POLITECNICO DI MILANO
DEPARTMENT OF AEROSPACE SCIENCE AND TECHNOLOGY
DOCTORAL PROGRAMME IN AEROSPACE ENGINEERING

DESIGN, INTEGRATION AND CONTROL OF MULTIROTOR UAV PLATFORMS

Doctoral Dissertation of:
Mattia Giurato

Supervisor:
Prof. Marco Lovera

Tutor:
Prof. Pierangelo Masarati

The Chair of the Doctoral Program:
Prof. Pierangelo Masarati

2019 – XXXII

Mattia Giurato
Dipartimento di Scienze e Tecnologie Aerospaziali
Politecnico di Milano
Via Giuseppe La Masa, 34
20156 Milano, Italy
E-mail: mattia.giurato@polimi.it

Al viso di ogni Nonno che proietta amore.

Acknowledgments

Ed eccoci qui, alla fine di un lungo percorso. Che poi visto sotto un altro punto di vista non è una fine, ma l'inizio di qualcosa di nuovo, ancora tutto da scrivere.

Eppure arrivare fino a questo punto non è stato facile, e mi è costato davvero tanto. Ovviamente non si parla in termini economici, ma piuttosto di energie, fatica, salute e rapporti umani.

In questi ultimi anni ho avuto la fortuna di dedicare la mia vita alla mia passione, e di trovare quindi la forza di alzarmi e lavorare perchè spinto da una forza che pochi possono capire. Eppure i momenti in cui avrei voluto prendere e mandare tutto all'aria ci sono stati, e non sono stati pochi.

In questi ultimi anni la mia vita è stata totalmente stravolta. Ho avuto la fortuna di conoscere un sacco di gente nuova, di amici, di persone che credono in me fermamente e che mi hanno spronato nei momenti più difficili. Ho avuto anche la sfortuna di perdere dei capisaldi della mia vita, dei punti di riferimento, e di dovermi dunque ridimensionare da zero, ma sono questi i momenti che più di ogni altra cosa ci permettono di crescere e di andare avanti.

Non voglio rispolverare il solito cliché per cui nei ringraziamenti bisogna citare una lista infinita di nomi di persone che in un modo o nell'altro hanno significato qualcosa per te. Nel mio piccolo, voglio solo ringraziare

tutte le persone che in questi anni mi hanno permesso di sorridere, di ubriacarmi, di avermi fatto ridere a squarciagola, di avermi fatto perdere ore di sonno, di avermi fatto superare limiti, di avermi fatto scoprire cose nuove, di avermi fatto ballare, di avermi fatto superare paure e che più di tutto mi abbiano insegnato qualcosa. Tutte queste persone rimarranno per sempre incise in maniera indelebile nella mia vita, e la mia unica speranza è in qualche modo di aver lasciato qualcosa dentro di loro che valga la pena di essere ricordato.

Infine gli unici ringraziamenti che mi sento di dedicare esplicitamente sono quelli rivolti alla mia Famiglia, che mi ha permesso di diventare l'uomo che sono, che ha sempre creduto in me, nelle mie capacità e nelle mie decisioni circondandomi sempre di affetto sincero.

E più di ogni altra cosa vorrei dedicare il risultato di queste mie fatiche a mio Nonno, che sono sicuro stia seguendo le mie avventure seduto tra i posti migliori. Ogni giorno la mia speranza è quella di essere diventato l'uomo che lui avrebbe voluto, che lo avrebbe reso fiero di me.

Abstract

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

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

In spite of the very large number of projects aimed to the design and development of multirotor platforms currently ongoing, it seems that no systematic approach to the problem has been proposed, in the sense of a consistent process flowing from mission requirements to actual platform implementation. This is quite surprising as the currently available components would enable a very rapid deployment of almost arbitrary configurations, following a dedicated conceptual and preliminary design and optimization stage. The main goal of this thesis is to present a systematic approach in order to realize a general UAV starting from the mission requirements. A deep analysis on the multirotor modelling, including non

conventional platforms, on the mechanical and aerodynamic aspects results then mandatory in this thesis.

Moreover, the dissertation covers the aspects of the attitude and position controller, describing both linear and nonlinear approaches.

Finally an application involving two multirotor UAVs which could be adopted in the future for the range extension of the multirotors in critical operations (*e.g.*, reconnaissance, patrolling, search and rescue, *etc.*) will be presented.

Contents

List of Figures

List of Tables

Introduction

An Unmanned Aerial Vehicle (UAV) is an aircraft without a pilot aboard, which is able to fly autonomously or could be piloted remotely from the ground. Usually called *drones*, in recent years these vehicles have met great interest both in civil and military fields thanks to their wide range of applications, including precision agriculture, photography, patrolling and surveillance, search and rescue, entertainment, product delivery, aerial inspection and many others. Furthermore, these platforms are low-cost test beds to verify many control approaches. When referring to drones, one usually refers to the category of multi-rotor Vertical Take-Off and Landing (VTOL) vehicles of small/medium size provided with a number of rotors greater than two and remotely controlled. In the simplest configuration, the motion is controlled by adjusting the angular speed of each propeller which, in turn, changes the thrust and torque generation. This mechanical simplicity has established the success of platforms such as the co-planar rotors configuration, both in commercial and research applications.

This thesis has the main purpose to contribute on the multirotor modelling both for the conventional platforms (*i.e.*, quadrotors) and the non-conventional ones (*i.e.*, thrust vectoring multirotors). It is possible to exploit two different modelling approaches for the multirotors: control-oriented and simulation-oriented ones. Control-oriented modelling re-

sults in a simpler representation of the multirotor's dynamical behaviour, while on the other hand simulation-oriented modelling includes effects (*i.e*., dynamical aerodynamic effects on the propellers) which are usually neglected in the control design. Particular attention is also given to the ground effect. In the literature very few publications which study such effect on the multirotor UAVs are present and but it is presented with the classical models from the rotorcraft background that have been proven not to be reliable on the small scale propellers of the multirotors taken into account. In this thesis an extensive campaign aimed to identify and model the ground effect on the multirotors is presented.

Despite the large number of ongoing projects, no systematic approach to the design and realization of these platform has been proposed, in the sense of a consistent process flowing from mission requirements to actual platform implementation. This is quite surprising as the currently available components would enable a very rapid deployment of almost arbitrary configurations, following a dedicated conceptual and preliminary design and final optimization stage. At present, the most used approaches to select the drive components are based on previously built multirotor configurations and the component data notified by the manufacturer, which may be inaccurate. These design procedures may shift away from the actual requirements that the designed vehicle must satisfy. So, an appropriate design method should be developed to cover all the performance requirements of the built vehicle. Currently, a variety of design methods have been proposed by several academics. A design process to select the drive components that makes use of Blade Element Momentum Theory is proposed by [1], but it does not perform an optimization regarding performance objectives. [2] documents a parametrization technique to estimate the masses of the drive components and a design tool to size the propulsion system of a multirotor UAV. This thesis makes use of the same mass estimation process combined with BEMT and electrical parametrization techniques reported by [3] and [4] to develop a design and optimization tool for multirotor UAVs, which mainly eliminates the need of a database for the drive components and contains the capability to optimize a configuration that takes into account the performance requirements for the vehicle in-design.

The second goal of this thesis is to contribute to the development of methods and tools to support the design process of multirotor UAVs. In particular two approaches will be presented: the former can provide an estimation of the mission performance given the description of the multirotor's components, the latter instead can provide a component's parameters list given the mission requirement. For this purpose, a complete description of the multirotor UAVs modelling is provided. In particular, the dynamical model which describes the behaviour of three multirotor UAVs is presented. Moreover, the aerodynamics of the multirotor is taken into account in order to improve the model of the UAV. Nevertheless, the electrical description of the components which constitute the UAV is presented and considered in the design process.

An important element in the design process of the multirotor platforms is the control architecture. In this thesis are then presented some innovative results in the advanced control architecture of multirotor UAVs. First and foremost the linear controllers for the attitude and position dynamics are taken into account. The results on the inversion-based methods (introduced by [5]) are extended and compared with the classical hierarchical approach (*i.e.*, cascaded PIDs) which is the most common approach adopted in the commercial platform. Consequently, given the non-linear nature of the multirotor UAVs dynamics, the limitations of the linear control approaches are clear. Non-linear controllers result as a solution to overcome such limitations. The first nonlinear controller is the adaptive augmentation of the attitude control system for a multirotor UAV. The presented approach allows to seamlessly combine a linear robust controller with an adaptive one and to disable or enable the adaptive controller when needed, in order to take the advantages of both the controllers. Furthermore, the proposed architecture allows not only to make use of a robust controller, but any generic baseline controller that guarantees stability of the closed-loop system in nominal conditions. Adaptive control is the natural candidate to face the external disturbances (*e.g.*, actuator degradation, actuator faults, *etc.*) and uncertainties because of its ability to provide high performance tracking in presence of uncertainties.

The other control techniques presented in the thesis have been specifically designed for thrust vectoring multirotors. Such platforms can achieve

a larger degree of actuation compared to coplanar multirotor UAVs since both thrust and torque can be oriented within the airframe. This feature makes thrust-vectoring UAVs capable of performing complex full-pose manoeuvres, which is particularly attractive for inspection-like applications that may require, for instance, navigation in a constrained environment. Moreover, being able to deliver both force and torque in any direction enhances the UAV interaction capabilities with the environment, which is especially desirable in aerial manipulation tasks. These fully actuated call for new methods to control them efficiently and to reliably cope with the added complexity of the platforms and of the larger set of tasks in which they can be employed, when compared to standard collinear multirotors. For this reason two full-pose control algorithms have been presented and tested in this thesis.

Finally, an innovative application of the multirotor UAVs is presented: the air-to-air automatic landing. When surveillance, reconnaissance and search-and-rescue missions are considered, these small-scale UAVs suffer of low endurance, being usually powered by batteries. For this reason, to extend mission endurance a drone can be used as carrier for lighter and smaller drones (followers) that can take-off from and land on it. This procedure requires the interaction between the two UAVs. Recent research activities deals with the interaction between more aircraft during flight, especially thinking about formation flight and air-to-air refuelling, studying the possibility to remotely command many drones together, following the same path or performing many tasks at the same time. In this thesis a new set of guidance, navigation and control laws enabling air-to-air UAV landing, are developed and validated. In particular the thesis focuses on the design of guidance laws aimed at providing a small multirotor with a reference descent trajectory.

The multirotors described in this thesis (*i.e.*, QUAD-X, TILT-X, ANT-1, ANT-R, CARRIER-1) have been developed adopting the above cited tools from the Author with the joint effort of the *M.Sc.* students who have been involved in the activities of the Aerospace System and Control Laboratory. In particular, the developed design approaches result to be reliable also for the design of unconventional platforms. Exception is made for the FAST-Hex which has been initially designed and built by Markus

Ryll and Davide Bicego (see [6]) in which the Author contributed in the final hardware design, parameter estimation and software implementation in order to make the experimental campaign possible.

Structure and contributions

The thesis is organized as follows:

- Chapter 1 presents the basic concepts adopted for the UAV modelling. Moreover the model of three different platforms, *i.e.*, QUAD-X, TILT-X and FAST-Hex, are described.
- Chapter 2 is a review of the fundamentals of rotor aerodynamics from the literature. Such effects are usually neglected in the UAV modelling since the multirotors are mostly studied in hovering conditions and in the literature few works adopt them. Nevertheless it is proven that this kind of effects are present in many flight conditions and considering them in the modelling oriented to the simulations may lead to more accurate results.
- Chapter 3 presents the results obtained in an analysis and research on the ground effect which is acting on the multirotor propellers while flying in proximity of the ground. The main classic models adopted in the literature are presented and the obtained experimental results are described in detail.
- Chapter 4 describes two approaches which have been developed aimed to the design of the multirotors. The presented algorithms help the design of the multirotor UAVs starting from the mission requirements providing then a list of parameters for the components sizing and *vice-versa*.
- Chapter 5 is a review of two linear position and attitude control schemes. In particular, the first one is based on the hierarchical control loops logic, which is widely adopted on most of the commercial UAVs. The second one instead, is based on the model-inversion approach which is more common in the rotorcraft field. Both control techniques are presented and validated on-flight under laboratory conditions.

- Chapter 6 presents the results on the nonlinear control of multirotor UAVs. The first achievement is related to the adaptive attitude control which can handle both the model uncertainties and the disturbance rejection. Then, two nonlinear full-pose control techniques for thrust vectoring UAV platforms are described in detail. All the presented nonlinear controllers have been tested in dedicated experimental campaigns.
- Chapter 7 presents an application involving two interacting UAVs: the air-to-air automatic landing. In particular, the horizontal synchronisation and the relative vertical motion estimation between the follower and the target drone are described and tested on an experimental campaign.

Published works and M.Sc. theses

Part of the material presented in this dissertation has been published in the following works:

1. A. Rivolta, M. Giurato, F. Cuzzocrea, F. Rovere, and S. Farí. *Low Cost MEMS IMU Calibration For Aerospace Student Activities*. 1st Symposium On Space Educational Activities, 2015;
2. M. Giurato, M. Lovera. *Quadrotor attitude determination: A comparison study*. 2016 IEEE Conference on Control Applications (CCA), Buenos Aires, Argentina, 2016;
3. D. Del Cont Bernard, M. Giurato, F. Riccardi, M. Lovera, *Ground effect analysis for a quadrotor platform*. 4th CEAS Specialist Conference on Guidance, Navigation & Control, Warsaw, Poland, 2017;
4. F. Haydar, M. Giurato, M. Lovera, G. Sechi. *Very high accuracy attitude determination for LOS steering*. 10th International ESA Conference on Guidance, Navigation & Control Systems, Salzburg, Austria, 2017;
5. A. Russo, D. Invernizzi, M. Giurato, M. Lovera. *Adaptive augmentation of the attitude control system for a multirotor UAV*. 7th Euro-

pean Conference for Aeronautics and Space Sciences, Milano, Italy, 2017;

6. D. Di Bacco, M. Giurato, M. Lovera. *Accurate positioning of multirotor UAVs for civil infrastructure monitoring*. 7th European Conference for Aeronautics and Space Sciences, Milano, Italy, 2017;

7. D. Del Cont Bernard, F. Riccardi, M. Giurato, M. Lovera. *A dynamic analysis of ground effect for a quadrotor platform*. 20th IFAC World Congress, Toulouse, France, 2017;

8. M. Giurato, S. Panza, M. Lovera. *Robust filtering for very high accuracy attitude determination*. 58th Israel Annual Conference on Aerospace Sciences, Tel Aviv and Haifa, Israel, (accepted), 2018;

9. D. Invernizzi, M. Giurato, P. Gattazzo, M. Lovera. *Full pose tracking for a tilt-arm quadrotor UAV*. IEEE Conference on Control Technology and Applications, Copenhagen, Denmark, 2018;

10. G. Bressan, A. Russo, D. Invernizzi, M. Giurato, S. Panza, M. Lovera. *Adaptive augmentation of the attitude control system for a multirotor UAV*. Journal of Aerospace Engineering, 2018;

11. D. Del Cont Bernard, F. Riccardi, M. Giurato, M. Lovera. *Ground Effect Analysis for a Quadrotor Platform*. Advances in Aerospace Guidance, Navigation and Control, Springer, 2018;

12. P. Giuri, A. Marini Cossetti, M. Giurato, D. Invernizzi, M. Lovera. *Air-to-air automatic landing for multirotor UAVs*. 5th CEAS Conference on Guidance, Navigation and Control, Milano, Italy, 2019;

13. G. Roggi, M. Giurato, M. Lovera. *A computer vision line-tracking algorithm for UAV GNSS-aided guidance*. XXV AIDAA, International Congress, Roma, Italy, 2019;

14. M. Giurato, P. Gattazzo, M. Lovera. *UAV Lab: a multidisciplinary UAV design course*. 21st IFAC Symposium on Automatic Control in Aerospace, Cranfield, UK, 2019.

Moreover, the Author has been involved in a number of *M.Sc.* theses as a tutor and co-advisor. Much of the work presented in this dissertation has been obtained as the result of the joint effort between the Author and the master students, and has been the basis for further developments. The Author's contributions are clarified in the following:

1. M. Ferronato. *Identificazione e controllo della dinamica verticale di un elicottero quadrirotore*. Automation and Control Engineering, *M.Sc.*, 2016;
2. C. Micheli. *Design, identification and control of a tiltrotor quadcopter UAV*. Automation and Control Engineering, *M.Sc.*, 2016;
3. A. De Angelis, A. Sorbelli. *Software architecture, estimators and control for multirotor UAVs*. Computer Science and Engineering, *M.Sc.*, 2016;
4. D. Di Bacco. *Distance control from vertical surfaces of a multirotor UAV designed for structural health monitoring of civil infrastructures*. Aeronautics Engineering, *M.Sc.*, 2016;
5. D. Del Cont Bernard. *Ground effect analysis for a quadrotor platform*. Aeronautics Engineering, *M.Sc.*, 2016;
6. A. Russo. *Adaptive control of multirotor UAVs*. Automation and Control Engineering, *M.Sc.*, 2016;
7. T. J. E. Chupin. *Data-driven attitude control design for multirotor UAVs*. Automation and Control Engineering, *M.Sc.*, 2016;
8. D. P. Carelli. *Nonlinear attitude and position control for a quadrotor UAV*. Automation and Control Engineering, *M.Sc.*, 2017;
9. D. Chevallard. *Design, identification and control of a micro aerial vehicle*. Aeronautics Engineering, *M.Sc.*, 2017;
10. A. Delbono. *Consensus based control for a unmanned aerial vehicle formation*. Computer Science and Engineering, *M.Sc.*, 2017;
11. P. Gattazzo. *Nonlinear control of a tilt-arm quadrotor UAV*. Automation and Control Engineering, *M.Sc.*, 2017;

12. R. Salbati. *Ground effect compensation for multicopter UAV*. Exchange student from the Université de Liège (ULg), Aeronautics Engineering, *M.Sc.*, 2017;

13. S. Farì. *Guidance and control for a fixed-wing UAV*. Automation and Control Engineering, *M.Sc.*, 2017;

14. S. Musacchio. *Optimal and robust UAV state estimation based on Gps and optical flow*. Aeronautics Engineering, *M.Sc.*, 2018;

15. P. Giuri, A. Cossetti. *Air-to-air automatic landing for multirotor UAVs*. Aeronautics Engineering, *M.Sc.*, 2018;

16. M. Maccotta. *Multirotor UAVs for fugitive emissions detection : sizing, modelling and control system design*. Automation and Control Engineering, *M.Sc.*, 2018;

17. E. Balcioglu. *Design and optimization tool for multirotor unmanned aerial vehicles*. Aeronautics Engineering, *M.Sc.*, 2018;

18. G. Bressan. *Hardware-software architecture, code generation and control for multirotor UAVs*. Computer Science and Engineering, *M.Sc.*, 2018;

19. A. Zangarini. *Data-driven multivariable attitude control design of multirotor UAV platforms*. Aeronautics Engineering, *M.Sc.*, 2019;

20. U. Arshad. *Adaptive control implementation to include ground effect on UAV simulator*. Aeronautics Engineering, *M.Sc.*, 2019;

21. A. Gatti. *Modelling, identification and control of a fixed-wing UAV*. Aeronautics Engineering, *M.Sc.*, 2019;

22. G. Gozzini. *UAV autonomous landing on moving aerial vehicle*. Aeronautics Engineering, *M.Sc.*, 2019;

23. D. Migliore. *Model identification and inversion-based control for multi-rotor UAVs*. Aeronautics Engineering, *M.Sc.*, 2019;

24. G. Roggi. *A computer vision line-tracking algorithm for UAV GNSS-aided guidance*. Aeronautics Engineering, *M.Sc.*, 2019.

CHAPTER 1

Modelling of multirotor UAVs

In this chapter are described all the adopted conventions and the dynamical model which will be used throughout the thesis to describe the behaviour of a multirotor UAV. Moreover three platforms are described in details: the QUAD-X, the TILT-X and the FAST-Hex, which are respectively an under-actuated, an over-actuated and a fully-actuated UAV.

1.1 Multirotor kinematics and dynamics

1.1.1 Reference frames

Despite overwhelming evidence to the contrary, from now on the Earth is assumed to be flat and non-rotating. The Earth reference frame is then considered inertial. Although its origin is usually arbitrary, the first two axes are aligned with the geographic North and East, while the third component is pointing Down to the center of the Earth. The Earth frame is

then defined as $\mathscr{F}_E = (O_E, N, E, D)$, where O_E is the origin of the frame, and N, E and D are respectively the three orthonormal vectors pointing North, East and Down.

The body reference frame instead is a right-handed frame attached to the body of the UAV and it is centred in its Center of Mass (CoM). The first axis lies in the plane of symmetry and it points forward, the second axis points to the right and the third axis points down. The body frame is then defined as $\mathscr{F}_B = (O_B, x_B, y_B, z_B)$. As for the Earth frame, O_B is the origin of the frame, while x_B, y_B, z_B are three orthonormal vector.

For each rotor, a rotor-fixed frame is defined as $\mathscr{F}_{R_i} = (O_{R_i}, x_{R_i}, y_{R_i}, z_{R_i})$ where O_{R_i} coincides with the center of the *i-th* rotor. The unit vector x_{R_i} lies in the rotor disk plane, it is aligned with the respective multirotor arm and it points outward, while z_{R_i} is aligned with the rotor shaft and points downward. Then, y_{R_i} completes a right-hand frame.

Hereafter $e_1 \in \mathbb{R}^3$, $e_2 \in \mathbb{R}^3$, and $e_3 \in \mathbb{R}^3$ will denote the three orthonormal vectors of the canonical basis of $\mathbb{R}^3$,

In Figure 1.1 are reported such reference frames with the body frame attached to the CoM of a generic quadrotor.

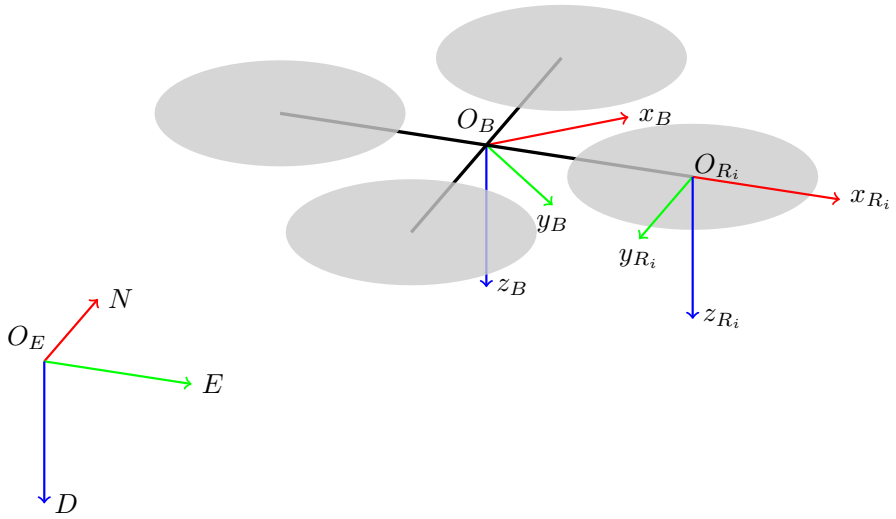


Figure 1.1: *Reference frames.*

1.1.2 Attitude parametrisations

The *attitude* of a UAV (also referred as *orientation*) is defined as the rotation matrix $R \in \mathrm{SO}(3)$, that describes the orientation of $\mathscr{F}_B$ with respect to $\mathscr{F}_E$. In particular, given a vector $v_E \in \mathbb{R}^3$ expressed in the Earth frame, it can be rotated into the body frame according to the equation

$$v_B = R v_E, \tag{1.1}$$

where $v_B \in \mathbb{R}^3$ is the vector v_E resolved in body frame.

Many parametrisations exist for the attitude matrix, in particular, in the following are described the Euler angles and the unit quaternion.

Euler angles

The orientation of one Cartesian coordinate system with respect to another can always be described by three successive rotations around the orthogonal coordinate axes, and the angles of rotation are called the Euler angles (see [7, 8]). The axes and the order of the rotations are chosen in various ways in different fields of science. When we rule out two successive Euler rotations about the same axis, there are twelve possibilities, six without repetition of an axis (counting both forward and reverse) and six with repetition. In flight dynamics, one encounters a specific order of rotation using angles named ψ (Yaw), θ (Pitch) and ϕ (Roll), which represent rotation about the Z axis, the new intermediate Y axis and the newer X axis respectively. The matrix that perform this specific action has the following definition:

$$R(\phi, \theta, \psi) = R_X(\phi) R_Y(\theta) R_Z(\psi), \tag{1.2}$$

where R_X, R_Y and R_Z are the three canonical rotation matrices in $\mathrm{SO}(3)$, in particular

$$R_X(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & \sin(\phi) \\ 0 & -\sin(\phi) & \cos(\phi) \end{bmatrix}, \tag{1.3}$$

$$R_Y(\theta) = \begin{bmatrix} \cos(\theta) & 0 & -\sin(\theta) \\ 0 & 1 & 0 \\ \sin(\theta) & 0 & \cos(\theta) \end{bmatrix}, \tag{1.4}$$

$$R_Z(\psi) = \begin{bmatrix} \cos(\psi) & \sin(\psi) & 0 \\ -\sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{1.5}$$

The defined ranges for the rotation angles are

$$-\pi < \phi \leq \pi \tag{1.6}$$

$$-\pi/2 \leq \theta \leq \pi/2 \tag{1.7}$$

$$-\pi < \psi \leq \pi \tag{1.8}$$

If the pitch angle had been allowed to have a $\pm 180^\circ$ range then the UAV could be inverted and heading South with the roll and heading angles at zero, which is obviously undesirable from a human factors viewpoint.

It is also often useful to extract the Euler angles from the attitude matrix to display to a human operator the Roll, Pitch and Yaw angles. Taking into account the chosen angular ranges, the Euler angles can be computed as

$$\phi = \tan^{-1}\left(\frac{r_{2,3}}{r_{3,3}}\right), \tag{1.9}$$

$$\theta = -\sin^{-1}(r_{1,3}), \tag{1.10}$$

$$\psi = \tan^{-1}\left(\frac{r_{1,2}}{r_{1,1}}\right), \tag{1.11}$$

where $r_{i,j}$ are the elements of the attitude matrix R placed at the *i-th* row and *j-th* column.

The angular orientation of the UAV changes with time when the UAV is manoeuvring. It is then needed to define the Euler angles rates related to

the Yaw-Pitch-Roll rotation sequence as $\dot{\psi}$ (Yaw rate), $\dot{\theta}$ (Pitch rate) and $\dot{\phi}$ (Roll rate). To relate such quantities to the components of the angular rate expressed in the body frame

$$\omega_B = \begin{bmatrix} p \\ q \\ r \end{bmatrix}, \tag{1.12}$$

one has to consider the rotations individually.

The final transformation is given by

$$\omega_B = \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} + R_X(\phi) \left(\begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + R_Y(\theta) \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} \right), \tag{1.13}$$

which can be equivalently written as,

$$\omega_B = E(\phi,\theta) \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\sin(\theta) \\ 0 & \cos(\phi) & \sin(\phi)\cos(\theta) \\ 0 & -\sin(\phi) & \cos(\phi)\cos(\theta) \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}. \tag{1.14}$$

To get the Euler rates in terms of the body rates one must invert the transformation matrix E.

$$E^{-1}(\phi,\theta) = \begin{bmatrix} 1 & \sin(\phi)\tan(\theta) & \cos(\phi)\tan(\theta) \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi)/\cos(\theta) & \cos(\phi)/\cos(\theta) \end{bmatrix}. \tag{1.15}$$

Unfortunately since E is not orthogonal the inverse of E is not its transpose, worse yet, the inverse is singular at pitch angles of $\pm 90^\circ$. This singularity is called *gimbal lock* which is caused by a collinearity of the physical rotation axis vectors of the first and third rotations in the sequence. All the attitude parametrisations that involve three parameters as the Euler angles have a singularity which can be avoided increasing the number of parameters to four. To avoid the gimbal lock one has then to use a four variables attitude parametrisations such as the quaternions.

Unit quaternion

Quaternions are based on a complex number algebra invented by Sir William Rowan Hamilton in 1844 in an attempt to generalize ordinary complex

numbers to three dimensions (see [9]). A quaternion is then a four-dimensional representation of a three-dimensional vector, in particular, it is composed by a three-dimensional vector and a scalar

$$q = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} = \begin{bmatrix} \rho \\ q_4 \end{bmatrix}, \tag{1.16}$$

and it is constrained to have unitary norm

$$\|q\|^2 = \|\rho\|^2 + q_4^2 = 1. \tag{1.17}$$

The attitude matrix is then related to the quaternion by a homogeneous quadratic function of the components of a quaternion (see [10])

$$R(q) = \left(q_4^2 - \|\rho\|^2\right) I_3 + 2\rho\rho^T - 2q_4[\rho\times], \tag{1.18}$$

where I_3 is the 3×3 identity matrix, and the cross product matrix is given by

$$[\rho\times] = \begin{bmatrix} 0 & -q_3 & q_2 \\ q_3 & 0 & -q_1 \\ -q_2 & q_1 & 0 \end{bmatrix}. \tag{1.19}$$

It is easy to see that the attitude matrix is quadratic in the elements of the quaternion vector and also does not involve transcendental functions.

The identity quaternion which correspond to the rotation matrix $R(q) = I_3$ is defined as

$$q = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}. \tag{1.20}$$

It is of course also possible to extract from an attitude matrix R the corresponding quaternion using a modification of Shepperd's algorithm (see [11]). First one has to consider four 4-components vectors

$$x^{(i)} = 4q_i q \quad \text{for} \quad i = 1,\ldots,4 \tag{1.21}$$

where q_i is the *i-th* component of the quaternion q. Each of the four components of each $x^{(i)}$ is given by the Shepperd's algorithm, so these vectors are easily computable from the components of the rotation matrix R. Explicitly,

$$x^{(1)} = \begin{bmatrix} 1+R_{11}-R_{22}-R_{33} \\ R_{12}+R_{21} \\ R_{13}+R_{31} \\ R_{23}-R_{32} \end{bmatrix}, \tag{1.22a}$$

$$x^{(2)} = \begin{bmatrix} R_{21}+R_{12} \\ 1+R_{22}-R_{33}-R_{11} \\ R_{23}+R_{32} \\ R_{31}-R_{13} \end{bmatrix}, \tag{1.22b}$$

$$x^{(3)} = \begin{bmatrix} R_{31}+R_{13} \\ R_{32}+R_{23} \\ 1+R_{33}-R_{11}-R_{22} \\ R_{12}-R_{21} \end{bmatrix}, \tag{1.22c}$$

$$x^{(4)} = \begin{bmatrix} R_{23}-R_{32} \\ R_{31}-R_{13} \\ R_{12}-R_{21} \\ 1+R_{11}+R_{22}+R_{33} \end{bmatrix}. \tag{1.22d}$$

Equation (1.21) shows that each of the $x^{(i)}$ is a scalar multiple of q, so one can obtain the unit quaternion by computing and normalizing any one of the $x^{(i)}$

$$q = \pm \frac{x^{(i)}}{\left\| x^{(i)} \right\|}. \tag{1.23}$$

As in Shepperd's method, choosing the $x^{(i)}$ corresponding to the maximum value of q_i^2 minimizes numerical errors. This selection is made by Shepperd's procedure of finding the largest of R_{ii} and $tr(R)$ (the trace of R).

Another advantage of quaternions is that successive rotations can be accomplished using quaternion multiplication

$$R(q')R(q) = R(q' \otimes q), \tag{1.24}$$

where $\otimes$ is the Hamiltonian product, which can be computed as follows

$$q' \otimes q = \begin{bmatrix} q'_4\rho + q_4\rho' - \rho' \times \rho \\ q'_4 q_4 - \rho' \cdot \rho \end{bmatrix}. \tag{1.25}$$

Moreover, the conjugate of a quaternion is defined by

$$q^* = \begin{bmatrix} -\rho \\ q_4 \end{bmatrix}. \tag{1.26}$$

Note that $q \otimes q^* = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}^\top$.

The inverse of any quaternion having non-zero norm is defined by

$$q^{-1} = \frac{q*}{\|q^*\|}, \tag{1.27}$$

which means that for unit quaternions the conjugate corresponds to the inverse.

Furthermore, the quaternion kinematics equation is given by

$$\dot{q} = \frac{1}{2} q \otimes \begin{bmatrix} \omega_B \\ 0 \end{bmatrix} = \frac{1}{2} \Omega(\omega_B) q, \tag{1.28}$$

where

$$\Omega(\omega_B) = \begin{bmatrix} -[\omega_B \times] & \omega_B \\ -\omega_B^\top & 0 \end{bmatrix}, \tag{1.29}$$

$$[\omega_B \times] = \begin{bmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{bmatrix}, \tag{1.30}$$

and ω_B is the body angular rate defined in equation (1.12). The major advantage of using quaternions is that the kinematics equation is linear in the quaternion and it is also free of singularities.

Finally, one can obtain the quaternion related to the Yaw-Pitch-Roll Euler angles sequence by defining the quaternions related to each indi-

vidual rotation

$$q_X(\phi) = \begin{bmatrix} \sin\left(\frac{\phi}{2}\right) \\ 0 \\ 0 \\ \cos\left(\frac{\phi}{2}\right) \end{bmatrix}, \quad q_Y(\theta) = \begin{bmatrix} 0 \\ \sin\left(\frac{\theta}{2}\right) \\ 0 \\ \cos\left(\frac{\theta}{2}\right) \end{bmatrix}, \quad q_Z(\psi) = \begin{bmatrix} 0 \\ 0 \\ \sin\left(\frac{\psi}{2}\right) \\ \cos\left(\frac{\psi}{2}\right) \end{bmatrix}, \tag{1.31}$$

and then compute the compound orientation as

$$q(\phi, \theta, \psi) = q_X(\phi) \otimes q_Y(\theta) \otimes q_Z(\psi). \tag{1.32}$$

To obtain instead the Roll, Pitch and Yaw Euler angles related to a given quaternion, one can compute the attitude matrix as in equation (1.18) and then compute the angle as in equations (1.9),(1.10) and (1.11). Finally note that for small angles the vector part of the quaternion is approximately equal to half angles so that $\rho \approx \begin{bmatrix} \phi/2 & \theta/2 & \psi/2 \end{bmatrix}^\top$ and $q_4 \approx 1$.

1.1.3 Flight dynamics

The UAV flight dynamics is modelled by the Newton-Euler equations which describe the combined translational and rotational dynamics of a rigid body. Such equations can be written considering as a reference frame both the inertial frame and the body frame, in particular, for both the linear and angular motion the body frame $\mathscr{F}_B$ is taken into account.

Linear motion

To write the linear motion equations one has to take into account Newton's second law.

$$f_B = \frac{\mathrm{d}(m v_B)}{\mathrm{d}t}, \tag{1.33}$$

where $f_B = \begin{bmatrix} f_x & f_y & f_z \end{bmatrix}^\top$ is the external force applied to the UAV, $v_B = \begin{bmatrix} u & v & w \end{bmatrix}^\top$ is the linear velocity and $m \in \mathbb{R}_{>0}$ is the body mass.

Equation (1.33) can be expanded as

$$\begin{aligned} f_B &= \left(\frac{\mathrm{d}m}{\mathrm{d}t}\right) v_B + m\left(\frac{\partial v_B}{\partial t} + \omega_B \times v_B\right) \\ &= \dot{m} v_B + m\dot{v}_B + \omega_B \times (m v_B)\,. \end{aligned} \tag{1.34}$$

Since the considered UAVs are propelled by electrical motors and the energy is stored in batteries, it is possible to neglect the term related to the variation of mass over time, then

$$f_B = m\dot{v}_B + \omega_B \times (m v_B)\,. \tag{1.35}$$

The obtained result represents the rate of change of momentum as a result of the applied force.

Angular motion

Newton's second law for the angular motion states

$$\tau_B = \frac{\mathrm{d}(J\omega_B)}{\mathrm{d}t}, \tag{1.36}$$

where $\tau_B = \begin{bmatrix} l & m & n \end{bmatrix}^\top$ is the external torque applied to the UAV and $J \in \mathbb{R}^{3\times 3}$ is the body inertia tensor.

As for the linear motion, it is possible to expand equation (1.36) while neglecting the mass (and then inertia) variation over time

$$\tau_B = J\dot{\omega}_B + \omega_B \times (J\omega_B)\,. \tag{1.37}$$

Overall states of the UAV

After the definition of the linear and angular dynamic equations it is possible to write the overall state vector containing all the variables considered so far

$$x = \begin{bmatrix} v_B \\ p_E \\ \omega_B \\ q \end{bmatrix}, \tag{1.38}$$

where $p_E = \begin{bmatrix} n & e & d \end{bmatrix}^\top$ is the position of the rigid body expressed in $\mathscr{F}_E$. Therefore, the overall nonlinear dynamic model of the UAV flight dynamics can be expressed as follows

$$\dot{x} = \begin{bmatrix} \dot{v}_B \\ \dot{p}_E \\ \dot{\omega}_B \\ \dot{q} \end{bmatrix} = \begin{bmatrix} (-\omega_B \times v_B + f_B/m) \\ R(q)^\top v_B \\ J^{-1}(-\omega_B \times J\omega_B + \tau_B) \\ \frac{1}{2} q \otimes \begin{bmatrix} \omega_B \\ 0 \end{bmatrix} \end{bmatrix}. \tag{1.39}$$

1.1.4 Applied wrench

The external wrench (*i.e.*, f_B and τ_B) applied to the rigid body of the multirotor is what affects its motion while flying. The most obvious force applied to the multirotor UAV is the gravity force, which will be expressed as

$$f_g = mge_3, \tag{1.40}$$

of course such force is expressed in Earth frame $\mathscr{F}_E$ and it must be rotated into body frame $\mathscr{F}_B$ by the attitude matrix $R(q)$.

It is then common to consider as an external wrench the aerodynamic damping, which of course it is proportional to the body velocity

$$f_d = -K_{f,v} v_B - K_{f,\omega} \omega_B, \tag{1.41}$$
$$\tau_d = -K_{\tau,v} v_B - K_{\tau,\omega} \omega_B, \tag{1.42}$$

where $K_{f,v}, K_{f,\omega}, K_{\tau,v}, K_{\tau,\omega} \in \mathbb{R}^{3\times 3}$.

Finally one has to take into account the controlled wrench produced by the propellers, which has to be expressed in propeller frame $\mathscr{F}_{R_i}$ as follows

$$f_{p_i} = -f_i e_3, \tag{1.43}$$
$$\tau_{p_i} = \varepsilon_i \tau_i e_3, \tag{1.44}$$

where $f_i \in \mathbb{R}_{>0}$ and $\tau_i \in \mathbb{R}_{>0}$ are respectively the thrust and torque produced by the *i-th* propeller and the simplest model that relates such quan-

tities to the controlled propeller angular speed $\omega_i \in \mathbb{R}_{>0}$ is the following

$$f_i = k_f \omega_i^2, \tag{1.45}$$

$$\tau_i = k_\tau \omega_i^2. \tag{1.46}$$

with $k_f \in \mathbb{R}_{>0}$ and $k_\tau \in \mathbb{R}_{>0}$ the thrust and torque coefficient respectively. It is common also to express the torque as a function of the thrust

$$\tau_{p_i} = \varepsilon_i k_f^\tau f_i, \tag{1.47}$$

with $k_f^\tau = k_\tau / k_f$. Moreover, the term $\varepsilon_i \in \{-1, 1\}$, which will be defined accordingly to the multirotor configuration, is used to define the direction of the torque produced by the *i-th* propeller. Figure 1.2 shows the wrench produced by the *i-th* propeller. Due to the medium-small size of the UAVs

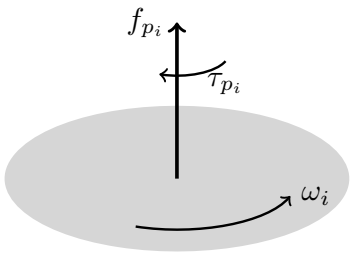


Figure 1.2: *Wrench produced by the i-th propeller.*

taken into account it is possible to neglect for the control-oriented modelling effects such as the gyroscopic effect and the blade flapping of the small propellers as well as the rotor induced drag and the ground/ceiling effects. The controlled forces and moments produced by each propeller are then combined by means of the multirotor configuration in order to obtain the control force and torque ($f_c \in \mathbb{R}^3$ and $\tau_c \in \mathbb{R}^3$ respectively) applied to the rigid body. In the next section three different configurations of multirotors will be described.

The total external wrench applied to the multirotor is expressed as

$$f_B = R(q) f_g + f_d + f_c, \tag{1.48}$$

$$\tau_B = \tau_d + \tau_c. \tag{1.49}$$

1.2 Multirotor configurations

Usually, it is possible to classify the multirotor configurations in three classes: under-actuated, fully-actuated and over-actuated. The first class of multirotor includes the majority of the configurations that can be found both in the literature and on the market. These multirotors usually have all the propellers pointing upward, which means they can produce moments in all the directions, but only a body-vertical force (thrust). This limitation eventually leads to an intrinsic coupling between the translational dynamics and the rotational one (*i.e.*, in order to change the position the multirotor has to change its attitude). The second class of multirotors includes the fully-actuated ones which can instantly achieve decoupled forces and moments in all the directions due to the configuration of the propellers. This feature allows such platforms to control the position and the orientation in a decoupled fashion. Finally there are the over-actuated platforms which have a number of actuators which is greater than the number of degrees of freedom one has to control. These multirotors have the same features of the fully-actuated multirotors but given their intrinsic actuation redundancy they can achieve the same pose with different actuation combinations.

Nowadays multirotors come in a variety of configurations which depend on the number of installed propellers and on which geometry has been chosen for the rotor hub location. These two variables eventually affect the possible wrench that the multirotor can produce by means of the propellers. Among the different technological solutions, multirotor UAVs can be endowed with thrust vectoring capabilities by employing tiltable propellers (see [12]) or by mounting the propellers in a fixed, non-coplanar fashion (see [13] and [14]). While mechanically complex, the tiltable propellers architecture is more efficient than the fixed-tilted one for which a larger amount of power is required to maintain hover. In [15] it is possible to find an interesting study on the degree of coupling between the total force and total moment generated by the propellers and the ability to robustly fly completely still in place after the loss of one or more propellers, in the case of mono-directional propellers. As well, [16] investigates the influence on propulsion system efficiency in hover due to

the configuration of these propellers, taking into account the influence of pusher or puller configuration of the propeller, number of blades, shape and dimensions of the arm, coaxial and overlapping propellers.

In the following are described three different platforms which will be studied for different applications in the next chapters.

1.2.1 QUAD-X

The first introduced platform is a classical X-shaped quadrotor (QUAD-X), with all the propellers pointing upward. Such platform is an under-actuated multirotor since it can only produce a vertical thrust and three torques around the body axes. In Figure 1.3 a scheme representing the numeration and the rotational direction of each propeller is reported.

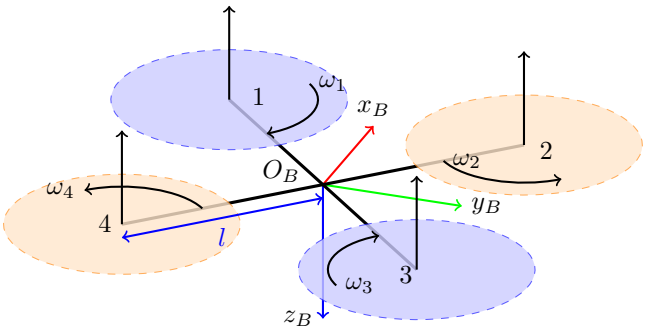


Figure 1.3: *QUAD-X scheme.*

It is possible to write the *wrench map* $F \in \mathbb{R}^{4\times4}$ (a matrix which relates the propeller forces to the achievable body wrench) without involving any roto-translation formalism as follows

$$\begin{bmatrix} f_{c,z} \\ \tau_{c,x} \\ \tau_{c,y} \\ \tau_{c,z} \end{bmatrix} = F \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{bmatrix}, \tag{1.50}$$

where

$$F = \begin{bmatrix} -1 & -1 & -1 & -1 \\ \frac{l}{\sqrt{2}} & -\frac{l}{\sqrt{2}} & -\frac{l}{\sqrt{2}} & \frac{l}{\sqrt{2}} \\ \frac{l}{\sqrt{2}} & \frac{l}{\sqrt{2}} & -\frac{l}{\sqrt{2}} & -\frac{l}{\sqrt{2}} \\ -k_f^\tau & k_f^\tau & -k_f^\tau & k_f^\tau \end{bmatrix}, \tag{1.51}$$

and $l \in \mathbb{R}_{>0}$ is the arm-length (*i.e.*, the distance between the center of mass and the rotor hub). It is then possible to define the *mixer matrix* (also called *wrench mapper*) as the matrix that relates the forces and moments requested by the controllers to the desired thrust of each propeller. Such matrix can be simply defined as the inverse of the wrench map

$$\begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{bmatrix} = F^{-1} \begin{bmatrix} f_{c,z} \\ \tau_{c,x} \\ \tau_{c,y} \\ \tau_{c,z} \end{bmatrix}, \tag{1.52}$$

where

$$F^{-1} = \begin{bmatrix} -\frac{1}{4} & \frac{\sqrt{2}}{4l} & \frac{\sqrt{2}}{4l} & -\frac{1}{4k_f^\tau} \\ -\frac{1}{4} & -\frac{\sqrt{2}}{4l} & \frac{\sqrt{2}}{4l} & \frac{1}{4k_f^\tau} \\ -\frac{1}{4} & -\frac{\sqrt{2}}{4l} & -\frac{\sqrt{2}}{4l} & -\frac{1}{4k_f^\tau} \\ -\frac{1}{4} & \frac{\sqrt{2}}{4l} & -\frac{\sqrt{2}}{4l} & \frac{1}{4k_f^\tau} \end{bmatrix}. \tag{1.53}$$

Finally, one can compute the desired angular speed of each propeller (according to equation (1.45)) as follows

$$\omega_i = \sqrt{\frac{f_i}{k_f}}. \tag{1.54}$$

1.2.2 TILT-X

TILT-X is a quadrotor made by a central rigid body and four arms that can be tilted independently by means of servo-actuators attached to the main body. This particular configuration has been presented in [12], [17] and [18], while in the literature it is possible to find similar configurations but with more tiltable propellers (see [19]). In Figure 1.4 a scheme

representing the numeration and the rotational direction of each propeller is reported. Each mechanical component made by an arm and a rotor is

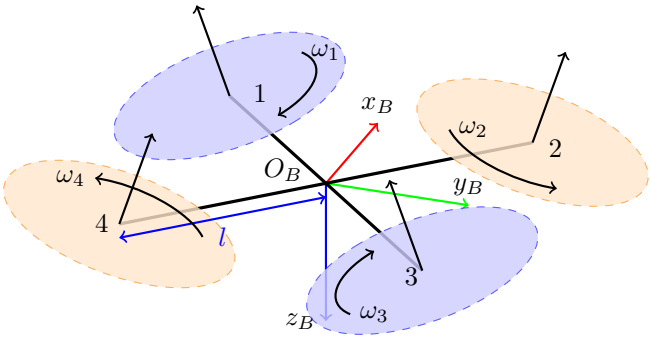


Figure 1.4: *TILT-X scheme.*

referred to as tilt-arm and in Figure 1.5 it is possible to see a detailed scheme. The wrench induced by the propulsive system is dependent on

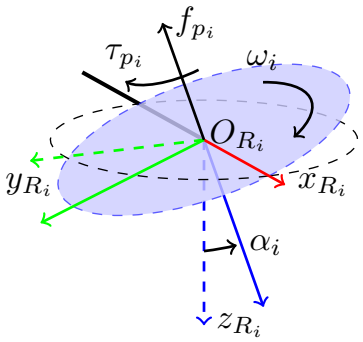


Figure 1.5: *Tilt-arm detail.*

the tilting angle of each propeller ($\alpha_i \in (-\pi, \pi]$) and it can be written as:

$$f_c = \sum_{i=1}^{4} -f_i R_{R_i}^B(\alpha_i)\, e_3, \tag{1.55}$$

$$\tau_c = \sum_{i=1}^{4} \left(x_{BR_i}^B \times \left(f_i R_{R_i}^B(\alpha_i)\, e_3 \right) + \varepsilon_i k_f^\tau f_i R_{R_i}^B(\alpha_i)\, e_3 \right), \tag{1.56}$$

with

$$x_{BR_i}^B = l R_Z \left(\frac{\pi}{4} - \frac{\pi}{2}(i-1) \right) e_1, \tag{1.57}$$

$$R_{R_i}^B(\alpha_i) = R_Z \left(\frac{\pi}{4} - \frac{\pi}{2}(i-1) \right) R_X(\alpha_i), \tag{1.58}$$

where $x_{BR_i}^B$ is the vector from O_B to O_{R_i}, describing the position of the center of the *i-th* propeller (expressed in $\mathscr{F}_B$), $R_{R_i}^B$ is the rotation representing the orientation of $\mathscr{F}_{R_i}$ with respect to $\mathscr{F}_B$ and $\varepsilon_i = (-1)^i$.

The control wrench $w_c = \begin{bmatrix} f_c^\top & \tau_c^\top \end{bmatrix}^\top$ applied to the rigid body is related to the physical inputs

$$u = \begin{bmatrix} \omega_1 & \omega_2 & \omega_3 & \omega_4 & \alpha_1 & \alpha_2 & \alpha_3 & \alpha_4 \end{bmatrix}^\top, \tag{1.59}$$

through a static nonlinear map. Furthermore, by defining a generalized vector-valued function of the inputs $f_u \in \mathbb{R}^8$

$$f_u = \begin{bmatrix} f_{u,1} & f_{u,2} & f_{u,3} & f_{u,4} & f_{u,5} & f_{u,6} & f_{u,7} & f_{u,8} \end{bmatrix}^\top, \tag{1.60}$$

$$f_{u,i} = f_i \cos(\alpha_i), \tag{1.61}$$

$$f_{u,i+4} = f_i \sin(\alpha_i), \tag{1.62}$$

the mapping from f_u to w_c is compactly expressed as

$$w_c = F f_u, \tag{1.63}$$

where $F \in \mathbb{R}^{8\times 6}$ is a full row rank matrix (see [20]) and the linear system (1.63) admits ∞^2 solutions

$$f_u = F^\dagger w_c + \left(I_8 F^\dagger F \right) w \tag{1.64}$$

in which $F^{\dagger} = F^{\top}\left(FF^{\top}\right)^{-1}$ is the right pseudo-inverse of F and $w \in \mathbb{R}^8$ is an arbitrary vector. Then, to compute the required angular velocity of each propeller one can simply use the following equations

$$f_i = \sqrt{f_{u,i}^2 + f_{u,i+4}^2}, \tag{1.65}$$

$$\omega_i = \sqrt{\frac{f_i}{k_f}}, \tag{1.66}$$

while for the angle of each servomotor one has

$$\alpha_i = \tan^{-1}\left(\frac{f_{u,i+4}}{f_{u,i}}\right), \tag{1.67}$$

for $i = 1, \ldots, 4$.

1.2.3 FAST-Hex

The Fully–Actuated by Synchronized–Tilting Hexarotor (FAST-Hex) is a novel UAV concept which is able to smoothly change its configuration from under-actuated to fully-actuated by using only one additional motor that tilts all the propellers at the same time. FAST-Hex has been first introduced in [6]. In Figure 1.6 it is possible to see a CAD of this UAV while in Figure 1.7 a detail of the turret tilting mechanism is shown. This platform can adapt to the task at hand by finely tuning its configuration from the efficient (but under-actuated) flight to the full-pose-tracking (but less efficient) flight, which is attainable by non-coplanar multi-rotors. Its simplest model which is composed by a rigid-body and six massless orientable propellers is presented in [6] nevertheless here a more accurate model will be described.

