flight attitude experiments

angular roll reference while maintaining almost null translational velocity by tilting the arms 1 and 3 in order to compensate the translational force generated by the not null roll attitude. Equally, in Figure 8.7 the velocity setpoint tracking while requiring null attitude is depicted.

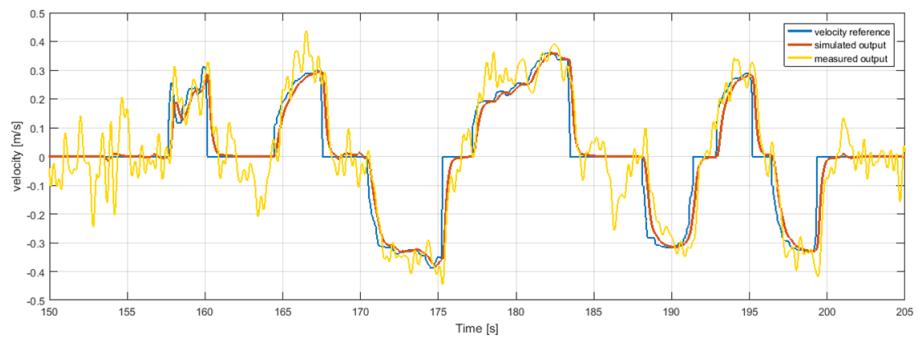


Figure 8.4: Translational simulated model vs real model comparison

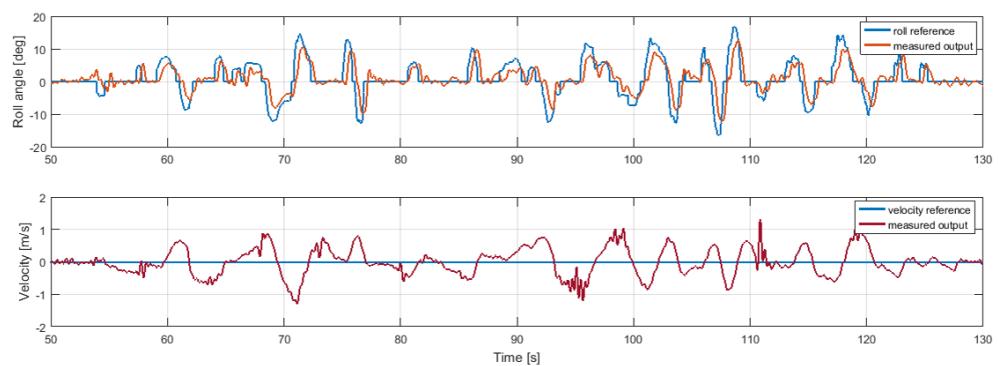
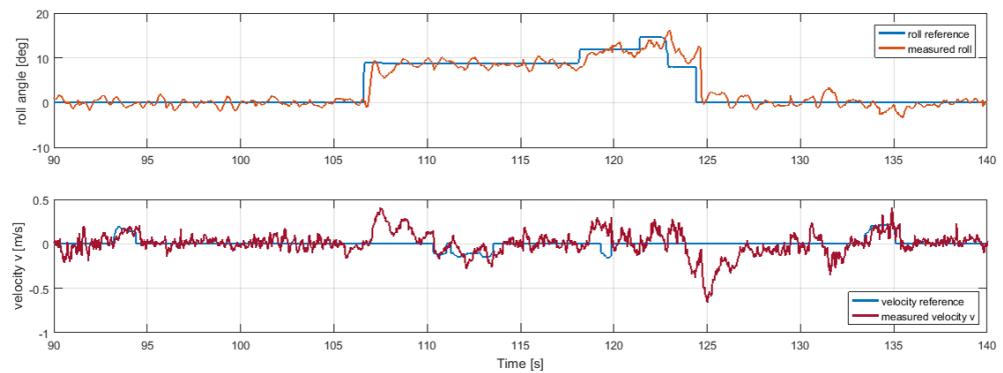
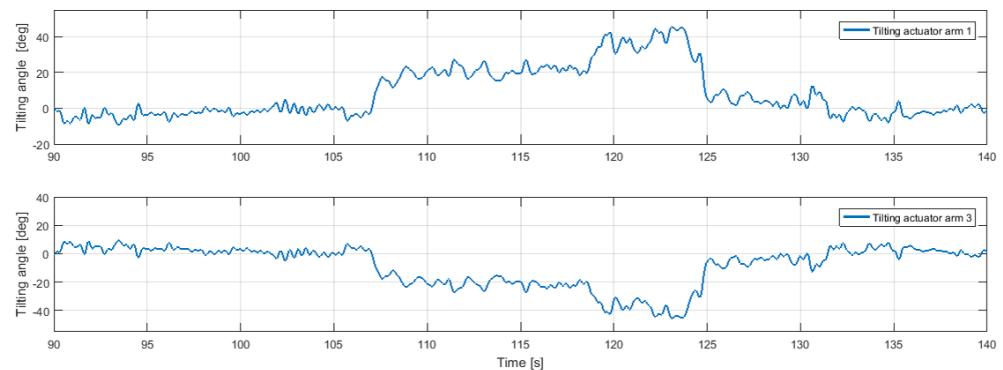