First of all, a new frame $\mathscr{F}_G = (O_G, x_G, y_G, z_G)$ has to be defined, with O_G centred in the geometric center of the UAV (as shown in Figure 1.8). The orientation of this new frame is the same of the body frame, so that the transformation to express $\mathscr{F}_G$ with respect to $\mathscr{F}_B$ is

$$T_{BG} = \begin{bmatrix} I_3 & x_{BG}^B \\ 0_{3\times 1} & 1 \end{bmatrix}, \tag{1.68}$$

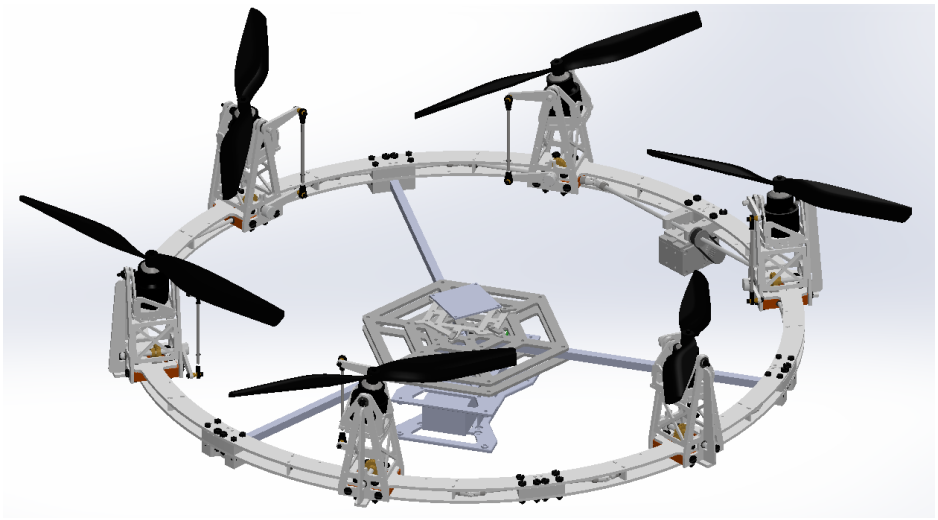


Figure 1.6: *FAST-Hex CAD.*

Figure 1.7: *FAST-Hex turret detail.*

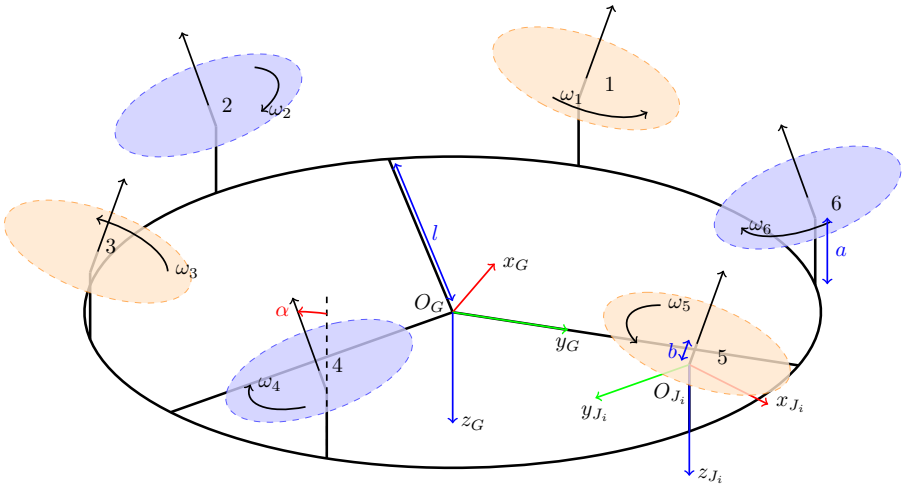


Figure 1.8: *FAST-Hex scheme.*

with $x_{BG}^B \in \mathbb{R}^3$ the vector from O_B to O_G, describing the position of the geometric center (expressed in $\mathcal{F}_B$). For each turret, a fixed frame is defined as $\mathcal{F}_{J_i} = (O_{J_i}, x_{J_i}, y_{J_i}, z_{J_i})$ where O_{J_i} coincides with the center of the *i-th* rotational joint. The unit vector x_{J_i} lies on the joint rotational axis and points outward, while z_{J_i} points downward. Then, y_{J_i} completes the right-hand frame. The *i-th* frame $\mathcal{F}_{J_i}$ is expressed with respect to $\mathcal{F}_G$ as follows

$$T_{GJ_i} = \begin{bmatrix} R_{J_i}^G & x_{GJ_i}^G \\ 0_{3\times 1} & 1 \end{bmatrix}, \tag{1.69}$$

where $x_{GJ_i}^G = \begin{bmatrix} l & 0 & -a \end{bmatrix}^\top$, with $a \in \mathbb{R}_{>0}$ the vertical distance between O_G and O_{J_i} (expressed in $\mathcal{F}_G$), and $R_{J_i}^G = R_Z\left(\frac{\pi}{3}(i-1)\right)$. Moreover, one can express the transformation between $\mathcal{F}_{J_i}$ and $\mathcal{F}_{R_i}$ as

$$T_{J_iR_i}(\alpha) = \begin{bmatrix} R_{R_i}^{J_i}(\alpha) & x_{J_iR_i}^{J_i} \\ 0_{3\times 1} & 1 \end{bmatrix}, \tag{1.70}$$

with $x_{J_iR_i}^{J_i} = \begin{bmatrix} 0 & 0 & -b \end{bmatrix}^\top$, $b \in \mathbb{R}_{>0}$ the vertical distance between O_{J_i} and O_{R_i} (expressed in $\mathcal{F}_{J_i}$), $R_{R_i}^{J_i} = R_X\left((-1)^{i-1}\alpha\right)$ and $\alpha \in [0, \pi/6]$ the synchronized tilting angle of each propeller.

Finally, it is possible to compute the transformation between $\mathscr{F}_B$ and $\mathscr{F}_{J_i}$ as follows

$$T_{BR_i}(\alpha) = T_{BG}T_{GJ_i}T_{J_iR_i}(\alpha) = \begin{bmatrix} R^B_{R_i}(\alpha) & x^B_{BR_i}(\alpha) \\ 0_{3\times 1} & 1 \end{bmatrix}, \tag{1.71}$$

where, as for the TILT-X case, $x^B_{BR_i}$ is the vector from O_B to O_{R_i}, describing the position of the center of the *i-th* propeller (expressed in $\mathscr{F}_B$), $R^B_{R_i}(\alpha)$ is the rotation representing the orientation of $\mathscr{F}_{R_i}$ with respect to $\mathscr{F}_B$.

The wrench induced by the propulsive system can be written as:

$$f_c = \sum_{i=1}^{6} -f_i R^B_{R_i}(\alpha)\, e_3, \tag{1.72}$$

$$\tau_c = \sum_{i=1}^{6} \left(x^B_{BR_i}(\alpha) \times \left(f_i R^B_{R_i}(\alpha)\, e_3 \right) + \varepsilon_i k^\tau_f f_i R^B_{R_i}(\alpha)\, e_3 \right), \tag{1.73}$$

where $\varepsilon_i = (-1)^{i-1}$.

The control wrench $w_c = \begin{bmatrix} f_c^\top & \tau_c^\top \end{bmatrix}^\top$ applied to the rigid body is related to the physical inputs

$$u = \begin{bmatrix} f_1 & f_2 & f_3 & f_4 & f_5 & f_6 \end{bmatrix}^\top, \tag{1.74}$$

through the nonlinear map

$$w_c = F(\alpha) f_u, \tag{1.75}$$

where $F(\alpha) \in \mathbb{R}^{6\times 6}$ is the allocation map.

Since the structural properties of the allocation map $F(\alpha)$ changes with the tilting angle α (*i.e.*, with $\alpha = 0$ the allocation map becomes singular or for values of $\alpha \approx 0$ it may be ill-conditioned) the computation of the wrench mapper is not trivial and the use of a simple inversion is not possible. In [21] different approaches aimed at modifying the original ill-posed via estimation problem with the goal of stabilizing the solution and/or obtaining a meaningful solution are presented, these approaches are known as *regularisation*. In particular, the adopted method is called *Tikhonov regularisation* which compute the solution in closed form as

$$f_u = F(\alpha)^\ddagger w_c, \tag{1.76}$$

$$F(\alpha)^{\ddagger} = \left(F(\alpha)^{\top} F(\alpha) + \gamma I_6\right)^{-1} F(\alpha)^{\top}, \tag{1.77}$$

where $\gamma \in \mathbb{R}_{>0}$ is a properly chosen regularization parameter. Of course, for $\alpha \gg 0$ the allocation matrix is full-rank, then the Tikhonov regularisation is not needed anymore. It is then mandatory to parametrise $\gamma = \gamma(\alpha)$ in order to make its contribution significant for $\alpha \approx 0$ and negligible for $\alpha \gg 0$. For this purpose, a hyperbolic curve has been adopted

$$\gamma(\alpha) = \frac{k_1}{\alpha + k_2}, \tag{1.78}$$

with $k_1 \in \mathbb{R}_{>0}$ and $k_2 \in \mathbb{R}_{>0}$ properly chosen.

FAST-Hex has two structurally different configurations

1. $\alpha = 0 \rightarrow rank\,(F\,(\alpha)) = 4$;
2. $\alpha \in (0, \pi/6] \rightarrow rank\,(F\,(\alpha)) = 6$.

In configuration 1) all propellers of the FAST-Hex are coplanar or, equivalently, their spinning axes are all collinear. The system then degenerates to an ordinary hexarotor platform which means that in such configuration the internal forces during hovering are zero. In the configurations of type 2), on the other hand, the internal forces in hovering are greater than zero, which means that the system is wasting more energy than in configuration 1). The larger $|\alpha|$ the larger the internal forces. Due to the fact that α is a slowly changeable parameter, the change of α is delegated to a high-level slow-rate controller/planner or to a human operator. The high-level controller can gently tune α while flying, thus continuously changing the platform between configuration 1) and any of the configurations of type 2) in order to adapt to the particular task being executed. For example, configuration 1) can be chosen when a pure horizontal hovering is requested while a type 2) configurations can be selected when hovering with non-zero roll and pitch is needed.

CHAPTER 2

Rotor aerodynamics

The forces and moment produced by each propeller have been presented in equations (1.45) and (1.46). These equations are adequate for a control stand-point but they are not suitable to describe the thrust and torque variation with respect to the flight regime. In this chapter the aerodynamic effects which have been included to describe the physical model of the rotors in order to improve the modelling for the simulation are described. In particular, the well-known theories from the helicopter literature which will be applied to the small scale multirotor case are described. Moreover, the ground effect modelling and the extensive experimental campaign for its characterisation will be described.

2.1 Forward flight rotor aerodynamics

As is well known in the helicopter literature (see [22, 23]), the thrust produced by a rotating propeller depends on many factors (*i.e.*, flapping,

lagging and pitching motion of the blades).

Depending on the flight condition (*e.g.*, hover, vertical flight, forward flight, etc.) the rotor performance change. Anyway, the results obtained for a forward flight condition summarise all the cases of practical interest.

In the next subsections the basic definition needed and the well-known aerodynamic theories applied in the helicopter field will be described.

2.1.1 Basic definitions

In the literature, the adopted convention is to assume that the rotor rotation direction is counter-clockwise (viewed from above). The natural reference length scale for the rotor is the blade radius r, and the natural reference time scale is the rotor rotational speed ω [rad/s]. The variables $\tilde{r}$ and ψ usually refer to the radial and azimuthal position of the blade, but they can also be used as polar coordinates for the rotor disk.

Assuming the helicopter flying forward, the relative velocity with the air is defined as v_r, while the angle of attack is α. The rotor induced velocity v is assumed to be normal to the propeller plane while the advance ratio μ and the total inflow ratio λ are the dimensionless velocity components parallel to and normal to the propeller plane, respectively

$$\mu = \frac{v_r \cos(\alpha)}{\omega r}, \tag{2.1}$$

$$\lambda = \frac{v_r \sin(\alpha) + v}{\omega r} = \mu \tan(\alpha) + \lambda_i = \lambda_c + \lambda_i, \tag{2.2}$$

where λ_c is the climb inflow ratio and λ_i is the induced inflow ratio. In Figure 2.1 the quantities introduced so far are summarised.

Moreover, with reference to Figure 2.2, one has to define as f the rotor thrust, which is normal to the disk plane and positive when directed upward, f_h the rotor drag force in the disk plane (positive when directed rearward, opposing the forward velocity of the helicopter), τ rotor shaft torque, which is positive when an external torque is required to turn the rotor (helicopter operation) and p the rotor shaft power, which is positive when power is supplied to the rotor. In Figure 2.2 the rotor forces with reference to the dimensionless velocities are reported. In coefficient form based on air density (ρ), rotor disk area (a), and tip speed (ωr) these

quantities are defined as follows

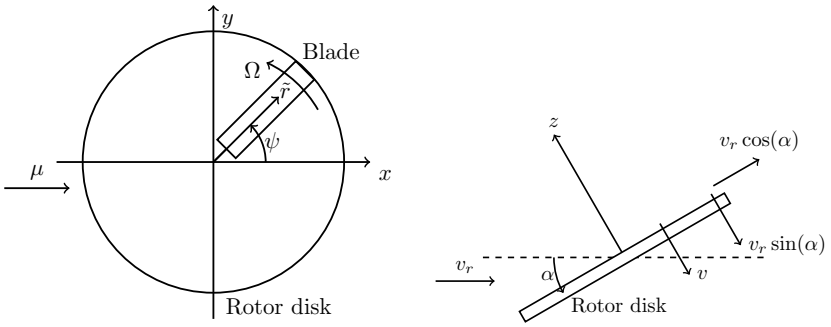
$$c_T = \text{thrust coefficient} = \frac{f}{\rho a (\omega r)^2}, \tag{2.3}$$

$$c_H = \text{H force coefficient} = \frac{f_h}{\rho a (\omega r)^2}, \tag{2.4}$$

$$c_Q = \text{torque coefficient} = \frac{\tau}{\rho a (\omega r)^2 r}, \tag{2.5}$$

$$c_P = \text{power coefficient} = \frac{p}{\rho a (\omega r)^3}. \tag{2.6}$$

The rotor shaft power and torque are related by $p = \omega\tau$ so the coefficients are equal, $c_P = c_Q$.



(a) Top view (b) Side view

Figure 2.1: *Rotor disk in forward flight.*

2.1.2 Momentum theory

Momentum Theory (MT) explains the rotor performance from a global point of view, considering the rotor disk as a unique lifting surface. This basic assumption describes the rotor thrust thanks to the actuator disk theory.

Consider a rotor operating at velocity v_r, with angle of attack α between the free stream velocity and the rotor disk (Figure 2.3). The in-

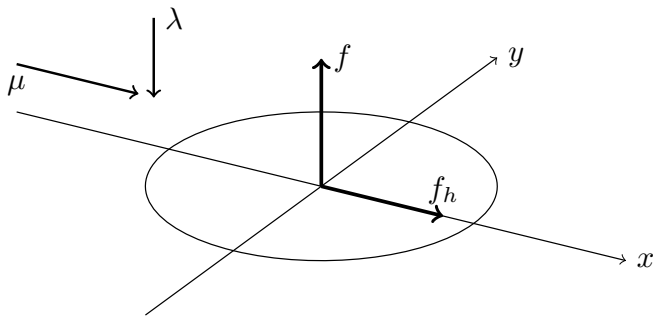


Figure 2.2: *Rotor forces and dimensionless velocity.*

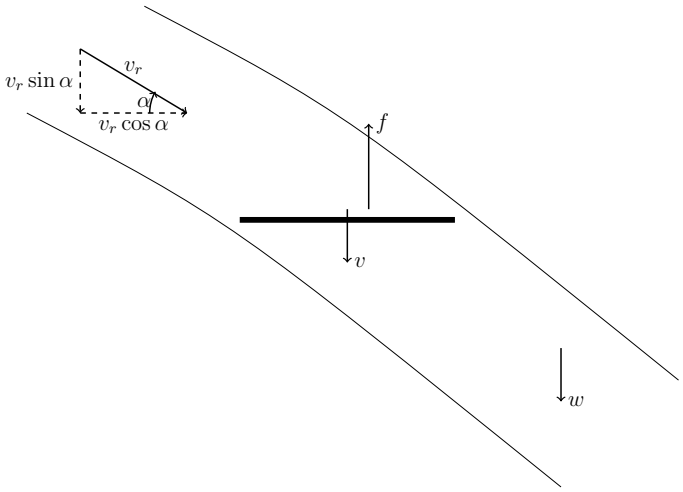


Figure 2.3: *Flow model for momentum theory analysis of rotor in forward flight.*

duced velocity at the disk is v, while in the far wake the velocity is $w = 2v$ (from the momentum and energy conservation relations, see [22]) which is assumed to be parallel to the rotor thrust vector. Momentum conservation gives the rotor thrust $f = \dot{m}2v$, with the mass flux $\dot{m} = \rho a u$ and the resultant velocity u given by

$$u^2 = (v_r \cos(\alpha))^2 + (v_r \sin(\alpha) + v)^2 = v_r^2 + 2v_r v \sin(\alpha) + v^2. \tag{2.7}$$

Hence,

$$f = 2\rho a v u = 2\rho a v \sqrt{v_r^2 + 2v_r v \sin(\alpha) + v^2}. \tag{2.8}$$

Notice that for hovering flight ($v_r = 0$), equation (2.8) reduces to

$$f = 2\rho a v^2 = 2\rho a v_h^2; \tag{2.9}$$

as forward flight speed increases such that $v_r \gg v$

$$f = 2\rho a v v_r, \tag{2.10}$$

which is called Glauert's "high-speed" approximation.

It is also possible to compute the rotor power thanks to the energy conservation, which gives

$$\begin{aligned} p &= \dot{m}\left(\frac{1}{2}\left[(v_r \sin(\alpha) + 2v)^2 + (v_r \cos(\alpha))^2\right] - \frac{1}{2}v_r^2\right) \\ &= f\left(v_r \sin(\alpha) + v\right) = f v_r \sin(\alpha) + f v. \end{aligned} \tag{2.11}$$

The first term on the right-hand side of the above equation (2.11) is the power required to propel the rotor forward and also to climb, while the second term is the induced power..

To finally compute the induced velocity, in [22] it is assumed that the thrust in forward flight (or in vertical flight as well) is similar to the hovering one (*i.e.*, $v_h^2 \approx vu$). Therefore, induced velocity is computed as

$$v = \frac{v_h^2}{\sqrt{v_r^2 + 2v_r v \sin(\alpha) + v^2}}, \tag{2.12}$$

where the induced velocity in hover is $v_h^2 = f/2\rho a$ (see also equation (2.9)).

Now, recalling the advance and inflow ratio definitions from equations (2.1) and (2.2), an analytic equation for induced inflow ratio is obtained

$$\lambda_i = \frac{c_T}{2\sqrt{\mu^2 + \lambda^2}}. \tag{2.13}$$

Finally, the solution for the inflow ratio λ is

$$\lambda = \mu \tan(\alpha) + \frac{c_T}{2\sqrt{\mu^2 + \lambda^2}}. \tag{2.14}$$

While analytic solutions to this equation can be found, in practice a simple numerical procedure is usually used to solve for λ. Such inflow equation is widely employed for practical calculations involving rotors in climbing and descending flight in both axial and forward flight. However, a non-physical solution will always be obtained if there is a descent (upward) component of velocity normal to the rotor disk that is between 0 and $2v$ (*i.e.*, if $-2v \leq v_r \sin(\alpha) \leq 0$ in level flight). Under these conditions there can always be two possible directions for the flow and there can be no well-defined slipstream boundary as was assumed in the physical model. Therefore, momentum theory cannot be applied under these conditions. For example, this can occur when the rotor disk is at steep angles of attack.

2.1.3 Blade element theory

Blade Element Theory (BET) provides a thrust expression from a differential point of view, analysing the thrust of an infinitesimal portion of each blade. It is possible to find a fully developed description of BET in [23]. In Figure 2.4 the rotor blade section is represented, with the blade section pitch θ, and the components of the airspeed relative to the blade. In particular, the components of the airspeed are the in-plane velocity u_T (tangential to the disk plane, positive toward the trailing edge) and u_P (perpendicular to the disk plane, positive downward). The resultant velocity u and the relative inflow angle (or induced angle of attack) ϕ of the

section are

$$u = \sqrt{u_T^2 + u_P^2}, \tag{2.15}$$

$$\phi = \tan^{-1}\left(\frac{u_P}{u_T}\right). \tag{2.16}$$

Then, the section angle of attack is

$$\alpha = \theta - \phi. \tag{2.17}$$

In Figure 2.4 are also depicted the aerodynamic forces on the blade section. The aerodynamic lift and drag (dl and dd) are, respectively, normal to and parallel to the resultant velocity u. The components of the section lift and drag resolved in the reference plane are df_z and df_x (normal and in-plane, respectively).

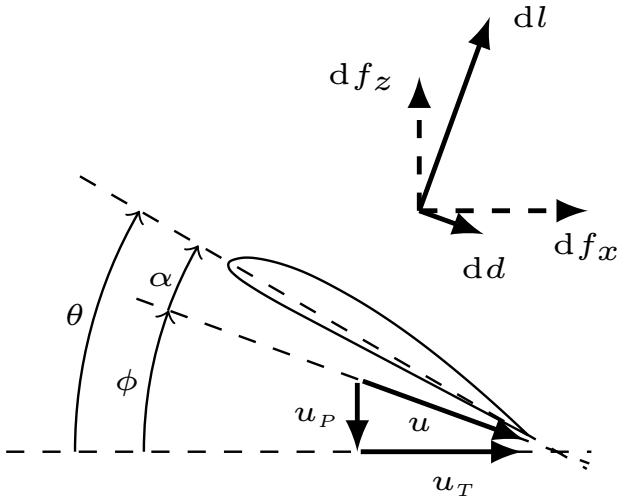


Figure 2.4: *Aerodynamics of the rotor blade section (looking outboard).*

Since in general u_P is small with respect to u_T, it is possible to rewrite

equations (2.15) and (2.16) as follows

$$u \approx u_T, \tag{2.18}$$

$$\phi \approx \frac{u_P}{u_T}. \tag{2.19}$$

The resulting incremental lift dl per unit span on the blade element is

$$\begin{aligned} \mathrm{d}l &= \frac{1}{2}\rho u^2 c c_L \mathrm{d}\tilde{r} = \frac{1}{2}\rho u_T^2 c c_{L\alpha}\left(\theta - \phi\right)\mathrm{d}\tilde{r} \\ &= \frac{1}{2}\rho u_T^2 c c_{L\alpha}\left(\theta - \frac{u_P}{u_T}\right)\mathrm{d}\tilde{r} \\ &= \frac{1}{2}\rho c c_{L\alpha}\left(\theta u_T^2 - u_P u_T\right)\mathrm{d}\tilde{r}, \end{aligned} \tag{2.20}$$

and the incremental drag is

$$\mathrm{d}d = \frac{1}{2}\rho u_T^2 c c_D \mathrm{d}\tilde{r}, \tag{2.21}$$

where c is the local blade chord, c_L and c_D are the lift and drag coefficients respectively. These forces can be then resolved in the rotor disk plane coordinates

$$\begin{aligned} \mathrm{d}g_z &= \mathrm{d}l\cos(\phi) - \mathrm{d}d\sin(\phi) \approx \mathrm{d}l \\ &= \frac{1}{2}\rho c c_{L\alpha}\left(\theta u_T^2 - u_P u_T\right)\mathrm{d}\tilde{r}, \end{aligned} \tag{2.22}$$

and

$$\begin{aligned} \mathrm{d}g_x &= \mathrm{d}l\sin(\phi) + \mathrm{d}d\cos(\phi) \approx \phi\,\mathrm{d}l + \mathrm{d}d \\ &= \frac{1}{2}\rho c c_{L\alpha}\left(\theta u_P u_T - u_P^2 + \frac{c_D}{c_{L\alpha}}u_T^2\right)\mathrm{d}\tilde{r}. \end{aligned} \tag{2.23}$$

In forward flight the blade element velocity components are periodic at the rotor rotational frequency. As for the hover case, there is an in-plane velocity component because of blade rotation about the rotor shaft, but now there is a further free-stream (translational) part such that

$$u_T\left(\tilde{r}, \psi\right) = \omega\tilde{r} + v_r\sin\left(\psi\right) = \omega\tilde{r} + \mu\,\omega r\sin\left(\psi\right). \tag{2.24}$$

As well, the velocity perpendicular to the disk can be written as

$$u_P = \lambda \omega r. \tag{2.25}$$

Finally, it is possible to compute the contributions to the thrust, H-force and torque of the rotor. The incremental thrust is given by

$$\mathrm{d}f = \mathrm{d}f_z = \frac{1}{2}\rho c c_{L\alpha}\left(\theta u_T^2 - u_P u_T\right)\mathrm{d}\tilde{r}, \tag{2.26}$$

which can be re-written taking into account the definition of the periodic velocity components

$$\mathrm{d}f = \frac{1}{2}\rho c c_{L\alpha}\omega^2 r^2\left[\theta\left(y+\mu\sin(\psi)\right)^2 + \mu\left(y+\mu\sin(\psi)\right)\right]\mathrm{d}y. \tag{2.27}$$

To find the total rotor thrust it is necessary to calculate the average thrust of a blade taken round the disc and multiply by the number of blades (n_b). Then, given the blade geometry, inflow, and section drag, this expression can be integrated numerically over the blade span. Unfortunately, it is not possible to perform a detailed numerical approach since the propeller's manufacturers usually do not provide enough informations about the blade geometry. It is then common to impose some assumptions and perform the integration analytically. For simplicity (see [24]), the chord is considered constant along the blade (rectangular blade assumption) as well as the pitch angle ($\theta = \theta_0$) and the inflow is considered uniform ($\lambda =$ constant). The total thrust is then given by

$$f = \frac{1}{4}n_b\rho c c_{L\alpha}\omega^2\tilde{r}^3\left[\frac{2}{3}\theta_0\left(1+\frac{3}{2}\mu^2\right) - \lambda\right], \tag{2.28}$$

which leads to the thrust coefficient

$$c_T = \frac{1}{4}\sigma c_{L\alpha}\left[\frac{2}{3}\theta_0\left(1+\frac{3}{2}\mu^2\right) - \lambda\right], \tag{2.29}$$

where $\sigma = n_b c/\pi r$ is the solidity ratio. The same approach can be applied to compute the H-force and the torque coefficients starting from the incremental definitions

$$\mathrm{d}f_h = n_b \mathrm{d}f_x \sin(\psi), \tag{2.30}$$

$$\mathrm{d}\tau = n_b \mathrm{d}f_x \tilde{r}, \tag{2.31}$$

which leads to

$$c_H = \frac{1}{4}\sigma c_{L\alpha}\left(\frac{\mu c_D}{2c_{L\alpha}} + \frac{1}{2}\mu\lambda\theta_0\right), \tag{2.32}$$

$$c_Q = \frac{1}{8}\sigma c_D\left(1+4.7\mu^2\right) + c_T\lambda - c_H\mu. \tag{2.33}$$

2.1.4 Blade element momentum theory

Momentum Theory and BET differ in their approach to the rotor aerodynamic problem. In order to take advantage of both, the combined version of these theories is presented in literature. The Blade Element Momentum Theory (BEMT) for hovering rotors is a hybrid method that combines the basic principles from both the blade element and momentum theory approaches. The principles involve the invocation of the equivalence between the circulation and momentum theories of lift. With certain assumptions, the BEMT allows the inflow distribution along the blade to be estimated.

Blade element theory describes the differential thrust on an annulus of the disk (on all n_b blades) of width dr at radial station r as

$$\mathrm{d}c_T = \frac{\sigma c_{L\alpha}}{2}\left(\theta r^2 - \lambda r\right)\mathrm{d}r. \tag{2.34}$$

Instead, the differential form of momentum theory is

$$\mathrm{d}f = 2\rho\mathrm{d}avu, \tag{2.35}$$

which normalised leads to

$$\mathrm{d}c_T = 4\sqrt{\mu^2+\lambda^2}\lambda_i r\,\mathrm{d}r. \tag{2.36}$$

Equating the blade element and momentum theory expressions for dc_T it is possible to find the distribution of the non-uniform inflow λ as a function of $\tilde{r}$.

Equations (2.34) and (2.36) provide an analytical starting point to the numerical analyses for the inflow ratio λ. The initial inflow ratio for the prescribed radial position can be found by solving the quadratic function

arising from the equality. In hovering condition, the initial inflow ratio is

$$\lambda_i = \frac{\sigma c_{L\alpha}}{16}\left(\sqrt{1+\frac{32}{\sigma c_{L\alpha}}\theta r}-1\right), \tag{2.37}$$

where, as previously described, $c_{L\alpha}$ is the lift slope of the airfoil in use and θ is the blade twist at the radial position r.

After the determination of the starting inflow value, the procedure continues by calculating the angle of attack, the aerodynamic coefficients and the relative velocity of the blade element. Again invoking the equality between the incremental thrust coefficient representations, a new inflow ratio can be obtained. New aerodynamic variables are computed using the found inflow ratio and the loop continues to find a final value for the inflow. The loop quits when the error between the successively found inflow values are below a certain threshold. Subsequently, the final aerodynamic forces and moments are computed to estimate the performance of the rotor.

2.1.5 Dynamic inflow

As described in [23], when a rotor moves forward an estimate of inflow distribution should be taken into account in order to estimate rotor performance. Many inflow models have been developed in the literature, aimed to better describe rotor loads in forward flight. The most famous model present in the literature is the one proposed by Pitt and Peters in [25]. In [26], instead, the authors improve the dynamical model of the rotor inflow due to all velocity components approaching the rotor blades. A dynamic inflow model is a set of first-order differential equations relating inflow variables and aerodynamic loading variables. The simplest model has three inflow states, consisting of uniform (λ_0) and linear (λ_c, λ_s) perturbations of the wake-induced downwash at the rotor disk

$$\lambda\left(\tilde{r},\bar{\psi}\right) = v_0 + v_s\frac{\tilde{r}}{r}\sin\left(\bar{\psi}\right) + v_c\frac{\tilde{r}}{r}\cos\left(\bar{\psi}\right), \tag{2.38}$$

where $\bar{\psi} = \psi - \delta$ is a needed transformation to relate the wind-axes coordinates to the rotor coordinates introduced in Figure 2.1.

The loading variables are the integrated section lift of all blades: thrust c_T, roll moment c_1 and pitch moment c_2 (aerodynamic contributions only). The dynamic inflow equations are

$$M \begin{bmatrix} \dot{v_0} \\ \dot{v_s} \\ \dot{v_c} \end{bmatrix} + L_{nl}^{-1} \begin{bmatrix} v_0 \\ v_s \\ v_c \end{bmatrix} = \begin{bmatrix} c_T \\ -c_1 \\ -c_2 \end{bmatrix}_{aero} . \tag{2.39}$$

Here, M is the mass matrix (a time delay effect due to the unsteady wake) obtained from the virtual mass of an impermeable disk in translation or rotation

$$M = \begin{bmatrix} \frac{8}{3\pi} & 0 & 0 \\ 0 & \frac{16}{45\pi} & 0 \\ 0 & 0 & \frac{16}{45\pi} \end{bmatrix}, \tag{2.40}$$

L_{nl}^{-1} is the nonlinear version of the inflow gains matrix

$$L_{nl}^{-1} = LV^{-1} \tag{2.41}$$

with

$$L = \begin{bmatrix} \frac{1}{2} & -\frac{15\pi}{64}\sqrt{\frac{1-\cos(\alpha)}{1+\cos(\alpha)}} & 0 \\ \frac{15\pi}{64}\sqrt{\frac{1-\cos(\alpha)}{1+\cos(\alpha)}} & \frac{4\cos(\alpha)}{1+\cos(\alpha)} & 0 \\ 0 & 0 & \frac{4}{1+\cos(\alpha)} \end{bmatrix}, \tag{2.42}$$

and

$$V = \begin{bmatrix} v_T & 0 & 0 \\ 0 & \frac{\mu^2+\lambda(\lambda+\lambda_i)}{v_T} & 0 \\ 0 & 0 & \frac{\mu^2+\lambda(\lambda+\lambda_i)}{v_T} \end{bmatrix}, \tag{2.43}$$

where $v_T = \sqrt{\mu^2+\lambda^2}$ is the resultant flow through the rotor. Through a transformation matrix T, dependent on the angles between wind-axes and natural rotor coordinates, it is possible to rotate the quantities to the rotor

disk plane

$$\begin{bmatrix} v_0 \\ v_s \\ v_c \end{bmatrix} = T \begin{bmatrix} \lambda_0 \\ \lambda_s \\ \lambda_c \end{bmatrix}, \tag{2.44}$$

$$\begin{bmatrix} c_T \\ -c_1 \\ -c_2 \end{bmatrix} = T \begin{bmatrix} c_T \\ c_l \\ -c_m \end{bmatrix}, \tag{2.45}$$

where

$$T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\delta) & \sin(\delta) \\ 0 & -\sin(\delta) & \cos(\delta) \end{bmatrix}, \tag{2.46}$$

and the parameters c_l and c_m represent the lateral roll and the longitudinal pitching moment coefficients. Substituting equations (2.44) and (2.45) into equation (2.39), one obtains the following first-order differential equation representing the nonlinear theory of dynamic inflow with respect to the rotor disk plane

$$M \begin{bmatrix} \dot{\lambda}_0 \\ \dot{\lambda}_s \\ \dot{\lambda}_c \end{bmatrix} + \hat{L}_{nl}^{-1} \begin{bmatrix} \lambda_0 \\ \lambda_s \\ \lambda_c \end{bmatrix} = \begin{bmatrix} c_T \\ c_1 \\ -c_2 \end{bmatrix}, \tag{2.47}$$

where

$$\hat{L}_{nl}^{-1} = VT^{\top}L^{-1}T = V\tilde{L}^{-1} \tag{2.48}$$

$$\tilde{L} = \begin{bmatrix} \frac{1}{2} & -a_1\sin(\delta) & -a_1\cos(\delta) \\ a_1\sin(\delta) & a_2\cos^2(\delta)+a_3\sin^2(\delta) & (a_3-a_2)\sin(\delta)\cos(\delta) \\ a_1\cos(\delta) & (a_3-a_2)\cos(\delta)\sin(\delta) & a_2\sin^2(\delta)+a_2\cos^2(\delta) \end{bmatrix} \tag{2.49}$$

$$\begin{aligned} a_2 &= \frac{4}{1+\sin(\alpha_d)}, \quad (a_2-a_3) = 4\frac{1-\sin(\alpha_d)}{1+\sin(\alpha_d)}, \\ a_3 &= \frac{4\sin(\alpha_d)}{1+\sin(\alpha_d)}, \quad a_1 = \frac{15\pi}{64}\sqrt{\frac{1-\sin(\alpha_d)}{1+\sin(\alpha_d)}}, \end{aligned} \tag{2.50}$$

and $\alpha_d = \tan^{-1}(|\lambda|/\mu)$ is the wake angle with respect to the rotor disk (always positive, wheter the flow comes from above or below). Finally,

the normal induced inflow (λ_i) due to the effect of the rotor thrust (c_T) is

$$\lambda_i = \frac{1}{2} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}^\top \tilde{L}^{-1} \begin{bmatrix} \lambda_0 \\ \lambda_s \\ \lambda_c \end{bmatrix}. \tag{2.51}$$

The instantaneous uniform inflow λ_i determined from equation (2.51) is then used with the instantaneous angle α to calculate v_T, V, and $\hat{L}_{nl}$. Thus, equations (2.47)-(2.51) represent the complete nonlinear theory of dynamic inflow.

In summary, the above derivation gives both the nonlinear dynamic inflow model and the perturbation dynamic inflow model in a general coordinate system (not necessarily wind-axes). Before application, the governing equations may be switched to their dimensional forms since time is normalized on rotor speed, *i.e.*,

$$\dot{\lambda}_0 = \frac{\mathrm{d}\lambda_0}{\mathrm{d}t^*} = \frac{\mathrm{d}\lambda_0}{\mathrm{d}t\,\omega} = \frac{1}{\omega}\frac{\mathrm{d}\lambda_0}{\mathrm{d}t}. \tag{2.52}$$

2.2 Multirotor aerodynamic modelling

The flight regime of a small-sized UAV is not always the hovering condition. The generally accepted control-oriented nonlinear dynamic model of a generic multirotor is based (as presented in Chapter 1) on thrust and torque models with constant thrust and torque coefficients derived from static tests

$$f = k_f \omega^2, \tag{2.53}$$

$$\tau = k_\tau \omega^2. \tag{2.54}$$

Even though such model is suitable for control law design, it is no longer valid when the vehicle undertakes dynamic manoeuvres that involve significant velocities. In such cases, besides the continuous changes in the propellers angular speed, each rotor is affected by the unsteady aerodynamic field surrounding the rotor itself. Thus, thrust aerodynamic variations must be taken into account during the modelling. In the literature it is possible to find few references that properly describe the aerodynamics

of multirotors, see [27], [28] and [29]. Before proceeding further some quantities need to be defined. In the rotor coordinates it is possible to define three velocity components denoted by μ_x, μ_y and λ_c. The components μ_x and μ_y represent the non-dimensional forward and side-ward rotor velocities and are combined to define a resultant forward velocity μ in the wind-axes system

$$\mu = \sqrt{\mu_x^2 + \mu_y^2}. \tag{2.55}$$

Thus in the wind-axes system the rotor encounters two velocity components, μ and λ_c. Then, the total inflow through the rotor is represented by $\lambda = \lambda_c + \lambda_i$. Finally, one has to relate the wind-axes coordinates with respect to the rotors one. Given δ the angle between rotor fixed reference frame and wind-axis coordinates, it is possible to define

$$\sin(\delta) = \frac{\mu_y}{\mu}, \tag{2.56}$$

$$\cos(\delta) = \frac{\mu_x}{\mu}. \tag{2.57}$$

2.2.1 Thrust and inflow

The equations presented in Section 2.1 are written in the fixed rotor reference frame $\mathscr{F}_{R_i}$. According to the small dimensions of the multirotors taken into account, many reasonable assumptions can be made to simplify the aerodynamic model of the rotors. The main assumption is to consider just the uniform variation of inflow due to c_T variations, because the smaller the rotor and the more negligible are the lateral and longitudinal inflow contributions. It is then possible to re-write equation (2.47) as

$$\begin{aligned} M_{(1,1)}\dot{\lambda}_0 + \hat{L}^{-1}_{nl(1,1)}\lambda_0 &= c_T, \\ \frac{8}{3\pi}\frac{1}{\omega}\frac{\mathrm{d}\lambda_0}{\mathrm{d}t} + v_T\left(\frac{2a_3}{2a_1^2 + a_3}\right)\lambda_0 &= c_T, \end{aligned} \tag{2.58}$$

and equation (2.51) as

$$\lambda_i = \frac{1}{2}\tilde{L}^{-1}_{(1,1)}\lambda_0 = \frac{1}{2}\left(\frac{2a_3}{2a_1^2 + a_3}\right)\lambda_0. \tag{2.59}$$

Equations (2.58) and (2.59) show how to compute λ_i but not how to express the forcing term c_T. The most reasonable approach is to express the thrust coefficient as the one resulting from BET in forward flight as presented in equation (2.29). The fully nonlinear dynamic model for the uniform inflow variation is obtained as

$$\frac{8}{3\pi}\frac{1}{\omega}\frac{\mathrm{d}\lambda_0}{\mathrm{d}t} + v_T\left(\frac{2a_3}{2a_1^2+a_3}\right)\lambda_0 = \frac{1}{4}\sigma C_{L\alpha}\left[\frac{2}{3}\theta_0\left(1+\frac{3}{2}\mu^2\right)-\lambda\right]. \tag{2.60}$$

Finally the thrust force can be computed as

$$f = c_T \rho a r^2 \omega^2. \tag{2.61}$$

2.2.2 Torque and drag forces

Once the thrust coefficient and the inflow have been computed as described in the previous subsection, it is possible to compute first the H-force coefficient c_H and then the torque coefficient c_Q using equations (2.32) and (2.33) from the BET approach as shown in subsection 2.1.3. Finally, the H-force and the torque can be computed as

$$f_h = c_H \rho a r^2 \omega^2, \tag{2.62}$$

$$\tau = c_Q \rho a r^3 \omega^2. \tag{2.63}$$

2.3 Geometric parametrisation of the propeller

It is common to characterise the propellers widely used on multirotor unmanned aerial vehicles by their radius, pitch and number of blades. Besides these parameters, the airfoil, the chord and the twist of the propeller must be described to make use of the aerodynamic theories. Having identified the radial distribution of the chord and the twist with the propeller airfoil, the calculation of the produced aerodynamic forces and moments becomes straightforward.

From now on, it is assumed to use for the chord and twist distributions the ones from GWS SlowFly propellers assuming also a constant airfoil (NACA0015) throughout the blade. The distributions and the analytic relationship between them and the propeller radius are documented in [30] and [2].

2.3.1 Chord & twist distribution

Brandt [30] reports a series of wind tunnel tests done on low Reynolds number propellers, which are widely used in multirotor applications, to document their performance characteristics. The dimensional data of the propellers have been measured throughout this procedure. Figure 2.5 shows the chord and twist distribution of a GWS SlowFly propeller.

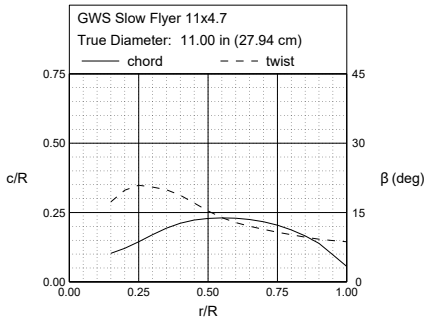


Figure 2.5: *Geometric data of GWS SlowFly* 11×4.7 *[30]*

The relationship between the blade radial position and its chord and twist at that exact position has been parametrized by Bershadsky [2]. This approach eases the use of propellers with different dimensions. Equation (2.64) describes the nondimensional chord while equation (2.65) demonstrates the physical twist angle, measured in degrees, with respect to the nondimensional radial position:

$$c_R = p_1 {r_R}^3 + p_2 {r_R}^2 + p_3 r_R + p_4, \tag{2.64}$$

where p_1, p_2, p_3 and p_4 are equal to -0.2872, -0.1637, 0.4551 and 0.05648 respectively.

$$\beta = b_1 {r_R}^3 + b_2 {r_R}^2 + b_3 r_R + b_4, \tag{2.65}$$

where b_1, b_2, b_3 and b_4 are equal to 30.322, -64.731, 23.008 and 20.558 respectively.

Bershadsky [2] also states that the physical twist of a propeller blade depends on the propeller pitch value. Propeller pitch is known as the distance covered by the propeller along its rotation axis by one revolution,

measured in inches. The twist of the propeller is effected by its pitch and needs to be scaled accordingly. The twist of GWS propellers at 0.6 nondimensional radial position for different values of propeller pitch are shown in Figure 2.6. Scaling the propeller twist with the propeller pitch indicates the correct twist distribution of the propeller in use and this procedure is vital to estimate the angle of attack of the blade accurately in every radial position.

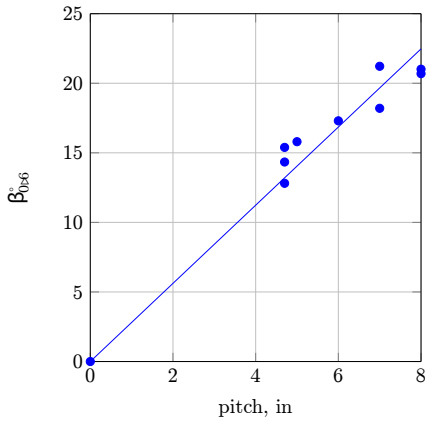


Figure 2.6: *Twist of GWS propellers at* 0.6*r with respect to different propeller pitch values [2]*

2.3.2 Airfoil selection

Another aspect that has a significant effect on the performance of a propeller is its airfoil. The propellers in the UAV market differ a lot at airfoils in use; furthermore, a lot of them do not contain a single airfoil but the shape varies throughout the blade. Hardened by these concerns, selecting a constant airfoil shape for all kind of modeled propellers in such a design tool is not the most accurate but the most ideal way. Different airfoil usage demands a wide range of airfoil data and implementation issues occur during the process. The creation of such a data set is not the best choice from the computational point of view. Selecting an airfoil which one can access to its data easily is advantageous to reduce the effort of development. The NACA0015 airfoil is used in this thesis to model the propeller

airfoil.

Constructing the propeller with an airfoil which is not the exact one used in the real situations causes the problem of inaccurate computation of the aerodynamic forces and moments. NACA0015 differs from most airfoils used on the propellers in the market from the camber point of view. NACA0015 is a camberless, symmetric airfoil whereas the propellers in the market are usually composed of airfoils with non-zero cambers, especially high-pitch propellers (see [2]). The computed thrust and power by the aerodynamic loop may become greatly inaccurate when the propeller modelled does not have a similar airfoil to NACA0015 in real.

Being easy-to-access and used by the ancestor tools makes NACA0015 a suitable choice for the developed tool in this thesis. The lift and drag coefficients of every single radial position is found by the corresponding angle of attack values considering NACA0015. The data are taken from Sheldhal [31] and a web database [32].

CHAPTER 3

Ground effect

In the rotorcraft literature, ground effect is defined as the increase in thrust (at constant power) of a rotor operating near the ground. While ground effect has been studied extensively for conventional helicopters (see [22, 23, 33]), the phenomenon has received limited attention as far as small-scale multirotor helicopters are concerned, even though there is significant evidence that the performance of quadrotors degrades when operating close to the ground surface, both in terms of stability (see [34]) and attitude/position control. In the literature on multirotors, this degradation is generically attributed to ground effect, even though this interaction is often neglected at the modelling stage. However during take-off and landing and, possibly, near ground operation, ground effect is not avoidable and must be handled if one has to achieve satisfactory flight performance in the case of autonomous operation. In this Chapter the results obtained in an experimental campaign aimed at characterising ground effect for a

quadrotor platform will be reported. In particular, the main concern is to assess the mean value of thrust as a function of altitude from ground, *i.e.*, to carry out a classical characterisation of ground effect in the sense of the early literature from the '30s and '40s. More precisely, a number of results have been obtained, in different conditions: first with the platform constrained both in position and attitude, then with one attitude degree of freedom released.

3.1 Classical modelling for ground effect

Ground effect has been studied both numerically and experimentally for almost a century (see [22, 23, 33]). In particular, experiments usually focus on the following conditions:

- Variation of power required at constant thrust: $\left[\frac{p_{IGE}}{p_{OGE}}\right]_f = k_{\bar{f}}$
- Variation of thrust at constant power: $\left[\frac{f_{IGE}}{f_{OGE}}\right]_p = k_{\bar{p}}$,

where coefficients $k_{\bar{f}}$ e $k_{\bar{p}}$ are related by

$$k_{\bar{f}} = \frac{p_{IGE}}{p_{OGE}} = \frac{v_{i_{IGE}}}{v_{i_{OGE}}} = \left(\frac{f_{IGE}}{f_{OGE}}\right)^{-3/2} = k_{\bar{p}}^{-3/2}, \tag{3.1}$$

while IGE and OGE stand for In Ground Effect and Out of Ground Effect respectively. Furthermore, experimentally a dependence on:

- Distance from ground h/r
- Disk loading c_T/σ
- Blade geometry (*e.g.*, taper and twist)

is observed. Some classical expressions for the above mentioned coefficients are provided hereafter

$$k_{\bar{p}} = \left[1 - \frac{1}{(4h/r)^2}\right]^{-1} \qquad \text{Cheeseman \& Bennet,} \tag{3.2}$$

$$k_{\bar{p}} = \left[0.9926 + \frac{0.03794}{(h/2r)^2}\right]^{2/3} \qquad \text{Hayden.} \tag{3.3}$$

3.2 Ground effect in multirotors

While there is increasing attention to ground effect in multirotors, only a small number of references is available at the moment. In [35] it is suggested to use the radius of a rotor equivalent to the sum of the areas of the individual rotor as a geometric parameter to apply classical results for single rotor helicopters. In [36] a model for ground effect based on the expression $f_{input}/f_{output} = 1 - \kappa(r/4h)^2$ is used, where κ is a parameter estimated from experimental data. For the identification of κ, hovering tests at different altitudes have been carried out and it was noted that the effect was measurable for distances up to $h/r = 4$. In [37] it is claimed that ground effect is measurable up to almost $h/r = 6$. It is also suggested to use the classical formulas for conventional helicopters by replacing the main rotor radius with the distance from the outer tip of one rotor to the vehicle's center. In [38] a commercial Draganflyer X8 multirotor, having 4 pairs of coaxial, counter-rotating rotors, is considered. Unlike other studies, the requirement of carrying out the experiments at constant power so as to obtain results comparable with the Cheeseman formula, has been enforced. However, a characterisation in terms of the rotor *RPM* is used. It is concluded that ground effect can be seen for distances up to $h/r = 6$. [34] reports frequency-domain analyses of roll dynamics as a function of altitude and a slow-down of roll dynamics as the distance from ground decreases are reported. [39] presents an extensive analysis on the ground effect both on simulation and on real outdoor flights, in particular the *partial ground effect*, a situation in which one or some of the rotors of the multirotor (but not all) are under the ground effect, is characterised. [40] studies the ground effect for small scale UAVs in forward flight. In [41, 42] is studied the disturbance rejection of the ground effect, while in [43] is presented and tested a model reference adaptive controller for the attitude loop in order to mitigate the ground effect on small quadcopters. Finally, a number of references dealing with the problem of the experimental characterisation of propellers for small-scale multirotors is available, see for example [44] and [45, 46].

3.3 Experimental set-up

Before getting to the multirotor case with four interacting rotors and wakes, the behaviour of one single isolated rotor is studied. This analysis also makes possible a comparison with the above-mentioned classical results from the rotorcraft literature. For the purpose of this analysis, a rotor was removed from the QUAD-X and mounted on a test bed with a load cell (see Figure 3.1) in order to keep a constant distance from the ground (up to $h/r = 7$) and adjust it during the different experiments. The sensors

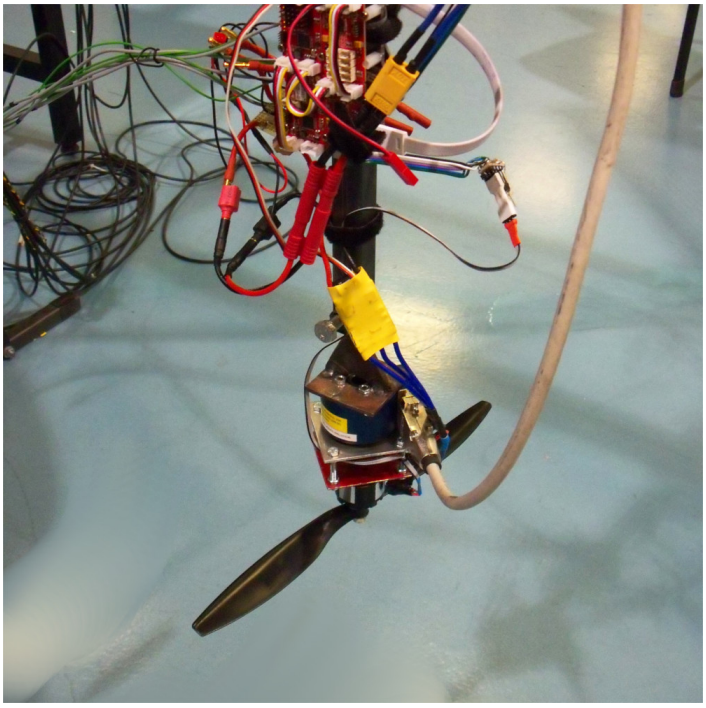


Figure 3.1: *Experimental setup for single-rotor tests.*

used during these experiments were:

- Load cell: a six degrees of freedom load cell has been used in order to measure the vertical force (f) and the longitudinal torques (τ).
- Tachometer: a custom tachometer has been used to measure the ro-

tational speed (ω) of each motor/propeller.

- Current sensor: in order to measure the power electrical consumption (p_E), each motor has been equipped with a voltage and current sensor.

All the data were logged by means of a LabVIEW visual interface and a PXIe-1078 system from National Instruments using two different data acquisition boards: a PXI-6284 for the load cell and current sensors and a PXI-6123 for the tachometer.

For each altitude taken into account, sequences of throttle step commands ($th_{\%}$) were sent to the Electronic Speed Controller (ESC) (see Figure 3.2), and the response in terms of angular velocity, loads, current and voltage on the ESC were recorded, stored, and later processed. In Figure 3.3 an example of the raw data collected for the single rotor out of ground effect over the time is shown. Such data have been used in the

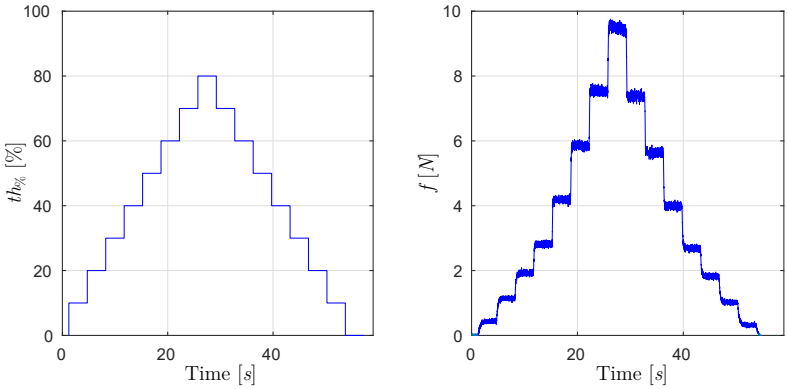


Figure 3.2: *Imposed throttle sequence.*

time domain to evaluate the step response of the rotor. Then, from each of the throttle steps, averages were taken of all relevant variables, in order to obtain a complete and systematic characterisation of the static behaviour at a given altitude. In addition, the energy efficiency of the motor+ESC assembly can be easily evaluated as follows

$$\eta = \frac{p_M}{p_E}, \tag{3.4}$$

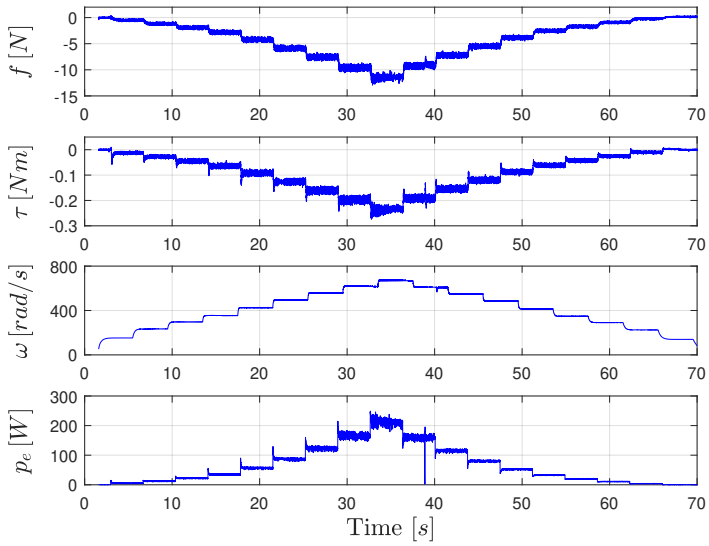


Figure 3.3: *Example of raw data logged during the experiment.*

where the mechanical power can be calculated as $p_M = \omega r$ (as described in the Chapter 1).

Figure 3.4 shows an example of the processed data of the static case of a single rotor. In particular, with a square symbol it has been displayed the relative value corresponding to the hovering condition of each plot.

Instead, for the multirotor case, the QUAD-X platform has been rigidly attached to the test bed (see Figure 3.5). The measurement set-up is the one described is the same presented for the single rotor case. Note that during the tests practical issues with the available set-up (*i.e.*, limitations in the current supply and saturation of the load cell) prevented from reaching 100% throttle, so that results for throttle up to 80% for the isolated rotor and up to 70% for the complete quadrotor are presented.

Finally, the quadrotor was placed on a 1 DoF test-bed set-up (see Figure 3.6) which constraints all the degrees of freedom except for pitch rotation and able to put the vehicle at different heights with respect to the ground. This indoor setup is representative of the actual pitch attitude dynamics in flight for near hovering conditions, see [47]. All the measurements come directly from the on-board Inertial Measurement Unit

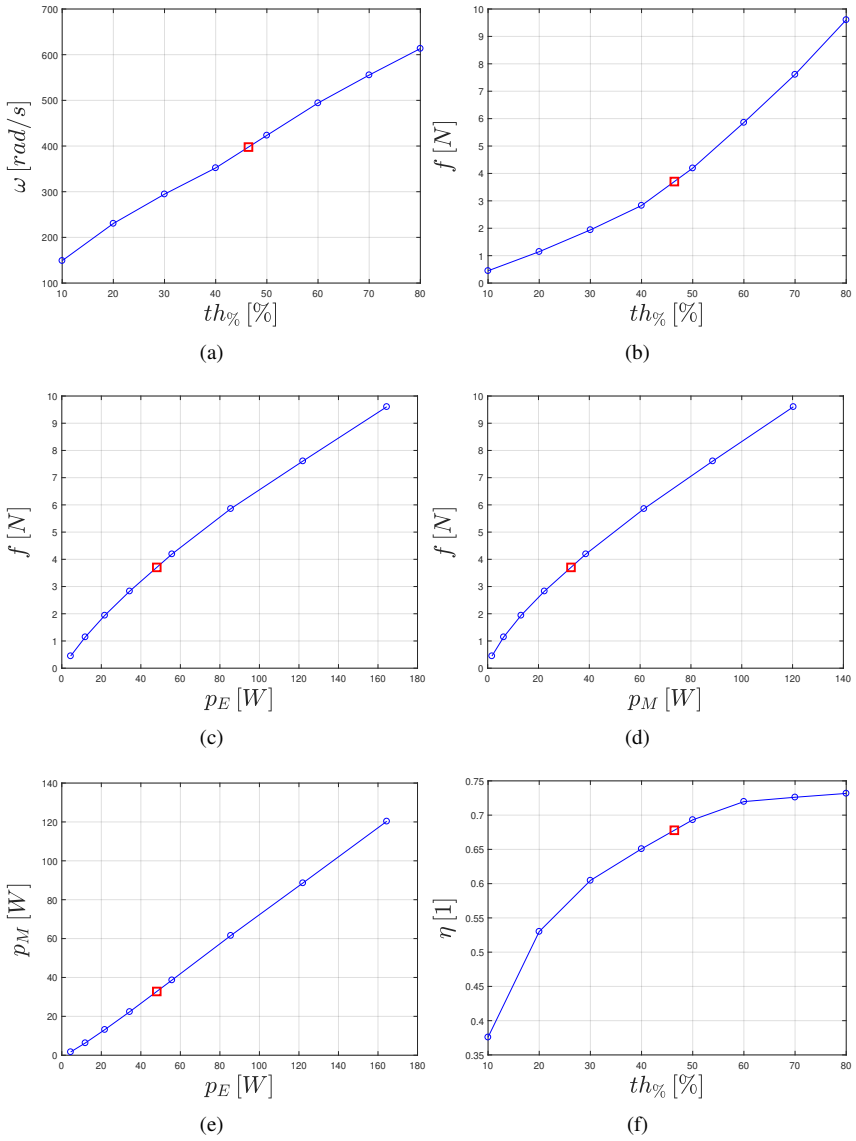


Figure 3.4: *Example of processed data for the single rotor case out of ground effect.*

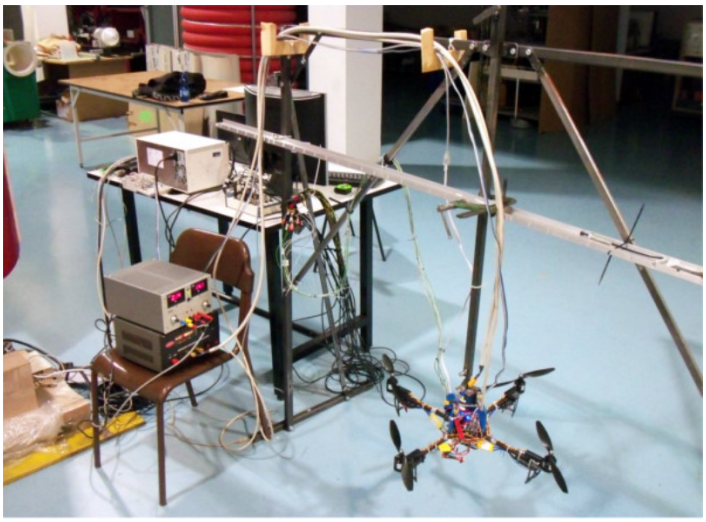


Figure 3.5: *Complete view of the experimental setup.*

(IMU). Control commands were sent to the Flight Control Unit (FCU) directly from a PC connected through a serial port.

3.4 Single rotor results

The resulting data, obtained applying the previously described work flow, can be plotted in graphs where any of measurements between powers, current, voltage, forces and moments can be expressed depending on altitude and throttle. For example, in Figure 3.7, one can see how the thrust (one of the most important parameters) changes with altitude and throttle.

In order to compare the performance of the system at different altitudes, it is useful to keep constant some of the parameters. Traditionally, ground effect has been studied at constant thrust or constant mechanical power ([22, 23]). In this case, along with power and thrust, other variables can be held constant, for example throttle or the input electrical power. In Figure 3.8, one can compare these different ways of seeing the thrust change with altitude, with different variables being fixed. Figure 3.8 was obtained by linearly interpolating the results obtained in dozens of tests. For each altitude, power and thrust or thrust and throttle or throttle and

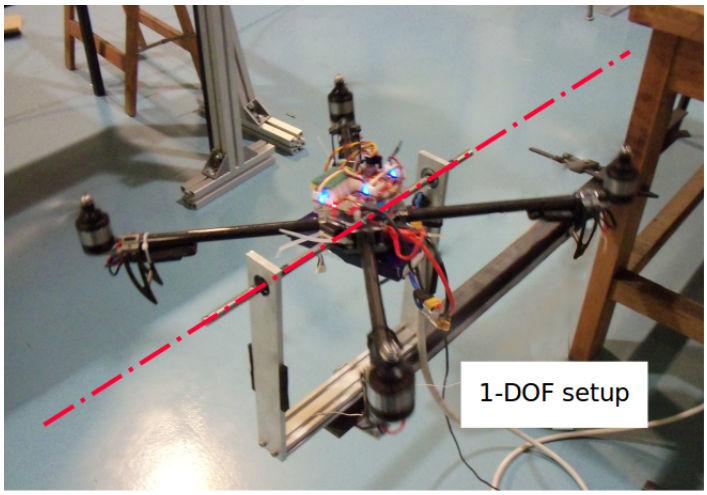


Figure 3.6: *1-DOF set-up.*

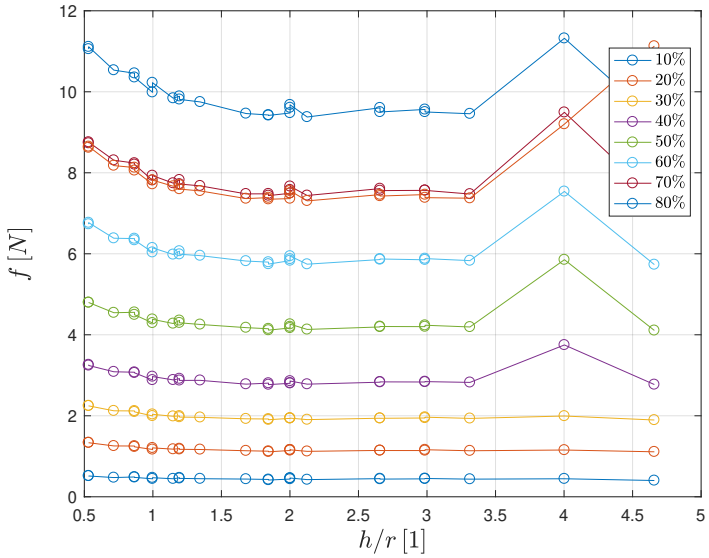


Figure 3.7: *Single rotor: thrust vs throttle and altitude.*

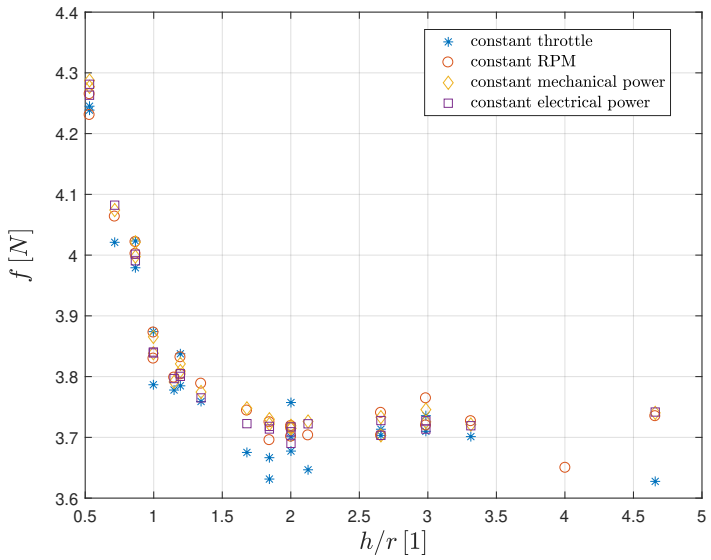


Figure 3.8: *Single rotor: comparison between different ways of presenting the results.*

angular speed data were interpolated; the query points for the interpolation were chosen to be the same required for hover. In the following, for the sake of simplicity all the results are presented at constant electrical power, since the mechanical power cannot be evaluated in the quadrotor configuration due to moments cancellation of opposing motors. From Figure 3.8 it is obvious that keeping constant the mechanical power is nearly equivalent to keeping constant the electrical power or the throttle.

Now with all these data available, a comparison with the classical models from the rotorcraft literature becomes possible. More precisely, in Figure 3.9 one can see a plot of the thrust at constant electrical power, compared with the estimates of Cheeseman & Bennet and Hayden models (see Section 3.1 for details). It should not come as a surprise that there isn't a perfect agreement between models and experimental data; in fact the models are based on hypothesis of inviscid potential flow whereas the experimental data is concerning a rotor working at very low Reynolds numbers. One can also notice that for $h/r = 4$ there is an outlier which is the result of a measurement error and must not taken into account for a physical analysis. The main conclusion one can draw is that for altitudes

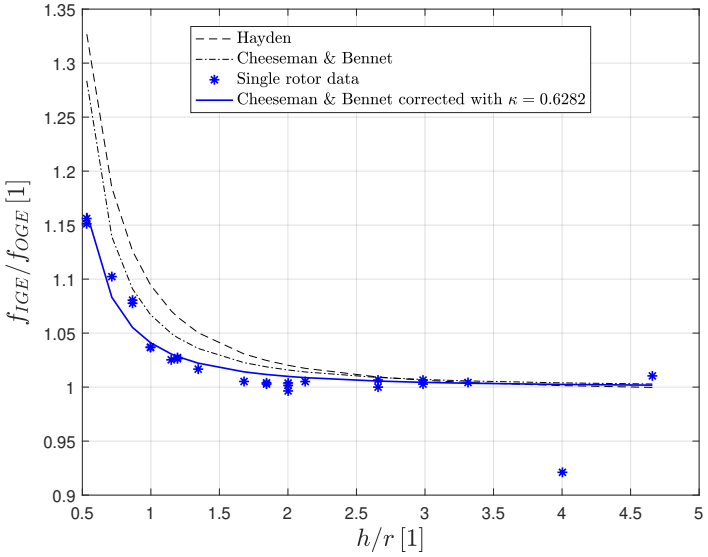


Figure 3.9: *Single rotor: comparison between experimental data and classical curves, at hovering power.*

greater than $h/r = 2$, ground effect is nearly negligible (less than 1%).

Similarly to [36] it has been decided to find a model that may match the data acquired better than the classical ones. To do so, to the Cheeseman & Bennet model a gain $\kappa \in \mathbb{R}_{>0}$ has been introduced which has been identified using a least squares method

$$k_{\bar{p}} = \left[1 - \frac{\kappa}{(4h/r)^2}\right]^{-1}. \tag{3.5}$$

The fit of the data obtained with the model in equation (3.5) is shown in Figure 3.9.

Furthermore, a dual way to express the ground effect, as previously discussed, is to evaluate the mechanical power necessary to hover at different altitudes (constant thrust). In Figure 3.10 the data acquired compared with the classical models and the improved one is shown.

Finally, for what concerns the dynamic analysis of the single rotor, in Figure 3.11 the transients of thrust f and torque τ for a single rotor, following the application of a step of throttle percentage are shown. As can

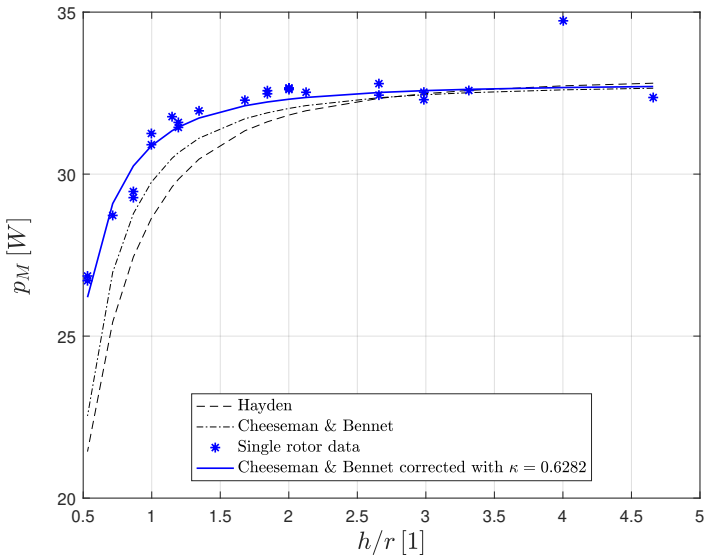


Figure 3.10: *Single rotor: comparison between experimental data and classical curves, at hovering thrust.*

be observed, the thrust response can be modelled as a first-order dynamics. Note that this is clearly not true for the torque response: a possible reason for this discrepancy could be associated with the torsional flexibility of the structure connecting the quadrotor to the load cell.

Focusing on the identification of first order models for the thrust response, a least squares approach has been used, to fit the analytical response for thrust f, given by

$$f = A_0 + \sum_{i=1}^{N} step(t - T_i)\, A_i e^{\frac{t-T_i}{\tau_i}}, \tag{3.6}$$

with the measured data, where A_0 is the static thrust level, T_i the starting time of each step, A_i its amplitude and τ_i the time constant to be identified. Figure 3.12 shows the identified function superimposed to the experimental data. In Figure 3.13 the values of the estimated time constants for each step (rising and decreasing), obtained at different heights, are presented. From such results one can see that the time constants for positive and negative steps are significantly different: the main reason for this behaviour

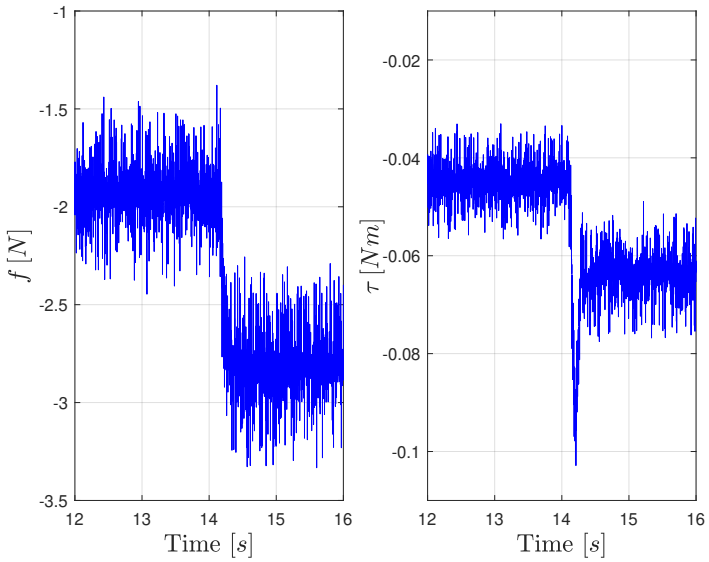


Figure 3.11: *Transients of rotor forces (left) and moments (right) applying a motor throttle step.*

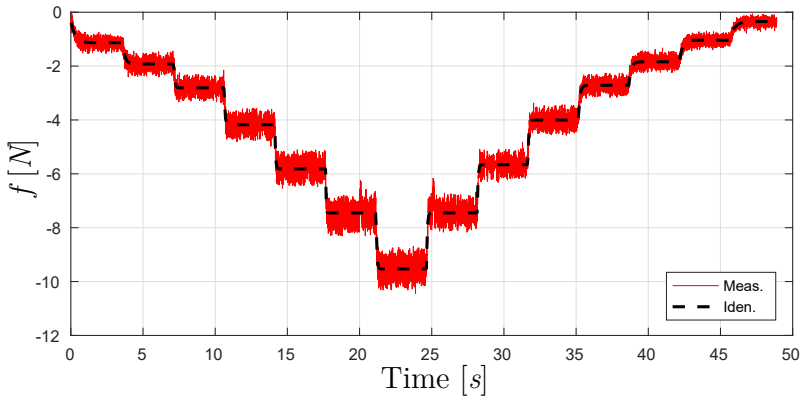


Figure 3.12: *Data acquired in a campaign at fixed height (red line) and identified response (blue dashed line).*

is that the motor ESCs do not have any active brake implemented so when the throttle command decreases the time constant of the thrust response is slower. The dynamics also depends on the throttle level: the higher the throttle, the faster the motor response. Finally, no clear dependence of the time constants on height appears from the data.

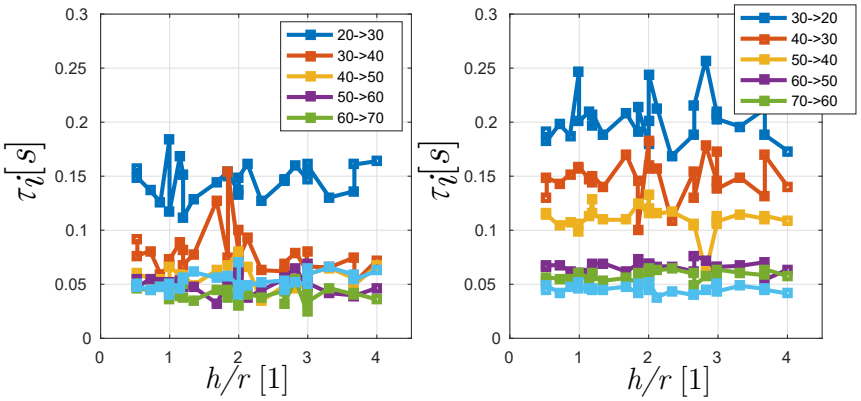


Figure 3.13: *Identified time constants, single rotor case: rising steps (left) and decreasing steps (right).*

3.5 Complete quadrotor results

In this section the experimental results obtained considering the complete quadrotor will be presented and discussed. The aim is to characterise the ground effect on global rotor performance parameters and put in evidence the differences respect to the previous analysis conducted on the isolated rotor.

The QUAD-X is assembled using COTS low-cost components then it is not guaranteed that the four propulsion systems composed by ESC, brushless motor and rotor are close to each other in terms of performance; in fact the noticed discrepancies are not negligible. In particular the motors appear to be the components with the wider variability. In order to evaluate these differences, the same experiments carried out on the isolated rotor were repeated on the quadrotor but activating only one rotor at a time (on a limited range of throttle, in order to not overload the load

cell). The results for the four rotors are collected in Figure 3.14. Tak-

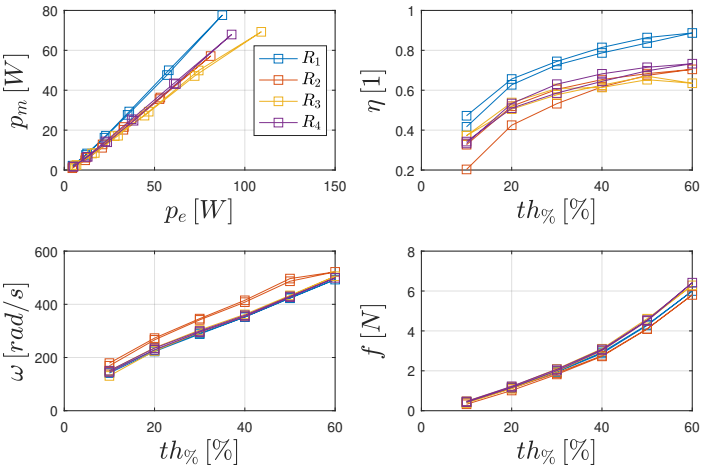


Figure 3.14: *Complete quadrotor: comparison between each ESC and motor+rotor system performance.*

ing into account the above issue and the necessity to define some global parameters able to characterise the quadrotor working state for different altitudes from ground, the following assumptions will be adopted hereafter:

- *Throttle*: the input command to the ESC is identical for each propulsion unit;
- *Power*: it is considered as the sum of the power required by each rotor, $p_E = \sum_{i=1}^{4} p_{E_i}$;
- *Thrust*: it is considered as the sum of each rotor thrust, $f = \sum_{i=1}^{4} f_i$;
- *Angular velocity*: it is considered as the average of the angular velocity of each rotor, $\omega = \frac{1}{4}\sum_{i=1}^{4} \omega_i$.

Using the above global quantities instead of the single ones, the obtained results are clearly repeatable as shown in Figure 3.15, where the results of more than 100 experiments at different altitudes are reported.

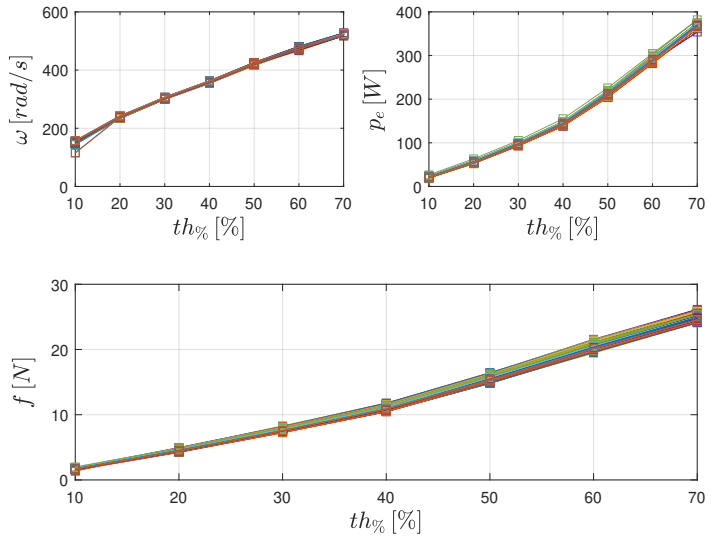


Figure 3.15: *Complete quadrotor: superimposed data of more than* 100 *experiments at different heights.*

The same test procedure previously described for the isolated rotor characterisation was adopted for the complete quadrotor case: at each altitude from ground different values of throttle (from 10% to 70%, step 10%) were tested and all the available measures were logged. For example, the rotor thrust as a function of altitude and throttle is shown in Figure 3.16.

As reported in Figure 3.17, the complete quadrotor case is quite different from the isolated rotor one: this is clearly due to the aerodynamic interaction effects among the four rotors. As a results, the effect of the ground on the thrust is extended up to almost 4 rotor radii of height, doubling the limit of about $h/r = 2$ found for the isolated rotor test. As well as in the single rotor case, a model fitting the acquired data has been identified by means of a least squared method with a sixth order polynomial, which result has been reported in Figure 3.17. It is worth to observe that the rotors aerodynamic interaction is present in spite of the expected strong wake contraction characterizing very small and hence low Reynolds rotors such as the considered ones (up to $4:1$ wake area contrac-

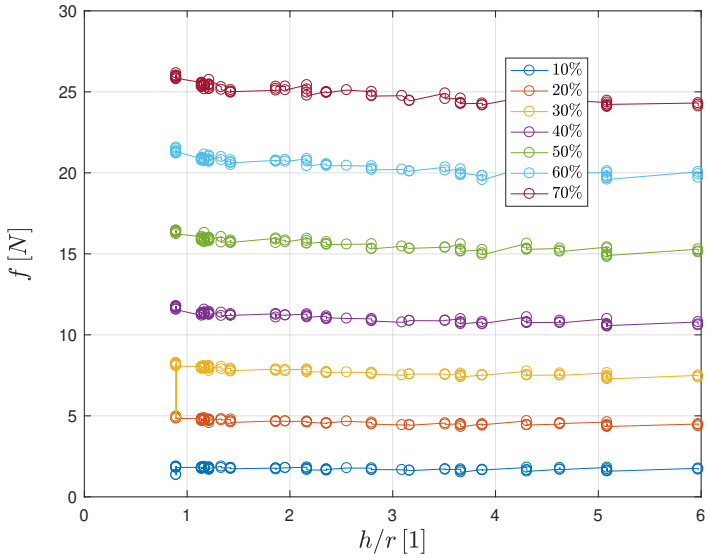


Figure 3.16: *Complete quadrotor: thrust vs throttle and altitude.*

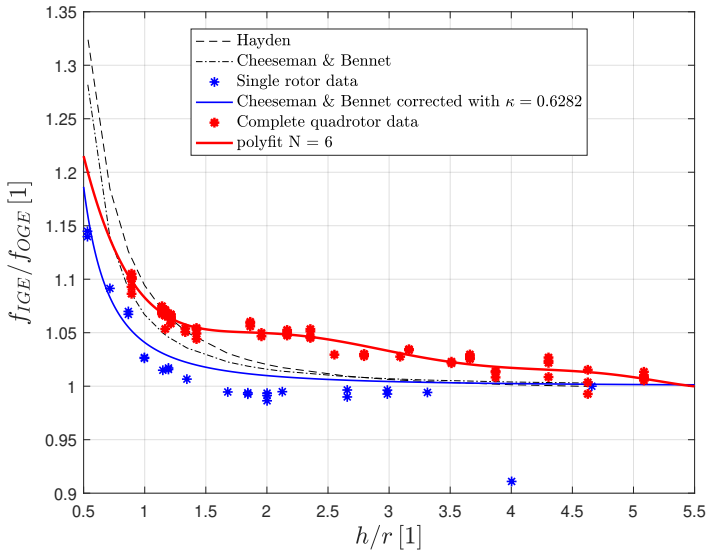


Figure 3.17: *Non-dimensional rotor thrust at constant power as a function of non-dimensional altitude from ground.*

tion with respect to rotor area, instead of classical 2 : 1 value for full-scale helicopters). It is likely that a different behavior may be observed varying the Reynolds number. Moreover it can be observed that the experimental data qualitatively follows the trend of the classical formulations in the isolated rotor case while for the complete quadrotor this is no longer true: hence it seems that the classical formulas valid for full-scale helicopters are not able to model correctly the ground effect for small multi-rotor vehicles. Moreover it does not appear straightforward to extend the isolated rotor results to the case of interacting rotors. Further experimental activities have to be carried out, varying also the distance among the four rotors, in order to characterize the rotors interaction impact on the ground effect phenomena of small quadrotors.

A secondary effect not considered in this preliminary characterization study but potentially affecting the ground effect behavior is the download phenomena due to the rotors inflow that impinges on the airframe elements. An interesting and exhaustive work about the topic, but considering a quad tilt-rotor architecture was presented in [48] concerning the experimental campaign and in [49, 50] about the corresponding CFD analysis: the various mutual interactions between rotors-fuselage/wing-ground were investigated, characterizing the transition from the penalizing download affecting the rotorcraft OGE (14% of total thrust) to the beneficial upload (5% of total thrust) entering in ground effect.

The presence of the quadrotor airframe (fundamentally the multirotor arms) surely influences the obtained results but the amount is proportional to the ratio of the impinged surface under the rotor and the rotor area: clearly for the quad tilt-rotor the ratio is particularly large (because of the wing and fuselage) while in the quadrotor case it is likely weak, therefore this secondary effect can be neglected as first instance when the aim is characterising the global dynamic response of the quadrotor in ground effect for control design purposes.

In conclusion an interesting validation of the obtained results on the complete quadrotor can be performed referring to one of the few works available about quadrotors in ground effect, [51]: the vehicle was smaller with respect to the considered one (rotor radius equal to 4 cm) and the data was gathered directly in flight, simply measuring the rotors angular

velocity reduction to maintain the hover when approaching the ground. The same plot was obtained from the data collected in this work, see Figure 3.18, representing the average rotors angular velocity as a function of height at constant thrust.

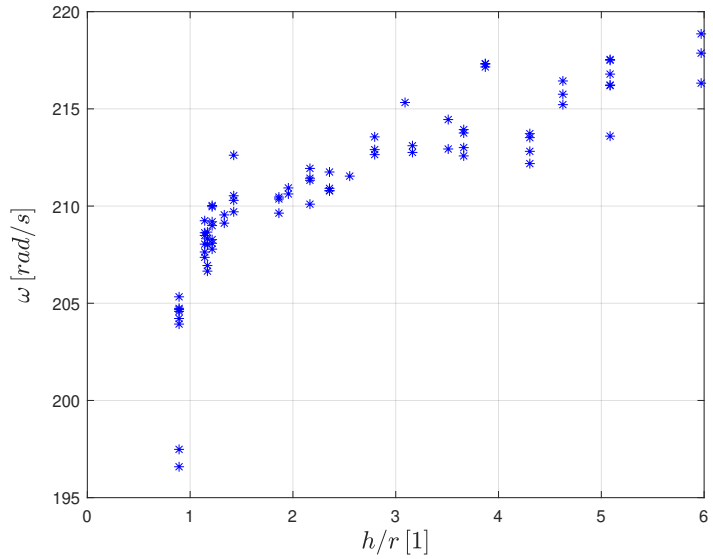


Figure 3.18: *Average rotors angular velocity as a function of non-dimensional height at constant thrust*

Comparing Figure 3.18 with the results of [51] in Figure 3.19 it is evident that the trend is qualitatively the same, in particular:

- for $h/r \geq 6$ the angular velocity remains almost constant varying the altitude from ground;
- for $4 \leq h/r \leq 5$ a small and abrupt angular velocity discontinuity occurs;
- for $1 \leq h/r \leq 4$ an almost linear variation of the angular velocity with the height from ground is evident;
- for $h/r \leq 1$ a steep angular velocity reduction approaching the ground occurs.

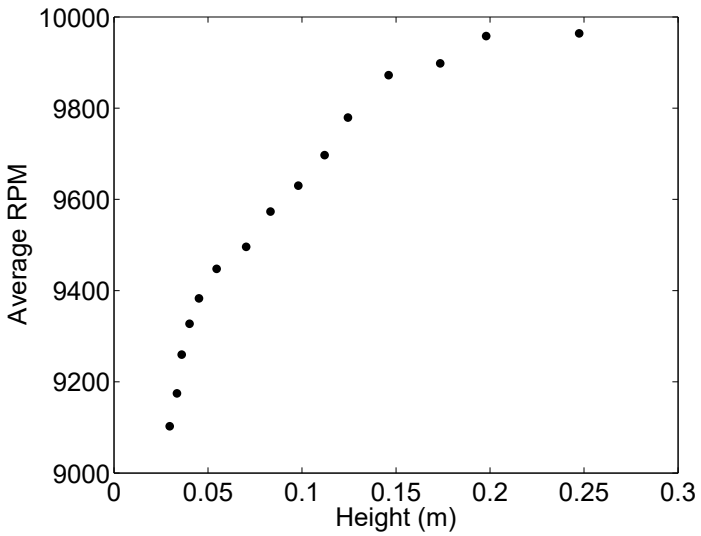


Figure 3.19: *Angular velocity at constant thrust. Each data point is based on 10 seconds of hover data for a vehicle with propellers with* $r = 4\,\text{cm}$ *[51].*

An analogous behaviour characterized by distinct segments can be recognized also in the previous Figure 3.17.

Finally, the same dynamical model identification performed on the data collected with the single rotor has been repeated for the four rotors mounted on the quadrotor frame and the results are reported in Figure 3.20. The same comments for the single rotor tests apply also to this case. In particular, it is not possible to recognize an evident dependency of the time constants from the height from ground. Moreover comparing the single rotor case with the four rotor one, no effect of the aerodynamic interaction between the rotors and the interference of the quadrotor airframe appears on the motors dynamics.

3.6 Pitch attitude dynamics in ground effect

In this section the results of the conducted quadrotor pitch attitude dynamics identification varying the height from ground will be presented and discussed. The experiments have been carried out in laboratory con-

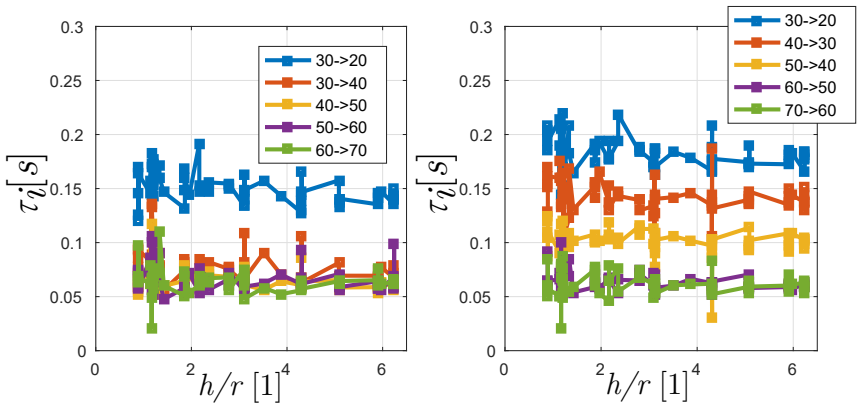


Figure 3.20: *Identified time constants, four rotors case: rising steps (left) and decreasing steps (right).*

ditions, with the quadrotor placed on the test-bed described in Section 3.3.

A characterization of the pitch dynamics for the considered quadrotor platform was performed in [52], in OGE conditions: as excitation signal a Pseudo Random Binary Sequence (PRBS, see [53]) was selected and applied in open-loop experiments. A grey-box approach was adopted to identify the unknown parameters of the *a-priori* assigned first order structure for the SISO model from the control variable to the pitch angular rate.

In the considered experimental framework however the open-loop approach turned out to be critical, as the bench test attitude limits were frequently reached by the quadrotor and there was no way to ensure that the vehicle maintains on average a null pitch angle during the oscillations, hence on average the same distance of the rotors from the ground plane. Therefore, while in the present study a PRBS excitation sequence was still used, it was applied in closed-loop. Regarding the identification algorithm, a black-box approach was chosen, in particular the PBSID (Predictor Based System IDentification) subspace model identification method (see [54]), in view of its ability to deal with data generated in closed-loop: a short overview of this technique is reported in Appendix A.

3.6.1 Identification: experiments and procedures

The PRBS excitation signal was applied according to the block diagram in Figure 3.21, combined to the attitude controller feedback action (with a null pitch angle set-point). The control variable is defined as in

$$u = \frac{1}{2}\left[(th_{\%,1} + th_{\%,2}) - (th_{\%,3} + th_{\%,4})\right] \tag{3.7}$$

where $th_{\%,i}$, $i = 1,\ldots,4$ is the throttle percentage command to the ESC of the *i-th* rotor.

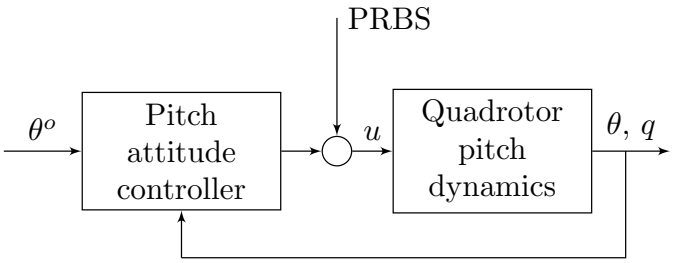


Figure 3.21: *Excitation signal introduction in the control scheme.*

The parameters of the PRBS, *i.e.*, signal amplitude and min/max switching interval, were tuned to obtain an excitation spectrum consistent with the expected dominant dynamics. An example of excitation sequence is shown in Figure 3.22, together with the total control variable and the measured quadrotor pitch angular velocity response. A key metric to verify that the system has been properly excited across the entire range of interest is the coherence function: as can be seen in Figure 3.23 the result is satisfactory (greater than 0.6) on a reasonably wide and adequate range of frequencies.

The 25 s long excitation was applied for 20 test repetitions for each of the 11 considered values of height from ground, logging with a sampling time of 0.01 s the manipulated variable u and the pitch angular rate q. Then the PBSID algorithm has been applied to identify SISO models having as input u and as output q. The investigated distance from ground ranges from 38 *cm* ($h/r = 2.49$, the lowest possible value in order to en-

sure a sufficient pitch angle range on the test-bed) to 120 cm ($h/r = 7.87$, fully OGE).

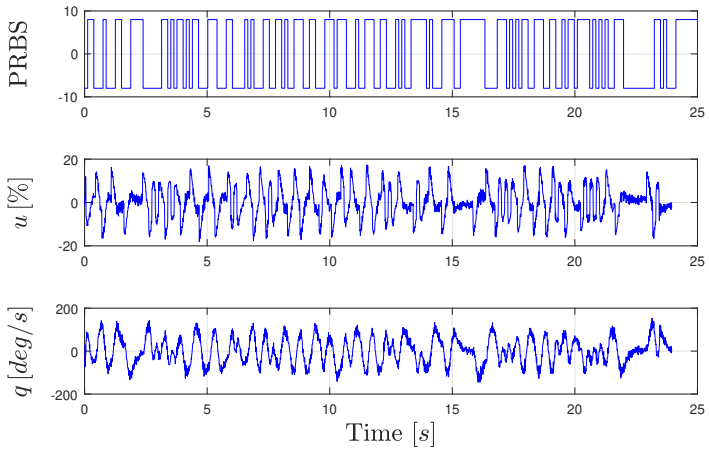


Figure 3.22: *PRBS excitation sequence, total control variable u (PRBS + attitude controller feedback action) and measured quadrotor pitch angular rate q, logged during an identification test.*

The presence of a time delay in the plant dynamics has been accounted for by means of a forward shift of a proper number of samples on data-set input signal u. The overall delay of the control loop implemented on board (from IMU measurements, through acquisition and processing, to actuation of rotors angular velocity) was estimated in 0.05 s (5 samples), evaluating the initial inverse behaviour of an identified model step response without any input data shift. The order of the identified models was fixed to 2 based on the inspection of the singular values in (A.17), with a cross-validation approach. Similarly, in order to select the parameters of the PBSID algorithm, *i.e.*, past window length p and future window length f, the identification on each experimental data-set has been carried out for several values, in a predefined range, for p and f; the performance of each obtained model has been assessed in terms of the Variance Accounted For (VAF) indicator

$$VAF = max\left\{1 - \frac{var(q - q_{est})}{var(q)}, 0\right\} \times 100\% \tag{3.8}$$

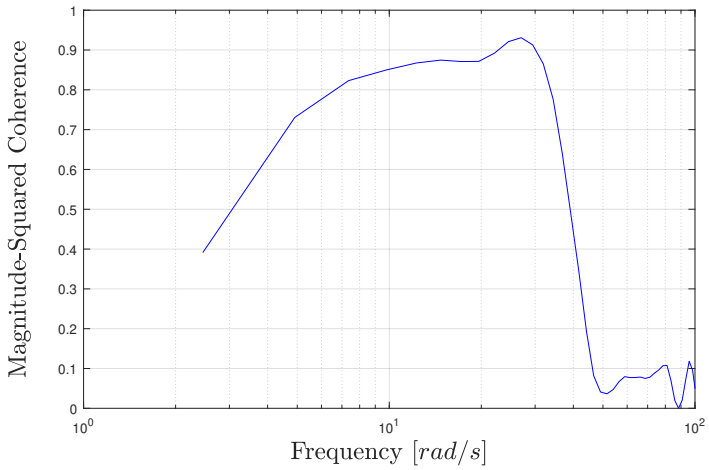


Figure 3.23: *Coherence function of a measured dataset.*

for the simulated response q_{est} of the estimated models respect to the measured angular rate q. Then, the combination of past and future window length that maximizes the VAF has been retained.

In Figure 3.24 is reported an example of one of the best matching in time domain between the estimated and measured angular rate on a cross-validation data-set.

3.6.2 Results

The analysis of the results aims at verifying if a variation of the identified dynamics as a function of the distance from ground is present, and, if so, at characterizing it. The Bode plots of the frequency response functions for the identified models for all tested heights from ground are shown in Figure 3.25, considering only the results obtained from experimental data-sets that guarantee a VAF greater than 85%. It is evident that all the models are close to each other in terms of magnitude, especially in the range of frequencies where the coherence function is higher and hence the best model accuracy can be expected. The phase plots, on the other hand, show that the location of the poles varies from one test result to another.

Analysing the location of the poles of the identified second order mod-

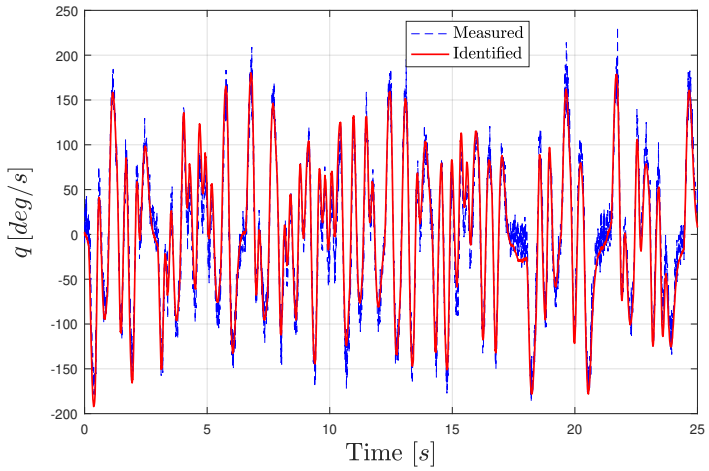


Figure 3.24: *Example of cross-validation of an identified model: VAF=92%.*

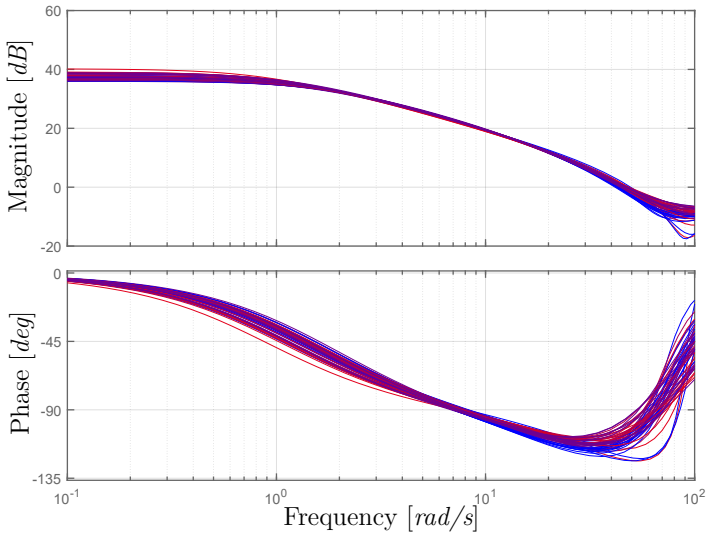


Figure 3.25: *Bode diagram of the most reliable identified models for all tested heights from ground (color scale: blu=OGE, red=IGE).*

els as a function of the distance from ground (again, considering for each height only data-sets producing a VAF$>85\%$) the results reported in Figure 3.26 are obtained. The dominant pole, representing the pitch attitude dynamics, becomes slower when reducing the distance from ground. On the other hand, the second pole, corresponding to the actuators dynamics, becomes faster when the height decreases. Therefore, it appears that besides affecting the rotors in a static sense, as is well known in the rotorcraft literature and also verified in Sections 3.4 and 3.5, the distance from ground has an impact also on the attitude dynamics (although in a limited way and at very close distances from ground). In ground effect a decrease in the rotor height produces a rotor thrust increase that acts as a spring against the height change, with increasing stiffness as the ground approaches: considering the side-by-side rotors configuration of the quadrotor, the antisymmetric height change of opposite rotors associated with pitch (or roll) rotation implies an alteration of the pitch (or roll) attitude dynamics as a function of height in ground effect. Such results have been also confirmed by the numerical results provided by [55] which adopts an analytical model of the attitude dynamics including the dynamic inflow in ground effect.

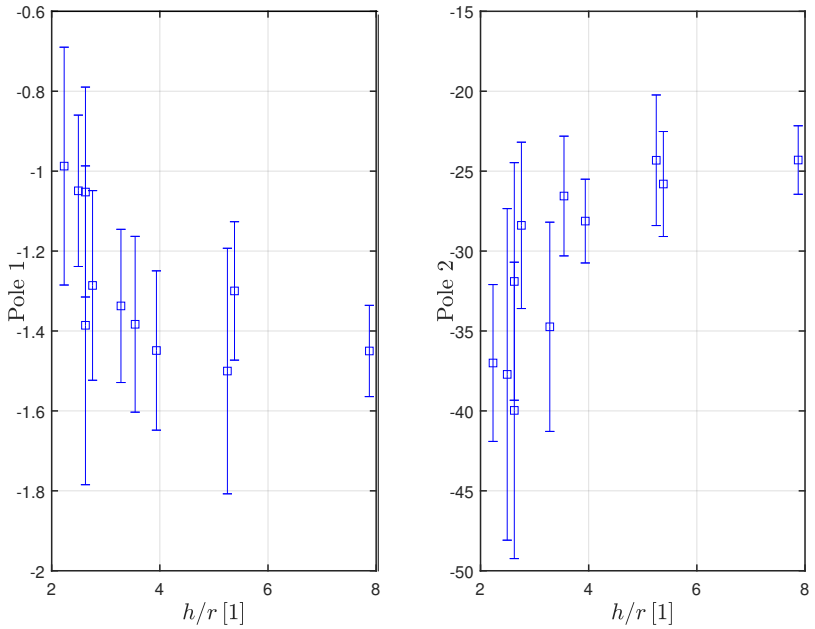


Figure 3.26: *Poles of the the most reliable identified models varying the distance from ground: the bar represents the standard deviation around he mean value of considered model for each height.*

CHAPTER 4

Design of multirotor UAVs

The goal of this chapter is to present an approach, and a corresponding tool, to support the design of multirotor UAVs.

In particular, first an analysis approach, which estimates the performance capabilities of the vehicle by the prescribed drive components is presented. Subsequently, the analysis method is used to formulate and solve design problems, *i.e.*, to find a suitable configuration which is able to achieve the prescribed performance and weight requirements. These two approaches can be considered as antipodes and one can be used to validate the other.

4.1 State of the art

Nowadays, most of the multirotor UAV designers use heuristic approaches to construct these platforms. These design procedures can only take advantage of previously constructed, similar platforms' data and can be vali-

dated through experimental procedures without preliminary aerodynamic, structural and electrical analysis.

The most important part of designing a multirotor unmanned aerial vehicle is the selection of the so-called *drive components*. These components, which are the electric motors, Electronic Speed Controllers (ESCs), batteries and propellers, are the most significant parts that affect the performance. There are several aspects that need to be considered during the selection of these components to achieve the desired performance.

Currently, a variety of design methods have been proposed by several academics. A design process to select the drive components that makes use of Blade Element Momentum Theory is proposed by [1], but it does not perform an optimization regarding performance objectives. [2] documents a parametrization technique to estimate the masses of the drive components and a design tool to size the propulsion system of a multirotor UAV. This thesis makes use of the same mass estimation process combined with BEMT and electrical parametrization techniques reported by [3] and [4] to develop a design and optimization tool for multirotor UAVs, which mainly eliminates the need of a database for the drive components and provides the capability to optimize a configuration that takes into account the performance requirements for the vehicle in-design.

4.2 Component parametrisation

Multirotor unmanned aerial vehicle drive systems involve four components: electric motors, Electronic Speed Controllers (ESCs), batteries and propellers. These four components can be named as *drive components*, the parts which are necessary for a multirotor UAV to fly. The selection of these components is the main consideration while designing a multirotor vehicle to achieve the desired performance capabilities.

Parametrizing drive components to estimate the total capability of multirotors is the principal design consideration for UAV applications. These components have one or more parameters that describe their masses and performance capabilities. Several researches (see [2] and [3]) have been made to express the components' masses by their critical parameters. In particular, [2] documents a method to characterize the masses of drive

components by use of a market search and curve-fitting of all the data to find a general equation that hinges upon the main parameters of the components. This thesis makes use of this method to estimate the masses of the drive components.

Moreover, the electrical model of the vehicle must be specified to calculate the combined maximum capability of drive components. In order to model the circuit, the internal resistance of drive components must be found. Researches have been made in order to parametrize not only the mass but the internal resistance of such components. This thesis makes use of the parametrization process documented by [3] and [4].

4.2.1 Electric motors

Brush-Less Direct Current (BLDC) motors are the primary choice of electric motors for multirotor applications. This type of electric motors is more efficient than brushed configurations from the electrical-mechanical energy conversion point of view (see [2]). There are two types of BLDC motors: Out-Runner (OR) and In-Runner (IR). The difference between these two types is that the OR configuration produces more torque than the IR configuration even though it covers more area. Consequently, OR motors are more suitable for large vehicles and IR motors are best for lighter and smaller vehicles.

The most important parameter of BLDC motors is the speed constant, k_v. The speed constant states the angular rate the motor can reach for a given voltage applied to it (units are usually given as RPM/V). As an example, a BLDC motor with a low k_v value must spin faster than one with a higher k_v to produce the same amount of thrust but too high values of k_v may cause an inability to handle the loads at high throttle inputs. High k_v motors are preferable for fast-manoeuvring flights while low k_v motors mounted under a large radius/low pitch propeller are more efficient.

Mass parametrisation

Figure 4.1 illustrates the relationship between the speed constant of a motor and its mass both for IR and OR BLDC motors. As reported in [2], 991 OR and 696 IR motors have been considered in this survey. The blue and

red lines show the curve-fit of the data and an equation has been obtained. According to [2], the mass of the motor, in grams, can be predicted by

$$m_m = 10^{p_1} {k_v}^{p_2}, \tag{4.1}$$

where coefficient p_1 is 4.0499 for OR motors and 4.4482 for IR motors and p_2 is -0.5329 for OR motors and -0.5242 for IR motors. It is also stated that the fit does not capture quite accurately the mass of the motors between 0 and 500 k_v, which are the mostly used type for "hobby-sized" vehicles (see [2]). In order to closely estimate the mass of the motors belonging to this range, a correction has been made in terms of curve-fitting for OR motors between 0 and 6000 k_v. Figure 4.2 shows the k_v-to-mass relationship after modification. OR H line corresponds to the modified fit. In this thesis, the modified fit is used to estimate the OR motor masses.

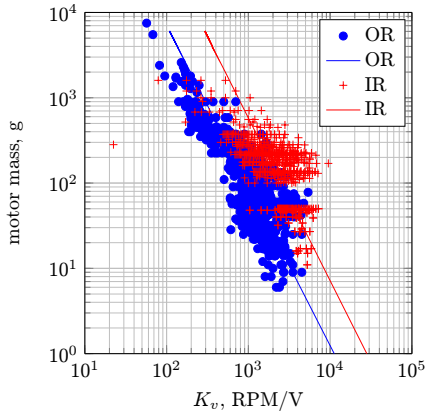


Figure 4.1: *BLDC mass vs speed constant [2]*

Electrical model

The internal resistance of a BLDC motor, R_{motor}, can be represented as a function of its speed constant k_v or zero-load current i_0. Figure 4.3, that shows the comparison of motor internal resistance to zero-load current of the motor, has been used to obtain equation (4.2) which expresses the

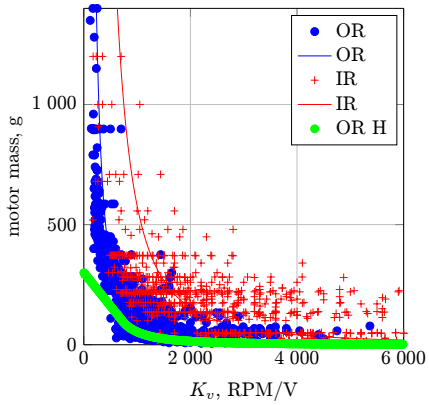


Figure 4.2: *BLDC mass vs speed constant with modified OR fit [2]*

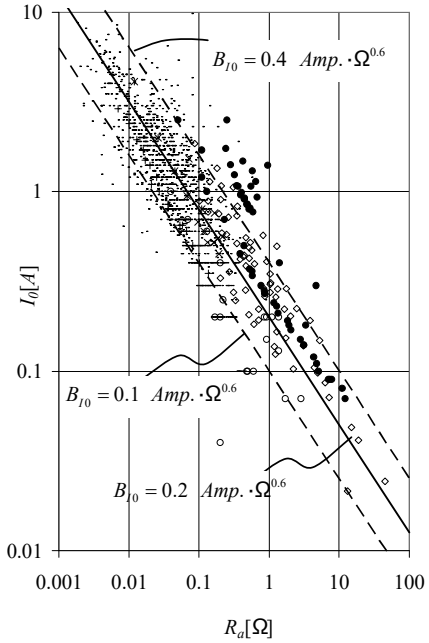


Figure 4.3: *Motor internal resistance to zero-load current [3]*

relation between them analytically:

$$i_0 = \frac{b_0}{{R_{motor}}^{0.6}}. \tag{4.2}$$

The zero-load current parameter, b_0, is a representative parameter on estimating the internal resistance of a BLDC motor. b_0 is assumed to be equal to $0.25\,\mathrm{A}\,\Omega^{0.6}$ for this tool.

4.2.2 Electronic Speed Controllers (ESCs)

ESCs are the control units for electric motors. To change the speed of a motor, the corresponding ESC takes the given pulse width modulation signal and varies the switching rate of field effect transistors (FETs), the motor spins by the frequency of switching (see [2]).

Electronic speed controllers are selected considering the parameter called maximum rated amperage, A_m, which describes how much current the ESC can withstand.

Mass parametrisation

The market survey done by Bershadsky includes 20 electronic speed controllers which have maximum rated amperage values up to 100 A (see [2]). Figure 4.4 shows the relationship between the ESC A_m and its mass with the obtained fit. It is stated that the mass of the ESC, measured in grams, making use of its maximum rated amperage can be obtained by

$$m_e = 0.8421 A_m. \tag{4.3}$$

Electrical model

Ampatis [4] documents that the internal resistance of an electronic speed controller depends on its transistor drain-to-source resistance, R_{DSON}, when it is in "ON" state. ESCs used with BLDC motors typically have three-state transistors to manage three phase currents. As a result, the internal resistance of an electronic speed controller, measured in mΩ, is

$$R_{ESC} = 3R_{DSON}. \tag{4.4}$$

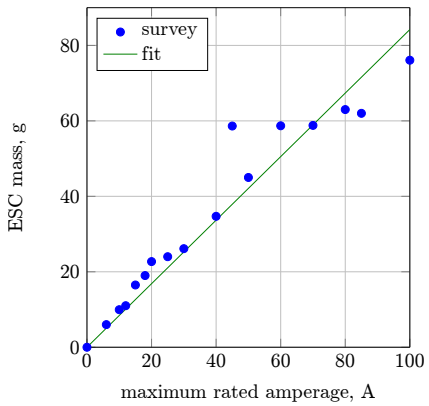


Figure 4.4: *ESC mass vs maximum rated amperage [2]*

The value of R_{DSON} varies between 3 mΩ and 15 mΩ. The tools presented hereafter assume that R_{DSON} is equal to 10 mΩ.