Figure 8.5: Coupled dynamics, translational control disabled

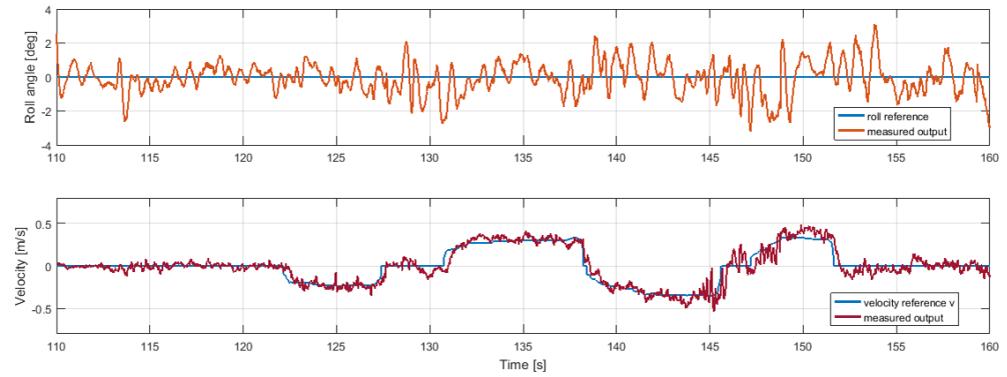


(a) Roll angle setpoint tracking while requiring null translational velocity  $v$

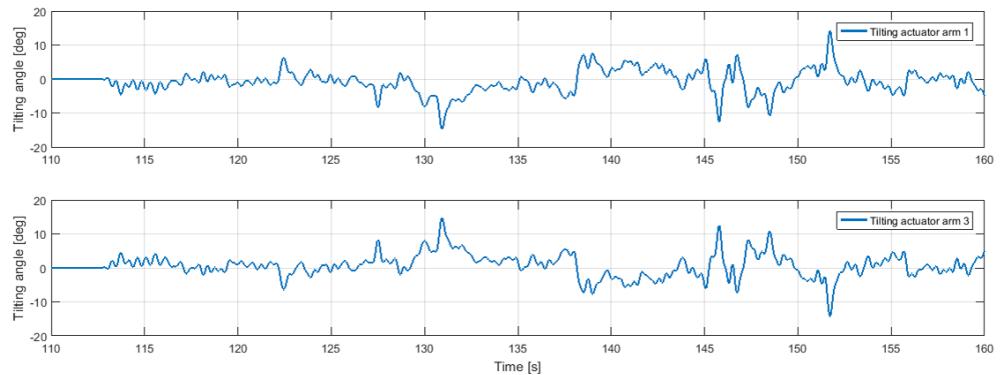


(b) Tilting actuators control variable

Figure 8.6: Decoupled dynamics, attitude setpoint tracking requiring null translational velocities



(a) Translational velocity v setpoint tracking while requiring null roll attitude



(b) Tilting actuators control variable

Figure 8.7: Decoupled dynamics, translational velocity setpoint tracking while requiring null attitude

# Chapter 9

## Conclusions

The main goal of this thesis work was the development of a novel tiltrotor quadcopter UAV. Starting from the prototype design, this work covered all the development phases that included the analytical model definition and identification, the design of a control architecture that allows the full control of the 6 DoFs of the vehicle in the space, an optimal controller parameters tuning through  $H_\infty$  synthesis and lastly experimental validation of the developed architecture.

More in detail, once the tiltrotor quadcopter mission requirements have been defined, a prototype frame has been designed and assembled (Chapter 2). In order to obtain a trustworthy system model, all the actuators used to build the tiltrotor quadcopter have been identified through different identification campaigns (Chapter 3).