4.2.3 Batteries

Batteries are the source of energy for electric multirotor UAVs. The most common type of battery used for UAV applications is the Lithium-Polymer (LiPo) one. These type of compositions have specific energy values of one-tenth of gunpowder and one-hundredth of kerosene, which is up to 250 W h kg^{-1} (see [2] and [56]).

The selection of the battery for such applications depends on the battery's cell configuration, cell capacity and the nominal voltage value. The internal system of a battery consists of several serial and parallel cells with corresponding cell capacities measured in mA h. Cell capacity combined with the internal system configuration gives information about the amount of current that can be supplied by the battery per hour. Because of that, the choice of the battery considering the amount of cell capacity and the configuration is significant for operation time goals. In addition, the propulsion and the other power sink systems must be fed by the battery with the adequate voltage to operate properly. Connecting the internal cells in series multiplies the voltage that the battery can supply while making these connections in parallel increases the total capacity of the

battery.

As an example, a battery which has a configuration of $3S1P$ (1 branch of 3 cells in series) with cell capacity of 600 mA h and nominal voltage of $3.7\,\mathrm{V}/cell$ supplies a total of 11.1 V and has a total capacity of 600 mA h. If the configuration becomes $4S2P$ (2 branches of 4 cells in series each), the total capacity increases from 600 to 1200 mA h because of the parallel connection and the total voltage supply increases to 14.8 V as a result of increasing number of cells connected in series. A battery with a capacity of 1200 mA h is able to provide 1.2 A of current for one hour if fully discharged. Decreasing the amount of discharge of a battery results in a longer battery life (see [2]). Manufacturers usually indicate the discharge rate of a battery by a parameter called *C-rating* which indicates how much steady discharge a battery can provide. In numbers, a battery with a total capacity of 500 mA h and $20C$ is capable of providing 10 A of current.

Mass parametrisation

The market search done by Bershadsky consists of 30 LiPo batteries which have internal cell configurations between two to six serial and one parallel (see [2]). A relationship between the total capacity of the battery and the mass is found by curve-fitting. Figure 4.5 illustrates the market search results and the corresponding fits. As a result of the fitting process, an analytical equation that describes the mass of the battery, measured in grams, considering its cell configuration and total capacity has been found to be

$$m_b = (p_1 s + p_2)c, \tag{4.5}$$

where p_1 is 0.026373 and p_2 is $2.0499e-05$. The variable s stands for the number of cells connected in series while C is the total capacity of the battery.

Electrical model

Ampatis [4] reports that the total internal resistance of a battery, $R_{battery}$, is related to the cell configuration of the battery and the internal resistance per cell, R_{cell}. Equation (4.6) shows the analytical representation of the

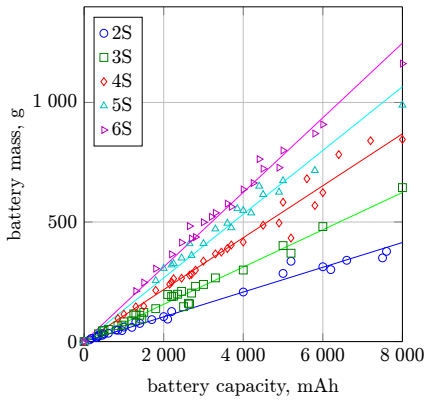


Figure 4.5: *Battery mass vs capacity [2]*

battery internal resistance.

$$R_{battery} = \frac{sR_{cell}}{p}. \tag{4.6}$$

R_{cell} is assumed to be equal to 10 mΩ for this tool.

4.2.4 Propellers

Propellers are the most important aerodynamic components of multirotor UAVs. They are characterized by their radius, pitch and number of blades. Propeller pitch can be described as the horizontal distance covered by the propeller when it has turned one full lap. These parameters have significant influences on the vehicle performance and they need to be taken into account carefully during the design process.

Furthermore, the material the propeller is made of counts as another important consideration on the choice of propellers. Generally, UAV propellers are made of wood, plastic, nylon reinforced plastic or carbon fibre. The choice of the material affects not only the mass but also the aerodynamic efficiency of the propeller. Propellers that are made of flexible materials tend to deflect from the plane of rotation at higher rotational speeds, which creates a reduction of the produced thrust (see [57]). During the propeller choice, this effect must also be considered.

Mass parametrisation

Bershadsky states that about 30 propellers are surveyed for the mass estimation problem and Figure 4.6 is obtained, curve-fitting is used to obtain the relationship between the diameter of the propeller and its mass (see [2]). The propellers in this search are built by four different type of materials which are mentioned before; wood, plastic, nylon reinforced plastic and carbon fibre. As a result of the curve-fitting process, the following equation is found to express the propeller mass in grams

$$m_p = p_1 D^2 + p_2 D + p_3. \tag{4.7}$$

Table 4.1 shows the values of the parameters in equation (4.7).

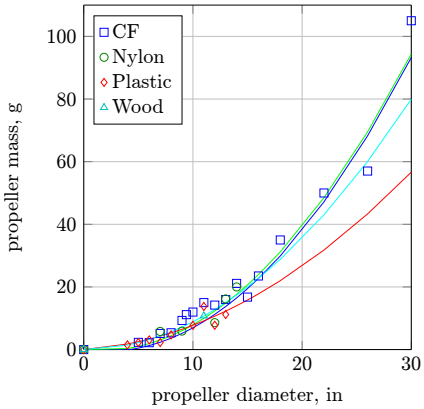


Figure 4.6: *Propeller mass vs diameter [2]*

Material	p_1	p_2	p_3
Wood	0.08884	0	−1.0510
Plastic	0.05555	0.2216	−1.6
Nylon-reinforced plastic	0.1178	−0.3887	0.1685
Carbon fibre	0.1207	−0.5122	2.4553

Table 4.1: *Parameters in equation* (4.7) *for different propeller materials*

4.2.5 Mass breakdown

As mentioned previously, multirotor unmanned aerial vehicles have four significant drive components to be considered in the design process. In addition to these components, there are several other parts that need to be counted while estimating the mass of the vehicle.

These other parts are named as *non-flight components* in this thesis. This means that these components are not critical in the flight process of the vehicle but their mass and also their need of electrical energy must be taken account in the process of design and optimization.

The first part is the structural part, which forms the base of the vehicle. The structural mass of an unmanned aerial vehicle ranges between 8% to 40% of its gross take-off weight (see [2]). To estimate the structural mass of the vehicle in design, the algorithm developed in this thesis does not use a constant ratio but offers a user-input option.

The second part to be considered is the wiring mass of the vehicle. The wiring mass, which contains all the signal and power line masses, is found to be equal to 5% of GTOW of all multirotor UAVs (see [2]). This ratio is used in this thesis to estimate the wiring mass.

Lastly, the mass of avionics and payloads must be taken into account. As these parts are added externally without the need of mass estimation, this thesis offers to enter the mass data of avionics and payloads inside the estimation process. Also, for performance calculations, the current drain of these components has to be entered by the user.

4.2.6 Electric circuit

Figure 4.7 shows the electric circuit of the vehicle in-design.

From now on, the wiring resistance of the vehicle must also be considered. The tools assume that the wiring is made of standard copper of 0.5 m length and 0.5 mm diameter.

The tools calculate the maximum angular velocity at which the motor is able to spin with considering the total battery voltage V_b, motor speed constant k_v, motor zero-load current i_0 and the total resistance of the circuit R_{total}. Ampatis [4] states that the maximum speed of a motor,

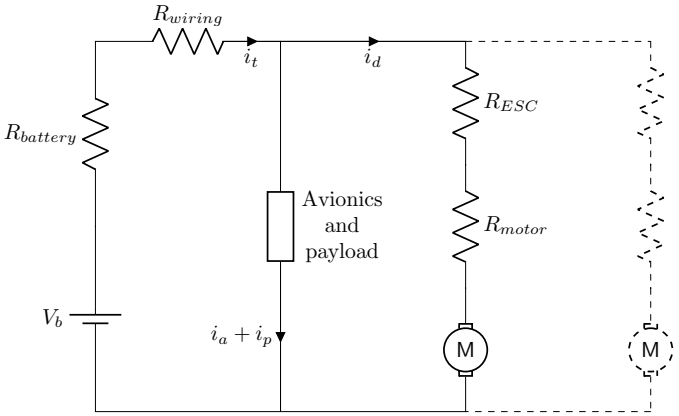


Figure 4.7: *Circuit model considered in the tools*

measured in *RPM*, can be represented as

$$\omega_{max} = (V_b - i_0 R_{total}) k_v, \tag{4.8}$$

and the total resistance of the circuit is given as

$$R_{total} = R_{battery} + R_{wiring} + \frac{R_{ESC} + R_{motor}}{n_m}, \tag{4.9}$$

where n_m is the number of motors.

In Figure 4.7 are also reported the currents i_t, i_a, i_p and i_d which are respectively the total, the avionics, the payload and the drive current. Such currents are related by the following equation

$$i_t = i_a + i_p + i_d. \tag{4.10}$$

It is then trivial to calculate the hovering time (t_h) as follows

$$t_h = \frac{c}{i_t}, \tag{4.11}$$

where c is the total capacity of the battery.

4.3 Design and optimisation tools

The tool developed in the thesis contains two major parts. The Inverse Design Tool (IDT) is the part in which the drive component data are the

main inputs. IDT aims to provide performance estimates about the vehicle in-design considering the provided data of the drive components. In addition, some more information about the vehicle must also be defined. Along with the performance capabilities, IDT has also the capability of analysing the hover time variation of the vehicle considering a change in one of the drive component parameters. This sensitivity analysis is only by request of the user. In Section 4.3.1, IDT is explained with all its features and inside definitions.

The Forward Design Tool (FDT) is the second part of the tool, in which the most significant inputs are the performance requirements. FDT has the purpose of determining the best possible configuration of drive components that can achieve the desired performance characteristics. It can be simplified as a combination of multiple IDT loops which try to find a feasible solution considering the requirements. All algorithms are explained in Subsection 4.3.2.

4.3.1 Inverse design

It is a straightforward algorithm that uses an inductive approach. Figure 4.8 shows the overview of the FDT algorithm.

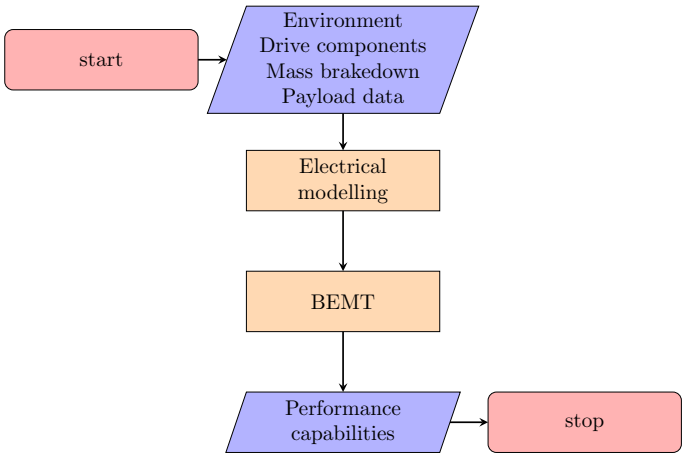


Figure 4.8: *Overview of IDT*

In the following the inputs are described needed from the IDT.

Environment IDT takes two inputs in the environment section. These are the flight altitude above the ground and the take-off altitude, both measured in meters. IDT calculates the corresponding air density to take into account in the BEMT algorithm.

Drive components Major inputs of IDT belong to this section. These inputs mainly describe the drive components that are used in the tool with the required additional data. The total list of the inputs in this section is

- Number of motors n_m
- Motor speed constant k_v
- Motor zero-load current i_0
- Motor type (IR/OR)
- ESC maximum rated amperage A_m
- Battery capacity per cell c_{cell}
- Battery number of cells in serial connection s
- Battery number of cells in parallel connection p
- Battery nominal voltage V_b
- Propeller diameter d
- Propeller pitch p_p
- Number of blades per propeller n_b
- Propeller material

Mass breakdown & payload data This section of the inputs is fundamental from the mass estimation point of view. IDT has two options in mass calculations; either the user can provide a gross take-off weight from the beginning of the analysis or IDT can estimate the total mass of the vehicle using the main drive component parameters that are explained in this Chapter. Besides, mass estimation has to take the structural mass into account. Structural mass is described as a fraction to gross take-off weight in the process. This fraction must be defined by the user if the mass estimation of the vehicle is desired. Also, the user has to provide the weight and the current drain of the avionics and -if present- the payload. Mass data must be considered while the estimation process and current drain have to be taken into account on endurance calculations.

BEMT model

The BEMT model of the tool is based on the theory in Chapter 2. The algorithm tries to find the adequate angular velocity that the propellers must rotate with to produce the required thrust, equal to the gross take-off weight of the vehicle for hover condition.

The procedure to estimate the correct angular velocity is an iteration process of the throttle input d_t. The outer loop iterates d_t to find the corresponding angular velocity while the inner loop uses this velocity to compute the aerodynamic forces and moments produced by a single blade. The results are scaled by the number of blades on the propeller and the number of motors on the vehicle to find the total thrust and torque.

BEMT must be fed with the propeller characteristics. The chord and twist distributions with the other dimensional data, such as propeller pitch and diameter, are inserted into the algorithm for the calculations.

In addition, the gross take-off weight of the vehicle must be specified or estimated. It is needed to terminate the loop.

Performance analysis

The main assignment of IDT is to estimate the hover endurance t_h for the prescribed configuration. IDT computes the required angular velocity through the BEMT model in an iterative fashion and, using this information, the required voltage for the motor is calculated considering the

speed constant of the motor. The drive current, i_{drive}, is calculated using the angular velocity and torque computed through the BEMT model. Combined with the current drain of the avionics and the payload, i_{drive} is used to compute the hovering time. t_h is found by means of equations

$$i_{total} = i_{drive} + i_{payload} + i_{avionics} \tag{4.12}$$

$$t_h = c/i_{total}. \tag{4.13}$$

Bershadsky [2] reports that a discharge rate less than 100 % for batteries is useful to maintain a longer battery life. IDT users can specify a discharge rate before the analysis. In this situation, the total capacity of the battery c is scaled by the discharge rate.

Hover analysis of IDT returns four outputs:

- Hover angular velocity [rad/s]
- Hover endurance [min]
- Hover current [A]
- Hover throttle input [%].

Apart from hover analysis, IDT uses forward flight aerodynamics combined with the procedure described by Bershadsky [2]. For the specified forward flight inputs, which are the lower and upper bound for forward velocity and velocity increment, IDT computes the forward flight endurance and the corresponding range. The prescribed forward velocities are included inside the BEMT algorithm to take the axial flow effect into account. The analysis are constrained by the requirement that the vehicle is not allowed to lose altitude during the forward motion.

In forward flight, rotor pairs of the vehicle spin with different angular velocities to tilt the vehicle in the forward direction. This tilt develops a force that introduces an axial motion. Because of different angular velocities, IDT computes an average angular velocity for endurance calculations.

Forward flight analysis of IDT produces five outputs for every forward velocity value:

- Vehicle tilt angle [deg]

- Flight time [min]
- Estimated range [m]
- Throttle input [%]
- Average angular velocity [rad/s].

Sensitivity analysis

IDT has the option to calculate the sensitivity of the hover time considering the key parameters of the drive components and the gross take-off weight. The user can define an increment to one of the key parameters (or GTOW) and IDT calculates the hover time of the new configuration changing the initial parameter value.

If the increment is defined for one of the drive component parameters, the algorithm finds two new configurations; one with the initial value minus the increment and the other with the initial value plus the increment. If the gross take-off weight of the vehicle is constrained from the beginning, the algorithm does not estimate the mass of the component with the new parameter but uses the constant GTOW. If not, the mass of the component is calculated again and the hover time is computed accordingly. To make the results reliable, it is useful to define the value of the increment as small as possible.

GTOW can also be used in the sensitivity analysis. If the user has started with a given GTOW, it is possible to define an increment for GTOW and analyze the hover time sensitivity. IDT calculates the hover times of the two new configurations; one with GTOW minus the increment, the other with GTOW plus the increment. The algorithm estimates the change in hover time by the change in GTOW and provides the result.

The user can make sensitivity analysis with respect to:

- Motor speed constant k_v
- Propeller diameter d
- Propeller pitch p_p
- Battery capacity c
- Gross take-off weight.

4.3.2 Forward design

As mentioned previously, forward design aims to find the best configuration of drive components to form the multirotor UAV considering the performance requirements. The performance requirements that can be specified for this tool are hover endurance t_h and maximum gross take-off weight W_{max}. FDT tries to find the lightest vehicle which can hover longer than t_h. Figure 4.9 shows the general algorithm of DO.

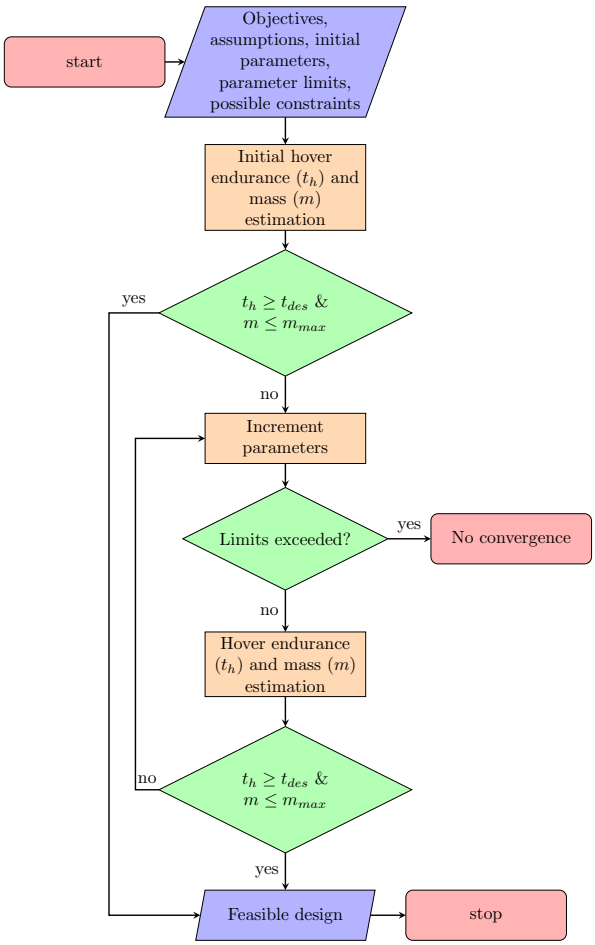


Figure 4.9: *Overview of FDT*

The configuration of drive components in FDT is totally dependent on the key parameters which are described in Section 4.2. These parameters are iterated up to a specified limit until the desired performance is obtained. The performance calculations are executed by the BEMT algorithm.

In the following the inputs needed from the IDT are described.

Objectives The objectives section contains three inputs, which are the flight altitude h, desired hover endurance t_h and the maximum gross take-off weight $GTOW_{max}$. The user defines these inputs considering the desired usage range of the vehicle in-design. The algorithm calculates the air density considering h and tries to find the configuration which can hover longer than t_h and weighs less than $GTOW_{max}$.

Assumptions As in FDT, a part of the total configuration must be specified before the algorithm is executed. For example, the user has to define how many motors form the vehicle and how many blades form the propeller. Also, the battery configuration must be defined before (s, p and V) with the structural fraction of the vehicle and payload-avionics data. Another aspect is the selection of the maximum allowable throttle input value for the hover condition. For example, assuming that the maximum allowable d_t is 80% for a generic multirotor, the selection of an upper bound of 60% for the DO analyses allows the user to grant a manoeuvrability margin of 20% d_t.

Initial parameters The key parameters of the drive components are speed constant k_v for motors, propeller pitch p_p and diameter d for propellers, maximum rated amperage A_m for ESCs and capacity c for batteries since the mass estimation depends essentially on these parameters. Moreover, the user must set an initial guess of these parameters and can optionally define a resolution step for the IDT iterative algorithm.

Parameter limits This input section consists in the limit values of the key parameters which the iteration takes place on until they reach these values. The ranges of the parameters is defined by the initial and limit values.

Constraints DO has the capability to make seven types of iterations. The user can either constrain a parameter or let the algorithm be free to iterate every parameter. The constraint options are:

- No constraints
- Constrained motor (constant k_v)
- Constrained battery (constant c)
- Constrained ESC (constant A_m)
- Constrained propeller diameter (constant d)
- Constrained propeller pitch (constant p_p)
- Constrained propeller (constant d and p_p)

For example, the user can use FDT to find a configuration without constraint on the drive components or a propeller configuration can be selected and used throughout the whole procedure.

Algorithm

The FDT procedure starts with forming the electrical model as in IDT. Then the process continues with the computation of the mass of the vehicle and the hover endurance with the initial parameters using the BEMT model. If the algorithm finds that the initial configuration is able to hover longer than t_h and is lighter than W_{max}, the procedure stops.

On the contrary, the algorithm starts to iterate the parameters considering the increments defined for each parameter and the selected constraint type. As previously stated in Section 4.2, the mass of the drive components increases with increasing parameter value except for motors; the mass of the motor decreases with increasing speed constant. Because of that, the FDT algorithm increases C, p_p and D while decreasing k_v. This is important in the range definition of the parameters; the user must select the initial values as the lower bounds and the limit values as the upper bounds, except for k_v: the initial value of k_v must be considered as the upper bound and the limit value as the lower bound.

Throughout the procedure, the algorithm calculates the gross take-off weight of the vehicle and hover endurance to compare with the objectives. The loop continues until the objectives are satisfied. If the configurations that are inside the parameter limits cannot achieve the prescribed objectives, the algorithm terminates. In this situation, the user must consider changing the configuration assumptions and the range of parameters. If a feasible configuration is found, FDT asks the user whether to terminate the loop or not. The user has the option to continue iterating or end the process.

It may not always be possible to find the drive components with the exact values of parameters on the market. Considering this possibility, FDT tries to define a range for k_v after finding the satisfying configuration. The configuration is assumed to be constrained except for the motor; the algorithm iterates k_v to find a lower bound which is useful for the user to select the motor comparing with the ones on the market.

4.4 Computational effort

Inverse Design Tool is a single run analysis tool that can perform 3 different analyses. A full run contains a single hover run, forward flight runs equal to the number of prescribed forward velocities and 5 sensitivity analyses. For example, a full run takes between 60 to 100 seconds depending on the given configuration of drive components, the mass estimation process, the selected forward flight velocity range and the prescribed increments for the sensitivity analyses. The main computational time is spent on the sensitivity analyses, which is - for a full run - equal to $80-90\%$ of the total time. Reducing the sensitivity increments reduces the computational effort significantly.

The computational effort significantly increases when the range of the parameters are too wide. As an example, the user can specify the range of k_v from 5000 rpm/V to 1000 rpm/V with an increment of 50 rpm/V. In this situation, the FDT construct a set of 800 motors. Considering the other parameter ranges for the battery and the propellers, the total number of configuration sets increases rapidly and DO must perform hover analysis until a set meets the requirements. Defining a narrow range always

decreases the computational time.

Furthermore, due to the desire of decreasing the computational effort, FDT's BEMT algorithm divides the propeller blade to sections which are less than the IDT's BEMT algorithm number of division. While this difference only results in a change in hover endurance between 0.1-0.3 minute for most cases, it significantly reduces the computational time.

4.5 Design example

In this section the available tools which are similar to IDT are introduced and performance results for different multirotor UAV configurations obtained by these tools are compared with the IDT results. Also, the IDT hover time results for specific multirotor configurations are compared with the experimental flight data of those vehicles and an example of the sensitivity analyses is presented for a generic multirotor.

Moreover, a set of already designed multirotor UAVs is used to validate the FDT by taking their experimental endurance and weight data and using them as requirements inside the FDT. The FDT solutions are compared with the exact configurations of the vehicles.

4.5.1 Inverse design results and analysis

There are several tools considering the preliminary design of multirotor UAVs in the market. In this thesis, the tools called Electric Multirotor Sizing Tool (EMST) and eCalc are introduced and their performance results are compared with the IDT results.

EMST is developed by the Georgia Institute of Technology UAV Research Facility [58] and mainly contains every analysis that IDT is capable to perform. EMST has the ability to perform hover, forward flight and sensitivity analyses such as IDT. The tool is described by Bershadsky [2]. The unique ability of EMST is the parametrization of drive components; IDT makes use of the same technique as well. The major differences between EMST and IDT are the construction of the electrical system and the electrical characteristic assumptions of the drive components. By reason of these aspects, the analyses performed by both tools give different results.

eCalc [59] is the most known online multirotor sizing tool. It has the capability to calculate the performance characteristics of a multirotor UAV with the requirement of the characteristics of the drive components, including their mass and electrical data, but has the unique option of selecting the drive components on the market by its own broad database of COTS components. A design process using eCalc is documented by Benito [60].

As it is obviously comprehensible, EMST and IDT use only the significant parameters of the drive components to estimate their mass and electrical characteristics while eCalc requires these data as inputs.

Comparison with ancestor tools

The comparison between the previously mentioned tools and IDT is made by selecting different configurations and obtaining corresponding results. Table 4.2 indicates the configurations for the analyses, Table 4.3 shows the hover endurance results and the differences between the tools for hover endurance results are designated in Table 4.4.

Vehicle	number of motors	k_v	A_m	d	p_p	s	p	c	V_b	GTOW
		rpm/V	*A*	*in*	*in*			*mAh*	*V*	*g*
UAV 1	4	920	30	9	4	4	1	5000	3.45	1250
UAV 2	4	350	30	13	4.5	6	1	4500	3.45	2935
UAV 3	4	950	30	9	4.5	3	1	5100	3.45	1282
UAV 4	4	1900	12	5	3	4	4	325	3.6	520
UAV 5	4	1000	30	9	4	4	1	5000	3.7	1300

Table 4.2: *UAV configurations used in the survey*

It is clear from the results that the difference between the tools is not constant for different configurations. However, for this set of configurations, IDT is closer to eCalc results than EMST in the sense of mean absolute difference.

The primary inference of this survey is that the parametrization of IDT and EMST is similar in nature but different in practice, especially for electrical system. Neither IDT nor EMST is closer to eCalc in every situation.

Vehicle	IDT hover end. *min*	EMST hover end. *min*	eCalc hover end. *min*
UAV 1	19.9	21.5	20.1
UAV 2	13.0	12.3	12.0
UAV 3	18.6	15.9	16.5
UAV 4	10.0	8.9	10.7
UAV 5	13.0	10.9	13.6

Table 4.3: *IDT hover endurance comparison with other tools*

Vehicle	IDT-EMST diff. %	IDT-eCalc diff. %	EMST-eCalc diff. %
UAV 1	-7.44	-0.99	6.96
UAV 2	5.69	8.33	2.50
UAV 3	16.98	12.72	-3.63
UAV 4	12.35	-6.54	-16.82
UAV 5	18.18	-4.41	-19.11
mean abs. difference	12.12	6.59	9.80

Table 4.4: *Hover endurance differences between the tools for the survey in Table 4.3*

Comparison with experimental flight data

Table 4.5 shows the analysis results compared with the hover endurance results obtained by IDT. Vehicles used in this specific work are developed by Aerospace Systems and Control Laboratory(ASCL), which is one of the scientific laboratories of the Department of Aerospace Science and Technology of Politecnico di Milano and all reported data are experimentally obtained.

In addition, Table 4.6 indicates the eCalc hover endurance results for ASCL multirotor vehicles and the error between reported data and eCalc for these vehicles.

Even though IDT is closer to the exact results than eCalc, both tools are in the $\pm 15\%$ error margin. These results validate the hover calculations

Vehicle	Reported hover endurance *min*	IDT hover endurance *min*	Error *%*
ANT-1	6	5.75	-4.1
ANT-1.1	8	7.4	-7.5
HEXA	15	13.7	-8.6
Quad-R2P	10	9.7	-3
Tilt-R2P	7.5	7.8	4
		mean abs. error	5.44

Table 4.5: *FDT hover endurance results compared with experimental flight data*

Vehicle	Reported hover endurance *min*	eCalc hover endurance *min*	Error *%*
ANT-1	6	6.8	13.3
ANT-1.1	8	7.6	-5
HEXA	15	11.6	-22.6
Quad-R2P	10	11.3	13
Tilt-R2P	7.5	7.6	1.3
		mean abs. error	11.04

Table 4.6: *eCalc hover endurance results compared with experimental flight data*

of IDT.

Forward flight comparison between IDT & EMST

The forward flight result schemes of FDT and EMST are equal in construction. Both tools calculate the flight endurance and range for specified forward velocities. Two different multirotor configurations are analyzed to compare the tools; the results are shown in Tables 4.7 and 4.8. These two vehicles are 3DR Iris+ and Quad Mini Generic which is developed by the Georgia Institute of Technology UAV Research Facility[58].

Sensitivity analyses scheme

As previously mentioned in Section 4.3.1, IDT performs sensitivity analyses considering the drive component parameters for hover endurance. Table 4.9 shows an example scheme of the sensitivity outputs of IDT.

Forward velocity	IDT flight end.	IDT range	EMST flight end.	EMST range	End. err.	Range err.
m/s	*min*	*km*	*min*	*km*	*%*	*%*
0.5	18.2	0.54	15.8	0.47	15.4	14.9
1	17.8	1.06	15.7	0.93	13.2	14.0
1.5	17.2	1.54	15.5	1.38	10.9	11.6
2	16.6	1.98	15.0	1.81	10.2	9.4
2.5	15.9	2.38	14.9	2.23	7.0	6.7
3	15.2	2.74	14.8	2.67	2.9	2.6
3.5	14.6	3.05	14.4	3.02	1.2	1.0
4	13.7	3.29	14.1	3.39	-2.9	-2.9
4.5	12.9	3.48	13.8	3.74	-6.8	-6.9
5	12.1	3.62	13.5	4.06	-10.7	-10.8
mean abs. error					8.12	8.08

Table 4.7: *3DR Iris+ inverse flight comparison*

Forward velocity	IDT flight end.	IDT range	EMST flight end.	EMST range	End. err.	Range err.
m/s	*min*	*km*	*min*	*km*	*%*	*%*
0.5	9.9	0.29	8.85	0.27	12.3	7.4
1	9.8	0.59	8.79	0.53	11.5	11.3
1.5	9.6	0.86	8.70	0.78	10.4	10.2
2	9.4	1.12	8.62	1.03	9.0	8.7
2.5	9.2	1.37	8.50	1.28	7.8	7.0
3	8.9	1.60	8.39	1.51	6.2	6.0
3.5	8.6	1.81	8.23	1.73	5.0	4.6
4	8.4	2.01	8.05	1.93	4.1	4.1
4.5	8.1	2.19	7.86	2.12	3.2	3.3
5	7.8	2.34	7.62	2.29	2.4	2.2
mean abs. error					7.19	6.48

Table 4.8: *Quad Mini Generic forward flight comparison*

Parameter	Δ	Unit sensitivity		Incremental sensitivity	
k_v	200	-0.00	*min/rpm/V*	-0.08	*min/200rpm/V*
d	0.5	-0.53	*min/in*	-0.26	*min/0.5in*
p_p	0.5	-0.17	*min/in*	-0.08	*min/0.5in*
c	350	0.01	*min/mAh*	2.12	*min/350mAh*
GTOW	50	-0.01	*min/g*	-0.26	*min/50g*

Table 4.9: *Sensitivity analyses for ASCL ANT-1*

Incremental sensitivity indicates the alteration caused by the increase in the parameter value exactly in the amount of the increment Δ. For example, the hover endurance increases with increasing battery capacity

and decreases while increasing the other parameters.

To validate the results of the sensitivity analyses, ANT-1 and ANT-1.1 are considered. These two vehicles differ only for used motors, battery, and the gross take-off weight. Corresponding increments for these parameters represents the configuration of ANT-1.1. The obtained endurance change is 1.78 minutes, which is close to the reported hover endurance difference of 2 minutes.

Reasons of inaccurate results

The majority of the errors of IDT is caused by its propeller parametrization. As previously discussed in Chapter 2, IDT uses the propeller parametrization described by Brandt [30] and Bershadsky [2], which is not always accurate for multirotor analyses because of the wide range of multirotor propellers.

Also, the airfoil selection is another reason behind large errors. The reasons for the deviation from exact results because of the airfoil assumption is already described in Chapter 2.

Furthermore, the estimation of the mass of the components is not always accurate due to the fact that the parametrization technique is a curve-fitting process. As already stated in Section 4.2.1, the estimation results sometimes diverge from the real results. Moreover, internal resistance parametrization of the components sticks to constant parameters such as, e.g., b_0 and R_{DSON}. These assumptions are not accurate for all of the components on the market.

4.5.2 Forward design results and analysis

For FDT results, three multirotor UAVs designed and built by ASCL are used. Taking as requirements the exact hover endurance and weight values of these vehicles, FDT obtains configurations to compare with the real configuration of the vehicles. The assumptions which have been mentioned in Section 4.3.2 are made according to the original configuration of the vehicle such as, *e.g.*, number of motors, battery serial and parallel connections and number of blades per propeller. FDT is operated in "no constraint" mode, which means none of the drive components are assumed to be constant.

Tables 4.10, 4.11 and 4.12 show the parameters of the vehicles and the corresponding FDT solutions with respect to the exact hover endurance and GTOW of the vehicles.

	ANT-R	FDT
k_v [rpm/V]	2300	1900
A_m [A]	30	30
d [in]	5	4
p_p [in]	4.5	3
c [mAh]	2650	2150
t_h [min]	13.0	13.1
GTOW [g]	733	625

Table 4.10: *ANT-R compared with FDT results*

	TILT-X	FDT
k_v [rpm/V]	2150	2300
A_m [A]	30	30
d [in]	6.5	5
p_p [in]	3.5	3
c [mAh]	5000	4500
t_h [min]	12.5	12.6
GTOW [g]	1523	1300.5

Table 4.11: *TILT-X compared with FDT results*

Using the same assumptions on the vehicles, FDT results correspond to the lightest configuration that can be used for the same performance characteristics. The found parameters do not match with the exact component parameters in every situation; however, they are adequately close. As mentioned in Section 4.5.1, due to the used technique for estimating

	CARRIER-1	FDT
k_v [rpm/V]	1600	1000
A_m [A]	35	30
d [in]	6.5	6
p_p [in]	3.5	3
c [mAh]	8000	7900
t_h [min]	18.0	18.1
GTOW [g]	2900	2795

Table 4.12: *CARRIER-1 compared with FDT results*

the physical and electrical characteristics of the components, corresponding deviations between the real case and optimizer case are understandable. Further validation of the FDT must be performed by constructing a multirotor UAV with the configuration obtained by the tool and testing for hover endurance and GTOW to acquire the error between the tool and experimental results.

Another run for ANT-R is made in "constrained motor" mode. Table 4.13 represents the results. As can be seen from Table 4.13, in "constrained motor" mode, FDT keeps the motor speed constant k_v fixed and runs the analyses accordingly.

	ANT-R	FDT
k_v [rpm/V]	2300	2300
A_m [A]	30	30
d [in]	5	3.5
p_p [in]	4.5	3.5
c [mAh]	2650	2200
t_h [min]	13.0	13.1
GTOW [g]	733	618.7

Table 4.13: *ANT-R compared with FDT results operated in "constrained motor" mode*

As previously stated in Section 4.3.2, it is not always possible to match the FDT solution with the COTS components, especially for BLDC motors. Table 4.14 shows an example of FDT solution that reports a lower bound for k_v to grant a range for the user to choose from within. The requirements and assumptions are stated in Table 4.15.

	Exact solution	Lower bounded solution
k_v [rpm/V]	4000	3450
A_m [A]	10	10
d [in]	3	3
p_p [in]	3	3
c [mAh]	900	900
t_h [min]	10.1	10.5
GTOW [g]	214.6	224.6

Table 4.14: *Lower bound example scheme of DO*

min. t_h	max. GTOW	n_m	i_0	$d_{t\,max}$	B	s	p	V_b
min	*g*		*A*	*%*				*V*
10	300	4	0.5	60	2	3	1	3.7

Table 4.15: *Performance requirements and assumptions of lower bound example in Table 4.14*

According to Table 4.14, the user can select the motors that have k_v values between 3450 rpm/V and 4000 rpm/V with 0.5 A of i_0 for the vehicle in-design.

CHAPTER 5

Linear attitude and position controllers

In this Chapter two different linear control approaches for both attitude and position dynamics will be described. As described in Chapter 1 the dynamical model describing the behaviour of a multirotor UAV is intrinsically nonlinear. In the first part of this chapter the so called "classical approach" is presented. Such control architecture assumes the attitude and the position dynamics as decoupled SISO models and it is based on a hierarchical control loops logic. The second part of this chapter describes a different approach which derives from the rotorcraft background while in the last part some experimental results are presented and discussed.

5.1 Classical approach

The objective of the control design that we consider is to stabilize the UAV at a constant position set-point. To tackle the underactuated nature of

vectored-thrust UAVs, we follow a hierarchical control strategy in which the attitude dynamics is used to stabilize the translational one [61].

5.1.1 Model

Starting from the dynamic equations derived in Chapter 1 and neglecting aerodynamic damping, the following expressions, written in the body frame, are obtained

$$m\dot{v}_B = -\omega_B \times (mv_B) + mgR(q)e_3 + f_c, \tag{5.1}$$
$$J\dot{\omega}_B = -\omega_B \times (J\omega_B) + \tau_c. \tag{5.2}$$

Since the body frame is at the center of mass, the equations of motion for control design, comprehensive of the kinematic equations, can be written as

$$m\dot{v}_E = mge_3 + R(q)^\top f_c, \tag{5.3}$$
$$\dot{p}_E = v_E, \tag{5.4}$$
$$J\dot{\omega} = -\omega_B \times (J\omega_B) + \tau_c, \tag{5.5}$$
$$\dot{q} = \frac{1}{2} q \otimes \begin{bmatrix} \omega_B \\ 0 \end{bmatrix}. \tag{5.6}$$

In this way, the translational motion evolves in the inertial frame, whereas the rotational motion in the body frame. The propulsive system of vectored thrust UAVs, like multirotors, can deliver a force only in the positive direction of $b_3 := R(q)e_3$. Its components in the body frame must satisfy

$$f_c = T_c e_3, \quad 0 < T_c \leq T_M, \tag{5.7}$$

where T_c is the thrust magnitude and T_M is the maximum available thrust. Under these constraints, the UAV dynamics is underactuated: the four control inputs can be used to track a desired position trajectory $p_E^o(t)$ and a desired rotation around the third body axis ψ^o [62].

5.1.2 Hierarchical position and attitude control

Starting from the position dynamic equation and substituting the definition of the control force of equation (5.7), the following expression is

obtained

$$m\dot{v} = mge_3 - T_c R(q)e_3. \tag{5.8}$$

Since the control force in the inertial frame, *i.e.*, $T_c R e_3$, cannot be delivered instantaneously in a desired direction, a widely adopted strategy is to introduce a virtual control variable $f_d \in \mathbb{R}^3$ in equation (5.8).

$$m\dot{v} = mge_3 + f_d - (f_d - R(q)T_c e_3), \tag{5.9}$$

where f_d should be selected so that the desired set-point is an equilibrium point of the dynamics, namely that:

$$m\dot{v} = mge_3 + f_d, \tag{5.10}$$

guarantees the tracking of the trajectory p_E^o. One possible solution, considering constant position reference $p_E^o(t) = p_E^o = const.$, is

$$f_d = \gamma_p(p_E^o - p_E, v_E, x_{pc}) - mge_3, \tag{5.11}$$

where x_{pc} is the controller state and γ_p is chosen as stabilizer of the position dynamics (5.10). Thus, γ_p should guarantee that ($p_E = p_E^o$, $v_E = 0$) is an asymptotically stable equilibrium point for the dynamics

$$m\dot{v} = \gamma_p(p_E^o - p_E, v_E, x_{pc}). \tag{5.12}$$

A common choice is a cascade controller for systems of the second order

$$v_E^o := K_{p_p}(p_E^o - p_E) \tag{5.13}$$

$$\gamma_p := \left(K_{p_v} + K_{i_v}\frac{1}{s}\right)(v_E^o - v_E) - K_{d_v}\frac{s}{1 + \frac{s}{N_v}}v_E, \tag{5.14}$$

Where, K_{p_p}, K_{p_v}, K_{i_v} and $K_{d_v} \in \mathbb{R}^{3\times 3}$ are positive definite matrices while $N_v \in \mathbb{R}_{>0}$ is the derivative filter constant. In Figure 5.1 is reported a block diagram representing the above described position control architecture. Note that the derivative action is applied only to the state variable and not to the error so that step-like references from the outer loop do not result in excessive peaks in control inputs. The inner loop controller can be written in state-space form as:

$$\begin{cases} \dot{x}_{pc} = A_{pc}x_{pc} + B_{pc}v_E^o \\ \gamma_p = C_{pc}x_{pc} + D_{pc}v_E^o. \end{cases} \tag{5.15}$$

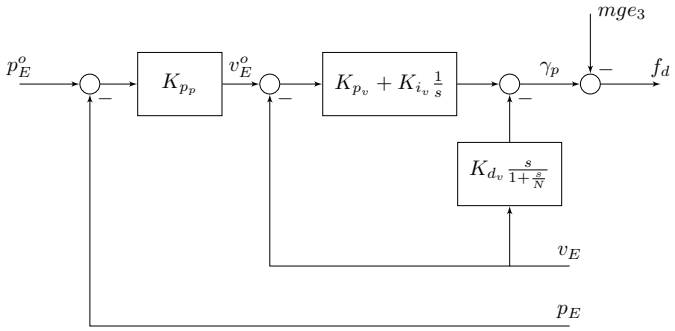


Figure 5.1: *Hierarchical position control architecture.*

The idea behind the hierarchical approach is to find a reference attitude R_d and a control thrust T_c such that

$$f_d - T_c R_p e_3 = 0. \tag{5.16}$$

Then, by exploiting the full actuation of the rotational dynamics (5.5)-(5.6), the control torque τ_c is designed so that the desired attitude motion $R_d(t)$ is asymptotically tracked. In this way, the mismatch term $f_d - T_c R(q) e_3$ will converge to zero. By ensuring additional properties to the stabilizer (see [63]), one can guarantee the convergence of the position trajectory to the desired set-point. It is easily seen that the solution to equation (5.16) is

$$R_d e_3 = \frac{f_d}{||f_d||}, \tag{5.17}$$

and

$$T_c = ||f_d||. \tag{5.18}$$

The other two columns of R_d can be selected to track a desired rotation around the UAV third body axis (see [62] for additional details). As mentioned above, in order to track the attitude reference that comes from the position controller, the control torque τ_c is exploited. The objective of the attitude control design is to track the desired attitude R_d so that $R(q)T_c e_3$ converges to $R_d T_c e_3$ and therefore the position dynamics asymptotically becomes

$$m\dot{v} = \gamma_p. \tag{5.19}$$

For lightweight multirotors in which the gyroscopic coupling is small, the attitude dynamics of equation (5.6) can be equivalently written as

$$\omega_B = \frac{1}{s} J^{-1} \tau_c, \tag{5.20}$$

which leads to a decoupled linear model. The inner loop computes the torque needed to track the desired angular velocity ω_B^o through a PID, which is fed by the outer loop using a purely proportional action.

$$\omega_B^o := K_{p_q} sgn(q_{4,e}) \rho_e \tag{5.21}$$

$$\tau_c := \left(K_{p_\omega} + K_{i_\omega} \frac{1}{s} \right) (\omega_B^o - \omega_B) - K_{d_\omega} \frac{s}{1 + \frac{s}{N_\omega}} \omega_B, \tag{5.22}$$

where $\rho_e \in \mathbb{R}^3$, $q_{4,e} \in \mathbb{R}$ are, respectively, the vectorial and the scalar part of the quaternion error q_e, which is computed as the Hamiltonian product between the desired quaternion q_d, obtained from matrix R_d, and the conjugate of the measured quaternion q, *i.e.*, $q_e := q_d \otimes q^*$. Furthermore, K_{p_q} is the proportional gain of the outer loop, K_{p_ω}, K_{i_ω} and K_{d_ω} are the proportional, the integral and the derivative gains of the inner loop respectively. In the considered analysis, for sake of simplicity, the system is considered as capable of providing the requested thrust and torques at each instant. Then, the linear inner loop controller can be written in state space form as

$$\begin{cases} \dot{x}_{ac} = A_{ac} x_{ac} + B_{ac} \omega_B^o \\ \tau_c = C_{ac} x_{ac} + D_{ac} \omega_B^o, \end{cases} \tag{5.23}$$

where x_{ac} are the controller states.

The controller presented above is by definition nonlinear but for near hovering flight conditions it can be assumed linear. Such assumption leads to

$$\omega_B^o = K_{p_q} sgn(q_{4,e}) \rho_e \quad \rightarrow \quad \omega_B^o = K_{p_q} \begin{bmatrix} \phi^o \\ \theta^o \\ \psi^o \end{bmatrix} - \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix}. \tag{5.24}$$

In Figure 5.2 a block diagram representing the above described attitude control architecture is reported.

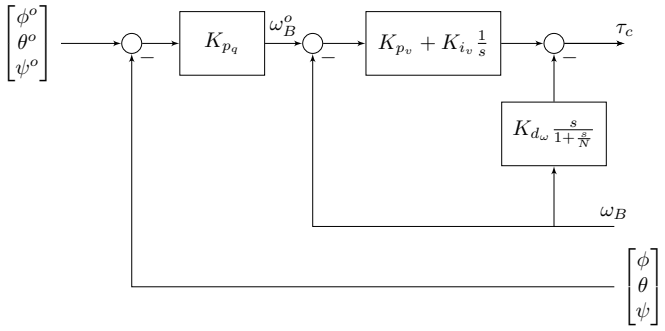


Figure 5.2: *Linearised hierarchical attitude control architecture.*

5.2 Inversion-based methods

In this section the the Dynamic Inversion (DI) and Explicit Model Following (EMF) control laws will be presented and discussed. These control systems have been much studied on full size and mid size helicopters in the last two decades (see [64], [65] and [66] for examples). On the contrary, few studies have been done on small scale quadrotors (see [5] and [67]) in which there are presented experimental results only on the identification phase while the controller performance are evaluated only by numerical results.

5.2.1 Model

The UAV flight dynamics is described by a nonlinear dynamical model as described by the equations (1.35), (1.37). Then, with the hypothesis of small perturbations about the hover (trim) condition (as described in [5]), the equations of motion expressed in the UAV body-fixed frame are derived

$$\dot{u} = X_u u + X_q q - g\theta + X_\delta \delta_{\tau_{c,y}}, \tag{5.25}$$

$$\dot{v} = Y_v v + Y_p p + g\phi + Y_\delta \delta_{\tau_{c,x}}, \tag{5.26}$$

$$\dot{w} = Z_w w + Z_\delta \delta_{f_{c,z}}, \tag{5.27}$$

$$\dot{p} = L_v v + L_p p + L_\delta \delta_{\tau_{c,x}}, \tag{5.28}$$

$$\dot{q} = M_u u + M_q q + M_\delta \delta_{\tau_{c,y}}, \tag{5.29}$$

$$\dot{r} = N_r r + N_\delta \delta_{\tau_{c,z}}. \tag{5.30}$$

The parameters X_u, X_q, M_u, M_q, Y_v, Y_p, L_v, L_p, N_r, Z_w are the dimensional stability derivatives with respect to velocities and rates, $\delta_{\tau_{c,x}}$, $\delta_{\tau_{c,y}}$, $\delta_{\tau_{c,z}}$, $\delta_{f_{c,z}}$ are the control inputs, X_δ, Y_δ, Z_δ, L_δ, M_δ, N_δ are the dimensional control derivatives with respect to the relative control input and finally g is the gravitational acceleration. It must be observed that these derivatives parameters contain mass and moments of inertia of the quadcopter, for example:

$$X_u = \frac{1}{m}\left(\frac{\partial f_x}{\partial u}\right), M_u = \frac{1}{J_{yy}}\left(\frac{\partial \tau_y}{\partial u}\right). \tag{5.31}$$

Lateral dynamics

The linearised system which describes the lateral dynamics is

$$\begin{bmatrix} \dot{v} \\ \dot{p} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} Y_v & Y_p & g \\ L_v & L_p & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} v \\ p \\ \phi \end{bmatrix} + \begin{bmatrix} Y_\delta \\ L_\delta \\ 0 \end{bmatrix} \delta_{\tau_{c,l}}. \tag{5.32}$$

Longitudinal dynamics

The linearised system which describes the longitudinal dynamics is

$$\begin{bmatrix} \dot{u} \\ \dot{q} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} X_u & X_q & -g \\ M_u & M_q & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} u \\ q \\ \theta \end{bmatrix} + \begin{bmatrix} X_\delta \\ M_\delta \\ 0 \end{bmatrix} \delta_{\tau_{c,x}}. \tag{5.33}$$

Yaw dynamics

In this case the linearised system is derived only from the moment equation (5.30))

$$\begin{bmatrix} \dot{r} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} N_r & 0 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} r \\ \psi \end{bmatrix} + \begin{bmatrix} N_\delta \\ 0 \end{bmatrix} \delta_{\tau_{c,z}}. \tag{5.34}$$

Vertical dynamics

The vertical dynamics can be extracted from (5.27) where w is assumed to be equal to $\dot{d}$ since the model is assumed to be linearised around the hover condition which means the attitude angles roll and pitch are negligible

$$\begin{bmatrix} \dot{d} \\ \ddot{d} \end{bmatrix} = \begin{bmatrix} Z_w & 0 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \dot{d} \\ d \end{bmatrix} + \begin{bmatrix} Z_\delta \\ 0 \end{bmatrix} \delta_{f_{c,z}}. \tag{5.35}$$

Complete plant model

The following complete plant model has been obtained by combining the models of each axis described above, since couplings have been neglected at low speed, see the state space system at equation (5.36), the state space matrices at equation (5.37) and equation (5.38), the state and input vectors at equation (5.39) and equation (5.40) respectively. The obtained plant model proved to be minimum phase and unstable around the x and y axes.

$$\dot{x} = Ax + Bu \tag{5.36}$$

$$A = \begin{bmatrix} Y_v & Y_p & g & 0 & 0 & 0 & 0 & 0 & 0 \\ L_v & L_p & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & X_u & X_q & -g & 0 & 0 & 0 \\ 0 & 0 & 0 & M_u & M_q & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & N_r & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & Z_w \end{bmatrix} \tag{5.37}$$

$$B = \begin{bmatrix} Y_\delta & 0 & 0 & 0 \\ L_\delta & 0 & 0 & 0 \\ 0 & X_\delta & 0 & 0 \\ 0 & M_\delta & 0 & 0 \\ 0 & 0 & N_\delta & 0 \\ 0 & 0 & 0 & Z_\delta \end{bmatrix} \tag{5.38}$$

$$x = \begin{bmatrix} v & p & \phi & u & q & \theta & r & \psi & \dot{d} \end{bmatrix}^\top \tag{5.39}$$

$$u = \begin{bmatrix} \delta_{\tau_{c,x}} & \delta_{\tau_{c,y}} & \delta_{\tau_{c,z}} & \delta_{f_{c,z}} \end{bmatrix}^\top. \tag{5.40}$$

5.2.2 Dynamic Inversion control law design

In this section the DI controller will be described in detail. DI inverts the equations which were used to obtain the plant model using feedback linearization. The procedure for the design of the control law (see [5] and [64]) can be divided in three steps:

1. Choice of the state vector x, input u and output y:

$$\dot{x} = Ax + Bu \tag{5.41}$$
$$y = Cx \tag{5.42}$$
$$x \in \mathbb{R}^n, u \in \mathbb{R}^m, y \in \mathbb{R}^m. \tag{5.43}$$

   DI can deal with square MIMO systems of any order as long as measurements of the states exist for feedback.

2. Differentiation of the output equation until the explicit dependence of the control is observed in the output:

$$\dot{y} = CAx + CBu. \tag{5.44}$$

3. Inversion of the output equation

$$u = (CB)^{-1}(v - CAx) \tag{5.45}$$
$$v = \dot{y}_f^o + Ke \tag{5.46}$$
$$e = y_f^o - y \tag{5.47}$$

   where v is the pseudo command vector and K is the compensator of e, which is the difference between the filtered desired output y_f^o and the measured output y.

The architecture instead consists in three blocks:

- feedback to achieve model inversion;
- command filter: the reference signal is filtered to obtain a smoother response to pilot command;
- feed-back compensation to govern disturbance rejection.

A schematic of the DI flight control system is shown in Figure 5.3.

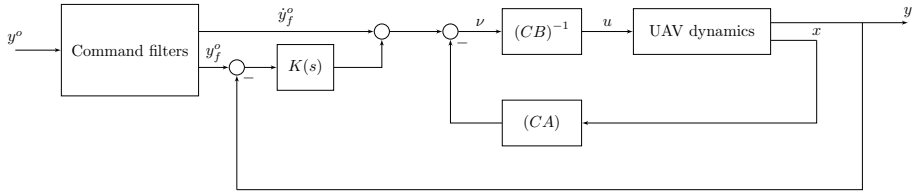


Figure 5.3: *Dynamic Inversion block diagram.*

Stability issues

The main problem of Dynamic Inversion could be the presence of zeros in the right-half complex plane (non minimum phase, NMP zeros). If equation (5.45) is substituted in equation (5.41):

$$\dot{x} = (I - B(CB)^{-1}C)Ax + B(CB)^{-1}\nu \tag{5.48}$$

it can be observed that the eigenvalues of the inverted system are the zeros of the initial one. Therefore, if the identified model is NMP, the controlled system is unstable. To solve this problem, the Approximate Dynamic Inversion method or simple inputs-outputs redefinition can be used (see [64] for more details).

Inner loop: attitude

The inner loop controls the roll, pitch, yaw and vertical dynamics. Following the procedure in the previous section:

1. States, inputs and outputs are chosen for the inner loop:

$$x = \begin{bmatrix} p & \phi & q & \theta & r & d \end{bmatrix}^\top \tag{5.49}$$

$$u = \begin{bmatrix} \delta_{\tau_{c,x}} & \delta_{\tau_{c,y}} & \delta_{\tau_{c,z}} & \delta_{f_{c,z}} \end{bmatrix}^\top \tag{5.50}$$

$$y^o = \begin{bmatrix} \phi^o & \theta^o & r^o & d^o \end{bmatrix}^\top \tag{5.51}$$

$$y = \begin{bmatrix} \phi & \theta & r & d \end{bmatrix}^\top. \tag{5.52}$$

2. A partition of matrix C is needed to see the control in the output equation since pitch and roll angle equations must be differentiated

twice while the yaw rate and vertical velocity equations only once:

$$C = \begin{bmatrix} C_1 \\ C_2 \end{bmatrix} \tag{5.53}$$

$$C_1 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \tag{5.54}$$

$$C_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.55}$$

$$\begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \dot{r} \\ \dot{w} \end{bmatrix} = \begin{bmatrix} C_1\hat{A}^2x + C_1\hat{A}\hat{B}u \\ C_2\hat{A}x + C_2\hat{B}u \end{bmatrix} \tag{5.56}$$

where $\hat{A}$ and $\hat{B}$ are the modified state space matrices:

$$\hat{A} = \begin{bmatrix} L_p & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & M_q & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & N_r & 0 \\ 0 & 0 & 0 & 0 & 0 & Z_w \end{bmatrix} \tag{5.57}$$

$$\hat{B} = \begin{bmatrix} L_\delta & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & M_\delta & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & N_\delta & 0 \\ 0 & 0 & 0 & Z_\delta \end{bmatrix}. \tag{5.58}$$

3. Inversion of the output equation:

$$u = \begin{bmatrix} C_1\hat{A}\hat{B} \\ C_2\hat{B} \end{bmatrix}^{-1} \left(\nu - \begin{bmatrix} C_1\hat{A}^2 \\ C_2\hat{A} \end{bmatrix} x \right) \tag{5.59}$$

with the pseudo-command vector ν defined as:

$$\nu = \begin{bmatrix} \nu_\phi \\ \nu_\theta \\ \nu_r \\ \nu_w \end{bmatrix} = \begin{bmatrix} \ddot{\phi}^o \\ \ddot{\theta}^o \\ \dot{r}^o \\ \ddot{d}^o \end{bmatrix} + \begin{bmatrix} e_\phi \\ e_\theta \\ e_r \\ e_w \end{bmatrix} K_P + \begin{bmatrix} \dot{e}_\phi \\ \dot{e}_\theta \\ 0 \\ 0 \end{bmatrix} K_D + \begin{bmatrix} \int e_\phi \\ \int e_\theta \\ \int e_r \\ \int e_w \end{bmatrix} K_I \quad (5.60)$$

and

$$e = y^o - y. \quad (5.61)$$

Outer loop: velocity

The outer loop tracks the lateral and the longitudinal velocities. Usually, these quantities are commanded in the Earth fixed frame (NED), while equations of motion are written in the body frame. A rotation matrix $R(q)$ from the Earth frame to body frame is so introduced in the flight control scheme (see Chapter 1 for more details about three dimensional rotations):

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = R(q) \begin{bmatrix} \dot{n} \\ \dot{e} \\ \dot{d} \end{bmatrix}. \quad (5.62)$$

Now the DI law for the outer loop can be designed:

1. The vectors of states, inputs and outputs are:

$$x = \begin{bmatrix} u & v \end{bmatrix}^\top, u = \begin{bmatrix} \theta^o & \phi^o \end{bmatrix}^\top \quad (5.63)$$

$$y^o = \begin{bmatrix} u^o & v^o \end{bmatrix}^\top, y = \begin{bmatrix} u & v \end{bmatrix}^\top. \quad (5.64)$$

2. Considering the lateral and longitudinal equations of motion, the output equations are:

$$\dot{v} = Y_v v + g\phi^o \quad (5.65)$$

$$\dot{u} = X_u u - g\theta^o. \quad (5.66)$$

3. Equations (5.66) and (5.65) are inverted:

$$\phi^o = 1/g\,(\nu_v - Y_v v) \quad (5.67)$$

$$\theta^o = -1/g\,(\nu_u - X_u u) \quad (5.68)$$

with the pseudo-command vector defined as:

$$\begin{bmatrix} \nu_u \\ \nu_v \end{bmatrix} = \begin{bmatrix} \dot{u}^o \\ \dot{v}^o \end{bmatrix} + \begin{bmatrix} e_u \\ e_v \end{bmatrix} K_P + \begin{bmatrix} \int e_u \\ \int e_v \end{bmatrix} K_I. \tag{5.69}$$

For both lateral and longitudinal planes, a first order command filter is used to obtain a more desirable reference signal and a PI controller compensates the error between the desired and the measured velocity.

Error dynamics

The equations of motion which are used to control the quadcopter are based on approximations. Model inversion in fact is not exact and, moreover, disturbances are present. For these reasons, error dynamics is introduced and it will be described in detail how to obtain controller gains for the inner attitude loop and the outer velocity loop starting from the definition of the pseudo-command vector

$$\nu = \ddot{y}^o + K_P e + K_D \dot{e} + K_I \int_0^t e \, d\tau, \tag{5.70}$$

using a straightforward pole assignment approach. For a DI controller

$$e^{(n)} = \nu - y^{o(n)} \tag{5.71}$$

where n is the order of differentiation. Substituting equation (5.71) in (5.70):

$$\ddot{e} + K_P e + K_D \dot{e} + K_I \int_0^t e \, d\tau = 0 \tag{5.72}$$

In the Laplace domain

$$E(s)\left(s^2 + K_D s + K_P + \frac{1}{s} K_I\right) = 0 \tag{5.73}$$

$$s^3 + K_D s^2 + K_P s + K_I = 0. \tag{5.74}$$

Consider now a third order system constituted by a pair of complex conjugate poles (introduced with the damping ratio ξ and the natural frequency

ω_n) and a real pole p:

$$(s^2 + 2\xi\,\omega_n s + {\omega_n}^2)(s+p) = 0 \tag{5.75}$$

$$s^3 + (p + 2\xi\,\omega_n)s^2 + (2\xi\,\omega_n p + {\omega_n}^2)s + {\omega_n}^2 p = 0. \tag{5.76}$$

It is possible to match equations (5.74) and (5.76)

$$K_D = 2\xi\,\omega_n + p \tag{5.77}$$

$$K_P = 2\xi\,\omega_n p + {\omega_n}^2 \tag{5.78}$$

$$K_I = {\omega_n}^2 p. \tag{5.79}$$

Initial values for ω_n and ξ were set equal to those of the command filters while the real pole p was considered equal to $p = \omega_n/5$ for frequency separation from the complex conjugate ones. In a similar way it is possible to derive values for the gains of the PI controllers:

$$K_P = 2\xi\,\omega_n \tag{5.80}$$

$$K_I = {\omega_n}^2. \tag{5.81}$$

Note that the simple pole assignment tuning rules discussed above can be replaced with more advanced tuning methods taking also into account robustness requirements, such as, e.g., the structured H_∞ method.

5.2.3 Explicit Model Following control law design

Another flight controller which uses simplified model inversion is the Explicit Model Following. Unlike the previous one, Explicit Model Following is based on the idea of inverting the equations using feed-forward linearisation. This controller could be considered as a particular case of the controller described in [68]. The architecture in fact is composed by:

- feed-forward to achieve model inversion;
- command filters in order to have a desirable response for the pilot;
- feed-back compensation.

An example of the block diagram is shown in Figure 5.4. As for Dynamic Inversion, a detailed description of control laws design for the velocity and the attitude loops will be provided.

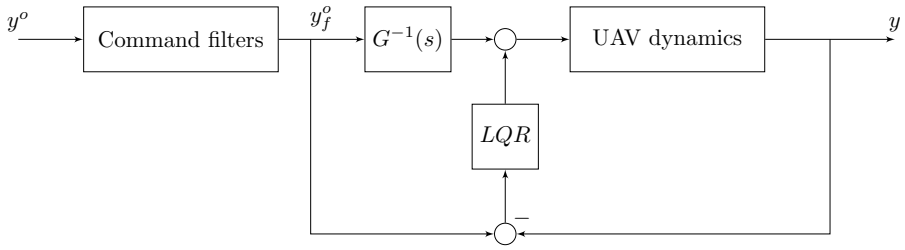


Figure 5.4: *Explicit Model Following block diagram.*

Inner loop: attitude

In order to construct the inner loop which tracks roll angle, pitch angle, yaw rate and velocity along the vertical axis, the following inverted transfer functions were obtained:

- Pitch:
$$\frac{\delta_{\tau_{c,y}}}{\theta} = \frac{s(s-M_q)}{M_\delta}; \tag{5.82}$$
- Roll:
$$\frac{\delta_{\tau_{c,x}}}{\phi} = \frac{s(s-L_p)}{L_\delta}; \tag{5.83}$$
- Yaw:
$$\frac{\delta_{\tau_{c,z}}}{r} = \frac{s-N_r}{N_\delta}; \tag{5.84}$$
- Heave:
$$\frac{\delta_{f_{c,z}}}{\dot{d}} = \frac{s-Z_w}{Z_\delta}. \tag{5.85}$$

Outer loop: velocity

Lateral and longitudinal velocities are controlled independently on each axis using LQR and the following first order decoupled linear models:

- Longitudinal velocity:
$$\frac{\theta^o}{u} = \frac{s-X_u}{-g}; \tag{5.86}$$

- Lateral velocity:

$$\frac{\phi^o}{v} = \frac{s - Y_v}{g}. \tag{5.87}$$

Disturbance rejection

The optimal regulators were obtained minimizing the cost function:

$$J(x,u) = \int_0^t \left(x^\top Q x + u^\top R u \right) d\tau \tag{5.88}$$

where matrices Q and R were related to the largest desired responses and inputs, as suggested in [5]:

$$Q = diag \left[\frac{{\alpha_1}^2}{{x_1}^2{}_{max}} \quad \dots \quad \frac{{\alpha_n}^2}{{x_n}^2{}_{max}} \right] \tag{5.89}$$

$$R = \rho \, diag \left[\frac{{\beta_1}^2}{{u_1}^2{}_{max}} \quad \dots \quad \frac{{\beta_m}^2}{{u_m}^2{}_{max}} \right] \tag{5.90}$$

where $x_{i_{max}}$ is the desired maximum displacement from equilibrium of state x_i and similarly $u_{i_{max}}$ is the desired maximum control action displacement from equilibrium of control input u_i, the constant values ρ, α_i, $i = 1, \dots, n$ and β_i, $i = 1, \dots, m$ are additional weights.

Finally, for the inner loop, the state vector and the modified state space

matrices for the LQR disturbance rejection are:

$$x^{\top}_{inner} = \begin{bmatrix} p & \phi & \int\phi & q & \theta & \int\theta & r & \int r & \dot{d} & \int\dot{d} \end{bmatrix} \tag{5.91}$$

$$A_{inner} = \begin{bmatrix} L_p & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & M_q & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & N_r & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & Z_w & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \tag{5.92}$$

$$B_{inner} = \begin{bmatrix} L_d & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & M_d & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & N_d & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & Z_d \\ 0 & 0 & 0 & 0 \end{bmatrix}. \tag{5.93}$$

For the outer loop, the state space matrices used for the LQR regulators are:

$$A_v = \begin{bmatrix} Y_v & 0 \\ 1 & 0 \end{bmatrix}, B_v = \begin{bmatrix} g \\ 0 \end{bmatrix} \tag{5.94}$$

for the lateral plane, and

$$A_u = \begin{bmatrix} X_u & 0 \\ 1 & 0 \end{bmatrix}, B_u = \begin{bmatrix} -g \\ 0 \end{bmatrix} \tag{5.95}$$

for the longitudinal plane. As for the case of dynamic inversion, also for explicit model following tuning methods including robustness requirements might be considered.

5.3 Experimental results

In the last section of this chapter the above mentioned controllers are compared. In particular, the two inversion-based approaches are compared and then the best controller is compared with hierarchical position and attitude controller. The physical parameters of the ANT-R quadrotor (*e.g.*, the stability and control derivatives) have been identified thanks to an identification campaign described in [69].

5.3.1 Comparison of inversion-based approaches

To compare the two different rotorcraft flight controllers, speed command doublets were chosen as inputs, in order to have a comparison of both the outer velocity loop and the inner attitude loop. In this case, off-board automated inputs were preferred rather than manual radio controls in order to have precise, fast, repeatable and therefore comparable commands. Figure 5.5 shows the response of the ANT-R quadrotor to a lateral speed command doublet of $1\,\mathrm{m\,s^{-1}}$ amplitude with a half-period of $2\,\mathrm{s}$.

The output velocity of the EMF flight controller features a slight overshoot with respect to the desired output while the DI control system follows correctly the set-point. In order to study these small differences of the two flight controllers, the control input actuator is studied once again in details. This control input actuator is the sum of the model inversion and the error compensation terms of both the inner and the outer loops. A comparison of these quantities is shown in Figure 5.6 and Figure 5.7 for the outer velocity loop, where data are expressed in *degrees* as inputs for the inner attitude loop and in Figure 5.8 and Figure 5.9 for the outer loop, where data are expressed in percentage of the maximum applicable moment as inputs for the mixer matrix.

Differences in the error compensation term for both loops are clearly due to the different control systems: PID regulators are present in the DI flight controller while LQR is used in the EMF flight controller. This could be solved by adjusting the LQR penalties in such a way that they would give results more similar to the Dynamic Inversion control law. More interesting are the model inversion terms: as can be seen in Figure 5.7 and Figure 5.9 the noise is an important factor in the results; in

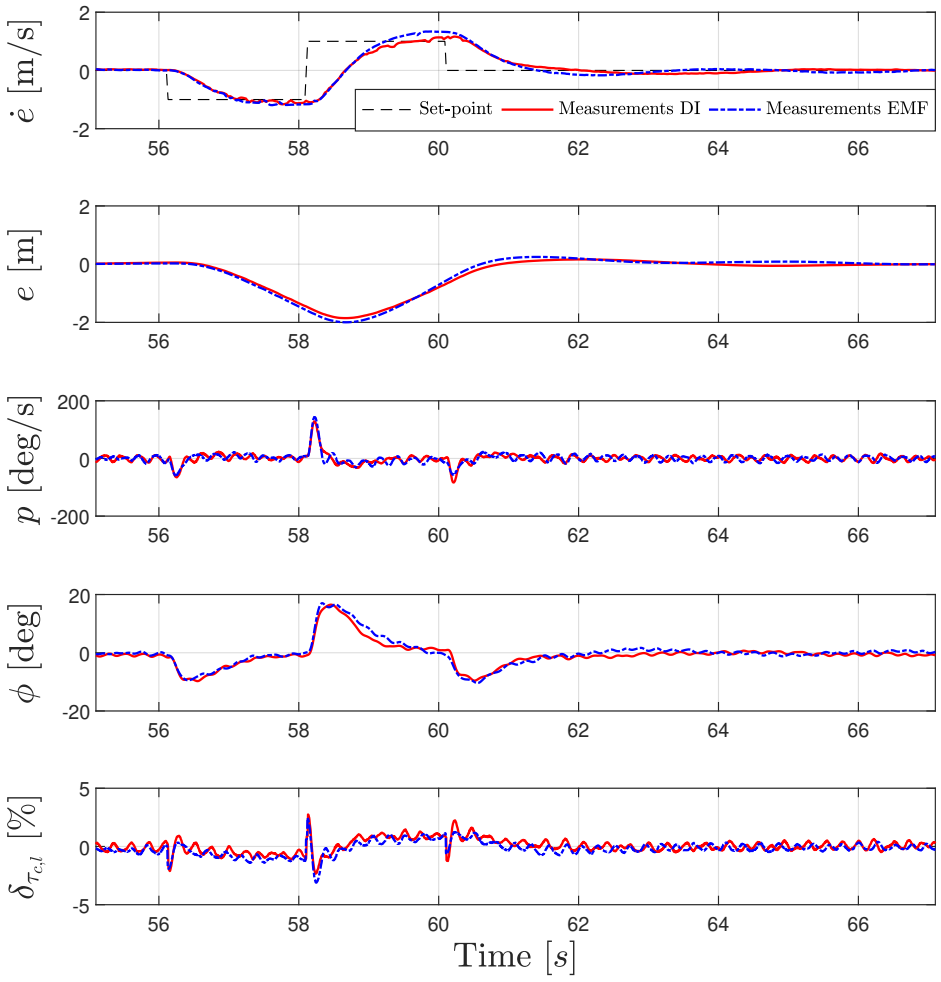


Figure 5.5: *DI and EMF comparison: lateral speed command doublet response (from top to bottom: lateral velocity $\dot{e}$, lateral position e, roll rate p, roll angle ϕ, lateral control actuator input $\delta_{\tau_{c,l}}$).*

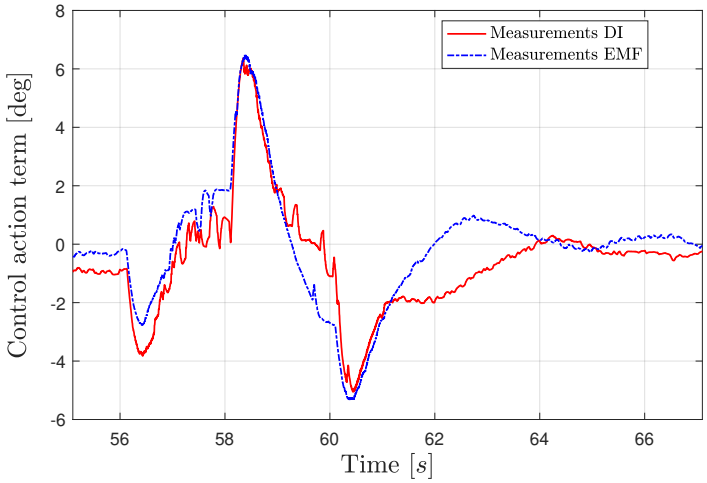


Figure 5.6: *DI and EMF comparison: outer loop control action term.*

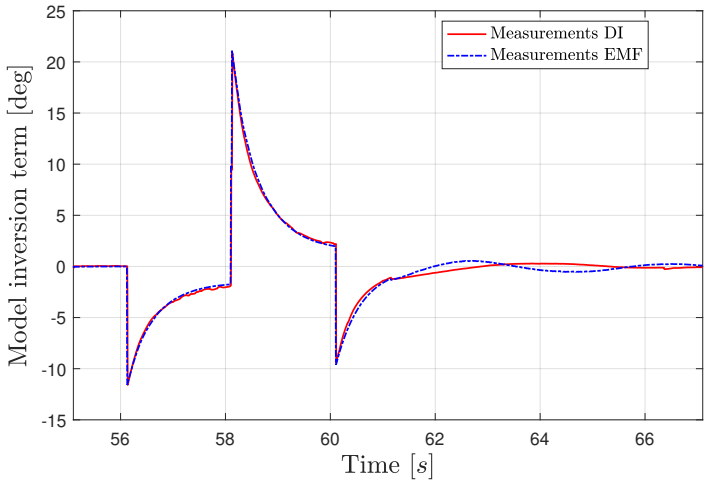


Figure 5.7: *DI and EMF comparison: outer loop model inversion term.*

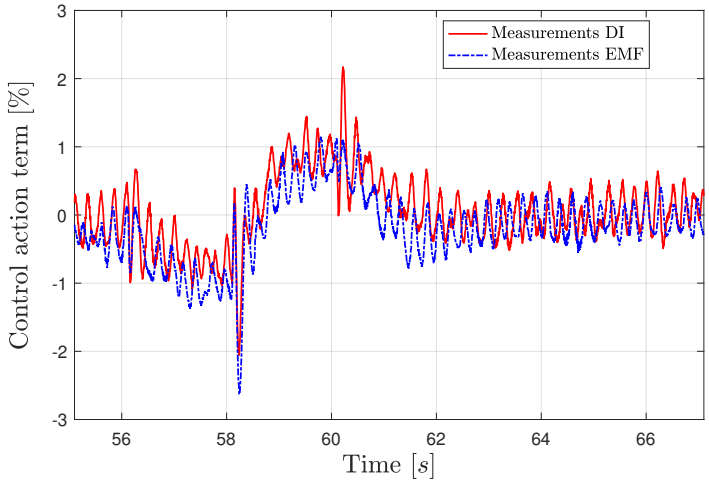


Figure 5.8: *DI and EMF comparison: inner loop control action term.*

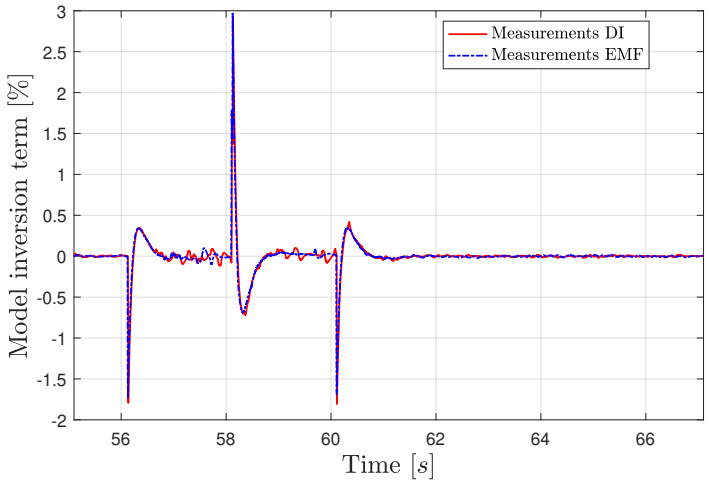


Figure 5.9: *DI and EMF comparison: inner loop model inversion term.*

fact despite having used the same equations of motion in both DI and EMF model inversions, the Dynamic Inversion flight controller introduces these terms in feedback and, as seen in the previous section, these signals come from the optical motion capture system and noise disturbances are inevitable. Moreover, these quantities must be rotated from the NED to the body reference system, introducing also possible numerical errors.

Finally, concerning the z axis, the quadrotor maintained correctly the altitude with both the implemented flight controllers, with no significant differences. Results along the vertical axis are reported in Figure 5.10.

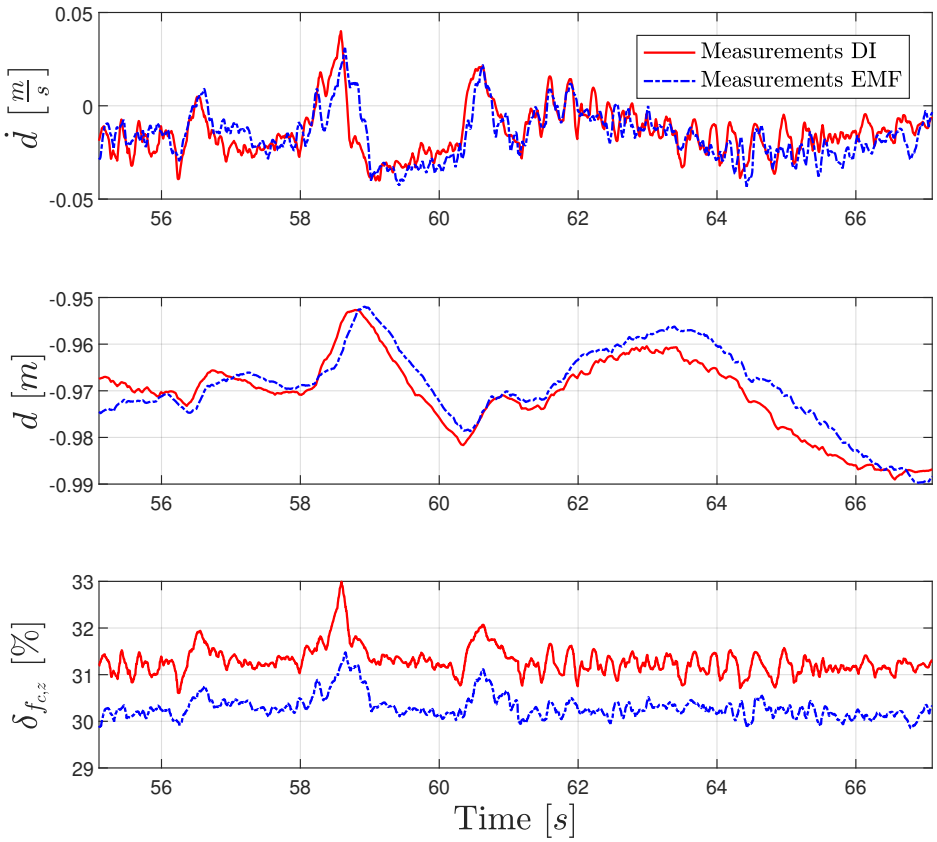


Figure 5.10: *DI and EMF comparison: vertical axis (from top to bottom: vertical velocity $\dot{d}$, vertical position d, vertical actuators control input $\delta_{f_{c,z}}$).*

5.3.2 Dynamic Inversion and stock cascaded PID comparison

Given the the good results obtained in the previous section, the Dynamic Inversion control law has been compared with the hierarchical PID control system. The same speed doublet command input and the same outputs described in Section 5.3.1 are used for the comparison and results are shown in Figure 5.11. Very similar measured outputs are obtained, except for the peak of the roll rate p which is higher with the DI flight controller. The vertical axis is studied in Figure 5.12; although differences are very small, the quadrotor with the Dynamic Inversion implemented onboard seems to better maintain the altitude.

5.4 Final considerations

The two different inversion-based control laws have been tested in flight and compared, giving very satisfactory results: the quadcopter proved to be stable and controllable by studying both off-board input responses and qualitative impressions of the pilot. Similar results were obtained in terms of response and tracking accuracy. A slight overshoot was observed in the velocity output of the EMF controller. Finally, the Dynamic Inversion control law has been compared with the PID hierarchical controller, observing appreciable improvements in the angular rates confirming the good levels of performance obtained with the designed controller.

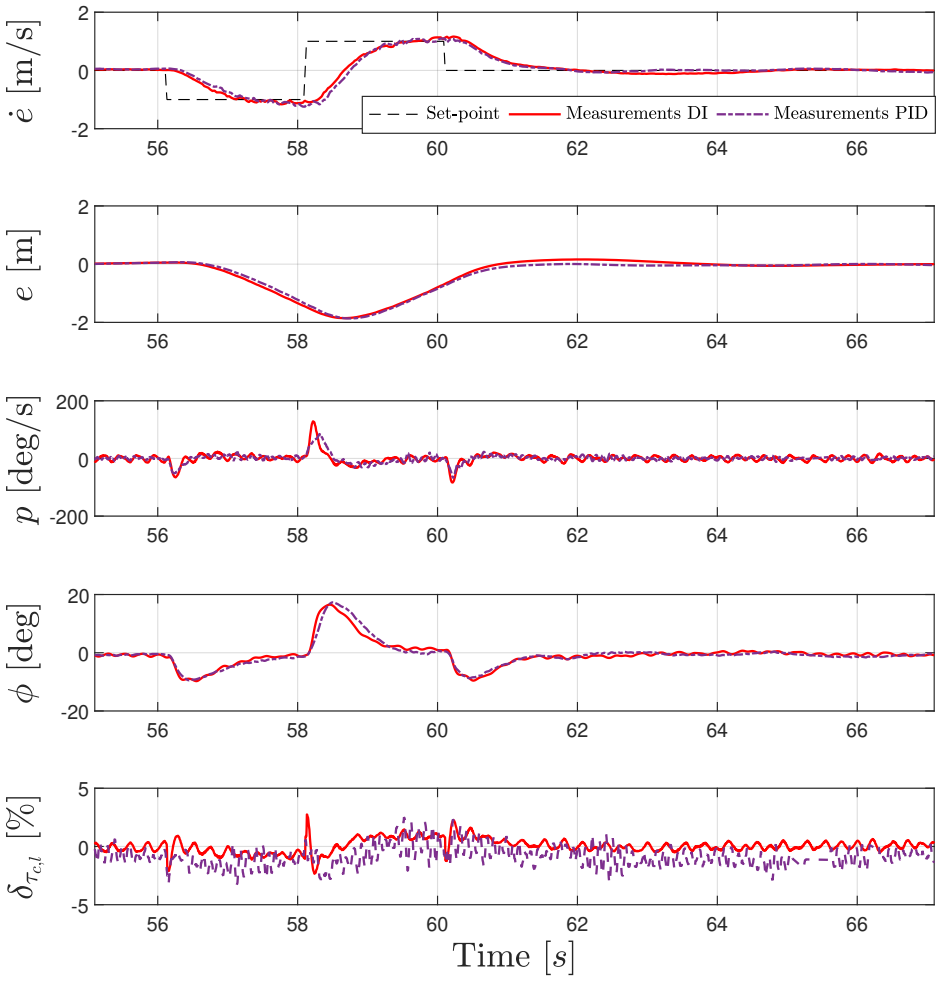


Figure 5.11: *DI and PID control laws comparison: lateral speed command doublet response (from top to bottom: lateral velocity $\dot{e}$, lateral position e, roll rate p, roll angle ϕ, lateral control actuator input $\delta_{\tau_{c,l}}$).*

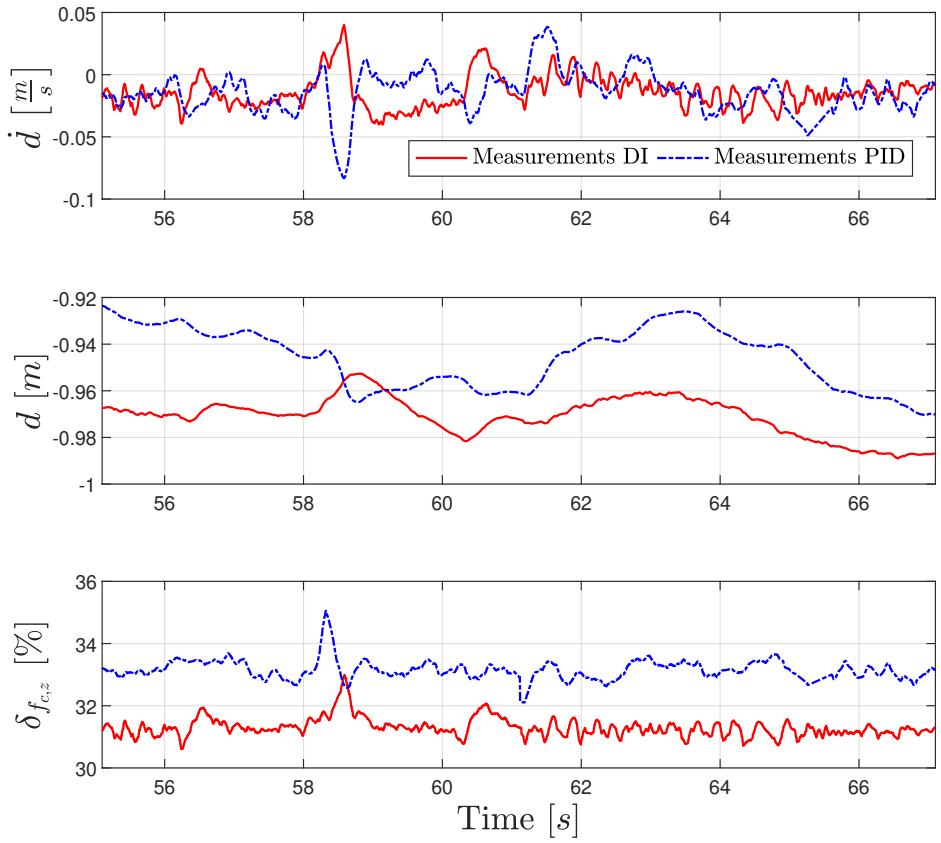


Figure 5.12: *DI and PID control laws comparison: vertical axis (from top to bottom: vertical velocity $\dot{d}$, vertical position d, vertical actuators control input $\delta_{f_{c,z}}$).*

CHAPTER 6

Nonlinear attitude and position controllers

In the previous chapter two approaches to design a linear controller for both the attitude and position dynamics have been presented. Unfortunately, the presented linear controllers cannot handle the intrinsic nonlinearities of the attitude dynamics or the possible model uncertainties. In this chapter an adaptive attitude control which can dynamically compensate model uncertainties and reject disturbances acting on the actuators of the quadrotor QUAD-X will be presented. Moreover, two full-pose controllers will be presented for the TILT-X and the FAST-Hex multirotors.

6.1 Adaptive attitude control

When dealing with nominal operation, fixed-gain linear or nonlinear controllers typically suffice to solve the problem in a satisfactory way [63]. If more challenging questions such as, *e.g.*, actuator degradation and faults,

severe external disturbances, parameter uncertainties and time delays, are to be considered, then more advanced approaches are needed. Adaptive control is an attractive candidate to face the mentioned disturbances and uncertainties for fault-tolerant or reconfigurable unmanned flight because of its capability of learning whilst operating.

Model reference adaptive control (MRAC) is the most widely known adaptive control technique and there are many interesting results in the control of multirotor UAVs. A common characteristic of MRAC schemes is the use of a baseline controller, usually a PI controller designed by means of LQR [70, 71], which is used to define the reference model. Both references [70, 71] show the superior performance of the augmented adaptive controller over the baseline one, in case of actuator and mass uncertainties, respectively. On the other hand, the standard MRAC approach is characterized by an inherent trade-off between good tracking performance and transient oscillations [72]. To overcome such undesired behavior, Combined/Composite direct and indirect MRAC (CMRAC) was proposed [73, 74], which has demonstrated significant improvement of transient performance using filtered control input and state, thereby allowing higher adaptive gains with smoother parameter estimates. Closed-loop MRAC [75], which uses closed-loop reference models, was introduced to reach the same goal. In the last decade, $\mathscr{L}_1$ adaptive control has emerged as a viable strategy to handle systems characterized by rapidly varying uncertainties and it has been validated experimentally in several aerospace applications (see, *e.g.*, [76, 77]). By including a low-pass filter in the scheme, a band-limited control signal is guaranteed, which allows fast adaptation rate, limited only by the available computational power. An $\mathscr{L}_1$ adaptive output feedback control design, tuned by minimizing a cost function based on the characteristics of the reference model and the low-pass filter, has been also proposed [78]. Flight test results showed that the augmented $\mathscr{L}_1$ adaptive system exhibits definite performance and robustness improvements. A different approach [79] uses a nominal backstepping controller to control the attitude of a quadrotor UAV, which is successively augmented with an $\mathscr{L}_1$ piecewise-constant adaptive controller. Performance is visibly improved with adaptation, since fast adaptation is now possible due to the low-pass filter introduced in the $\mathscr{L}_1$

methodology. These works show the control designer's ability to choose large adaptation gains for fast convergence without compromising robustness. Adaptive augmentation schemes based on neural networks have also been proposed in aerospace applications [80, 81, 82, 83]. The peculiarity of those schemes is the use of the concurrent learning modification, enabling a faster converge of the estimates to their true values.

In this thesis, see also the preliminary results in [84], the adaptive augmentation of the attitude control system for a multirotor UAV is considered. The proposed architecture can be applied to augment any baseline linear controller that guarantees asymptotic stability of the closed-loop system in nominal conditions. With respect to standard model reference approaches [70, 71], the proposed adaptive augmentation design does not require the explicit knowledge of the baseline controller structure, but only of the controller output. The approach allows to combine a baseline controller with an adaptive one and to disable or enable the adaptive controller when needed, in order to take the advantages of both controllers. The proposed adaptive scheme exploits an observer instead of a reference model and it is therefore different from the usual formulation that includes the dynamics of the reference closed-loop system. The key idea behind this strategy lies in the use of the observation error and not of the tracking error, as in the standard MRAC, to update the adaptive parameters. When the observation error is zero, either no uncertainties were present or the adaptive law has successfully compensated for them so that the system is operating in the nominal (desired) conditions. By properly tuning the observer gains, the adaptation law becomes more effective in learning the uncertainties. In this regard, such strategy, originally presented in [85], is similar to the mismatched-observer based MRAC proposed in [86], which makes use of both tracking and observation errors in adapting the parameters. Specifically, [86] shows the mechanism for which the observer-based design can effectively damp transient oscillations typical of the standard MRAC while achieving tight tracking performance.

The proposed observer-like MRAC scheme has been exploited to augment the angular velocity controller of a baseline cascade attitude control architecture and it has been validated through experiments on a small quadrotor UAV. In particular, a disturbance torque, mimicking the effect

of a propeller fault or a wind gust, was induced on the quadrotor in hovering conditions to compare the disturbance rejection capabilities of the observer-based MRAC with respect to the baseline controller alone. Furthermore, a modified version of the $\mathscr{L}_1$ controller proposed in [79] has been considered for comparison purposes since $\mathscr{L}_1$ adaptive control has been successfully flown on-board multirotor UAVs [78, 79, 76]. The metrics introduced to evaluate recovery performance are the amount of time the quadrotor takes to reach level attitude after the anomaly, the peak angle deviation and the required control effort. Both the tested adaptive schemes were significantly faster in reacting to the induced disturbance compared to the baseline controller alone.

6.2 Problem statement

The problem under study is to design an adaptive controller that can be implemented in an already existing control architecture, capable of controlling the angular velocity dynamics of a multirotor UAV. For that purpose consider the Euler equations describing the attitude dynamics of a rigid body resolved in the principal inertial axes frame as already described in equation (1.36)

$$J_{xx}\dot{p} + (J_{zz} - I_{yy})qr = l \tag{6.1}$$
$$J_{yy}\dot{q} + (J_{xx} - I_{zz})pr = m \tag{6.2}$$
$$J_{zz}\dot{r} + (J_{yy} - I_{xx})pq = n, \tag{6.3}$$

where J_{xx}, J_{yy}, J_{zz} are the principal moments of inertia, $\begin{bmatrix} l & m & n \end{bmatrix}^\top = \tau_B$ represents the total torque applied to the rigid body and $\begin{bmatrix} p & q & r \end{bmatrix}^\top = \omega_B$ is column vector containing the components of the body angular velocity. Letting $H_0 := \text{diag}(J_{zz} - J_{yy}, J_{xx} - J_{zz}, J_{yy} - J_{xx})$ and denoting with $J_0 := \text{diag}(J_{xx}, J_{yy}, J_{zz})$ the inertia matrix, the Euler equations can be written as

$$\dot{\omega}_B = J_0^{-1}(\tau_B - H_0 f(\omega_B)) \tag{6.4}$$

where $f(\omega)_B := \begin{bmatrix} qr & pr & pq \end{bmatrix}^\top$. The total torque acting on a multirotor UAV can be decomposed into the sum of three contributions: aerodynamic damping torque, torque due to propellers and torque due to external disturbances. For what concerns the aerodynamic damping torque,

we assume it to be proportional to ω_B, hence given by $\tau_d = A\omega_B$, where A is a constant uncertain matrix. The torque due to the propellers is just the control action (output of the actuators), which will be indicated by τ_c. Finally, the external disturbance torque is denoted as σ. Therefore we can write

$$\tau = \tau_d + \tau_c + \sigma, \tag{6.5}$$

and letting $K := J_0^{-1}, H := KH_0$, equation (6.4) becomes

$$\dot{\omega}_B = KA\omega_B + K\tau_c + K\sigma + Hf(\omega_B). \tag{6.6}$$

When hovering the term $f(\omega_B)$ is negligible, but uncertainties and disturbances are still acting on the system, which means that the matrices K, A, H are uncertain. As for the disturbance σ, in the following we will assume it to be either constant or slowly varying. Let now the subscript 0 denote nominal values, and the subscript δ the uncertainty. Then matrices A and H can be rewritten in the following way using the additive uncertainty form:

$$A := A_0 + A_\delta, \quad H := K(H_0 + H_\delta). \tag{6.7}$$

On the other hand, regarding K, it is usually preferable to express the uncertainty on the input gain in multiplicative form:

$$K := K_0\Lambda_K, \quad \Lambda_K := I_3 + K_0^{-1}K_\delta, \tag{6.8}$$

where I_3 is the identity matrix, so that equation (6.4) is rewritten as

$$\dot{\omega}_B = K_0\Lambda_K((A_0 + A_\delta)\omega_B + \tau_c + \sigma) + (H_0 + H_\delta)f(\omega_B). \tag{6.9}$$

As for the actuators, their effect is modelled as a low pass filter, with constant time delay and a zero-order hold, as

$$\tau_c = G(s)u, \tag{6.10}$$

with

$$G(s) := \frac{1}{\tau_n s + 1}\frac{1 - e^{st_s}}{st_s}e^{-st_s}. \tag{6.11}$$

Assume now that for the nominal system, *i.e.*, $A_\delta = 0, H_\delta = 0, \Lambda_K = I_3, \sigma = 0$, we design a baseline feedback controller $u_b := C_b(r, \omega_B)$ capable of asymptotically stabilizing the dynamics of ω_B, so that the DC-gain

from the reference input r to ω_B is unitary. In addition, we suppose to be in hover, so that the term $f(\omega_B) \approx 0$ is negligible.

The idea behind adaptive augmentation is that we want the system to operate mainly in nominal conditions, *i.e.*, to have adaptation active only when necessary. To that purpose, let the control input u be given as

$$u := u_b + u_a, \tag{6.12}$$

where u_b is the control action provided by the baseline controller, while u_a is the contribution to the control action given by the adaptive controller, to be designed based on the knowledge of the nominal one. Similarly, let $\tau_c := \tau_b + \tau_a$, where $\tau_b := G(s)u_b$ and $\tau_a := G(s)u_a$.

6.2.1 Observer-like MRAC augmentation design of attitude control

Plant model

In the plant model

$$\dot{\omega}_B = K_0\Lambda_K((A_0 + A_\delta)\omega_B + \tau_b + \tau_a + \sigma) + (H_0 + H_\delta)f(\omega_B) \tag{6.13}$$

the nominal part is given by

$$K_0(A_0\omega_B + \tau_b). \tag{6.14}$$

By adding and subtracting $K_0A_0\omega_B + K_0\tau_b$ to the right-hand side of (6.13) we get

$$\begin{aligned}\dot{\omega}_B =& K_0\Lambda_K((A_0 + A_\delta)\omega_B + \tau_b + \tau_a + \sigma) + (H_0 + H_\delta)f(\omega_B) \\ & \pm K_0(A_0\omega_B + \tau_b) \\ =& \underbrace{K_0(A_0\omega_B + \tau_b)}_{\text{Nominal part}} + \underbrace{K_0\Lambda_K(\alpha_1\omega_B + \alpha_2\tau_b + \tau_a + \sigma + \alpha_3 f(\omega_B))}_{\text{Uncertain part}}\end{aligned} \tag{6.15}$$

where

$$\alpha_1 := (I_3 - \Lambda_K^{-1})A_0 + A_\delta \tag{6.16}$$

$$\alpha_2 := I_3 - \Lambda_K^{-1} \tag{6.17}$$

$$\alpha_3 := (K_0\Lambda_K)^{-1}(H_0 + H_\delta). \tag{6.18}$$

Notice that Λ_K^{-1} always exists since

$$\Lambda_K = I_3 + K_0^{-1} K_\delta > 0,$$

as K_0 is positive definite and K_δ is positive semi-definite.

Control law

Based on (6.15), the adaptive control law u_a is defined as

$$u_a := -\hat{\alpha}_1 \omega_B - \hat{\alpha}_2 \tau_b - \hat{\sigma} - \hat{\alpha}_3 f(\omega_B) \tag{6.19}$$

where $\hat{\alpha}_1$ is the estimate of α_1, $\hat{\alpha}_2$ is the estimate of α_2, $\hat{\sigma}$ is the estimate of σ and finally $\hat{\alpha}_3$ is the estimate of α_3. Hence, letting

$$\Delta\alpha_i := \hat{\alpha}_i - \alpha_i, \quad i = 1,2,3 \tag{6.20}$$

$$\Delta\sigma := \hat{\sigma} - \sigma \tag{6.21}$$

we get

$$\dot{\omega}_B = K_0(A_0\omega_B + \tau_b) - K_n\Lambda_K(\Delta\alpha_1\omega_B + \Delta\alpha_2\tau_b + \Delta\sigma + \Delta\alpha_3 f(\omega_B)) \tag{6.22}$$

where the assumption that $\tau_a \approx u_a$ has been used for small adaptive gains.

Observer-like reference model and error dynamics

We now build an observer of the nominal part of the plant in the following way:

$$\dot{\hat{\omega}}_B = K_0(A_0\hat{\omega}_B + \tau_b) + Le \tag{6.23}$$

where

$$e := \hat{\omega}_B - \omega_B \tag{6.24}$$

and L is a Hurwitz matrix, added to assign the error dynamics, given by:

$$\dot{e} = (K_0A_0 + L)e + K_0\Lambda_K(\Delta\alpha_1\omega_B + \Delta\alpha_2\tau_b + \Delta\sigma + \Delta\alpha_3 f(\omega_B)). \tag{6.25}$$

Matrix L can be chosen in many ways: for instance it can be found by optimizing the $\mathscr{L}_\infty$ norm of the error e. In the following it is assumed that the solutions of the observer are globally uniformly bounded for any bounded reference signal r and any bounded observation error e. This assumption is easily satisfied for practical cases, *e.g.*, when the baseline controller is linear. In that case it is verified by requiring that the nominal closed-loop system is asymptotically stable.

Adaptive laws

Based on the control law and on the error equation, the following adaptive laws can be deduced:

$$\dot{\hat{\alpha}}_1 := \text{Proj}(\hat{\alpha}_1, -\Gamma_1 \omega_B e^\top PB) \tag{6.26}$$

$$\dot{\hat{\alpha}}_2 := \text{Proj}(\hat{\alpha}_2, -\Gamma_2 \tau_b e^\top PB) \tag{6.27}$$

$$\dot{\hat{\alpha}}_3 := \text{Proj}(\hat{\alpha}_3, -\Gamma_3 f(\omega_B) e^\top PB) \tag{6.28}$$

$$\dot{\hat{\sigma}} := \text{Proj}(\hat{\sigma}, -\Gamma_4 e^\top PB) \tag{6.29}$$

where $\text{Proj}(\cdot,\cdot)$ is the projection operator [72] and the initial conditions are set to $\hat{\alpha}_1(0) = \hat{\alpha}_2(0) = 0$, $\hat{\sigma}(0) = 0$ and $\hat{\alpha}_3(0) = K_0^{-1} H_0$. The symmetric matrix P is chosen so as to satisfy the Lyapunov equation:

$$A_e^T P + P A_e = -Q, \quad A_e := K_0 A_0 + L, \tag{6.30}$$

where $Q = Q^T > 0$ was chosen as $Q = I_3$. Note that by selecting a gain of the form $L = -k_L I_3$, where k_L is a positive scalar, the only non-null components of $\hat{\alpha}_i$ are the diagonal ones, thereby reducing significantly the order of the controller.

The proof of convergence of the observation error can be derived along the lines of the proof of [Theorem 1] [75], by first proving that the observation error is uniformly bounded (from which it follows that all the signals of the closed-loop system are uniformly bounded thanks to the boundedness assumption on the observer solutions) and finally by invoking Barbalat's Lemma, according to consolidated proof techniques in adaptive control.

Comparison with the mismatched-observer based adaptive design of Kim [86]

As mentioned before, an observer-based MRAC scheme has been recently proposed with the aim of improving transient performance of standard MRAC. The underlying development follows a similar reasoning to the one of our strategy and it is therefore interesting to point out how the two schemes are related. Specifically, in [86] the observation error is defined as

$$e := k_L (e_t - e_{t_f}), \tag{6.31}$$

where $e_t := \omega_B - \omega_r$ is the tracking error employed in the standard MRAC approach and e_{t_f} is a corresponding filtered version, the dynamics of which is assigned as:

$$\dot{e}_{t_f} := A_r e_t + ke \tag{6.32}$$

in which A_r is the reference closed-loop dynamics. By substituting equation (6.32) in the dynamics of the observation error, *i.e.*, $\dot{e} = k_L(\dot{e}_t - \dot{e}_{t_f})$, one obtains:

$$\dot{e} = -k_L e - k_L K_0 \Lambda_K (\Delta\alpha_1 \omega_B + \Delta\alpha_2 \tau_b + \Delta\sigma + \Delta\alpha_3 f(\omega_B). \tag{6.33}$$

With respect to (6.25), the only difference is the term $K_0 A_0 e$ that is missing from (6.33). Note that the negative sign in front of the mismatch term in (6.33) is due to the convention used in defining e_t but the contradiction is resolved thanks to the negative sign in the adaptation laws (6.26)-(6.29). Therefore, the two solutions would match if the observer gain were selected as $L = -k_L I_3$ in (6.25) and the term $-\frac{K_0 A_0}{k_L} e$ were included in the definition of e_{t_f} (6.32). It is worth remarking that for a sufficiently large gain k_L, which is expected when seeking high performance, the two approaches yield the same observation error dynamics.

6.2.2 $\mathscr{L}_1$ augmentation design of attitude control

In this section a predictor-based $\mathscr{L}_1$ adaptive control design is presented for comparison purposes. Consider again equation (6.9) and add and subtract the nominal part of the system:

$$\begin{aligned}\dot{\omega}_B =& K_0 \Lambda_K ((A_0 + A_\delta)\omega_B + \tau_c + \sigma) + (H_0 + H_\delta) f(\omega_B) \\ & \pm K_0 (A_0 \omega_B + \tau_b) \\ =& \underbrace{K_0 (A_0 \omega_B + \tau_b)}_{\text{Nominal part}} + K_0 (\alpha_1 \omega_B + \alpha_2 \tau_b + \Lambda_K \tau_a + \sigma + \alpha_3 f(\omega_B))\end{aligned} \tag{6.34}$$

where

$$\alpha_1 := (\Lambda_K - I_3) A_0 + \Lambda_K A_\delta \tag{6.35}$$

$$\alpha_2 := \Lambda_K - I_3 \tag{6.36}$$

$$\alpha_3 := K_0^{-1} (H_0 + H_\delta) \tag{6.37}$$

$$\sigma := \Lambda_K \sigma. \tag{6.38}$$

Further, we can make use of the fact that the actuator can be modelled as an uncertain input gain, and thus rewrite the previous equation as

$$\dot{\omega}_B = K_0(A_0\omega_B + \tau_b) + K_0(\alpha_1\omega_B + \alpha_2\tau_b + \lambda u_a(t) + \sigma + \alpha_3 f(\omega_B)), \tag{6.39}$$

where λ is a parameter used in $\mathscr{L}_1$ adaptive control to model the uncertain input gain [87]. Notice that Λ_K is included in λ.

Design of the filter $C(s)$

We know [87, 85] that we can design the filter $C(s)$ so that the reference system is stabilized. In our case the reference system is given by

$$\dot{\omega}_B = K_0(A_0\omega_B + \tau_b) + K_0(\alpha_1\omega_B + \alpha_2\tau_b + \lambda u_a + \sigma + \alpha_3 f(\omega_B)) \tag{6.40}$$

$$u_a := -\frac{1}{\lambda}C(s)(\alpha_1\omega_B + \alpha_2\tau_b + \sigma + \alpha_3 f(\omega_B)) \tag{6.41}$$

and $C(s)$ needs to be a proper stable filter with DC-gain $C(0) = 1$. Further, the reference system should be stable for all the possible unknown dynamics of the actuator. Let F_Δ denote the set of possible dynamics of the actuator, with $G(s) \in F_\Delta$, then $C(s)$ has the following structure:

$$C(s) := \frac{kF(s)D(s)}{1 + kF(s)D(s)} \tag{6.42}$$

with $F(s) \in F_\Delta, k > 0$ user chosen and $D(s)$ selected as

$$D(s) := \frac{1}{s} \tag{6.43}$$

to satisfy the assumption of DC-gain $C(0) = 1$. The actuator was modelled according to equation (6.11), whilst the nominal control is given by the output of a PID controller $R_{\text{PID}}(s)$

$$\tau_b := G(s)R_{\text{PID}}(s)(r - \omega_B), \tag{6.44}$$

with r being the reference signal. Let then

$$R(s) = G(s)R_{\text{PID}}(s), \tag{6.45}$$

and

$$H(s) := (sI - K_0A_0 + K_0R(s))^{-1}K_0, \tag{6.46}$$
$$M(s) := 1 - C(s) \tag{6.47}$$

from which it follows that

$$\begin{aligned}\omega =& H(s)R(s)r(s)\\ &+ H(s)M(s)(\alpha_1\omega_B + \alpha_2R(s)(r-\omega_B)\\ &+ \tilde{\sigma} + \alpha_3 f(\omega_B)).\end{aligned} \tag{6.48}$$

Next, define

$$G_1(s) := H(s)R(s) + H(s)M(s)R(s)\alpha_2, \tag{6.49}$$
$$G_2(s) := H(s)M(s)R(s) \tag{6.50}$$

and $G_3(s) := H(s)M(s)$. If the filter gain k in (6.42) is selected such that $G_d(s) := (I_3 + G_2(s)\alpha_2 - G_3(s)\alpha_1)^{-1}$ is a stable transfer function for all possible values of α_1, α_2, then the reference system is stable. It can be shown [85] that a sufficient condition for stability is the $\mathscr{L}_1$ norm of $G_d(s)$ be finite. Hence, an upper bound on the value of k can be found by checking up to which value $\|G_d(s)\|_{\mathscr{L}_1} < \infty$.

Predictor model and control law

Based on the expression of the plant model given in equation (6.39) the predictor model

$$\begin{aligned}\dot{\hat{\omega}}_B =& K_0(A_0\hat{\omega}_B + \tau_b)\\ &+ K_0(\hat{\alpha}_1\omega_B + \hat{\alpha}_2\tau_b + \hat{\lambda}u_a(t) + \hat{\sigma} + \hat{\alpha}_3 f(\omega_B))\\ &+ Le\end{aligned} \tag{6.51}$$

is used, where $\hat{\alpha}_1, \hat{\alpha}_2, \hat{\lambda}, \hat{\sigma}, \hat{\alpha}_3$ are the estimates of $\alpha_1, \alpha_2, \lambda, \sigma, \alpha_3$.

Further, let again L be a Hurwitz matrix, added to choose the convergence rate of the error dynamics, where the error is defined as $e := \hat{\omega} - \omega$,

then, based on that, the error dynamics is modelled by

$$\begin{aligned}\dot{e} =&(K_0A_0 + L)e \\ &+ K_0(\Delta\alpha_1\omega_B + \Delta\alpha_2 g_b \\ &+ \Delta\lambda u_a(t) + \Delta\sigma + \Delta\alpha_3 f(\omega_B)).\end{aligned} \tag{6.52}$$

Finally, the adaptive control law is defined as

$$u_a(s) := -kD(s)\eta(s), \tag{6.53}$$

$$\eta(s) := \hat{\alpha}_1\omega_B + \hat{\alpha}_2 g_b + \hat{\lambda}u_a + \hat{\sigma} + \hat{\alpha}_3 f(\omega_B), \tag{6.54}$$

where k and $D(s)$ are chosen as discussed in the previous subsection.