The identified actuators models, have been then implemented in the overall analytical model of the quadcopter, obtained by applying classical mechanics principles (Chapter 4). To obtain a more reliable model of the attitude dynamics of the quadcopter, the model parameters have also been identified through an experimental identification campaign with two identification methods: Grey-est and SRIVC.

In particular, during the roll attitude dynamics identification, it was highlighted how the performances of the grey-box identification procedure, compared to the SRIVC approach, were highly affected by the intrinsic time delay of the system. Once the delay was identified through the SRIVC method, both the Grey-est and SRIVC approaches converged to the same identification result (Chapter 5.1.6).

Thanks to the obtained model, a control architecture able to manage, in a decoupled way, all the 6 DoFs has been designed and evaluated in a simulation environment.

The designed system has been finally validated through an experimental campaign which took advantage of both testbed constrained experiments and in-flight experiments.

Experimental results highlighted how, thanks to the designed control architecture, the craft has reached the full control of the 6 DoFs, being able not only to translate while maintaining null attitude, but also to modify its attitude while maintaining its position as depicted in Figure 9.1.

As introduced at the beginning of this thesis, the tiltrotor quadcopter augmented motion capabilities, can be exploited in the UAVs industry to overcome the traditional multirotor helicopters motion limitations.

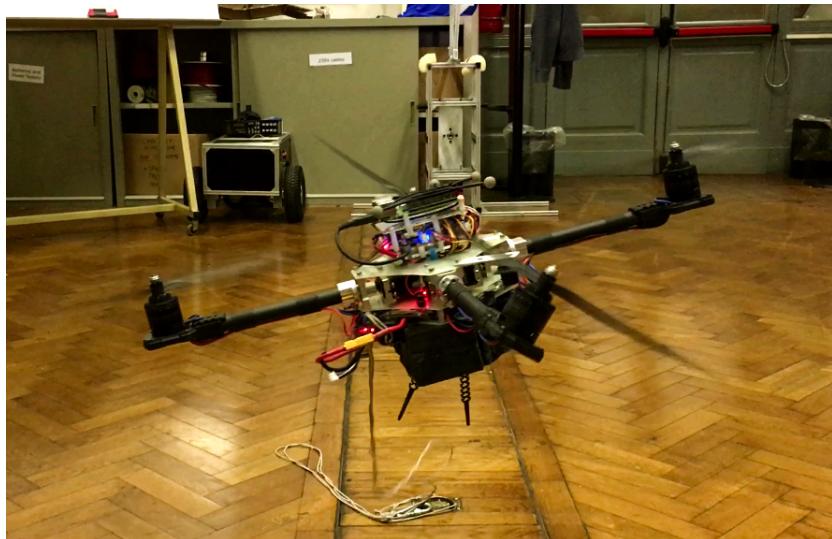


Figure 9.1: Tiltrotor quadcopter performing a decoupled dynamics maneuver

## 9.1 Further developments

Here are reported some further developments that could increase the performances of the developed architecture:

- Improve the tiltrotor quadcopter actuation system in order to reduce the overall vibrations transmitted to the FCU.
- Introduce in the analytical model translational aerodynamics forces to increase the simulator reliability.

- Exploit the overactuated tiltrotor quadcopter structure to overcome rotor faults.
- Exploit the overactuated tiltrotor quadcopter structure to increase the efficiency of the vehicle by optimizing resources usage.
- Migrate the control architecture to a more powerful control unit with better onboard sensors and computational power such that operations like visual odometry can be performed directly on the vehicle FCU.