Adaptive laws

Based on the error equation (6.52), the adaptive laws

$$\dot{\hat{\alpha}}_1 := \text{Proj}(\hat{\alpha}_1, -\Gamma_1\omega_B e^\top PB) \tag{6.55}$$

$$\dot{\hat{\alpha}}_2 := \text{Proj}(\hat{\alpha}_2, -\Gamma_2\tau_b e^\top PB) \tag{6.56}$$

$$\dot{\hat{\alpha}}_3 := \text{Proj}(\hat{\alpha}_3, -\Gamma_3 f(\omega_B) e^\top PB) \tag{6.57}$$

$$\dot{\hat{\lambda}} := \text{Proj}(\hat{\lambda}, -\Gamma_4 u_a e^\top PB) \tag{6.58}$$

$$\dot{\hat{\sigma}} := \text{Proj}(\hat{\sigma}, -\Gamma_5 e^\top PB), \tag{6.59}$$

are used. The initial condition of the uncertain parameter associated with λ, which is the equivalent gain of the propellers dynamics, is $\hat{\lambda}(0) = I_3$. The other initial conditions are the same used for the MRAC scheme.

6.3 Experimental results

Both the observer-based MRAC and $\mathscr{L}_1$ adaptive control laws were experimentally tested in flight on the ANT-1 small scale quadrotor described in Appendix B. The hierarchical PID control law described in Section 5.1 was used as a baseline to show performance improvement of the two adaptive laws in terms of capability of rejecting disturbances. The baseline controller gains have been tuned to achieve satisfactory performance

Table 6.1: *Baseline controller gains.*

Axis	K_{p_ω}	K_{i_ω}	K_{d_ω}	K_{p_q}
Roll	0.1	0.21	0.003	13
Pitch	0.14	0.21	0.003	13
Yaw	0.06	0.05	0.001	4

in terms of response to pilot commands, and are reported in Table 6.1 for the three axes.

The adaptive control laws were implemented and validated on all the three axes of roll, pitch and yaw; though, only results related to the pitch axis will be presented. Bounds on the adaptation coefficients are shown in Table 6.2; they are the same for both the MRAC and $\mathscr{L}_1$ adaptive laws (except for the bound on λ which is only applicable to $\mathscr{L}_1$). Similarly, gains on the adaptation coefficients are reported in Table 6.3; it can be noticed that, where applicable, the gains of the $\mathscr{L}_1$ law are larger than the MRAC ones: this is made possible by the presence of the filter in (6.53). As for the observer gain, a value of $L = 150I_3$ was chosen for the MRAC law while a higher value of $L = 300I_3$ was selected for the $\mathscr{L}_1$ law. As for the $\mathscr{L}_1$ filter constant, $k = 25$ has been chosen. Both the bounds and the gains values were tuned by trial and error. Note that for rotations occurring about one principal axis the $f(\omega_B)$ term is negligible, therefore the contribution of α_3 has not been included in (6.19) and (6.53).

Table 6.2: *Bounds on the adaptation laws coefficients.*

Coefficient	**Value**
α_1	10^{-2}
α_2	10^{-1}
λ	0.3
σ	0.2

An experiment was designed to verify the ability of the adaptive control laws to quickly recover level attitude in the case a disturbance occurs on the actuators. The experiment consists in taking the quadrotor to hovering; then, an artificial disturbance on the actuators is injected, *i.e.*, a step disturbance with an amplitude of half the maximum available control moment (in this case, the maximum pitch moment $\bar{m}$) is summed to the

Table 6.3: *Gains on the adaptation laws coefficients.*

Coefficient	Gain (MRAC)	Gain ($\mathscr{L}_1$)
α_1	10^{-3}	1
α_2	10^{-1}	10
λ	-	10
σ	2×10^{-2}	10^{-1}

control moment u, downstream the controller:

$$u_{cmd} = u + \Delta u \tag{6.60}$$

$$\Delta u := \begin{bmatrix} 0 & 0.5\bar{m} & 0 \end{bmatrix}^\top \tag{6.61}$$

so that the actual commanded actuator control action is u_{cmd}.

This experiment is representative of the effect of a (partial or total) loss of throttle on one or more motors, as this would produce a loss of thrust, which in turn produces a disturbance moment about the body axes of the quadrotor.

Figure 6.1 shows the pitch angle and pitch angular rate responses to the injected disturbance on the pitch axis. The disturbance occurs at time instant $t = 0$. It can be noticed that the two adaptive control laws outperform the baseline law in terms of attitude angle disturbance reaction time t_R, which is defined as the time needed to recover the initial attitude angle θ_0 within a range of $\pm1.5^\circ$. The disturbance reaction times for the three control laws are reported in Table 6.4, along with the peak e_{PK} and the RMS e_{RMS} of the error e_θ, where e_θ is defined as the difference between the pitch angle θ and its value θ_0 before the disturbance injection, namely:

$$e_\theta := \theta - \theta_0. \tag{6.62}$$

It can be noticed that, in all the cases, the pitch angle undergoes an initial sudden variation; then, control comes into play and the pitch angle returns to the initial value. In the baseline case, the angle slowly returns to zero attitude always remaining positive, while in the case of the two adaptive laws the attitude angle quickly returns to zero and during the transient also reaches negative values; this turns into an interesting "braking" effect of the quadrotor, and results in a small displacement from its initial

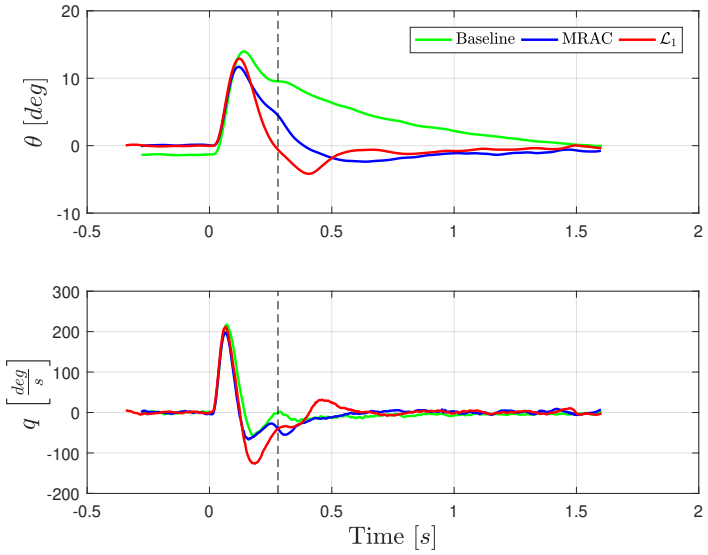


Figure 6.1: *Pitch angle and angular rate: response to input disturbance (comparison between baseline, MRAC and $\mathscr{L}_1$).*

Table 6.4: *Performance metrics: pitch response to input disturbance (in brackets the relative reduction with respect to baseline).*

Control law	t_R [s]	e_{PK} [deg]	e_{RMS} [deg]
baseline	1.496	15.3	6.19
MRAC	0.340	11.6	3.21
	(-77.3%)	(-24.1%)	(-48.2%)
$\mathscr{L}_1$	0.242	12.9	3.11
	(-83.8%)	(-15.6%)	(-49.7%)

position. The $\mathscr{L}_1$ adaptive law achieves a better disturbance reaction time with respect to MRAC.

As for the peak angle, the two adaptive laws achieve a smaller peak with respect to the baseline, with MRAC performing better than $\mathscr{L}_1$. The RMS of the error is also reduced with respect to the baseline, in the case of the two adaptive laws.

Figure 6.2 shows the respective control actions for the three control laws; the control action is dimensionless, *i.e.*, a value of $m = 1$ indicates the maximum control moment which can be produced about the pitch axis. The effect of adding the step disturbance downstream the controller can be noticed in that the steady-state value of the control action response counterbalances the disturbance (with equal magnitude and opposite value). Despite looking very similar, the three control actions actually produce very different results in term of pitch angle response; one difference between the baseline and the adaptive laws can be seen in the steady-state behaviour of the control action response, which has a slightly larger value in the case of the baseline law, resulting in the slow decrease of the attitude angle depicted in Figure 6.1. The RMS of the control action u_{RMS} for the three control laws is reported in Table 6.5. The RMS is similar in all the cases, showing that the control energy employed is comparable; the RMS in the adaptive laws is slightly larger than the baseline case.

Table 6.5: *Performance metrics: control action response to input disturbance.*

Control law	u_{RMS}
baseline	0.491
MRAC	0.509
$\mathscr{L}_1$	0.518

The contribution of the adaptation law to the control action is shown in Figures 6.3 and 6.4 respectively, for the MRAC and $\mathscr{L}_1$ adaptive laws, where the control action has been split into its components u_a and u_b (see equation (6.12)).

The adaptive contributions (namely, u_a) from MRAC and $\mathscr{L}_1$ have been compared and shown in Figure 6.5; it can be noticed that $\mathscr{L}_1$ presents a more prompt response of the adaptive contribution as the disturbance

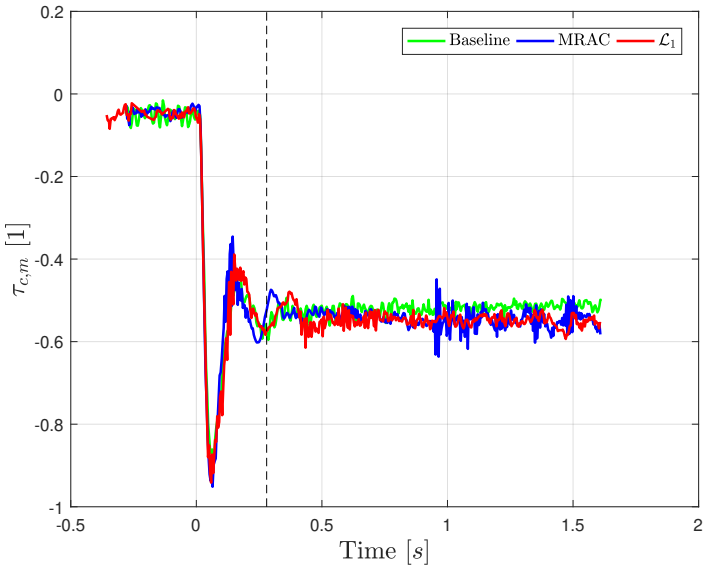


Figure 6.2: *Pitch control action: response to input disturbance (comparison between baseline, MRAC and $\mathscr{L}_1$).*

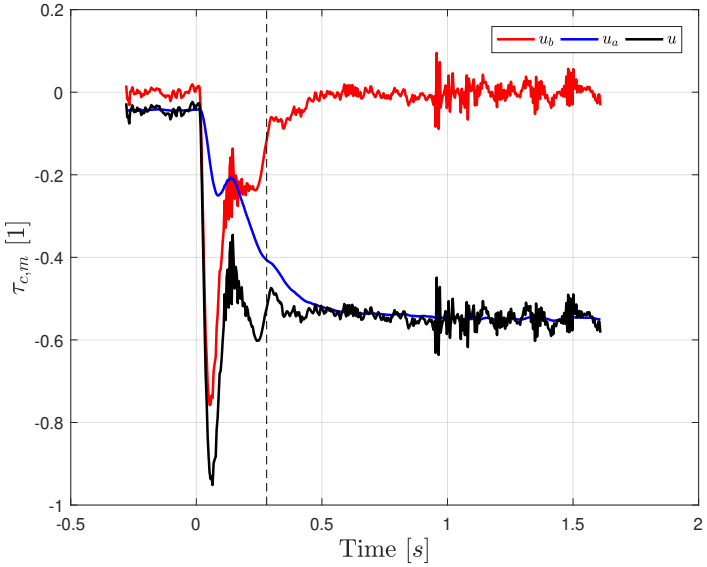


Figure 6.3: *Contributions of MRAC adaptive law: response to input disturbance.*

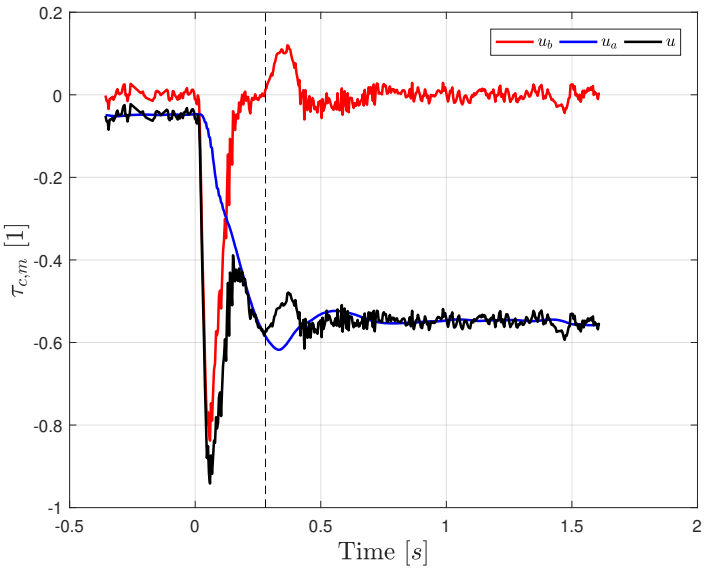


Figure 6.4: *Contributions of $\mathscr{L}_1$ adaptive law: response to input disturbance.*

shows up.

Finally, the response of the adaptive coefficients is shown in Figures 6.6 and 6.7 respectively for the MRAC and $\mathscr{L}_1$ control laws; the coefficients (solid lines) are shown along with their saturation bounds (dashed lines). Notice that these plots consider a wider time horizon than the previous plots. In these experimental runs, the disturbance is injected and removed several times. The presence of the disturbance is well represented by the plot of σ, which can be interpreted as an estimate of the external disturbance as it varies with time.

6.3.1 Final considerations

In this section an adaptive augmentation scheme that can be implemented on top of an existing controller and that has shown improvements in terms of disturbance rejection performance, with only a small increase of the required control power has been presented. A novel MRAC scheme, that includes an observer to estimate the uncertainties, has been presented and compared to similar approaches. The proposed design showed superior

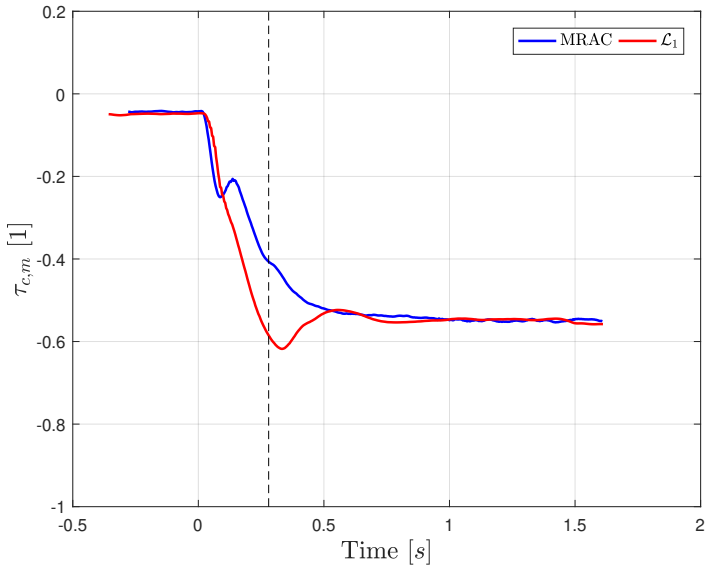


Figure 6.5: *Adaptive contributions of MRAC and $\mathscr{L}_1$: response to input disturbance.*

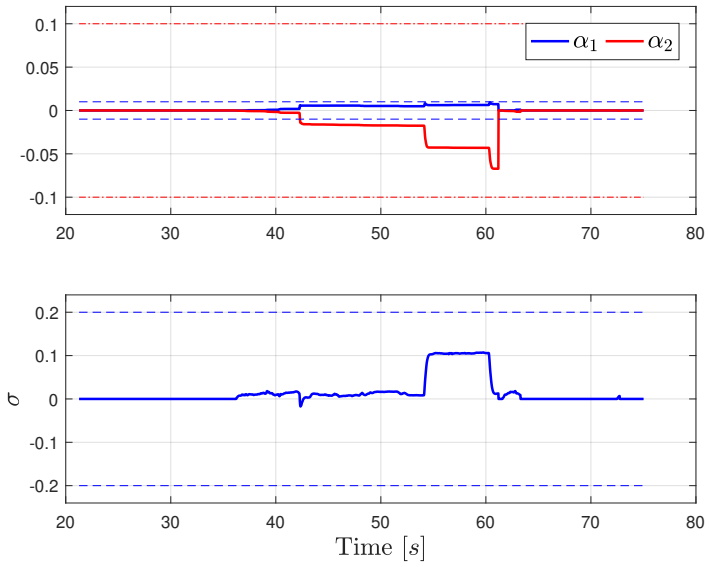


Figure 6.6: *Adaptation coefficients: MRAC.*

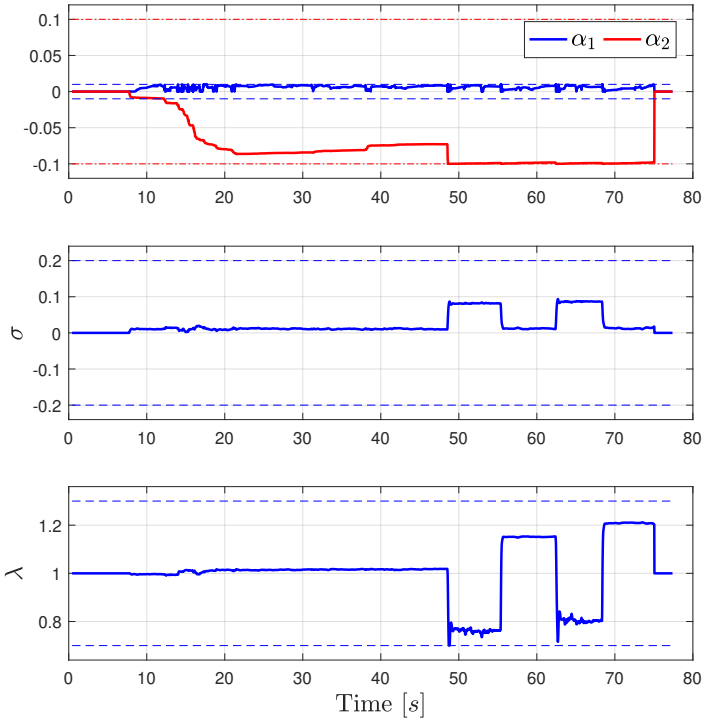


Figure 6.7: *Adaptation coefficients:* $\mathscr{L}_1$.

performance with respect to the nominal controller when operated in the presence of significant disturbances. The observer-based MRAC controller on a quadrotor UAV has been tested and the experimental results confirmed the overall performance improvement with respect to the baseline controller. Furthermore, similar performance were obtained with respect to a predictor-based $\mathscr{L}_1$ controller. In conclusion, the adaptive laws developed in this work offers an increased effectiveness in mitigating a loss-of-thrust like anomaly compared to the nominal controller alone.

6.4 Full-pose tracking control

In recent years the development of multirotor UAVs with thrust vectoring capabilities has received a growing interest. These systems can achieve a larger degree of actuation compared to coplanar multirotor UAVs since both thrust and torque can be oriented within the airframe. This feature makes thrust-vectoring UAVs capable of performing complex full-pose manoeuvres, which is particularly attractive for inspection-like applications that may require, for instance, navigation in a constrained environment. Moreover, being able to deliver both force and torque in any direction enhances the UAV interaction capabilities with the environment, which is especially desirable in aerial manipulation tasks.

Two main technological solutions have been proposed to endow multirotor UAVs with thrust vectoring capabilities: by employing tiltable propellers [12, 18, 88] and by mounting the propellers in a fixed, non-coplanar fashion [13, 14, 89]. As per the first kind of UAVs, the tilting of propeller groups is usually achieved by means of servo-actuators. Although being mechanically complex, this configuration is more efficient, in terms of power consumption, than the fixed-tilted one, as it does not lead to the generation of internal forces when hovering with null attitude. In this thesis, the focus is on tiltable propellers UAVs which can be considered as fully actuated platforms, as documented in recent works [17, 19].

For tiltable propellers platforms, one can consider rotor angular velocities and servo-actuators angles as physical inputs, under the assumption that there are sufficiently fast low-level controllers capable of track-

ing any desired set-point. The majority of control designs consider the UAV as a rigid body endowed with an actuation mechanism capable of producing any wrench, *i.e.*, the platform is assumed to be fully actuated. Based on this, the standard approach splits the control design in two parts. First, a control law in charge of computing the wrench required for full-pose tracking is developed according to different strategies (*e.g.*, feedback linearization, cascade control, Lyapunov-based design, *etc.*). Then, an allocation algorithm is used to assign the physical inputs required to deliver the control wrench, based on a nonlinear mapping that depends upon some basic information about the UAV. Although some care must be taken to deal with kinematic singularities and with the limited actuators bandwidth, the input mapping guarantees full actuation in most operating conditions, provided that the tiltable propellers are suitably placed. This strategy decouples the design of the control law and the choice of a specific actuation mechanism.

In this thesis a feedback linearization approach is proposed, inspired by [90], that does not make use of dynamic extension. The solution proposed in this thesis (which has been preliminary presented in [18]) is based on a geometric PID controller for attitude tracking [91] and a quasi-time optimal stabilizer [92] augmented with a conditional integrator [93] for position tracking. The use of nonlinear control laws is motivated by the non-trivial operating conditions in which fully actuated UAVs can be employed, involving possibly large initial errors and challenging tracking tasks that combine attitude and position motion. In such scenarios it is mandatory to guarantee global tracking properties while accounting for saturation of the actuators, in particular, for the limited control force that can be delivered by propellers.

6.4.1 Main assumptions

In this section the simplified dynamical model that is exploited for control design purposes in most of the works that deal with thrust-vectoring UAVs ([12, 13, 19]) is reported. One can start by presenting the main assumptions under which the full set of equations (1.39) can be approximated to have a sufficiently accurate representation of the system dynamics, in the flight regime of interest, and a manageable control problem.

Assumption 1. *Besides the assumption that the UAV components are rigid, one has also to consider the following:*

1. *the UAV is operated at low speed;*
2. *shape-changes due to the actuation mechanism (servo-actuators) do not affect significantly the geometric and dynamical properties of the system;*
3. *the external wrench* $w_e = \begin{bmatrix} f_e^\top & \tau_e^\top \end{bmatrix}^\top$ *is bounded and constant.*
4. *the control wrench* w_c *delivered by the actuators spans* $\mathbb{R}^3 \times \mathbb{R}^3$*, i.e., the system is fully actuated;*
5. *the actuator dynamics is sufficiently faster than the expected system dynamics;*
6. *full state measurements are available.*

The first assumption is reasonable for an indoor flight scenario as the one considered in this work. It is also the case for most outdoor operating conditions involving multirotor UAVs (*e.g.*, for surveillance, inspections, photography, *etc.*). The second assumption is admissible for the typical configuration in which the center of mass of each propeller groups lies almost on the axis of rotation of the corresponding arm and when the angular momenta of the rotors and the maximum tilting velocities imposed by servo-actuators are sufficiently small. The third assumption is again reasonable for the indoor flight regime that is considered in this work, in which no wind is present. Indeed, the major contributions to the disturbance wrench w_e are the almost time-invariant effect of unbalanced rotors, which may create a significant parasitic torque in the body frame, and battery discharge, which is equivalent to a slow mass increase, *i.e.*, a slow increase of the gravitational force. Under these premises, equations (1.39) take the following form:

$$m\dot{v}_E = -mge_3 + Rf_c + f_e \tag{6.63}$$

$$J\dot{\omega}_B = -\hat{\omega}_B J\omega_B + \tau_c + \tau_e \tag{6.64}$$

where with (f_e, τ_e) is denoted the constant disturbance wrench (here f_e is the vector containing the inertial components of the disturbance force).

The fourth assumption is valid when the propeller groups are at least three ($n \geq 3$) and properly spaced. In such case, one can directly use the wrench w_c as the input variable for control design, so that the problem reduces to the one of controlling a rigid body affected by constant disturbances. This strategy has the advantage of being independent from the specific actuation mechanism: the input allocation problem, *i.e.*, the computation of the physical inputs (θ_a, ω_r) from w_c, can be solved independently. Clearly, for this strategy to work, the closed-loop response of the servo-actuators and of the motors should be fast enough to track the desired input commands so that the control wrench can be almost instantaneously delivered (assumption five). The last assumption is required since most of the control laws exploit a full state feedback strategy to address the trajectory tracking problem. Full state information can be obtained by employing suitable state estimation techniques that exploit the available pose and velocity measurements, typically provided by a motion capture system (laboratory setup) or by on-board sensors (outdoor conditions).

By referring to the simplified dynamical model described by equations (6.63)-(6.64), one sees that the attitude dynamics (6.64) evolves independently from the translational one (6.63). Moreover, since the control force f_c can be delivered in any direction with respect to the airframe (thanks to full actuation), it is possible to use the virtual input $f_d := Rf_c$ as control variable for the position dynamics. By direct substitution, equation (6.63) can be written as follows:

$$m\dot{v} = -mge_3 + f_d + f_e. \tag{6.65}$$

Therefore, the control design for attitude and position tracking can be performed separately.

6.4.2 Control problem

In this section the full pose tracking problem for the system described by equations (1.39) is presented. The control objective is to make the main body of the UAV track a sufficiently smooth trajectory

$$t \mapsto (R_d(t), x_d(t), \omega_d(t), v_d(t))) \in \mathrm{SO}(3) \times \mathbb{R}^3 \times \mathbb{R}^3 \times \mathbb{R}^3, \tag{6.66}$$

where $\omega_d(t) = (R_d^T(t)\dot{R}_d(t))^\vee$ is the desired *body* angular velocity and $v_d(t) = \dot{x}_d(t)$ is the desired *inertial* translational velocity. The steady-state wrench, obtained by inverting the system dynamics evaluated on the desired reference for $(f_e, \tau_e) = (0,0)$, is:

$$f_d^{ss}(t) := m(\dot{v}_d(t) + ge_3) \tag{6.67}$$
$$\tau_c^{ss}(t) := J\dot{\omega}_d(t) + \hat{\omega}_d(t)J\omega_d(t). \tag{6.68}$$

It is usually assumed that the steady state wrench is bounded, *i.e.*, $f_d^{ss}(\cdot)$, $\tau_c^{ss}(\cdot) \in \mathscr{L}_\infty$, in order to have a well-posed tracking problem in the presence of actuators saturations. In this regard, the following assumption is made.

Assumption 2. *The desired trajectory* $t \mapsto (R_d(t), x_d(t), \omega_d(t), v_d(t)) \in \mathrm{SO}(3) \times \mathbb{R}^3 \times \mathbb{R}^3 \times \mathbb{R}^3$ *is such that* $v_d(\cdot) = \dot{x}_d(\cdot) \in \mathscr{C}^1$ *and that* $\omega_d(\cdot) = (R_d^T(\cdot)\dot{R}_d(\cdot))^\vee \in \mathscr{C}^1 \cap \mathscr{L}_\infty$. *Furthermore, the desired angular and translational acceleration are bounded, i.e.,* $\dot{\omega}_d(\cdot), \dot{v}_d(\cdot) \in \mathscr{L}_\infty$.

6.4.3 Cascade P/PID controller (CPID)

This control strategy is based on a nested loops architecture with a proportional action in the outer loop and a PID controller in the inner loop for both attitude and position, following the quite established scheme exploited, *e.g.*, by the PX4 autopilot for multirotor UAVs [94].

The attitude control law is defined by:

$$\omega_d := K_{po}\mathrm{sgn}(q_{e,4})\rho_e \tag{6.69}$$
$$\tau_c := K_{ff}\omega_d + \left(K_{pi} + K_i\tfrac{1}{s}\right)(\omega_d - \omega) - K_d\tfrac{s}{1+\frac{s}{N}}\omega, \tag{6.70}$$

where $\rho_e \in \mathbb{R}^3$, $q_{4,e} \in \mathbb{R}$ are, respectively, the vectorial and the scalar part of the quaternion error $q_e \in \mathbb{S}^3$, which is computed as the Hamiltonian product between the desired quaternion q_d and the conjugate of the measured quaternion q, *i.e.*, $q_e := q_d \otimes q^*$ [95]. Furthermore, K_{po} is the proportional gain of the outer loop, K_{ff}, K_{pi}, K_i and K_d are the feed-forward, the proportional, the integral and the derivative gains of the inner loop respectively.

The position controller has a nested architecture with the same structure as the attitude controller:

$$v_d := -K_{po} e_x, \tag{6.71}$$

$$f_c := R^T \left(K_{ff} v_d - \left(K_{pi} + K_i \frac{1}{s} \right) e_v - K_d \frac{s}{1+\frac{s}{N}} v + f_d^{ss} \right) \tag{6.72}$$

where f_d^{ss} is the steady-state force (6.67).

Remark 1. *The outer attitude loop* (6.69) *computes the desired angular velocity, which is sent as a reference to the inner loop* (6.70)*, based on the quaternion error and a switching strategy to avoid the unwinding phenomenon [95]. The use of the* sgn *function in* (6.69) *to avoid unwinding may result in chattering due to noise [96], although the discrete implementation should mitigate such misbehavior [95]. When assuming near hovering conditions, ρ_e is such that $\rho_e \approx \frac{1}{2}\delta\theta$, and the linearised closed-loop is described by three independent equations, one for each axis. Although the number of parameters involved in the control law is large, tuning techniques such as structured $\mathscr{H}_\infty$ can be employed to achieve satisfactory performance in terms of set-point tracking and disturbance rejection capabilities. The control design relies on the assumption that each axis is decoupled, which can be considered acceptable if the UAV is operated around hovering with small attitude angles and limited angular and translational velocities. Although this flight regime is typical of most operating conditions, it may be too restrictive for the applications in which fully actuated UAVs can be employed.*

6.4.4 The approach of Invernizzi et al. [18] (IQTO)

In this section the control strategy proposed in [18], which exploits the decoupled structure of the equations of motion on $\mathrm{SO}(3) \times \mathbb{R}^3$ is presented. The control law is based on the use of conditional integrators and nonlinear stabilizers, specifically a Quasi Time-Optimal law for position tracking and a geometric PID for attitude tracking. This architecture guarantees boundedness of the control force as well as global position tracking. Attitude tracking is instead guaranteed almost globally following the proof of [97], which is the best result one can hope for with a continuous time-invariant stabilizer. For a more detailed introduction to these control

techniques, the reader is referred to [92, 93]. The control wrench is given by:

$$f_d := \beta\left(e_x, e_v, e_I\right) + m\left(\dot{v}_d + g e_3\right), \tag{6.73}$$

$$\ddot{e}_{I_i} := -k_{I_{d_i}} \dot{e}_{I_i} + \mathrm{sat}_{M_{I_1}}\left(k_{I_{p_i}}\left(-e_{I_i} + \mathrm{sat}_{M_{I_2}}\left(\bar{e}_x\right)\right)\right) \tag{6.74}$$

$$\tau_c := -R_d^T e_R - K_\omega e_\omega - K_I e_I + J\dot{\omega}_d + \hat{\omega}_d J\omega \tag{6.75}$$

$$\dot{e}_I := R_d^T e_R + K_{\omega_I} e_\omega \tag{6.76}$$

for $i = 1, 2, 3$, where $\beta_i(e_{x_i}, e_{v_i}, e_{I_i}) := \mathrm{sat}_{\bar{M}_i}\left(u_i(\bar{e}_{x_i}, \bar{e}_{v_i}) - \ddot{e}_{I_i}, u_i(\bar{e}_{x_i}, \bar{e}_{v_i}\right) := -k_{x_i}\left(\bar{e}_{x_i} + \bar{e}_{v_i}\max\left(\frac{|\bar{e}_{v_i}|}{2\bar{M}_i}, \frac{k_{v_i}}{k_{x_i}}\right)\right)$ and $\mathrm{sat}_M(\cdot)$ is the standard symmetric saturation function, *i.e.*, given $s \in \mathbb{R}$, $\mathrm{sat}_M(s) := \max(-1, \min(1, s))$. Herein, the tracking errors are defined as $\bar{e}_{x_i} := e_{x_i} + e_{I_i}$, $\bar{e}_{v_i} := e_{v_i} + \dot{e}_{I_i}$, and k_{x_i}, k_{v_i}, $k_{I_{d_i}}$, $k_{I_{p_i}}$ are positive gains while $\bar{M}_i := M_i - 2M_{I_{1i}}$, where $M_{I_{1i}}$, $M_{I_{2i}}$ are suitably selected bounds [93, 18].

Remark 2. *The main challenge in applying the control law* (6.73)-(6.76) *is the tuning phase, due to the large number of parameters involved and the non-straightforward interpretation of their contribution on the system response. To this end, one can try to linearise the plant and the control laws for small tracking errors and then to apply suitable tuning techniques (e.g., structured $\mathscr{H}_\infty$). Note that due to the specific choice of the attitude error (left), the linearised control law is time-varying and one has to consider constant reference trajectories to have a linear time-invariant closed-loop system.*

6.4.5 Experimental results

In this section the results obtained by applying the two considered control schemes to a tilt-arm quadrotor UAV that was developed by Aerospace Systems and Control Laboratory (ASCL) at Politecnico di Milano (see Section 1.2.2) are presented. Two experiments are presented: an aggressive stabilization task, in which the UAV is required to go to a specific position with level attitude, and a full pose trajectory tracking task, in which the UAV has to follow a straight line back and forth with a predefined time law, while keeping a non-null pitch angle. The experimental campaign took place indoor in the facility described in Appendix C.

Controller tuning

The parameter tuning for the IQTO has been performed by requiring similar performance to a step reference (in terms of settling time and overshoot) applied to a simulation platform that was developed by relying on the identified parameters of the prototype. Since the yaw moment of inertia was not identified, the corresponding gains have been adjusted by repeated experiments until a satisfactory behaviour was achieved. In particular, the following values have been employed in the IQTO law.

$$\begin{gathered} K_R = \mathrm{diag}(0.5,0.5,0.4) \quad K_\omega = \mathrm{diag}(0.35,0.35,0.3) \\ K_I = \mathrm{diag}(0.01,0.01,0.01) \quad K_{\omega_I} = \mathrm{diag}(0.4,0.4,0.4) \\ k_x = (7,7,7) \quad k_v = (7,7,7) \quad k_{I_p} = (7,7,7) \\ k_{I_p} = (7,7,7) \quad M = (10,10,10) \quad M_{I_1} = M_{I_2} = (3,3,3). \end{gathered} \tag{6.77}$$

For what concerns the CPID architecture, the attitude controller parameters have been tuned using a structured $\mathscr{H}_\infty$ approach (Table 6.6, top) which allows to handle more easily the large number of parameters involved in the control law while optimizing the closed-loop performance. As for the position controller, the parameters obtained with structured $\mathscr{H}_\infty$ have been adjusted (Table 6.6, bottom) to achieve a step response comparable with the other controllers.

6.4.6 Stabilization: set-point tracking

The first experiment is intended to show the set-point tracking capabilities of thrust-vectoring UAVs in a typical scenario in which the UAV has to reach a given position starting from hovering. In particular, the UAV starts at $x(0) = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^\top$ m and has to reach $x_d = \begin{bmatrix} 1.2 & 0 & 1 \end{bmatrix}^\top$ m. Throughout the test, a zero-level attitude is required. Note that this motion could not be performed by coplanar platforms since they can translate only by changing their attitude. Even so, significantly superior performance is expected in terms of settling time, since translation is mainly achieved by tilting the propellers, which is much faster than changing the UAV attitude.

The results obtained with the different controllers are reported in Figures 6.8-6.9 for what concerns position tracking errors, while attitude er-

		k_F	k_P	k_I	k_D	N
Roll/Pitch	outer		2.56			
	inner	0.0266	0.287	0.836	0.007	5.79
Yaw	outer		1.536			
	inner	0.0266	0.287	0.836	0.007	5.79
North/East	outer		8			
	inner	0	4	0.002	0.996	5
Down	outer		8			
	inner	0	4	0.002	0.996	5

Table 6.6: *Cascade P/PID gains used in the experiments*

rors, in terms roll-pitch-yaw angle errors, are reported in Figures 6.10-6.11. The control force and torque computed by the proposed controllers are shown in Figures 6.12-6.13 and the corresponding percentage thrust of each motor and the tilt angles commanded to the servo-actuators are reported in Figures 6.14-6.15. From the Figures 6.12-6.15 it is possible to see a persistent oscillation on the signals; such oscillation is introduced by the gyroscope measurements which are affected by the structural vibrations of the multirotor. The performance obtained with the proposed controllers is satisfactory: the CPID architecture achieves the best overall results. Although the IQTO provides slightly worse results, the control law (6.73)-(6.74) guarantees a limited control force which results in a reduced disturbance torque: its attitude performance is basically the same as the one achieved by the CPID controller.

6.4.7 Trajectory tracking: combined position and attitude manoeuvre

The second experiment has the goal of comparing the tracking capabilities of the controllers and, at the same time, of showing the potential of thrust-vectoring UAVs to perform independent attitude and position manoeuvres. The desired trajectory is assigned as follows: the platform has to reach a desired pitch angle of 25° and then it is required to follow

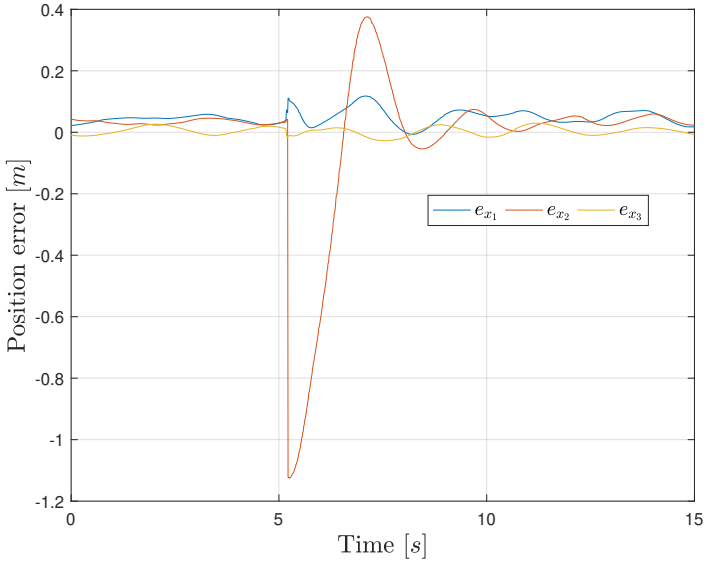


Figure 6.8: *Set-point tracking experiment (CPID): position tracking error - e_x.*

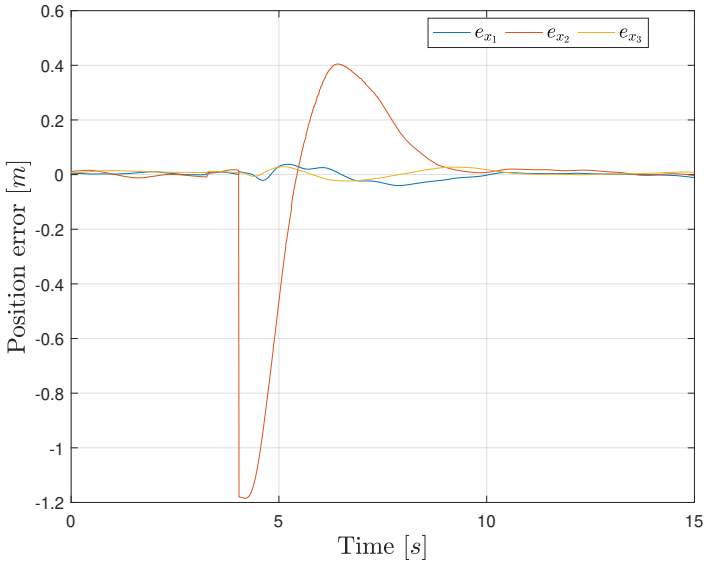


Figure 6.9: *Set-point tracking experiment (IQTO): position tracking error - e_x.*

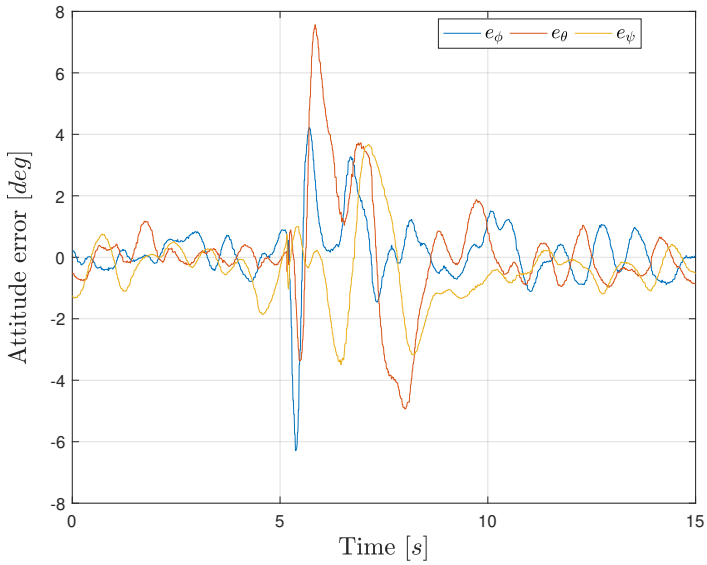


Figure 6.10: *Set-point tracking experiment (CPID): attitude error in terms of roll-pitch-yaw angle errors.*

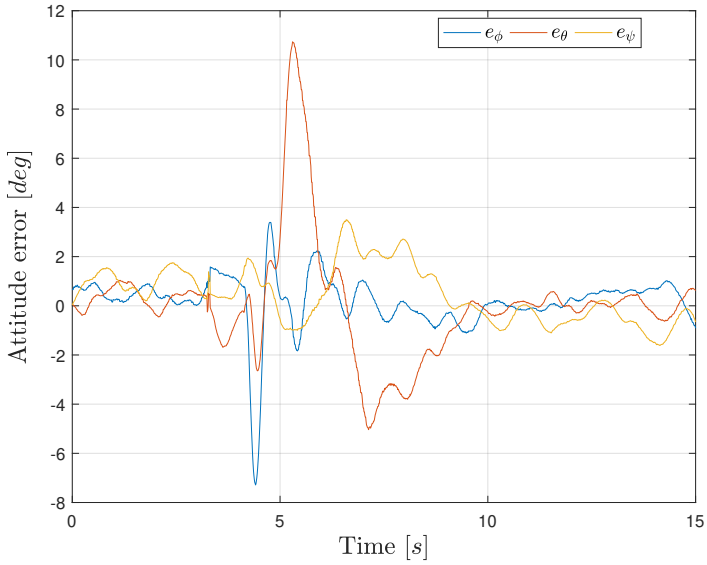


Figure 6.11: *Set-point tracking experiment (IQTO): attitude error in terms of roll-pitch-yaw angle errors.*

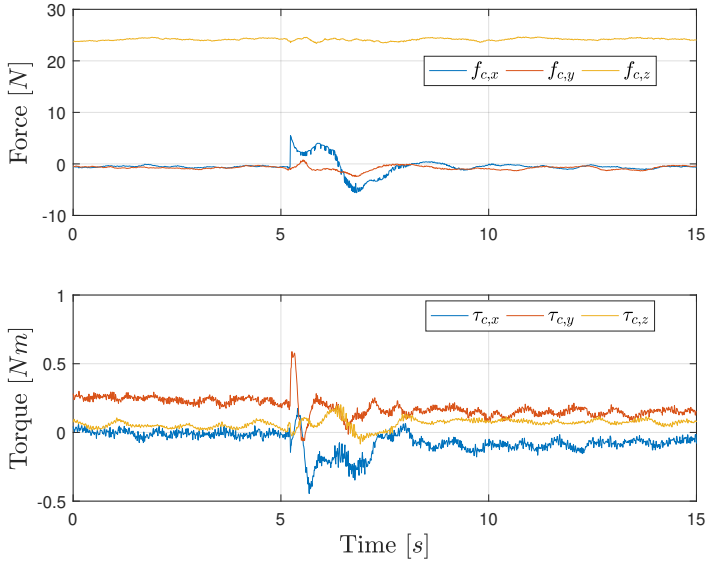


Figure 6.12: *Set-point tracking experiment (CPID): control force and torque -* (f_c, τ_c).

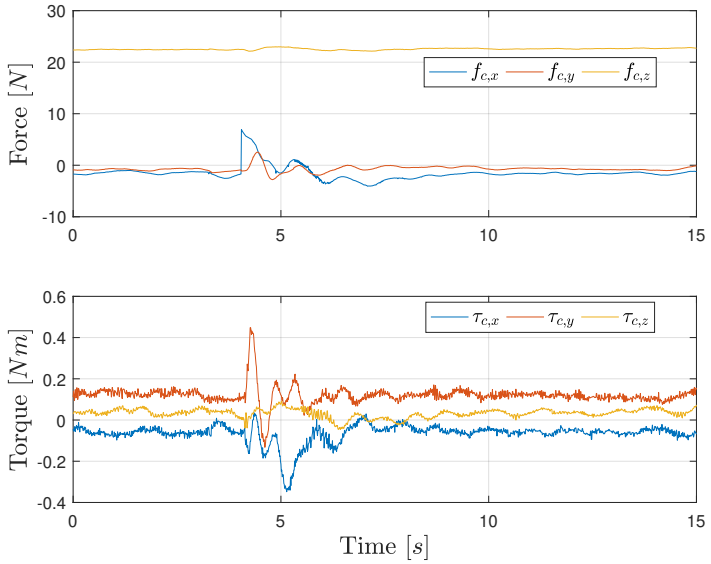


Figure 6.13: *Set-point tracking experiment (IQTO): control force and torque -* (f_c, τ_c).

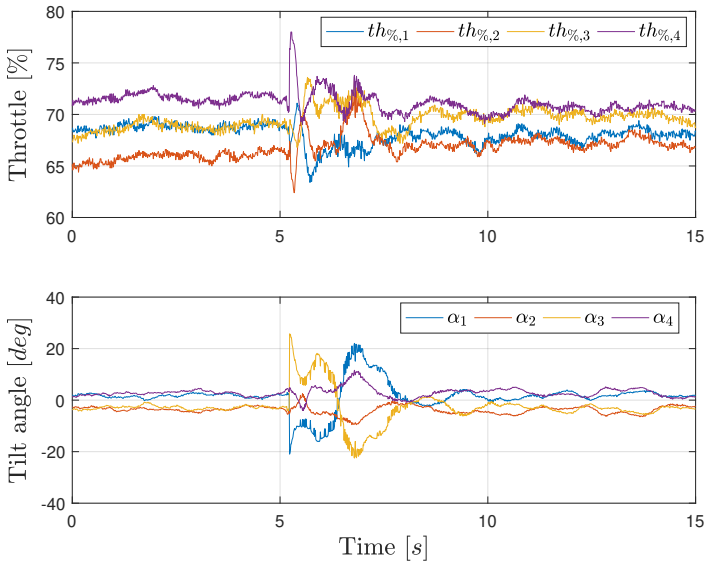


Figure 6.14: *Set-point tracking experiment (CPID): Percentage motors thrust (top) - tilt angles (bottom).*

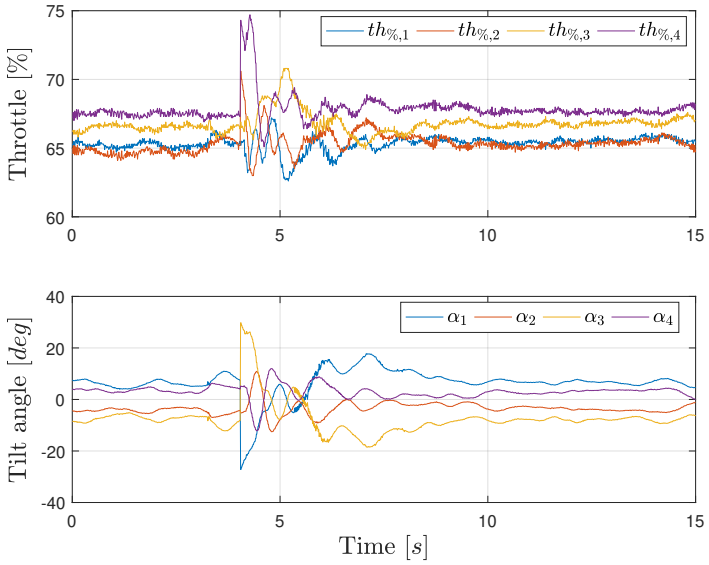


Figure 6.15: *Set-point tracking experiment (IQTO): Percentage motors thrust (top) - tilt angles (bottom).*

a straight segment of 2.5 m back and forth, at constant altitude, in 10 s, which is assigned as a fifth-order polynomial function of time. The maximum speed and acceleration along the trajectory are $v_M = 0.95\,\mathrm{m\,s^{-1}}$ and $\dot{v}_M = 0.6\,\mathrm{m\,s^{-2}}$.

As done for the first experiment, the following quantities have been collected: position tracking errors (Figures 6.16-6.17), attitude errors, in terms roll-pitch-yaw angle errors (Figures 6.18-6.19), the control force and torque (Figures 6.20-6.21) and the percentage thrust of each motor and the tilt angles commanded of the servo-actuators (Figures 6.22-6.23), as computed by the different controllers. As for the previous example, it is possible to see from Figures 6.20-6.23 the vibration noise introduced by the gyroscope measurements. Tables 6.7 and 6.8 report the mean and root mean square error (rmse) of the position and attitude tracking errors, respectively (therein e_ϕ, e_θ, e_ψ represents the error between the desired and actual roll, pitch and yaw angles). By inspecting the results in Table 6.7, it is clear that the CPID controller has significantly worse tracking capabilities. On the other hand, similar performance are achieved by the controllers in terms of attitude errors (see Table 6.8) for which a constant reference is assigned.