# Bibliography

- [1] A. Kulhare A. B. Chowdhury and G. Raina. Back-stepping control strategy for stabilization of a tilt-rotor UAV. *24th Chinese Control and Decision Conference (CCDC)*, pages 3475 – 3480, ,2012.
- [2] M. Migliavacca A. Bonarini, M. Matteucci and D. Rizzi. R2p: An open source hardware and software modular approach to robot prototyping. *Robotics and Autonomous Systems*, 62(7):1073–1084, 2014.
- [3] F. Cunha A. Moutinho, E. Mateos. The tilt-quadrotor: concept, modeling and identification. *IEEE International Conference on Autonomous Robot Systems and Competitions*, 2015.
- [4] E. N. Johnson B. L. Stevens, F. L. Lewis. *Aircraft Control and Simulation: Dynamics, Controls Design, and Autonomous Systems - Third edition*. Wiley, 2015.
- [5] Randal W. Beard. Quadrotor dynamics and control. *Brigham Young University, Quadrotor Team Topgun*, Jun 01, 2015.
- [6] S. Bouabdallah and R. Siegwart. Design and control of a miniature quadrotor. *Advances in Unmanned Aerial Vehicles*, page 171–210, 2007.
- [7] Dr. Ing. Zzung Yoon Dipl. Ing. Karsten Grojekatthöfer. Introduction into quaternions for spacecraft attitude representation. *Departement of Astronautics and Aeronautics, Berlin, Germany*, May 31, 2012.
- [8] N. dos Santos Fernandes. Design and construction of a multi-rotor with various degrees of freedom. Master’s thesis, Instituto Superior Técnico, Universidade Técnica de Lisboa, 2011.
- [9] P. Panizza F. Riccardi and M. Lovera. Identification of the attitude dynamics for a variable-pitch quadrotor uav. *40th European Rotorcraft Forum, At Southampton, UK*, 2014.

- [10] Matteo Ferronato. Identificazione e controllo della dinamica verticale di un elicottero quadrirotore. Master's thesis, Politecnico di Milano, 2015.
- [11] G. F. Trommer G. Scholz. Model based control of a quadrotor with tilttable rotors. *Gyroscopy and Navigation*, 7:1:72–81, 2016.
- [12] H. Gamier and P. Young. Time-domain approaches to continuous-time model identification of dynamical systems from sampled data. *Proceeding of the 2004 American Control Conference*, Boston. Massachusetts June 30 - July 2, 2004.
- [13] Hugues Garnier. Direct continuous-time approaches to system identification. overview and benefits for practical applications. *European Journal of Control*, 24:50–62, 2015.
- [14] Mattia Giurato. Design, integration and control of a multirotor uav platform. Master's thesis, Politecnico di Milano, 2015.
- [15] Tong Heng Lee Guowei Cai, Ben M. Chen. *Unmanned Rotorcraft Systems, Chapter 2*. 2011.
- [16] O. Cervellin. H. Garnier, M. Gilson. Latest developments for the matlab con- tsid toolbox. *IFAC. 14th IFAC Symposium on System Identification (SYSID'2006)*, pages 714–719, Newcastle, Australia. Elsevier, 2006.
- [17] H. H. Bülthoff M. Ryll and P. Giordano. A novel overactuated quadrotor unmanned aerial vehicle: Modeling, control, and experimental validation. *IEEE Transactions on Control Systems Technology*, Volume 23:540 – 556, March 2015.
- [18] A. Nemati and M. Kumar. Modeling and control of a single axis tilting quadcopter. *American Control Conference*, Portland, OR, 2014.
- [19] R. Mahony P. Pounds and P. Corke. *Modelling and control of a large quadrotor robot - Control Engineering Practice*, volume 18(7):691-699. 2010.
- [20] Raymond W Prouty. *Helicopter performance, stability, and control*. 1995.
- [21] Fabio Riccardi. *Model identification and Control of a variable pitch quadrotor UAVs*. PhD thesis, Politecnico di Milano, Departement of Aerospace Science and Technology, 2015.

- [22] P. Rocco. Kinematic redundancy - control of industrial robots, course slides.
- [23] R. Vaidyanathan S. Madgwick and A .Harrison. Automated calibration of an accelerometers, magnetometers and gyroscopes-a feasibility study. 2010.
- [24] Sciavicco L. Villani L. Oriolo G. Siciliano, B. *Robotics - Modelling, Planning and Control*. 2009.
- [25] B. L. Stevens and F. L. Lewis. *Aircraft Control and Simulation*. John Wiley & Sons, Inc., 2nd, 2003.
- [26] R. Kitayoshi I. Maruta T. Wada, M. Ishikawa and T. Sugie. Practical modeling and system identification of r/c servo motors. *18th IEEE International Conference on Control Applications Part of 2009 IEEE Multi-conference on Systems and Control*, Saint Petersburg, Russia, July 8-10, 2009.
- [27] Kimon P. Valavanis, editor. *Advances in Unmanned Aerial Vehicles - State of the Art and the Road to Autonomy*. Springer, 2007.
- [28] John Watkinson. *Art of the Helicopter*. Butterworth-Heinemann, 2003.