Measure	**CPID**	**IQTO**
$e_{x_{mean}}$	0.0815	0.0023
$e_{y_{mean}}$	0.0681	0.0017
$e_{z_{mean}}$	0.0062	0.0026
$e_{x_{rms}}$	0.0892	0.0117
$e_{y_{rms}}$	0.0992	0.0128
$e_{z_{rms}}$	0.0126	0.0117

Table 6.7: *Mean and root mean square error of the position tracking error components* [m]

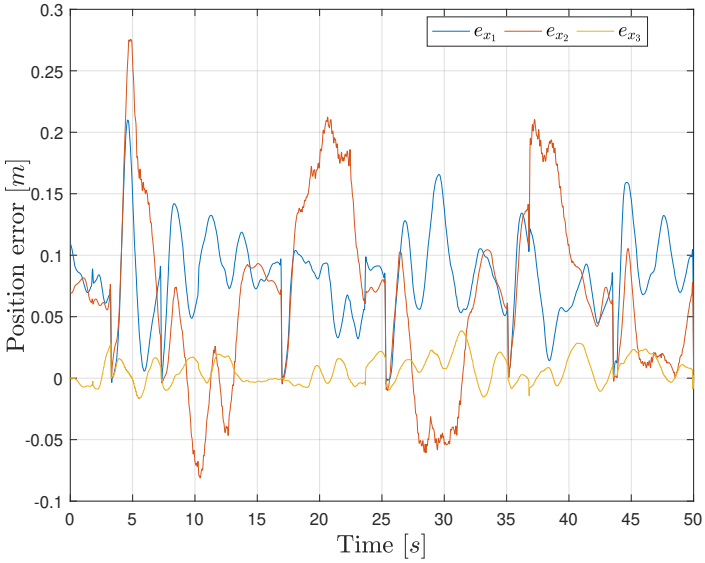


Figure 6.16: *Back and forth motion with 25° pitch (CPID): position tracking error - e_x.*

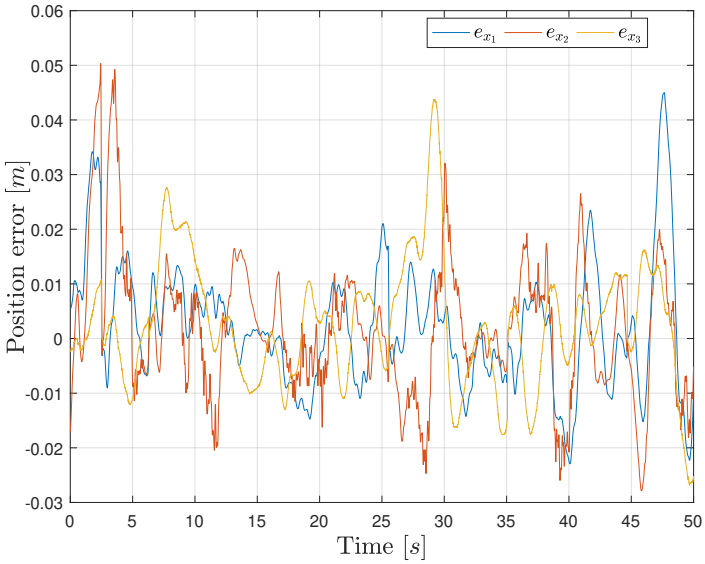


Figure 6.17: *Back and forth motion with 25° pitch (IQTO): position tracking error - e_x.*

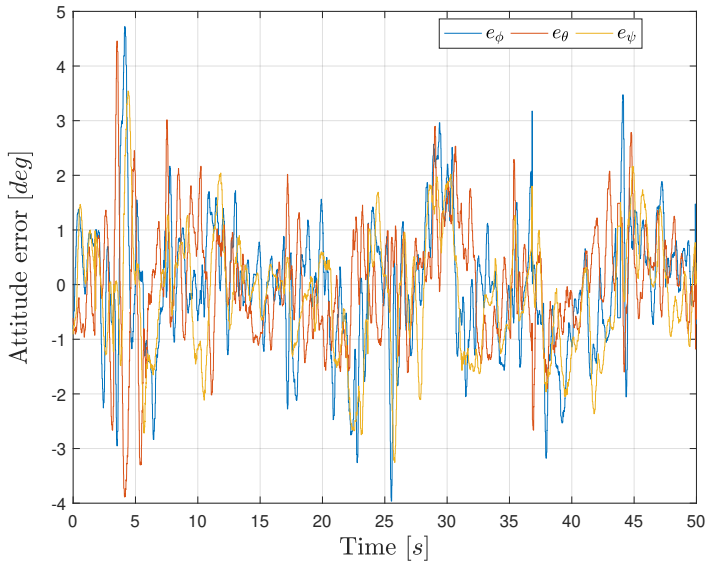


Figure 6.18: *Back and forth motion with 25° pitch (CPID): attitude error in terms of roll-pitch-yaw angles errors.*

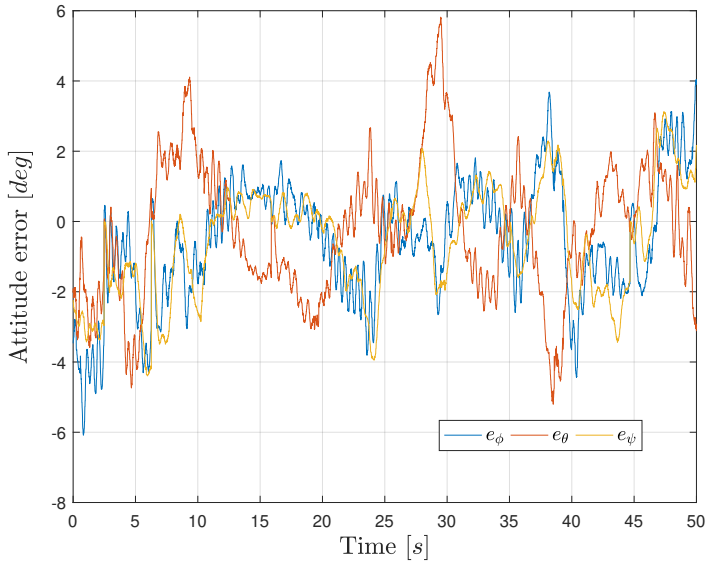


Figure 6.19: *Back and forth motion with 25° pitch (IQTO): attitude error in terms of roll-pitch-yaw angles errors.*

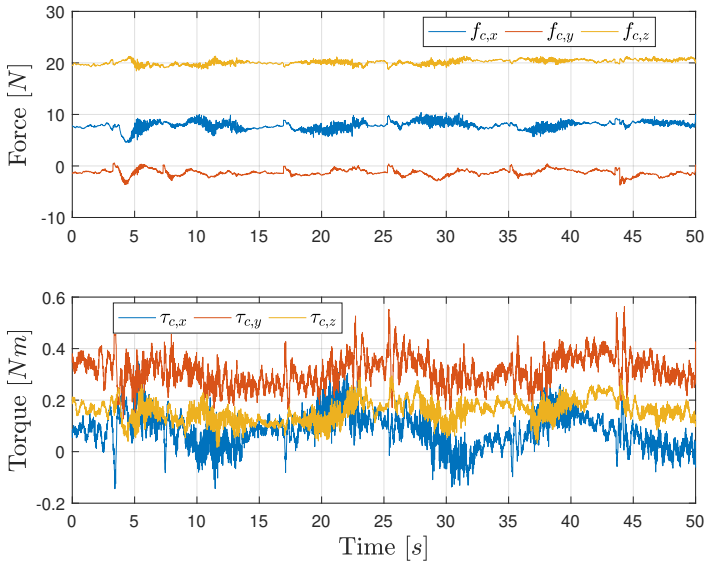


Figure 6.20: *Back and forth motion with 25° pitch (CPID): control force and torque - (f_c, τ_c).*

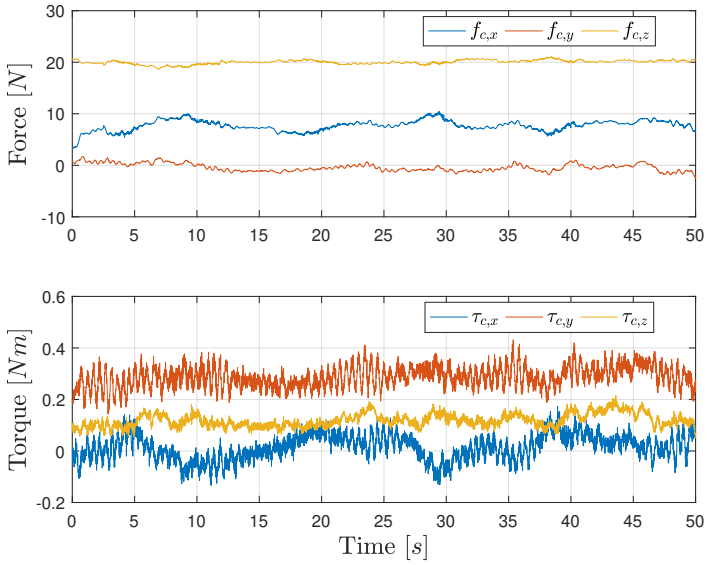


Figure 6.21: *Back and forth motion with 25° pitch (IQTO): control force and torque - (f_c, τ_c).*

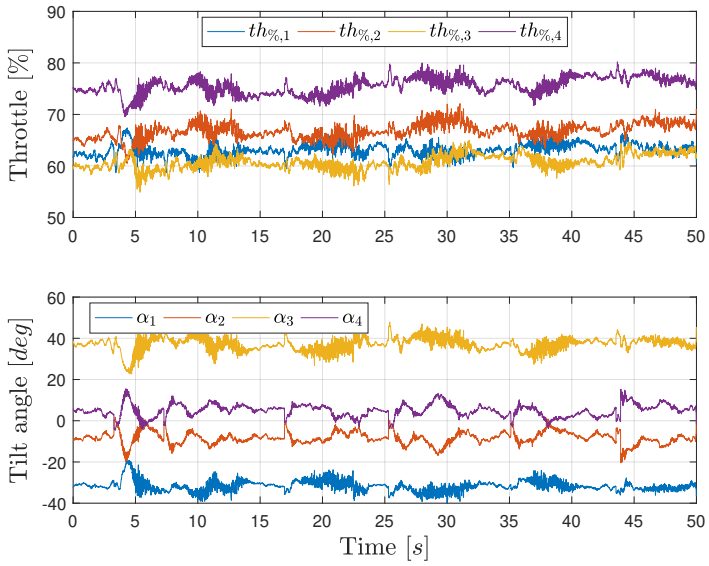


Figure 6.22: *Back and forth motion with 25° pitch (CPID): percentage motors thrust (top) - tilt angles (bottom).*

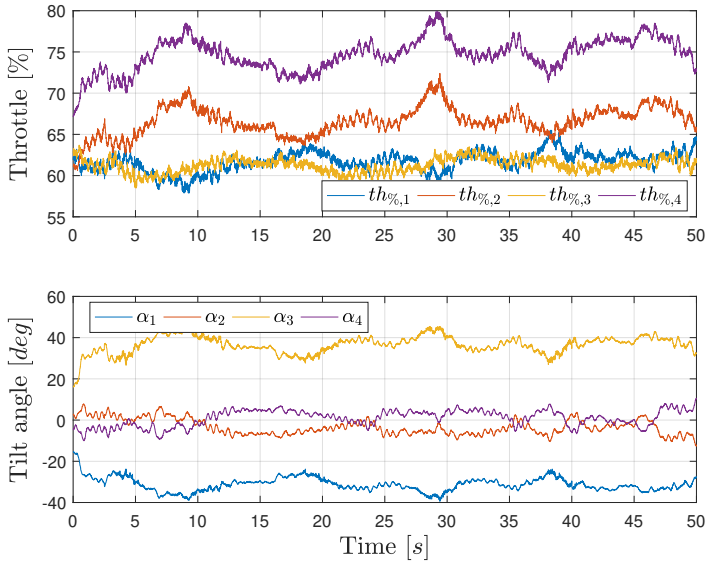


Figure 6.23: *Back and forth motion with 25° pitch (IQTO): percentage motors thrust (top) - tilt angles (bottom).*

Measure	CPID	IQTO
$e_{\phi_{mean}}$	0.0815	0.0023
$e_{\theta_{mean}}$	0.0097	−0.2455
$e_{\psi_{mean}}$	−0.1946	−0.5577
$e_{\phi_{rms}}$	1.1630	1.7880
$e_{\theta_{rms}}$	1.0400	1.9489
$e_{\psi_{rms}}$	1.0996	1.6596

Table 6.8: *Mean and root mean square error of the roll, pitch and yaw angle errors* [°]

6.4.8 Final considerations

The considered architectures have been analysed on theoretical as well as on practical aspects, with specific reference to the trajectory tracking problem. The results of two experiments have been presented: a set-point tracking example and a full-pose tracking example with time-varying reference. While the cascade controller (CPID) was easily tuned to achieve tight performance around hovering conditions, the IQTO controller has shown superior performance in a full-pose trajectory tracking scenario. Finally, it is worth mentioning that the IQTO tracking controller, which is based on a quasi-time optimal stabilizer that accounts by design for actuators saturation, is more suitable to handle cases with significant initial errors.

6.5 Full-pose tracking control with laterally-bounded input force

The fully-actuated platform as the ones described in this thesis need methods to control them efficiently and to reliably cope with the added complexity of the platforms and of the larger set of tasks in which they can be employed, when compared to standard collinear multirotors. To fill this gap, in [90] a method for controlling such platforms while taking into account the most limiting input bounds they have to cope with (*i.e.*, lateral input force) is presented. The proposed controller ensures, in nominal

conditions, the tracking of a full-6D pose reference trajectory (position plus orientation). If the reference orientation and the force needed to track the position trajectory do not comply with the platform constraints, the proposed strategy gives priority to the tracking of the positional part while also tracking the feasible orientation that is the closest to the reference one. In [90] the control technique is presented both theoretically and experimentally tested on the Tilt-Hex platform (see [13] for the details on the Tilt-Hex modelling) while in this thesis a modified version of such controller will be tested on the FAST-Hex platform (described in Section 1.2.3).

In this section the control architecture, which is composed by an inner *attitude controller* and an outer *position controller*, will be described. The controllers are then cascaded by the *wrench mapper* presented in Section 1.2.3.

6.5.1 Position control

The position controller takes as input the full-pose trajectory (p_r, $\dot{p}_r$, $\ddot{p}_r \in \mathbb{R}^3$ and $R_r = [b_{1r} b_{2r} b_{3r}] \in \mathrm{SO}(3)$), the measured position (p_E), the measured linear velocity ($\dot{p}_E$) and the measured attitude (R). It produces as output the desired orientation ($R_d \in \mathrm{SO}(3)$) and the desired control force (f_c).

Given the considered input one can define the position and velocity tracking errors respectively as follow

$$e_p = p_E - p_r, \qquad e_v = \dot{e}_p = \dot{p}_E - \dot{p}_r. \tag{6.78}$$

It is then possible to define the integral position tracking error as

$$e_{pi} = \int_0^t e_p d\tau. \tag{6.79}$$

The reference force vector is then computed as

$$f_r = m(\ddot{p}_r + g e_3) - K_p e_p - K_{pi} e_{pi} - K_v e_v, \tag{6.80}$$

where K_p, K_{pi}, $K_v \in \mathbb{R}^3$ are positive diagonal matrices.

Algorithm 1: Computation of R_d with the bisection method

Data: n_{it} (number of iterations $\propto$ solution accuracy)
Data: b_{3r}, f_r, $r_{xy}(\alpha)$
$\theta_{max} \leftarrow \arcsin\left(\frac{\|b_{3r}\times f_r\|}{\|f_r\|}\right)$, $\theta \leftarrow \frac{\theta_{max}}{2}$, $k \leftarrow \frac{b_{3r}\times f_r}{\|b_{3r}\times f_r\|}$;
for $i = 1$ *to* n_{it} **do**
  $b_{3d} \leftarrow b_{3r}c_\theta + (k \times b_{3r})\, s_\theta + k\,(k \cdot b_{3r})\,(1 - c_\theta)$;
  if $f_r^\top b_{3d} \geq \sqrt{\|f_r\|^2 - r_{xy}(\alpha)^2}$ **then** $\theta \leftarrow \theta - \frac{\theta_{max}}{2}\frac{1}{2^i}$;
  else $\theta \leftarrow \theta + \frac{\theta_{max}}{2}\frac{1}{2^i}$;
return θ

Such force vector is then rotated from the inertial to the body frame and saturated assuming a cylindric bounded force as described in [90] in order to obtain the desired control force

$$f_c = sat_{U_{xy}}\left((f_r{}^\top Re_1)e_1 + (f_r{}^\top Re_2)e_2\right) + (f_r{}^\top Re_3)e_3, \tag{6.81}$$

$$U_{xy}(\alpha) = \left\{ \begin{bmatrix} u_1 & u_2 \end{bmatrix}^T \in R^2 | u_1^2 + u_2^2 \leq r_{xy}^2(\alpha) \right\}, \tag{6.82}$$

where $r_{xy}(\alpha)$ will be described in Section 6.5.3.

The desired orientation instead is computed taking into account the request orientation, the reference force vector and the lateral force bound as described in Algorithm 1 which is an improvement of the algorithm presented in [90]. In particular, c_θ and s_θ are the cosine and sine of θ, respectively. Finally it is possible to compute the desired orientation as follows:

$$b_{3d} = b_{3r}c_\theta + (k \times b_{3r})\, s_\theta + k\,(k \cdot b_{3r})\,(1 - c_\theta) \tag{6.83}$$

$$R_d = \begin{bmatrix} \underbrace{(b_{3d} \times b_{1r}) \times b_{3d}}_{b_{1d}} & \underbrace{b_{3d} \times b_{1r}}_{b_{2d}} & b_{3d} \end{bmatrix}. \tag{6.84}$$

6.5.2 Attitude control

The attitude controller takes as input the desired orientation computed from the position controller (R_d), the measured orientation (R) and the measured angular rate (ω_b) to compute the desired control torque (τ_c).

The desired control torque is computed as

$$\tau_c = \omega_b \times J\omega_b - K_R e_R - K_{Ri} e_{Ri} - K_\omega \omega_b, \tag{6.85}$$

where K_R, K_{Ri}, $K_\omega \in \mathbb{R}^3$ are positive definite diagonal matrices and e_R is the orientation tracking error defined as

$$e_R = \frac{1}{2}\left(R_d^\top R - R^\top R_d\right)^\vee, \tag{6.86}$$

with $\bullet^\vee$ which is the Vee map from $\mathrm{SO}(3)$ to $\mathbb{R}^3$ and e_{Ri} the integral orientation tracking error computed as

$$e_{Ri} = \int_0^t e_R d\tau. \tag{6.87}$$

6.5.3 Lateral force saturation

The lateral force achievable by the FAST-Hex is directly proportional to the tilting angle (α) of the propellers. It has been then decided to define the lateral force bound used in the position controller r_{xy} as a function of α.

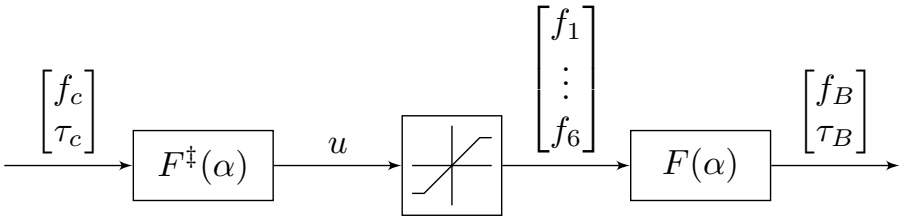


Figure 6.24: *Wrench allocation with actuators' saturation*

To do so, the scheme represented in Figure 6.24, in which the propellers' saturation has been taken into account, has been considered. The maximum achievable lateral force considering as input of the wrench mapper a desired lateral force (*e.g.*, 10 N) around the hovering conditions with a null desired moment for different values of α has been then numerically computed. The obtained saturated force is reported in Figure 6.25.

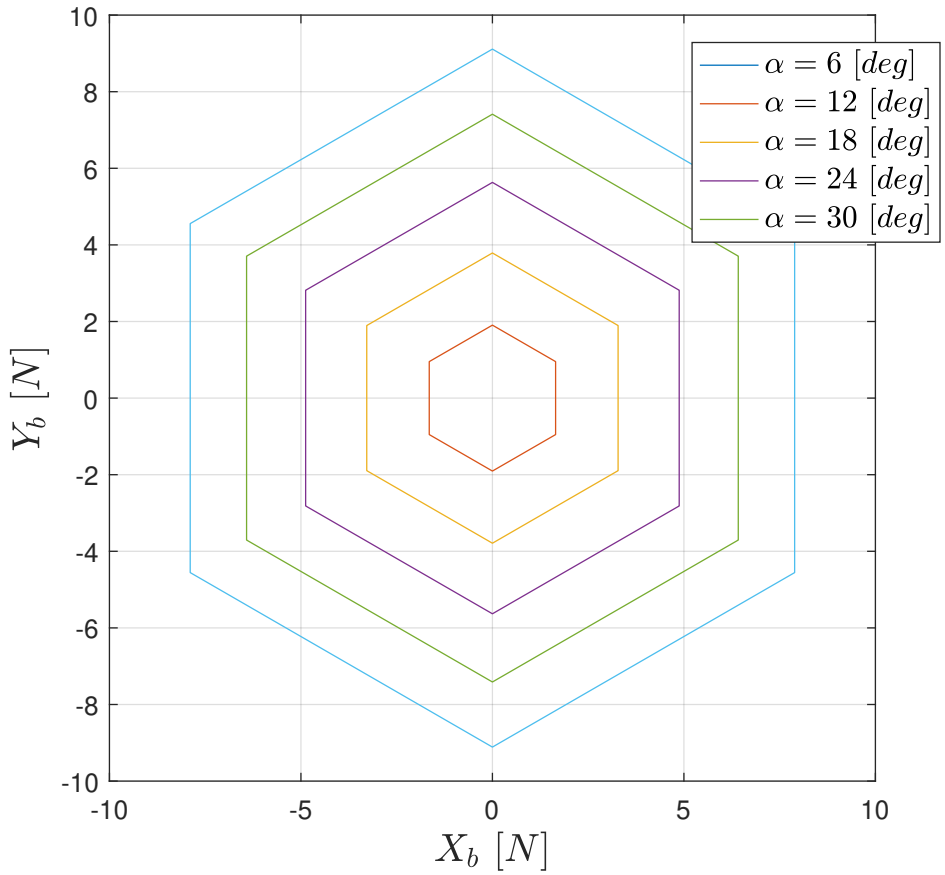


Figure 6.25: *Saturated lateral force for different values of* α.

Since the obtained planar lateral forces have a hexagon shape, for the sake of simplicity it has been decided to consider as a lateral bound the circle inscribed in each hexagon. The radius of the circles with reference to the tilt angle are reported in Figure 6.26.

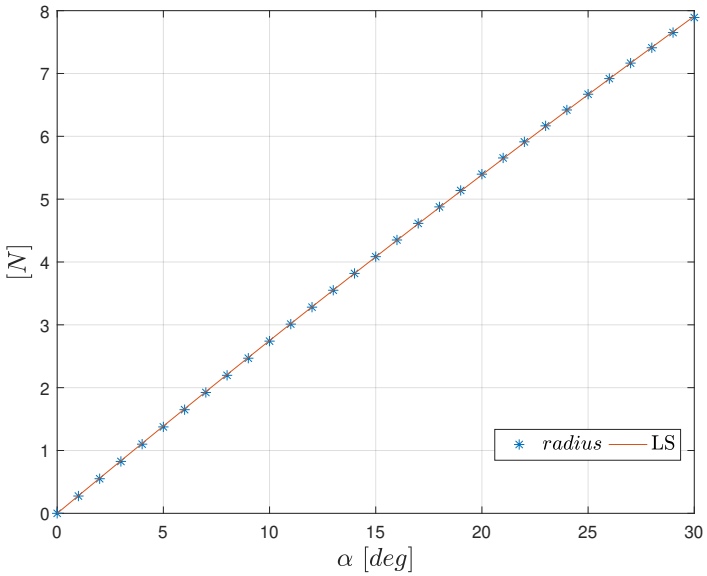


Figure 6.26: *Interpolated values of the inscribed circles' radius.*

To exploit r_{xy} as a function of α a Least Squares (LS) approach has been used to interpolate the obtained values with a degree two polynomial.

Finally, the obtained polynomial has been scaled down with a tunable gain leading to a more conservative lateral force bound. To asses the numerical problem related to the ill-conditioned pseudo inverse a dead-zone in proximity of $\alpha = 0$ has been introduced (see Figure 6.27).

6.5.4 Experimental results

The physical parameters of the FAST-Hex are reported in Table 6.9, while the controller gains and the actuator limits are collected in Table 6.10.

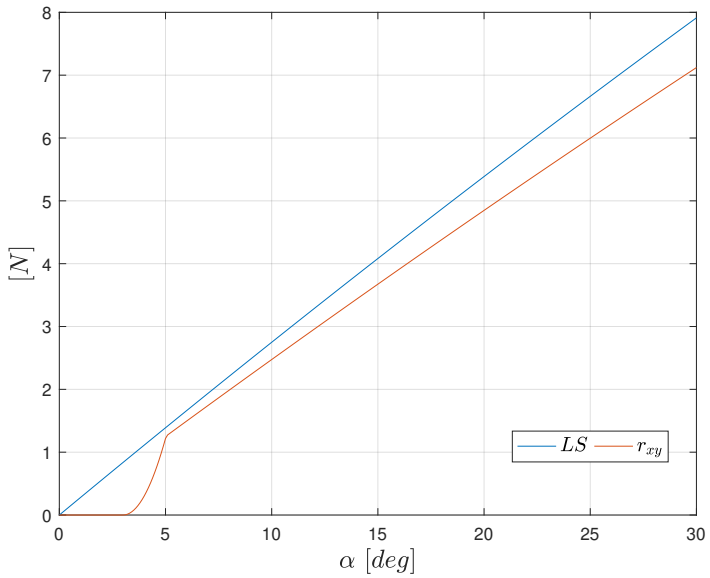


Figure 6.27: *Lateral force saturation function.*

Transition experiment

The reference trajectory for this experiment is characterized by a constant position, equal to the initial rest position of the UAV at the beginning of the test plus an offset along the vertical direction, and a constant level attitude. The reference linear and angular velocity are set to zero, accordingly. Furthermore, the reference α_r for the synchronized tilting angle of the actuators has a rectangular profile between the values $\alpha_1 = 0°$ and $\alpha_2 = 30°$. As a consequence, the UAV instantaneously switches its configuration between under-actuated and fully-actuated.

Motivation This reference trajectory is designed with the goal of testing the controller capability to safely drive the platform in the two configurations, maintaining stability also during transitions. Furthermore, through this test the controller robustness towards the un-modeled aerodynamic effects is assessed. Moreover, rotating the spinning axes of the propellers induces gyroscopic moments on the platform which, however, typically tend to cancel out during hovering conditions, thanks to the symmetric

Table 6.9: *Physical parameters of the FAST-Hex prototype.*

Tilt-Hex		
Parameter	Value	Unit
m	3.1	kg
$J(:,1)$	$[0.089 \ -0.001 \ -0.001]^\top$	$\mathrm{kg\,m^2}$
$J(:,2)$	$[-0.001 \quad 0.091 \quad 0]^\top$	$\mathrm{kg\,m^2}$
$J(:,3)$	$[-0.001 \quad 0 \quad 0.164]^\top$	$\mathrm{kg\,m^2}$
c_i	$(-1)^{i-1}$	[ ]
c_f^τ	$1.9e-2$	m
c_f	$9.9e-4$	$\mathrm{N/Hz^2}$
$\lvert\alpha\rvert$	$\in [0,\ 30]$	deg
β	0	deg
l	0.305	m

arrangement of the propellers.

Analysis and discussion of the results The reference set-points and the measurements related to the platform position, linear velocity and angular velocity are depicted, from top to bottom, in Figure 6.28. The plots show that the measured trajectory always keeps close to the reference one despite the variation of the tilting angle α, as confirmed by the plots of Figure 6.30, which depict the state errors used by the control algorithm to produce the input from the state feedback. Small oscillations in the linear and angular velocities can be observed around $t = 40\,\mathrm{s}$, when the configuration of the UAV progressively changes from fully-actuated to under-actuated. The evolution of α, which is represented in the first plot of Figure 6.32, allows to conceptually divide the experiment in three parts. During the initial and final parts of the plots, for $t \in [0,\ 18.6]\mathrm{s}$, and $t \in [40,\ 57]\mathrm{s}$, the UAV can generate a thrust force only along one direction in body frame, since $\alpha = 0$. As a consequence, the two rotational

Table 6.10: *Controller parameters used in the experiments.*

Parameter	Value	Unit
$K_p(j,j)\|_{j=1,2,3}$	50, 50, 50	[ ]
$K_{pi}(j,j)\|_{j=1,2,3}$	20, 20, 20	[ ]
$K_v(j,j)\|_{j=1,2,3}$	14.14, 14.14, 14.14	[ ]
$K_R(j,j)\|_{j=1,2,3}$	15, 15, 6	[ ]
$K_{Ri}(j,j)\|_{j=1,2,3}$	1, 1, 1	[ ]
$K_w(j,j)\|_{j=1,2,3}$	1.5, 1.5, 0.5	[ ]
$\overline{e}_{pi} = -\underline{e}_{pi}$	$[3\ 3\ 3]^\top$	m
$\overline{e}_{Ri} = -\underline{e}_{Ri}$	$[1\ 1\ 1]^\top$	rad
n_{it}	20	[ ]
$\overline{w_i}$	98	Hz
$\underline{w_i}$	20	Hz
$\overline{\dot{\alpha}} = -\underline{\dot{\alpha}}$	10	deg/s

degrees of freedom associated with the body roll and pitch rotations are directly coupled with the two degrees of freedom related to the translations on the horizontal plane. Therefore, the UAV is under-actuated, in the sense that it cannot be commanded to follow arbitrary trajectories in its full configurations space.

On the other hand, for $t \in [40,\ 57]$s, the UAV can exploit the fully-actuated capability to partially decouple the tracking of the rotational dynamics from the translational one. Due to the limitations of the servo-motor for the synchronized tilting of the actuators, the measured tilting angle follows a trapezoidal profile with maximum velocity $\overline{\dot{\alpha}} = -\underline{\dot{\alpha}} = 10^\circ\,\text{s}^{-1}$.

The plots of Figure 6.29 induce interesting considerations on the behaviour of the inner attitude control loop. Comparing the first and the

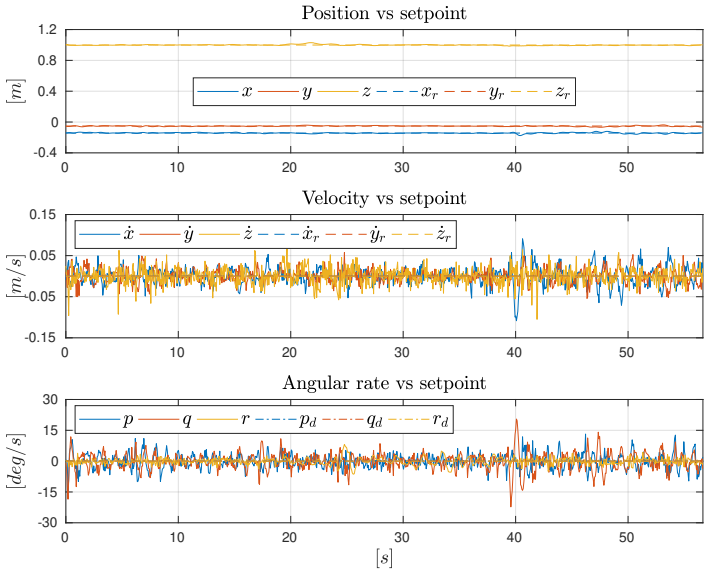


Figure 6.28: *Transition: plots of the state set-points and the state evolution.*

second plot, it is clear how the control algorithm re-computes the desired orientation for the system (from the reference one) during the first and the third phases. In this case, the desired orientation is continuously regulated to correct position errors. The non-zero mean for ϕ_d and θ_d is due to parameter mismatches between the model and the real systems, especially of those associated with the orientation of the actuators, and to external disturbances like the one induced by the serial cable. This is confirmed by the plots of Figure 6.31. Notably, the first one shows that the UAV applies a small force along the x body axis with a negative non-zero mean. Conversely, as soon as the angle α is large enough the UAV can exert lateral forces without the need of re-orient itself, see the second plot, and so the desired attitude can be constantly flat.

The second and the third plots of Figure 6.32 depict the evolution of the desired spinning velocities for the rotors, computed by the pose controller, and the measured forces produced by the propellers. As can be appreciated, the signals remain bounded by their limits, which demonstrates the controller's ability to comply with the actuator bounds. Furthermore, it is

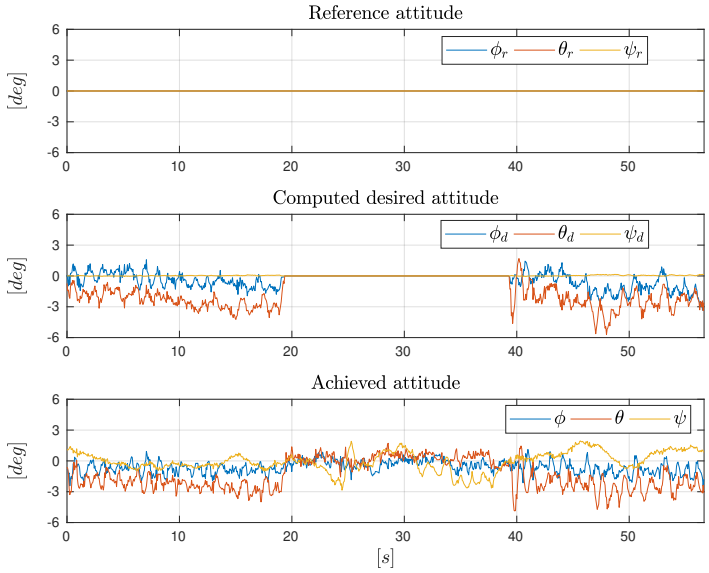


Figure 6.29: *Transition: plots of the state set-points and the state evolution.*

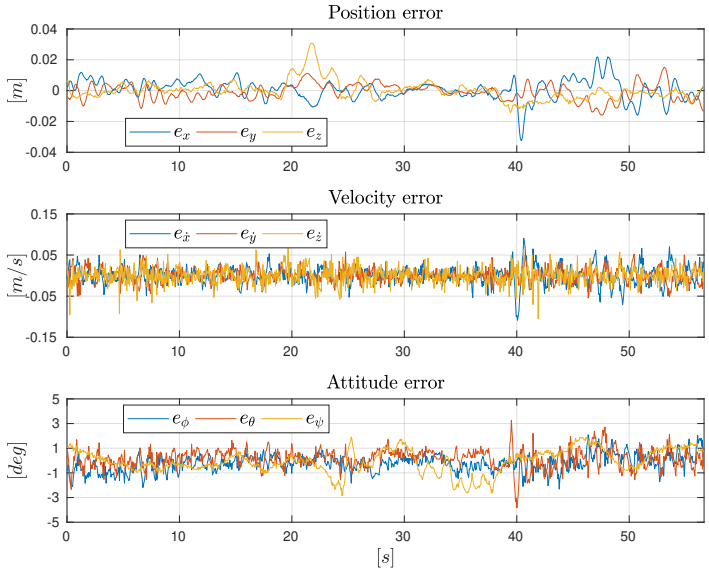


Figure 6.30: *Transition: plots of the controller errors.*

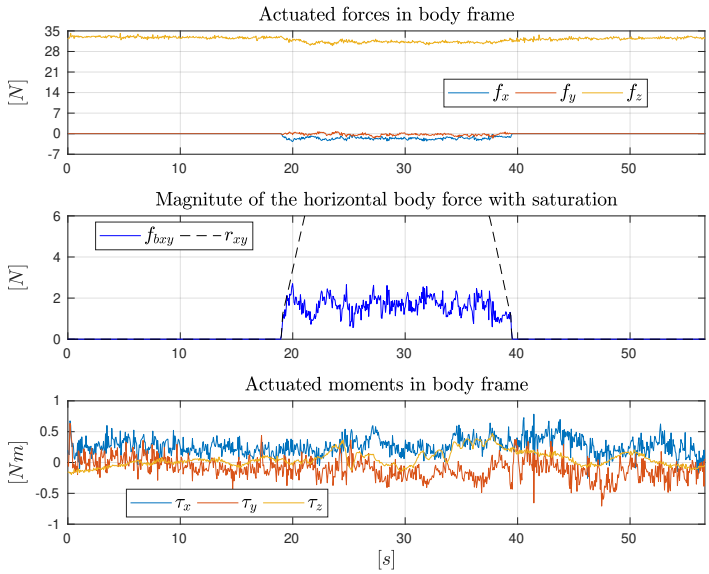


Figure 6.31: *Transition: plots of the UAV body wrench.*

worthwhile to observe the small discontinuities in the actuator commands during the changes of configuration.

Finally, the plots of Figure 6.33 and Figure 6.34 outline all the partial terms of the control laws for the translational and the rotational system dynamics, respectively. In particular, the second plots show how the integral terms, which never reach their saturations (cf. Table 6.10), help in reducing the steady-state errors.

Hovering with sinusoidal roll

In this second experiment the UAV is asked to hover in a fixed position, as can be seen from Figure 6.35, while oscillating in a sinusoidal fashion around the roll axis (see Figure 6.36). The α angle is initially set to zero and after 26 s it is slowly increased up to 30° following a ramp trajectory.

Motivation As has been already described above, the FAST-Hex can change its configuration from under-actuated to fully-actuated by changing the tilting angle of the propellers. The controller presented in this section

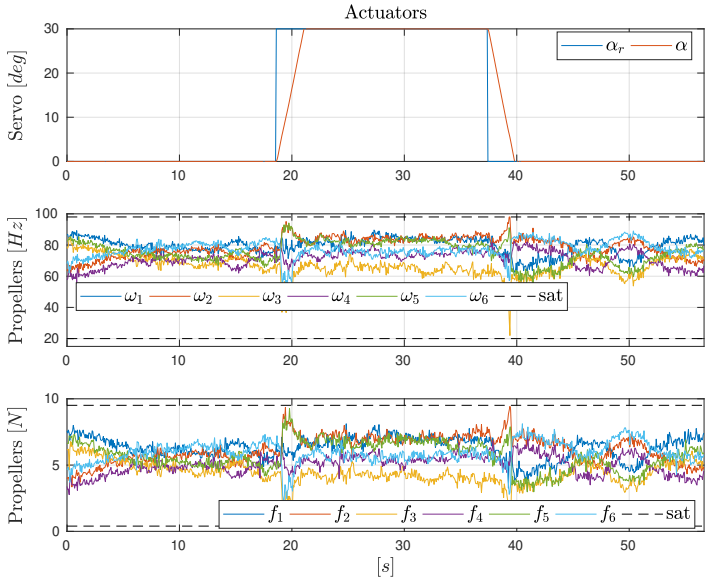


Figure 6.32: *Transition: plots of the actuators data.*

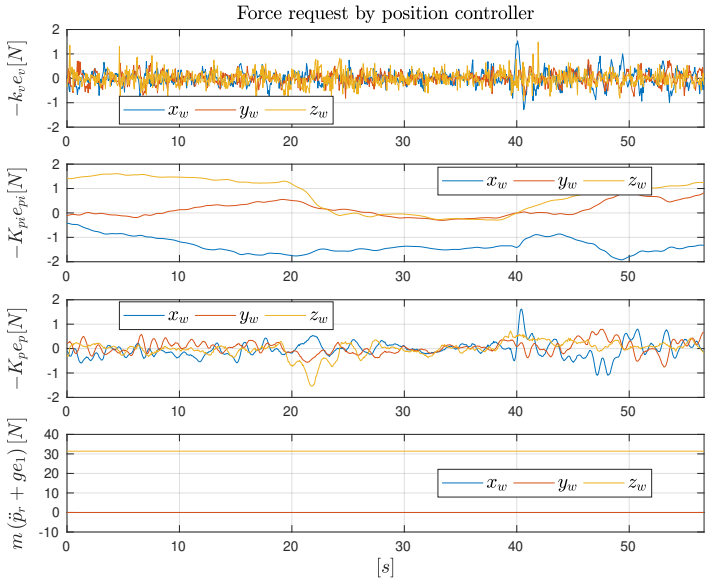


Figure 6.33: *Transition: plots of the different terms in the position control.*

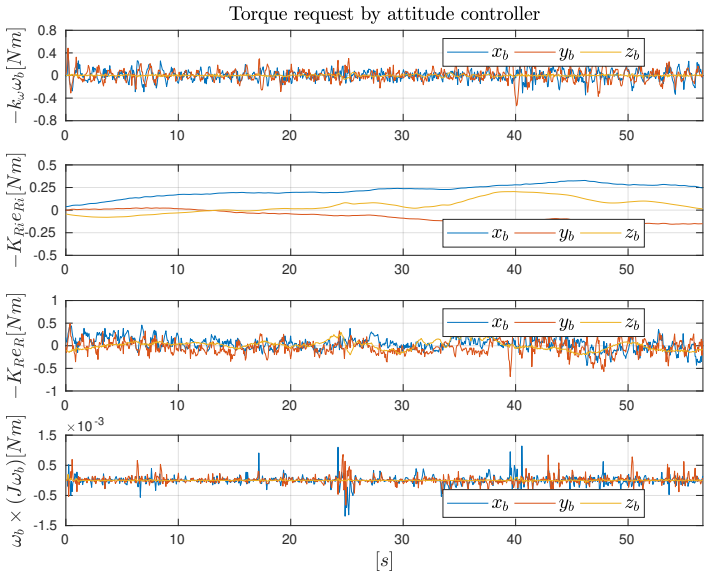


Figure 6.34: *Transition: plots of the different terms in the orientation control.*

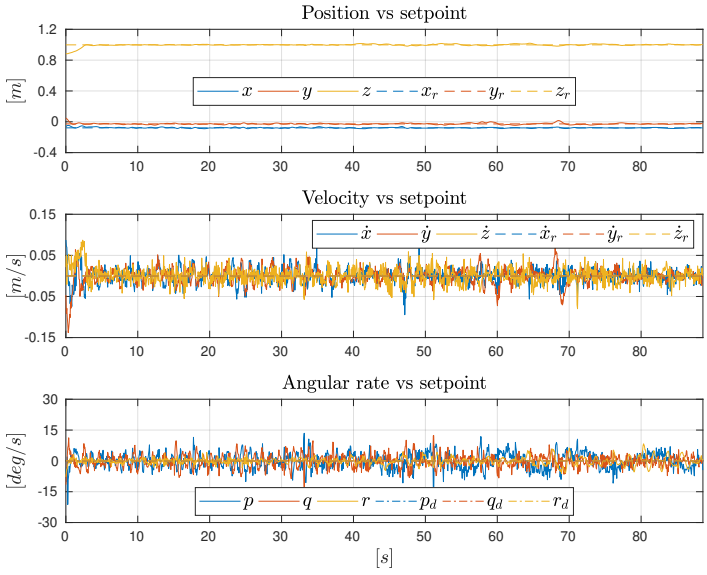


Figure 6.35: *Sinusoidal roll: plots of the state set-points and the state evolution.*

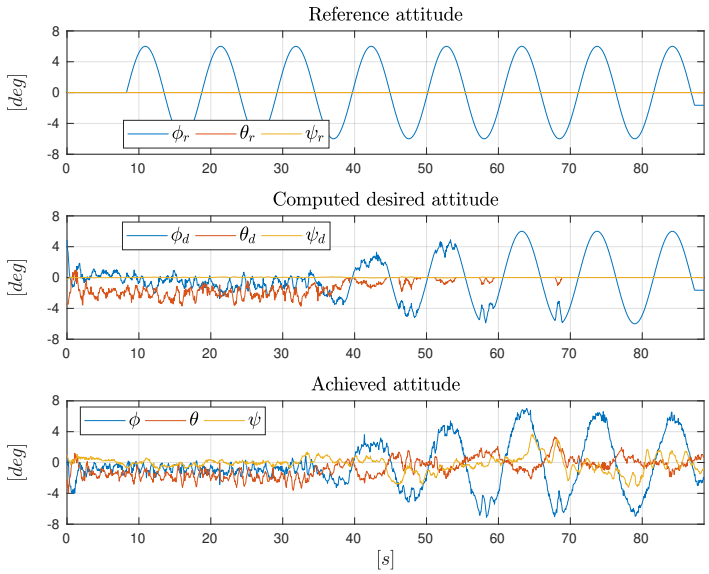


Figure 6.36: *Sinusoidal roll: plots of the state set-points and the state evolution.*

is a full-pose control structure with prioritisation of the position tracking, which means it can recompute the attitude set-point according to the achievable lateral force, *i.e.*, according to the force bound proportional to the tilting angle. This test is aimed to show the capability of the control algorithm to recompute the attitude set-point using the bisection method over the entire range of the achievable lateral forces.

Analysis and discussion of the results The first interesting result is reported in Figure 6.36 in which from top to bottom one can see the requested sinusoidal trajectory around the roll axis, the computed desired attitude and finally the achieved one. It can be seen as the computed desired attitude trajectory before 26 s is close to 0° since the platform is in under-actuated configuration due to the the tilting propellers in vertical shape, while it slowly starts to assume a sinusoidal shape when the lateral force bound allows the fully-actuation.

From Figure 6.37 one can see the good performance of the controller despite some loss of performance when the platform is fully-actuated.

Such degradation is caused by the saturation of the actuators, which is easily reached in full-actuation (see Figure 6.38). While the tilting angle is large an internal lateral force has to be compensated, which means forces produced by the propellers(*i.e.*, the angular speed) have to increase compared to the under-actuated case.

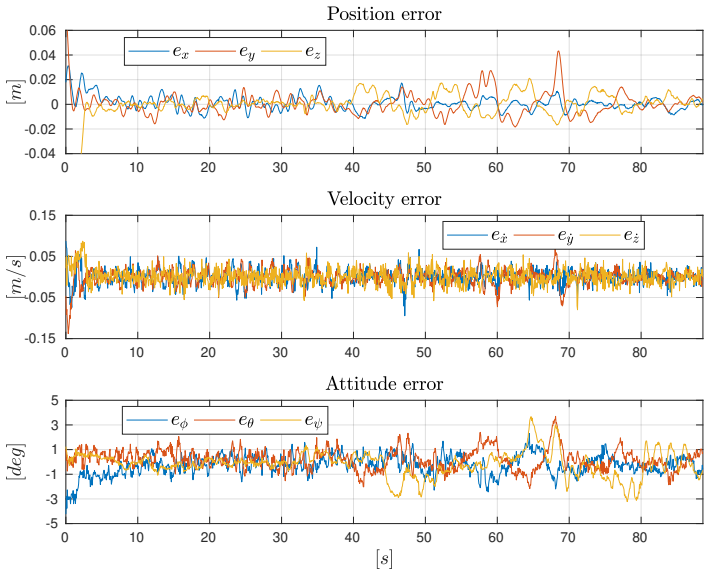


Figure 6.37: *Sinusoidal roll: plots of the controller errors.*

In Figure 6.39 one can see how the lateral force bound increases according to α. Furthermore, one can see when the lateral force hits the saturation corresponds to a re-computation of the desired attitude set-point shown in Figure 6.36.

Finally, as for the previous experiment, the plots of Figure 6.40 and Figure 6.41 outline all the partial terms of the control laws for the translational and the rotational system dynamics, respectively. In particular, the second plots show how the integral terms, which never reach their saturations (cf. Table 6.10), help in reducing the steady-state errors.

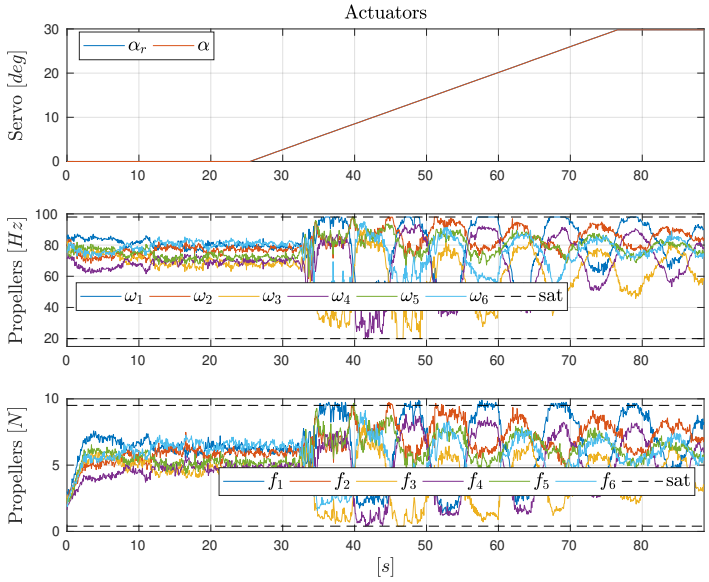


Figure 6.38: *Sinusoidal roll: plots of the actuators data.*

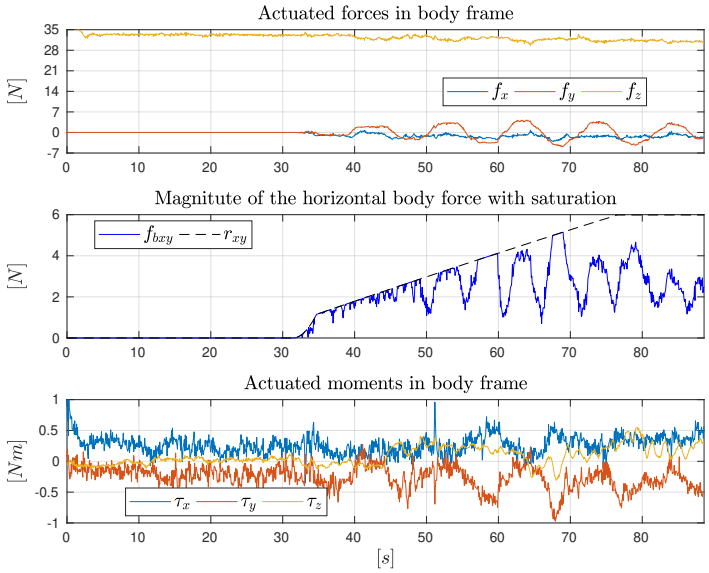


Figure 6.39: *Sinusoidal roll: plots of the UAV body wrench.*

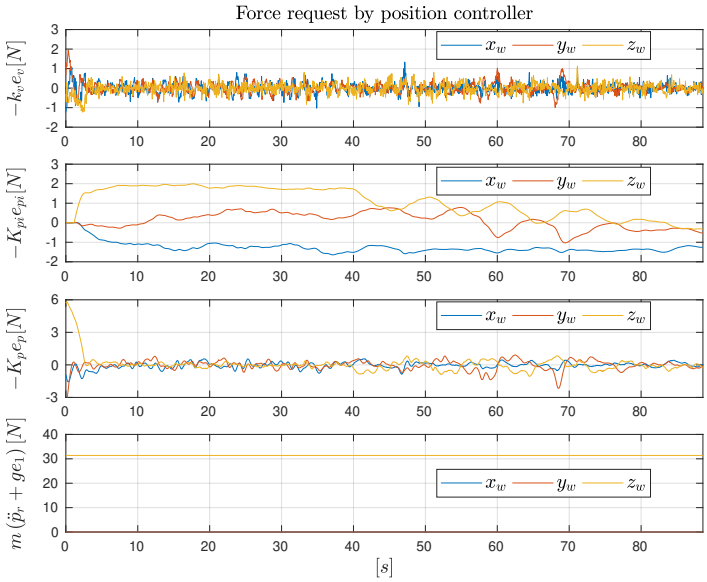


Figure 6.40: *Sinusoidal roll: plots of the different terms in the position control.*

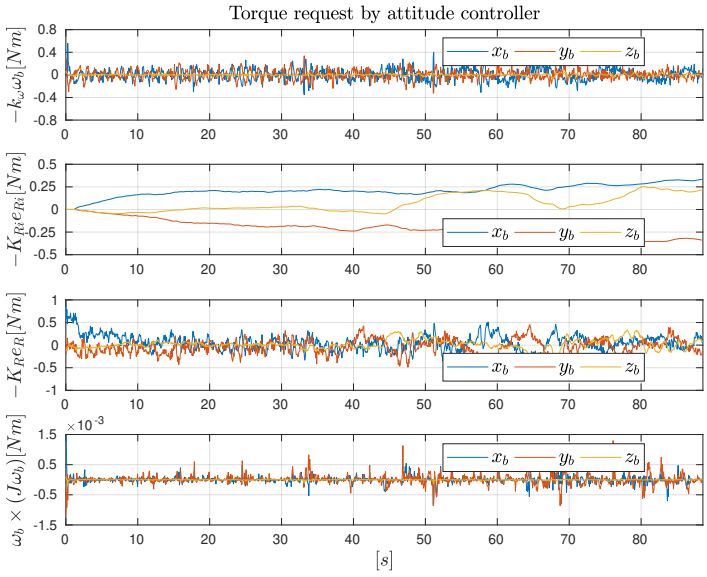


Figure 6.41: *Sinusoidal roll: plots of the different terms in the orientation control.*

6.5.5 Final considerations

In this section an improvement of the full-pose control technique with bounds on the lateral forces has been presented. Moreover such innovative control law has been tested experimentally on a platform which is able to smoothly change between an efficient but under-actuated aerial platform and a fully-actuated but less efficient one. The preliminary obtained results are satisfactory both in terms of trajectory tracking and in terms of the ability to cope with propeller spinning rate saturation.

CHAPTER 7

Interaction

When surveillance, reconnaissance and search-and-rescue missions are considered, small-scale UAVs suffer of low endurance, being usually powered by batteries. For this reason, to extend mission endurance a drone can be used as carrier for lighter and smaller drones (followers) that can take-off from and land on it. This procedure requires the interaction between the two UAVs. New research activities involve the possibility to remotely command many drones simultaneously, following a given path or performing a task in a cooperative way and autonomously, see [98, 99, 100]. Among the main sub-areas covering the cooperative control problem of UAVs, formation flight has attracted great interest and has been widely investigated. Besides multi-rotors formation flight, in the literature it is possible to find researches related to the Air-to-Air Automatic Refuelling (AAAR) involving fixed-wing drones (see [101, 102, 103]). Furthermore, also the landing of a multirotor on a moving platform is a consolidated

problem (see [104, 105, 106]) as well as the landing on a tilted platform (see [107]) or a vertically oscillating one (see [108, 109]) but none of them involves a flying platform.

Moreover, another interesting application in which the automatic landing algorithm could come in hand is the landing of a Vertical Take Off and Landing (VTOL) UAV on the Landing Helicopter Dock (LHD) of a moving ship (see [110, 108, 109]) since this kind of operation is by nature among the most difficult technological and human challenges. This operation is often challenging for the pilot because it forces him to stabilize the helicopter above a moving boat and to precisely position the helicopter on the deck. Weather conditions are often unfavourable, resulting in substantial movement of the ship. In this situation, the pilot's workload is very high, and the risk of collision is not negligible.

In this Chapter the objective is to develop and validate a new set of guidance, navigation and control laws enabling air-to-air UAV landing. In particular this Chapter focuses on the design of guidance laws aimed at providing a small multirotor with a reference descent trajectory. This kind of manoeuvre is as risky as dangerous. The wake of the propellers of a multirotor generates an unsteady flow field around it, so that when two UAVs fly in close proximity they perturb each other. During the landing operation the follower drone is always above the carrier, which constantly flies in a perturbed regime as stronger as the vehicles are closer.

Obvious considerations regarding the size and weights of the two UAVs must be taken into account, *i.e.*, it is necessary that the carrier drone must be larger and heavier than the follower, in order to be able to stand its weight after the touch down. Moreover the on-board controller must be able to reject at least part of the aerodynamic disturbances.

The hazardous nature of this manoeuvre makes the design of the trajectory complex, as the problem is three-dimensional and the carrier in movement. To overcome these issues a decoupled approach is adopted: first a horizontal synchronization between follower and target must be achieved, then the reference landing path is computed and sent to the follower. While the two drones are not in the same horizontal position, the landing is stopped until the synchronization is obtained again.

7.1 Notation

In this section the sign conventions and notation used in the rest of the chapter are presented. In particular the origin of the NED reference frame is located on the ground at the centre of the flying arena in which the experiments are conducted, the North axis points forward, the East axis points rightward and the Down axis downward. Using this convention when the UAVs are flying, the Down position will be negative. The same sign conventions are applied for velocity and acceleration: in particular when the UAV is moving up from the floor it has negative Down velocity $\dot{d}$, while when it descends it has a positive velocity, the downward acceleration is positive, while the upward acceleration is negative.

Moreover all relative quantities are defined as target drone measurements minus follower drones measurements, so relative position and velocity for example along the Down axis, are defined in this way:

$$d_r = d_t - d_f, \tag{7.1}$$

$$\dot{d}_r = \dot{d}_t - \dot{d}_f, \tag{7.2}$$

where the subscript $(\cdot)_r$ refers to relative quantities, $(\cdot)_t$ to target and $(\cdot)_f$ to follower.

We will refer to estimated quantities with the hat symbol $\hat{(\cdot)}$. Finally, the set-points given to a drone are identified by the $(\cdot)^o$ apex.

7.2 Overall control architecture

In this thesis a control architecture capable of performing an autonomous landing of an UAV on another one, while the second is flying, is proposed in two different conditions.

Two problems are investigated separately. The first one deals with the target drone that is moving in the plane following a circular trajectory at constant angular rate. The second one deals with the target that is moving out of the plane, following a sinusoid in the vertical direction. The proposed structure consists of three main modules (as shown in Figure 7.1), as in the previous works [108], [109] and [111], but with some improvements to make the landing possible also in the two conditions described

before.

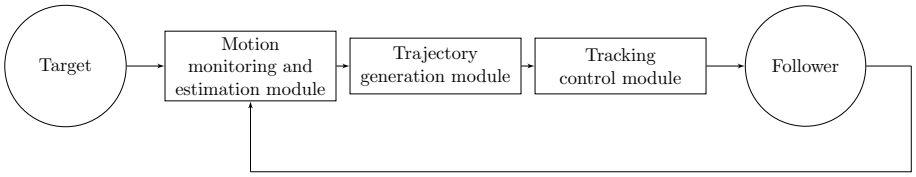


Figure 7.1: *Overall control architecture: three modules.*

7.2.1 Tracking control module

The tracking control module (Figure 7.2) is the same for the two problems and consists in the built-in controllers running on the Flight Control Units (FCUs) of follower and target. They are capable of tracking a given position and a yaw reference with good performance. The UAVs feature a nested control loops architecture as the one described in Chapter 5: an outer position control loop generates the thrust set-point ($f_{c,z}$) and the attitude set-point (q^o) for the inner attitude control loop. The position control loop is based on an inner Proportional-Integral-Derivative (PID) loop for the linear velocity and an outer Proportional (P) loop for the position itself. The attitude controller instead generates the required moments (τ_c) to achieve attitude tracking. This controller is also based on an inner PID loop for the angular velocity and a P loop for the angular attitude angle.

Finally a control allocator block (*mixer matrix*) is defined according to the multirotor frame/configuration in order to convert the force/moments set-points into propeller angular velocity commands. A module for position, velocity and attitude estimation running on the FCU provides then the state estimates which will be used as a feedback for the control loops described above.

7.2.2 Motion monitoring and estimation module

The motion monitoring module manages the feedback position measurements and ensures the safety of the whole procedure. In particular in the case of landing with carrier that is moving in the plane the motion capture

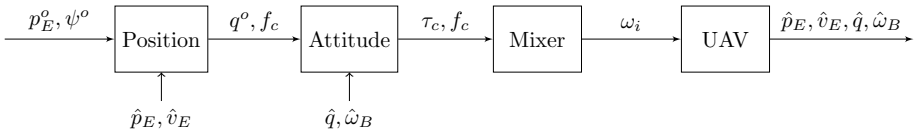


Figure 7.2: *Tracking control module.*

system is able to give position and velocity measurements used to check the safety of the landing procedure, while in the case of the sinusoidal landing an on-line estimation of the target motion is necessary. The latter problem will be analysed in Section 7.4.2.

The most important safety constraint is the in-plane synchronization. This is obtained with the simple and the augmented position control system described in Section 7.3.2 and Section 7.3.2. Only when the follower is in the neighbourhood of the target position, the landing manoeuvre can start. This constraint must be verified for the entire duration of the landing, otherwise the descent is stopped until repositioning and synchronization are re-obtained. Moreover, since the target may voluntarily vary its altitude or experience vertical oscillations caused by the aerodynamic perturbations originating from the wake of the follower's rotors, continuous feedback of the relative vertical distance as well is used to the design real-time reference landing trajectory.

7.2.3 Trajectory generation module

The trajectory generation module has to generate the follower's trajectory so as to perform first of all the synchronization in the plane and finally the landing on the target. To do so, it uses information given by the motion monitoring and estimation module.

The trajectory for the in-plane synchronization is generated in two different ways: the first one is related to a collaborative case, in which the follower knows the exact position of the target and synchronization is performed giving to the follower the target position as set-point (simple position control, Section 7.3.2); the second one is related to a non-collaborative case where only the in-plane relative distance is known and starting from it, the follower's acceleration set-point is computed through

an augmented position control presented in Section 7.3.2. This acceleration is integrated twice to obtain the set-point position. This can be interpreted as a feedback loop on the follower position, where the control variable is the follower acceleration set-point; this control loop is at a higher level with respect to the position controller implemented in the tracking control module.

The trajectory for the vertical approach (landing) is computed using the bang-zero-bang algorithm [112], with different features (bisection method and check on the estimate quality) in the case of vertical oscillating target. Also in this case the acceleration computed through the algorithm is integrated twice to obtain the position set-point for the follower, starting from measurements of the motion monitoring and estimation module.

The detailed description of trajectory generation process is presented in Section 7.3.3 and Section 7.4.3.

7.3 Landing on a circular trajectory

In this Section the procedure for the landing with target moving on a circular trajectory is described. The Section is divided in four parts: in the first one the problem is presented in general terms, in the second one two approaches to obtain the in-plane synchronization are formulated, in the third one the algorithm for the landing is presented and in the last part experimental results are finally shown.

7.3.1 Problem description

The objective of this work is to perform autonomous landing of the follower on the target while both are flying, in particular while the target is moving along a circular trajectory with constant velocity. This is done in order to simulate the possibility of landing a small multirotor UAV on a fixed-wing drone that has greater endurance than the multirotor target and might be more suitable for outdoor applications. As fixed-wing UAVs are unable to hover, loiter at low velocity is the easiest condition so as to land a small drone on it; the problem with the target moving on a circular trajectory is the most similar to loiter and for this reason it is investigated.

The problem is clearly three-dimensional but, for the sake of simplicity, it is decoupled: the first part deals with synchronization of the trajectories of the two UAVs in the horizontal plane, while the second one concerns the descent of the follower drone on the target.

In-plane synchronization is the first safety constraint. Only when the follower is in the same in-plane position as the target, or in its neighbourhood, the landing manoeuvre can start. This constraint must be verified for the entire duration of the descent, otherwise the landing procedure is paused until the constraint is verified again. The safety constraints do not concern just in-plane synchronization. As a matter of fact, continuous feedback of the vertical relative position is used to design the real-time reference landing trajectory. The main reason of this approach is that the target may vary its altitude or experience vertical oscillations caused by the aerodynamic perturbations, originating from the follower's rotors wake.

Obviously the follower must be more reactive and faster than the target to perform synchronization, otherwise the former is not able to follow and to land on the latter.

In the problem, limitations on performance of the follower are also considered. In particular constraints on maximum horizontal and vertical velocities, together with maximum in-plane acceleration, are taken into account during the whole procedure.

The two parts of the landing procedure, namely in-plane synchronization and landing algorithm, are described in detail respectively in Section 7.3.2 and Section 7.3.3.

7.3.2 In-plane synchronization

The objective of the in-plane synchronization procedure is that the follower, that starts in a different position with respect to the target, must be able to reach and continuously be in the neighbourhood of the target in-plane position, expressed into North and East coordinates.

Defining the in-plane position error in NED frame as

$$H_{tf}(t) = \begin{bmatrix} n_t(t) \\ e_t(t) \end{bmatrix} - \begin{bmatrix} n_f(t) \\ e_f(t) \end{bmatrix} = \begin{bmatrix} n_r(t) \\ e_r(t) \end{bmatrix} = \begin{bmatrix} e_n(t) \\ e_e(t) \end{bmatrix}, \tag{7.3}$$

the safety objective can be expressed as

$$| H_{tf}(t) | = \sqrt{e_n^2(t) + e_e^2(t)} \leq \varepsilon_{ne}(d_r), \tag{7.4}$$

where d_r is the relative vertical distance computed as the difference between the target and the follower Down positions. In equation (7.4) ε_{ne} represents the horizontal tolerance on the in-plane position error; when the error is within this tolerance, landing over the target is considered safe. This horizontal tolerance is defined as a function of the relative vertical distance, in order to restrict the safety area while the UAVs are approaching, in this way:

$$\varepsilon_{ne}(d_r) = m_s d_r + q_s, \tag{7.5}$$

where the coefficients are selected to be $m_s = 0.067$ and $q_s = 0.2\,\text{m}$ in order to obtain a circular safety area with radius of $0.2\,\text{m}$ at zero vertical relative distance and with radius of $0.3\,\text{m}$ at $1.5\,\text{m}$ of relative vertical distance. In this way the follower can land as close as possible to the centre of gravity of the target so as not to create large moments on it. A truncated cone defines the safety area for landing as can be seen in Figure 7.3.

The problem of in-plane synchronization has been studied in two different conditions: collaborative and non-collaborative cases.

In the collaborative case it is supposed that the target drone is able to communicate its position and velocity to the follower, hence simplifying the synchronization that can be performed with a simple position control architecture in which the target position is directly used as set-point for the follower trajectory tracking control module.

In the non-collaborative case the follower does not know the absolute position of the target, but starting only from relative position and velocity measurements is able to compute the follower position set-point to obtain synchronization, through a feed-back control law.

Simple position controller

Assuming to know the exact absolute position of the target (collaborative case), the control law for synchronization uses the current North and East positions of the target as the position set-point for the tracking control

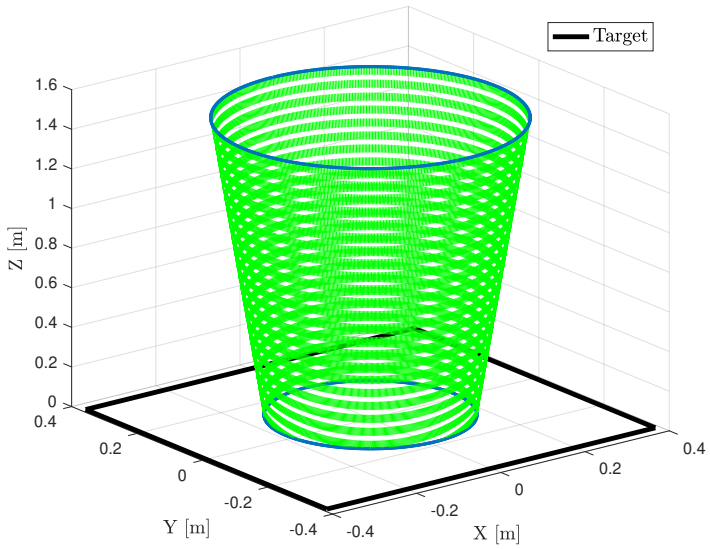


Figure 7.3: *Safety cone.*

module of the follower as outlined in Figure 7.4 and in equation (7.6) and equation (7.7):

$$n_f^o(t) = n_t(t), \tag{7.6}$$

$$e_f^o(t) = e_t(t). \tag{7.7}$$

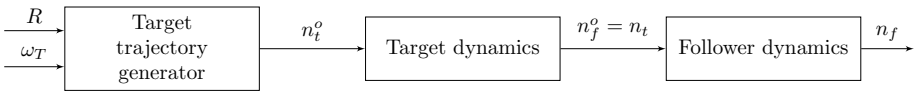


Figure 7.4: *Collaborative case control scheme.*

It is clear that in this way there is no control on the velocity and acceleration limits of the follower. In the worst cases, when the relative distance is large, the follower tracking control module receives an input that can cause dangerously large in-plane acceleration and velocity, obtained with high roll and pitch angles. In this case a low pass bandwidth

filter with time constant of 0.1 s on the set-point position is implemented so as to filter possible noise on target position measurements.

Augmented position controller

In the non-collaborative case the target does not provide the follower with information about its state, hence the follower has to estimate or measure the target motion, in particular all the information needed by the augmented position control architecture and by the landing algorithm, *i.e.*, relative position and velocity. This problem can be solved through a visual-based approach, *i.e.*, [113], [106] and [105]. Methods to obtain the measurements used as inputs in the algorithm are beyond the thesis objective. In this work Motion Capture (Mo-Cap) measurements are employed to get all quantities used in the algorithm. In particular, a centralized architecture is used, where a Ground Control Station (GCS) is able to communicate with all the UAVs. The Mo-Cap software running on the GCS measures the absolute position and velocity of all the UAVs; then, the relative position and velocity are computed and fed to the in-plane synchronization algorithm, which in turn sends the position set-point to the follower. The augmented position controller employed in the non-collaborative case consists in a feedback architecture ([113]) that uses only information about the relative in-plane position $H_{tf}(t)$, defined in equation (7.3) (in-plane position error), and the relative in-plane velocity in the NED frame, namely the in-plane velocity error, computed as:

$$\dot{h}_{tf}(t) = v_t(t) - v_f(t) = \begin{bmatrix} \dot{n}_t(t) \\ \dot{e}_t(t) \end{bmatrix} - \begin{bmatrix} \dot{n}_f(t) \\ \dot{e}_f(t) \end{bmatrix} = \begin{bmatrix} \dot{n}_r(t) \\ \dot{e}_r(t) \end{bmatrix} = \begin{bmatrix} \dot{e}_n(t) \\ \dot{e}_e(t) \end{bmatrix}. \quad (7.8)$$

The in-plane acceleration set-point a_f, directed along the line of sight between the two UAVs, is then computed with the following Proportional-Integral-Derivative control law:

$$a_f(t) = \begin{bmatrix} a_n(t) \\ a_e(t) \end{bmatrix} = k_p h_{tf}(t) + k_i \int_{t_0}^{t} h_{tf}(\tau) d\tau + k_d \dot{h}_{tf}(t). \quad (7.9)$$

Considering that the trajectory tracking control module requires a position set-point, the acceleration set-point must be integrated twice. Before

doing this, the limitations in follower performance must be defined. In this case saturations on the in-plane acceleration set-point and in-plane velocity of the follower can be taken into account. In particular when the in-plane acceleration set-point computed through equation (7.9) is greater than a maximum value (a_{max}), instead of using the acceleration value computed by equation (7.9), the acceleration set-point vector is scaled by a factor $1/a_{max}$ so as to reduce its magnitude while maintaining the same direction. When the in-plane velocity reaches the maximum value v_{max}, an input acceleration that decelerates the follower is used.

These switching conditions can be summarized as follows:

$$u_f(t) = \begin{bmatrix} u_n(t) \\ u_e(t) \end{bmatrix} = \begin{cases} a_f(t) = \begin{bmatrix} a_n(t) \\ a_e(t) \end{bmatrix}, \mid a_f(t) \mid \leq a_{max} \wedge \mid v_f(t) \mid < v_{max} \\ a_{max} \begin{bmatrix} \cos\gamma(t) \\ \sin\gamma(t) \end{bmatrix}, \mid a_f(t) \mid > a_{max} \wedge \mid v_f(t) \mid < v_{max} \\ -k_{brake} \begin{bmatrix} \dot{n}_f(t) \\ \dot{e}_f(t) \end{bmatrix}, \mid v_f(t) \mid \geq v_{max}, \end{cases} \tag{7.10}$$

where

$$\gamma(t) = \arctan \frac{a_e(t)}{a_n(t)} \tag{7.11}$$

and k_{brake} is the brake constant that defines the forces to be applied when the acceleration overcomes the limit.

In the equations (7.10), a_f is defined as the acceleration set-point computed through the PID law while u_f is the acceleration set-point actually implemented after saturation and control of the switching conditions in equation (7.10).

As already said the input saturated acceleration $u_f(t)$ is then integrated twice and the position set-point for the trajectory tracking control module

of the follower is obtained:

$$\begin{aligned}
\dot{n}_f(t) &= \dot{n}_f(t_0) + \int_{t_0}^{t} u_n(\tau) d\tau, \\
\dot{e}_f(t) &= \dot{e}_f(t_0) + \int_{t_0}^{t} u_e(\tau) d\tau, \\
n_f^o(t) &= n_f(t_0) + \int_{t_0}^{t} \dot{n}_f(\tau) d\tau, \\
e_f^o(t) &= e_f(t_0) + \int_{t_0}^{t} \dot{e}_f(\tau) d\tau.
\end{aligned} \tag{7.12}$$

In discrete time position and velocity are computed as:

$$\begin{aligned}
\dot{n}_f(k) &= \dot{n}_f(k-1) + u_n(k)\, t_{int}, \\
\dot{e}_f(k) &= \dot{e}_f(k-1) + u_e(k)\, t_{int}, \\
n_f^o(k) &= n_f(k-1) + \dot{n}_f(k)\, t_{int}, \\
e_f^o(k) &= e_f(k-1) + \dot{e}_f(k)\, t_{int}.
\end{aligned} \tag{7.13}$$

The integration initial conditions are the initial position and velocity of the follower when the chase begins, while t_{int} is the integration time step.

The position set-point just computed is sent to the follower at a specific rate, t_{sp}, namely set-point rate.

The augmented position control block diagram is shown in Figure 7.5 for the longitudinal direction, while the same applies to the lateral direction.

7.3.3 Landing algorithm

Once the in-plane synchronization is obtained, the landing algorithm can start. The objective of the landing algorithm is

$$\lim_{t \to t_f} \big(d_t(t) - d_f(t) \big) = \varepsilon_d. \tag{7.14}$$

The landing trajectory is computed starting from an acceleration command $u_d(t)$ in Down direction, integrated twice because UAVs have their own position controllers on-board which can receive a position set-point, and aim at minimizing position tracking errors.

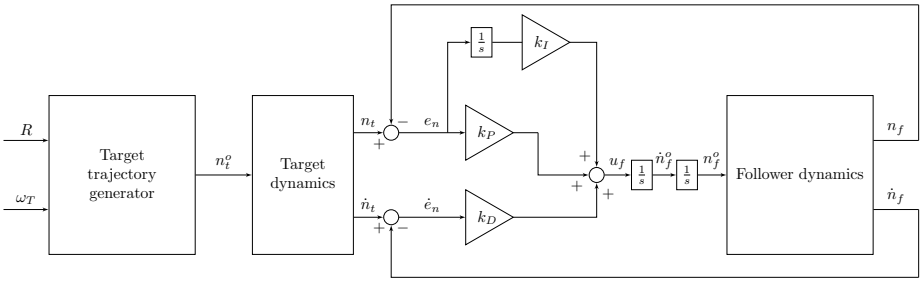


Figure 7.5: *Augmented position control block diagram for in-plane synchronization in longitudinal direction.*

To generate the acceleration command the three-states bang-zero-bang algorithm described in [108] is used. The problem reduces to the one of designing a control input to minimize the total time to land the follower on the target, under the following constraints:

- the absolute ([108]) input acceleration of the follower should be bounded by a given limit

$$a_{d_{min}} \leq u_d(t) \leq a_{d_{max}}, \tag{7.15}$$

- the absolute descending velocity ([108]) of the follower should also be bounded

$$\mid v_f(t) \mid \leq v_{d_{max}}. \tag{7.16}$$

The reference trajectory d^o_{ref}, like the whole procedure, is discretized and updated at sampling time t_{int}. Assuming as n_{stop} the number of iterations before the touchdown, the cost function is defined as in equation (7.17):

$$\min J = \min n_{stop}\, t_{int}. \tag{7.17}$$

Optimization constraints in discrete time are:

- initial and final state constraints: the initial states should be the same as the initial follower states and the final states should be the same as the final target states; in the following equations l is the sample

index at which the descent begins, calculated from the start of the entire algorithm (including the initial synchronization):

$$\begin{aligned} d^o_{ref}(l) &= d_f(l), \\ \dot{d}^o_{ref}(l) &= \dot{d}_f(l), \\ d^o_{ref}(l+n_{stop}) &= d_t(l+n_{stop}), \\ \dot{d}^o_{ref}(l+n_{stop}) &= \dot{d}_t(l+n_{stop}). \end{aligned} \tag{7.18}$$

- System kinematic constraints: the kinematic model of the double-integrator is:

$$\begin{aligned} &\dot{d}^o_{ref}(k+1) = \dot{d}^o_{ref}(k) + u_d(k)t_{int} \\ &d^o_{ref}(k+1) = d^o_{ref}(k) + \dot{d}^o_{ref}(k+1)t_{int} \quad \forall\, k = l,...,n_{stop}. \end{aligned} \tag{7.19}$$

- System input constraint: the acceleration input is bounded between the maximum downward and the maximum upward accelerations, respectively positive and negative according to the NED sign convention. These values are selected in order not to exceed the maximum reachable acceleration:

$$a_{d_{min}} \leq u_d(k) \leq a_{d_{max}} \quad \forall\, k = l,...,n_{stop}. \tag{7.20}$$

- Landing velocity constraints: we bound the follower Down velocity during landing:

$$|\, \dot{d}_f(k)\, | \leq v_{d_{max}} \quad \forall\, k = l,...,n_{stop}. \tag{7.21}$$

According to Pontryagin's principle [112], the optimal solution is a bang-bang type control law. The novelty is that, to avoid abrupt changes in the angular speed set-point of the rotors, a bang-zero-bang control law, involving three states instead of the classical two, is used. Starting from the initial conditions, the time τ needed to reach zero relative velocity, applying the maximum deceleration $a_{d_{min}}$ (upward acceleration), is computed solving the linear equation

$$\tau(k) = -\frac{\dot{d}_r(k)}{a_{d_{min}}}. \tag{7.22}$$

At this step there are two possibilities: in case $\tau(k)$ is negative, this means that the two UAVs are moving away; being the solution non-physical, $\tau(k)$ is set to zero. In case $\tau(k)$ is positive we proceed evaluating the relative distance at time $t+\tau$, where the shorthand notation $\hat{d}_r^{\tau}(k)$ will be used instead of $d_r(t(k)+\tau(k))$ that is defined as:

$$\hat{d}_r^{\tau}(k) = \hat{d}_t^{\tau}(k) - \hat{d}_f^{\tau}(k), \tag{7.23}$$

$$\hat{d}_t^{\tau}(k) = d_t(k), \tag{7.24}$$

$$\hat{d}_f^{\tau}(k) = d_f(k) + \dot{d}_f(k)\tau + a_{d_{min}}\tau^2(k). \tag{7.25}$$

Based on the value of the relative distance after time $\tau(k)$, the control action is chosen:

- if the future relative position is less or equal to the threshold on relative distance ε_d, it means that the follower "hits" the target, so the maximum upward acceleration (hence deceleration) $a_{d_{min}}$ must be applied to reduce the touch down velocity;
- if the future relative position is greater than ε_d there are two possible situations:
  - if the follower descent velocity limit $v_{d_{max}}$ has already been reached then zero acceleration is applied, in order to maintain the maximum descent velocity that minimizes the total landing time,
  - else the maximum downward acceleration must be applied to increase as much as possible the descent velocity and reduce the time to land.

It must be noticed that if the computed $\tau(k)$ is negative it will be bounded to zero. The procedure is summarized in the following equation (7.26):

$$u_d(k) = \begin{cases} a_{d_{min}} & \text{for } \hat{d}_r^{\tau}(k) \leq \varepsilon_d \\ 0 & \text{for } \hat{d}_r^{\tau}(k) > \varepsilon_d \text{ and } \mid \dot{d}_f(k) \mid \geq v_{d_{max}} \\ a_{d_{max}} & \text{for } \hat{d}_r^{\tau}(k) > \varepsilon_d \text{ and } \mid \dot{d}_f(k) \mid < v_{d_{max}}. \end{cases} \tag{7.26}$$

7.3.4 Experimental results

In this section the obtained experimental results are presented. In particular, the UAVs adopted for the tests are the ANT-R (follower) and the

CARRIER-1 (target) which are presented in Appendix B while the experimental facility and set-up are described in Appendix C. The feedback frequency has been lowered in order to emulate the measurements as provided from sensors as cameras or GPS in order to exploit the indoor tests considering future outdoor scenarios (see [114]). The thesis [114] describes also the possible degradation of the feedback measurements in terms of accuracy with the use of exogenous noise.

In-plane landing

The landing with target moving in the plane has been performed with the parameters reported in Table 7.1, this corresponds to a platform moving along a circular trajectory at $15\,\mathrm{cm\,s^{-1}}$. In particular the simple position controller and two sets of gains (obtained with heuristic approach) for the in-plane acceleration augmented position controller of the follower are tested.

Parameter	Measurement unit	Value
Integration time step T_{int}	$[s]$	0.02
a_{max}	$[m/s^2]$	1
v_{max}	$[m/s]$	2
$v_{d_{max}}$	$[m/s]$	0.3
$a_{d_{min}}$	$[m/s^2]$	-0.1
$a_{d_{max}}$	$[m/s^2]$	0.1
ε_d	$[m]$	0.02
Target initial position	$[m]$	$[1.5\ -2\ -1.3]^\top$
Follower initial position	$[m]$	$[0\ -3.5\ -2.8]^\top$
Centre of the circular trajectory	$[m]$	$[0\ -2\ -1.3]^\top$
Radius of the circular trajectory R	$[m]$	1.5
Pulsation for the circular trajectory ω_t	$[rad/s]$	0.1

Table 7.1: *Parameters for in-plane landing experiments.*

In paragraph 7.3.4 the results of the simple controller are shown while in paragraph 7.3.4 the performance of two sets of gains used for the augmented position controller is evaluated. Finally a comparison between the simple and augmented controllers under the same experimental conditions is done.

Collaborative case: simple position controller In Figure 7.6 and Figure 7.7 the in-plane positions and velocities of the target and of the follower are shown as example of synchronization using the simple position controller; in fact, in these two figures it can be noted that the follower position set-point almost coincides with the target position.

Figure 7.8 shows the Down position and velocity of the landing procedure, where the descent trajectory is computed with the bang-zero-bang algorithm (Figure 7.9).

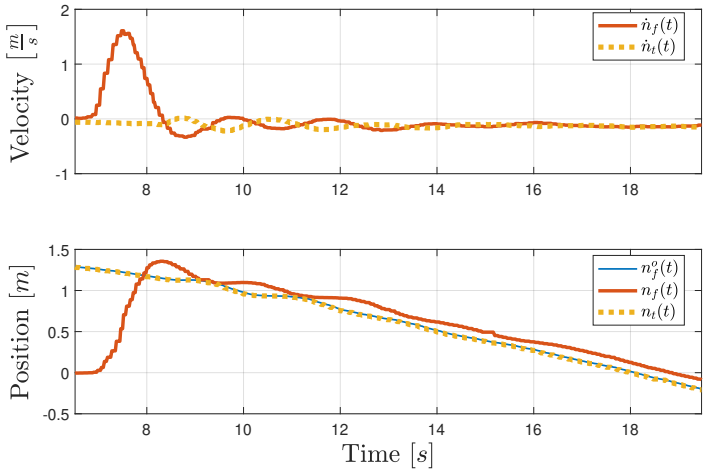


Figure 7.6: *North position and velocity from the beginning of the synchronization with simple position controller.*

Non-collaborative case: augmented position controller The gains in Table 7.2 have been tuned so as to get the largest achievable bandwidth (which is fundamental to get good tracking performance) without impairing stability of the augmented closed-loop system (*i.e.*, avoiding too many oscillations), and resulting in a low phase margin. Moreover, it has been decided to keep the integral gain to zero, since its contribution has been considered not fundamental to trajectory tracking, and it would instead further slow down the response. The first two sets of gains in Table 7.2 are the ones used for the augmented position controller. In Figure 7.10 and Figure 7.11 the in-plane positions and velocities of the target and of the follower

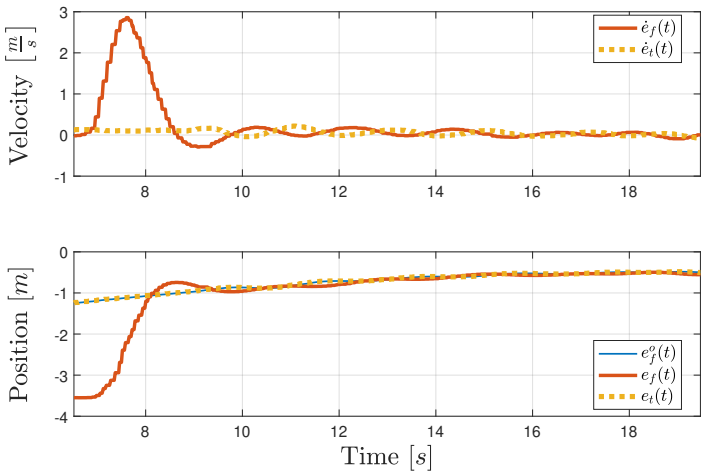


Figure 7.7: *East position and velocity from the beginning of the synchronization with simple position controller.*

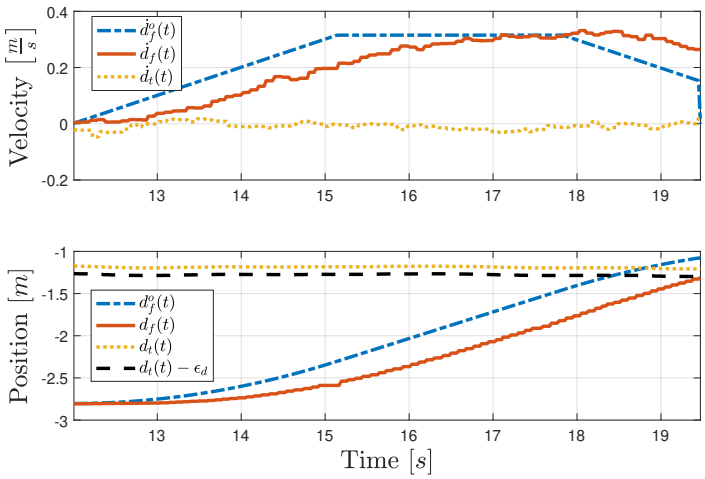


Figure 7.8: *Down position and velocity during landing with simple position controller.*

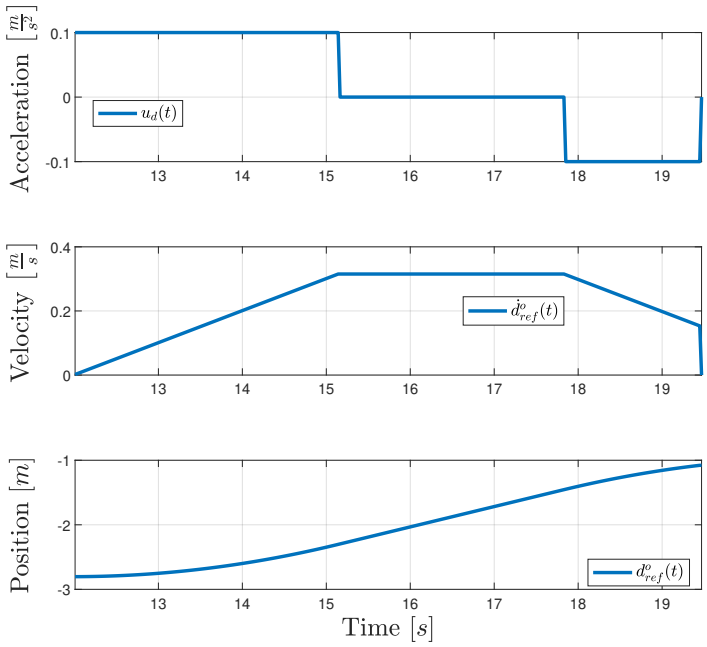


Figure 7.9: *Landing trajectory in the test with simple position controller.*

Gain-set	k_p	k_i	k_d	Phase margin $[deg]$	Cut-off frequency $[rad/s]$
A	0.3	0	1	30.4	1.09
B	0.4	0	1	24.8	1.12
C	0.5	0	1	19.4	1.15
D	0.5	0	0.9	19.4	1.07

Table 7.2: *PID gains for the follower longitudinal acceleration controller.*

are shown as an example. In Figure 7.12 are shown the Down position and velocity of the landing procedure, with the descent trajectory computed with the bang-zero-bang algorithm (Figure 7.13). It can be noted that around 13 s the algorithm is paused because the synchronization constraint is not satisfied. The same gain values are also successfully tested with target moving on the circular trajectory with constant velocity of $30\,\mathrm{cm\,s^{-1}}$ ($\omega_t = 0.2\,\mathrm{rad\,s^{-1}}$) but with longer time to land and larger horizontal final error values (in one case the follower landed on the safety cone limit). Results are summarized in Table 7.3.

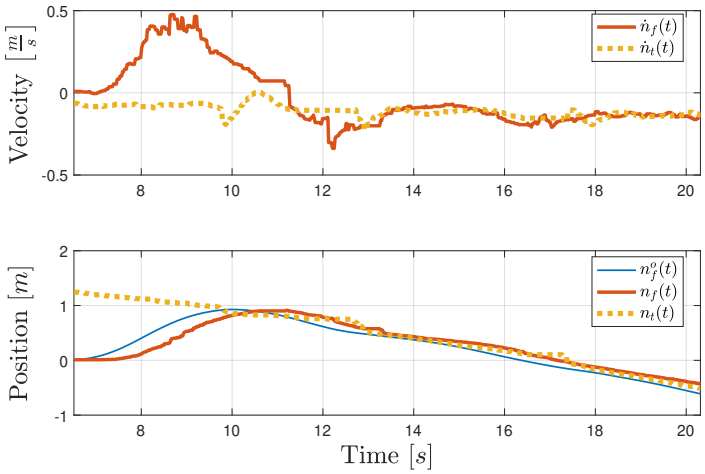


Figure 7.10: *North position and velocity from the beginning of the synchronization with the augmented position controller gain-set A (Table 7.2).*

Comparison of the position controllers A comparison between the simple and augmented in-plane position controllers can be performed in terms of horizontal error from the beginning of the synchronization to the touch down (Figure 7.14): the simple position controller is the faster one but has an error at steady state greater than the augmented position controller, the augmented position controller with the lowest value of the proportional gain (Table 7.2 Gain-set A) is the slowest one but it is also the most accurate in the low frequency case. It must be noted that the position controller with the largest value of the target trajectory frequency ($\omega_t = 0.2\,\mathrm{rad\,s^{-1}}$)

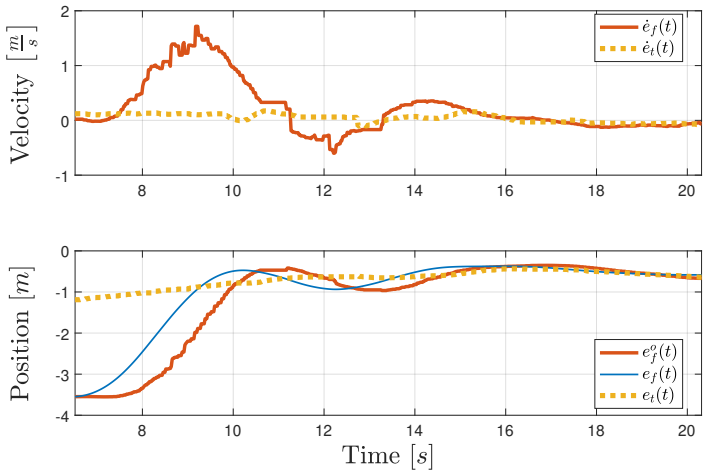


Figure 7.11: *East position and velocity from the beginning of the synchronization with the augmented position controller gain-set A (Table 7.2).*

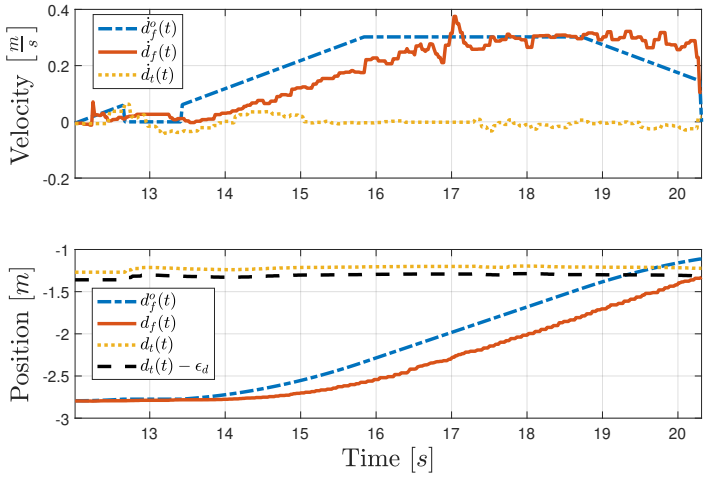


Figure 7.12: *Down position and velocity during landing with the augmented position controller gain-set A (Table 7.2).*

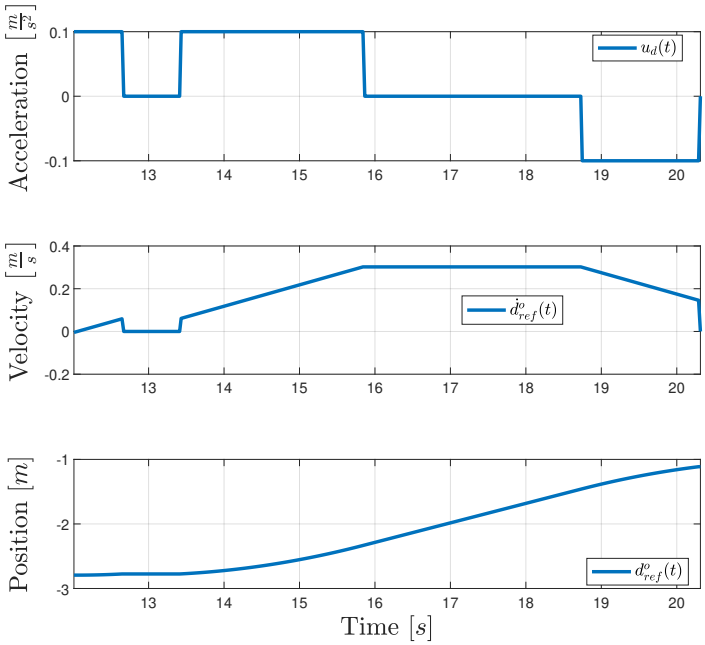


Figure 7.13: *Landing trajectory in the test with the augmented position controller gain-set A (Table 7.2).*

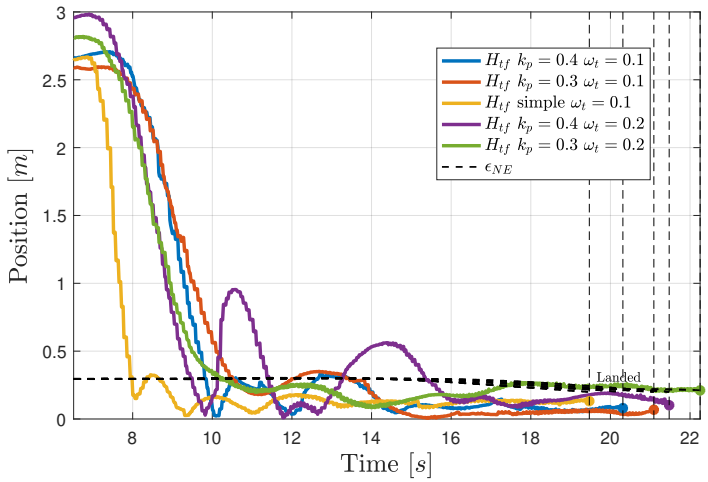


Figure 7.14: *Horizontal error comparison for the two controllers in different conditions (Table 7.2).*

Controller and gain-set	Final error $[m]$	$\dot{D}_r$ touch down $[m/s]$	Time to land $[s]$
Simple $\omega_t = 0.1$	0.1317	-0.2456	7.46
Augmented $\omega_t = 0.1$, A	0.0682	-0.2335	9.08
Augmented $\omega_t = 0.1$, B	0.0871	-0.2536	8.27
Augmented $\omega_t = 0.2$, A	0.1952	-0.2719	11.16
Augmented $\omega_t = 0.2$, B	0.1014	-0.3160	9.46

Table 7.3: *Landing performance comparison with gain-set of Table 7.2.*

has bad performance; in this case it would be better to use different values of the gain parameters. In Table 7.3 final horizontal distance, relative velocity at touch down and time to land for each test are reported.

7.4 Landing on an oscillating platform

In this Section the procedure for landing with the target oscillating vertically is described. The Section is divided in four parts: in the first one the problem is presented in general terms, in the second the motion estimation module necessary to perform the landing is described, in the third the landing algorithm is presented, highlighting the main differences with re-

spect to the one in the previous Section, and in the last part experimental results are finally shown.

7.4.1 Problem description

The problem of landing the follower on the target while the latter is oscillating vertically is quite different from the one in which it flies at constant altitude (*i.e.*, in-plane) and presents some critical issues.

The target is oscillating vertically as described by equation (7.27):

$$d_t(t) = h_t + \sum_{i=1}^{p} a_i \sin\left(2\pi f_i t + \beta_i\right), \tag{7.27}$$

where f_i is the *i-th* frequency, a_i is the i-th amplitude of oscillation, β_i is the i-th phase, p is the total number of sinusoidal components and h_t is the mean height around which the target is oscillating. In the tests conducted only one frequency component is considered in order to simplify the problem (*i.e.*, $p = 1$).

The problem, for the sake of simplicity, can be decoupled as in Section 7.3 in trajectory synchronization in the horizontal plane and vertical approach.

The objective of the synchronization in this problem is that the follower, that starts in a different position with respect to the target, must be able to reach and continuously be in the neighbourhood of the target North and East positions. Recalling the definition in equation (7.3), we want to verify that, after a transition time, the following relation is verified for all the time of vertical approach and landing:

$$h_{tf}(t) = \sqrt{e_n^2(t) + e_e^2(t)} \leq \varepsilon_{ne}(d_r), \tag{7.28}$$

where ε_{ne} has the same function of Section 7.3.2. The same safety cone of the Figure 7.3 is defined.

In this problem to keep the follower inside the safety area, *i.e.*, perform in-plane synchronization, the target in-plane position is given as set-point to the tracking control module of the follower, as described in Section 7.3.2. This has been done for the sake of simplicity, but the synchronization can be also performed with the augmented position controller described in Section 7.3.2.

Once the synchronization is obtained and maintained, the landing algorithm can start. The bang-zero-bang algorithm described in Section 7.3.3 is used with some modifications with respect to the in-plane problem because in this case the target is moving vertically. A motion estimation module is needed so as to apply the algorithm, as described later in Section 7.4.3.

7.4.2 Motion estimation module

In the sinusoidal landing case the trajectory generation module needs also a prediction of the target motion. This is obtained estimating on-line the amplitude and the phase of the oscillation and the height around which the target oscillates. To do this the Recursive Least Squares (RLS) algorithm is used ([110]) under the hypothesis to know the frequency of the oscillation *a priori*. Supposing to be in a partially collaborative case in which only the target position is known, the signal is represented exactly by the target position and is decomposed in a Fourier basis as in equation (7.29):

$$y(k) = [g(k)\cos(2\pi f t_s k) + h(k)\sin(2\pi f t_s k)] + z_0(k), \tag{7.29}$$

where t_s is the sampling period and for the frequency component f_i the amplitude is $a(k) = \sqrt{g(k)^2 + h(k)^2}$ and the phase is $\beta(k) = \arctan(g(k)/h(k))$. Some quantities must be defined at discrete time k: $y(k)$ is the signal that needs to be estimated, $\theta(k)$ is the vector of the real parameters containing the quantities that have to be estimated, the estimated parameter vector $\hat{\theta}(k)$, contains all the estimated parameters, *i.e.*, the coefficients of the Fourier decomposition and the mean height, and $\Phi(k)$ is the regression vector (equations (7.30)).

$$\begin{cases} \hat{\theta}(k) & = [\hat{g}(k)\ \hat{h}(k)\ \hat{z_0}(k)]^\top, \\ \Phi(k) & = [\cos(2\pi f t_s k)\ \sin(2\pi f t_s k)\ 1]^\top \end{cases} \tag{7.30}$$

Starting from these quantities the signal estimate can be computed through

$$\hat{y}(k) = \hat{\theta}^\top(k)\Phi(k). \tag{7.31}$$

and the error is the difference between real and estimated signal $\varepsilon_k = y(k) - \hat{y}(k)$.
The recursive least squares algorithm is implemented at the same rate of the set-point integration used in the landing algorithm, *i.e.*,$t_s = t_{int}$, in this way:

$$\begin{cases} \varepsilon_k &= y_k - \hat{\theta}_{k-1}^t \Phi_k, \\ g_k &= \frac{1}{\lambda_0}\left(g_{k-1} - \frac{g_{k-1}\Phi_k^\top \Phi_k g_{k-1}}{\lambda_0 + \Phi_k^\top g_{k-1}\Phi_k}\right), \\ \hat{\theta}_k &= \hat{\theta}_{k-1} + g_k \Phi_k \varepsilon_k, \end{cases} \tag{7.32}$$

where the subscripts are used as a shorthand notation for $(\cdot)(k)$ and indicate the step, ε_k is the a priori error, g_k is the adaptation gain and λ_0 is the exponential forgetting factor, typically chosen between 0.98 and 0.995 (0.98 in our case).

With the estimation procedure, starting from an initial guess, the amplitudes $\hat{A}$, the phases $\hat{\beta}$ and the mean height $\hat{z_0}$ are obtained at each time step, so that future values of the position and velocity can be computed starting from actual parameter values using equation (7.33) and equation (7.34):

$$\hat{d}_t^{\Delta t}(k) = \hat{z_0}(k) + \hat{A}(k)\sin\left[\omega(kt_s + \Delta t) + \hat{\beta}(k)\right], \tag{7.33}$$

$$\dot{\hat{d}}_t^{\Delta t}(k) = \hat{a}(k)\omega\cos\left[\omega(kt_s + \Delta t) + \hat{\beta}(k)\right], \tag{7.34}$$

where $\omega = 2\pi f$, and the velocity is obtained by analytical differentiation of the estimated position. The notation $\hat{d}_t^{\Delta t}(k)$ means that the vertical position of the target is predicted on a time horizon Δt based on the parameter estimate at time k.

The accuracy of the RLS method is strongly dependent on the initial guess.

7.4.3 Landing algorithm

In this section, the landing algorithm for the case of oscillating target is described. Once the follower is in the safety cone, the vertical descent can start. Also in the sinusoidal case the landing trajectory is computed

starting from an acceleration command $u_d(t)$ that, integrated twice, yields a position set-point used by the trajectory tracking module on-board the follower.

The acceleration command is generated through the three-states bang-zero-bang algorithm [108].

Starting from initial conditions, the maximum deceleration $a_{d_{min}}$ is applied (upward acceleration), the time instant $\tau(k)$ at which the estimated relative velocity becomes zero must be identified solving an equation that in this case is no longer linear in the unknown τ, as equation (7.22), and that requires the estimation of the future motion of the target obtained with RLS described in the previous Section 7.4.2:

$$\hat{d}_r^{\tau}(k) = \hat{d}_t^{\tau}(k) - \hat{d}_f^{\tau}(k) = 0, \tag{7.35}$$

with:

$$\hat{d}_f^{\tau}(k) = \dot{d}_f(k) + a_{d_{min}}\tau(k), \tag{7.36}$$

$$\hat{d}_t^{\tau}(k) = \hat{a}(k)\omega\cos\left[\omega(kt_s + \tau(k)) + \hat{\beta}(k)\right], \tag{7.37}$$

where $\tau(k)$ is the unknown variable.

To solve equation (7.35) the bisection algorithm is applied because thanks to its simplicity it does not increase too much the computational time. The same two possibilities presented in Section 7.3 may occur: in the case $\tau(k)$ is negative, *i.e.*, the two UAVs are moving away, being the solution non-physical, $\tau(k)$ is set to zero, while in the case $\tau(k)$ is positive we proceed evaluating the relative distance at instant $t+\tau$, $\hat{d}_r(t(k)+\tau(k))$, using for the target the estimated parameters:

$$\hat{d}_r^{\tau}(k) = \hat{d}_t^{\tau}(k) - \hat{d}_f^{\tau}(k), \tag{7.38}$$

$$\hat{d}_t^{\tau}(k) = z_0(k) + \hat{a}(k)\sin\left[\omega(kt_s + \tau(k)) + \hat{\beta}(k)\right], \tag{7.39}$$

$$\hat{d}_f^{\tau}(k) = d_f(k) + \dot{d}_f(k)\tau(k) + \frac{1}{2}a_{d_{min}}\tau^2(k). \tag{7.40}$$

Based on the value of $\hat{d}_r^{\tau}(k)$, the control action is chosen in the same way

Test	$\hat{g}(0)$	$\hat{h}(0)$	$\hat{z_0}(0)$
A	0	0.1	-1.3
B	0	0	-1.3

Table 7.4: *Initial guess* $\hat{\theta}(0)$ *for the estimation.*

described in Section 7.3. The procedure is summarized in this equation:

$$u_d(k) = \begin{cases} a_{d_{min}} & \text{for } \hat{d}_r^{\tau}(k) \leq \varepsilon_d \\ 0 & \text{for } \hat{d}_r^{\tau}(k) > \varepsilon_d \text{ and } \mid \dot{d}_f(k) \mid \geq v_{d_{max}} \\ a_{d_{max}} & \text{for } \hat{d}_r^{\tau}(k) > \varepsilon_d \text{ and } \mid \dot{d}_f(k) \mid < v_{d_{max}}. \end{cases} \tag{7.41}$$

7.4.4 Experimental results

In this section results of the landing on oscillating target experiments are shown. In the first part the experiments conducted to test the estimation module are presented, then the results for landing on oscillating target are shown.

Test for estimation module

First of all, the estimation module has been tested in two ways: in the first case post-processing flight data and in the second estimating in real time. During both experiments the target oscillates with frequency $\omega = 0.5\ rad/s$ and amplitude of oscillation $A = 0.1\ m$ for 60 s in undisturbed conditions, and then for the same period in disturbed conditions, *i.e.*, with the follower hovering over the target. The estimation tests have been performed with different values of the initial guess in order to evaluate the convergence of the RLS. Moreover, after a tuning procedure, the forgetting factor has been set to $\lambda_0 = 0.98$. In Table 7.4 two initial guesses are reported, the first consists in the exact values used to generate the set-point for target motion, while the second consists in incorrect values (except for the height that is supposed to be known).

In Figure 7.15 the position and velocity estimated by RLS with exact initial guess are compared with the measured data; it must be noted that actually they correspond to the parameter values used to generate the set-point target motion, but, when the follower is above the target, the motion

of the latter is quite different from the set-point because of aerodynamic disturbances, so also this initial guess is indeed "wrong" but it constitutes a good initial guess for the estimation.

In Figure 7.16 the results for wrong initial guess are shown; in this case the estimate needs more time to converge.

Figure 7.17 shows real and estimated target position and velocity; in the moment in which the follower arrives above the target, approximately at 70 s, the estimation gets worse but after a transient time it converges again. The error between estimated and real target position and velocity is shown in Figure 7.18. An increase is clearly visible in the disturbed case.

In Figure 7.19 the estimated parameters of the Fourier decomposition of the target motion are compared with the values that they would have in order to make equation (7.29) equal to equation (7.27). From the obtained results it follows that, for safety reasons, another check based on the estimation must be introduced in the landing algorithm: if the error ε_k, defined as the difference between measured and estimated velocity at the current instant, is greater than a chosen threshold, the landing procedure is paused and the follower maintains its position. In this way also in the case in which the follower is still, because of the poor quality of the estimate, at low relative distance from the target and the latter is moving upward, before it can touch the follower, the check on the vertical distance makes the follower disarmed and land without any problem.

Landing on oscillating target test

The landing algorithm with the oscillating target has been tested in the three cases reported in Table 7.5.

The parameters used are reported in Table 7.6. It must be noted that the

Test	Amplitude A $[m]$	Phase β $[deg]$	ω $[rad/s]$
LND A	0.1	0	0.5
LND B	0.2	0	0.5
LND C	0.1	0	0.7

Table 7.5: *Data for the sinusoidal target motion for each test.*

maximum descent velocity has been reduced with respect to the in-plane

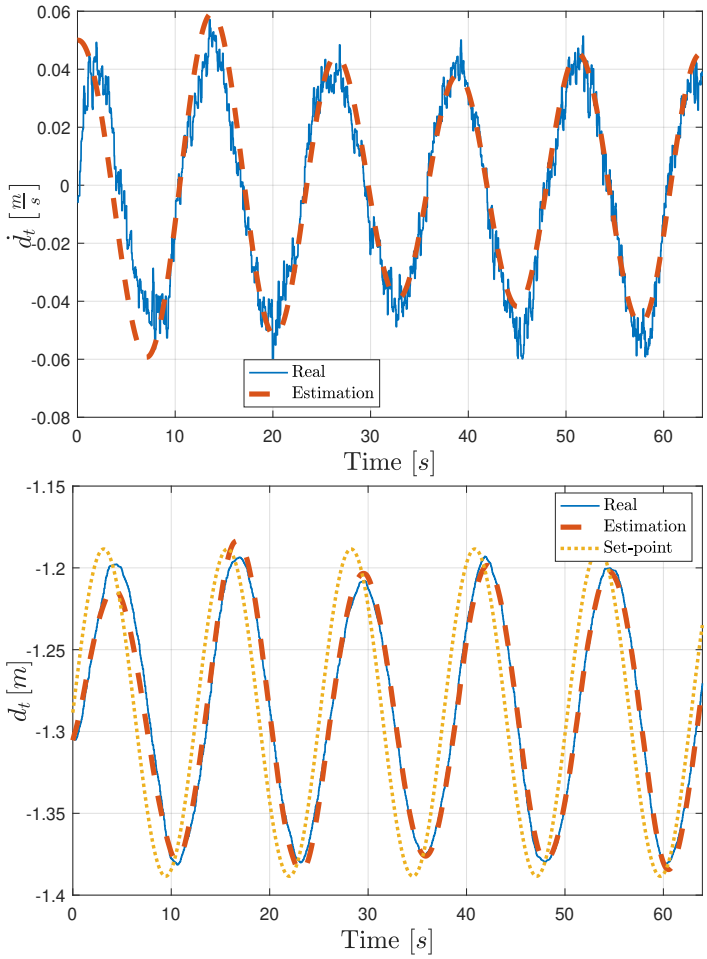


Figure 7.15: *Target velocity and position estimation with initial guess A ("exact") (Table 7.4).*

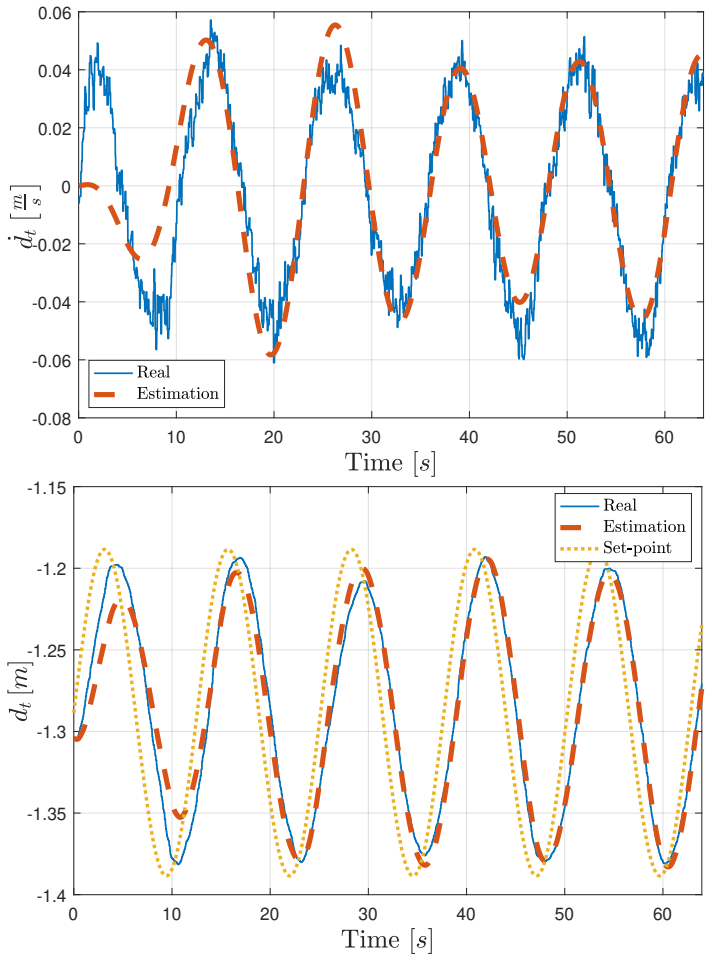


Figure 7.16: *Target velocity and position estimation with initial guess B ("wrong") (Table 7.4).*

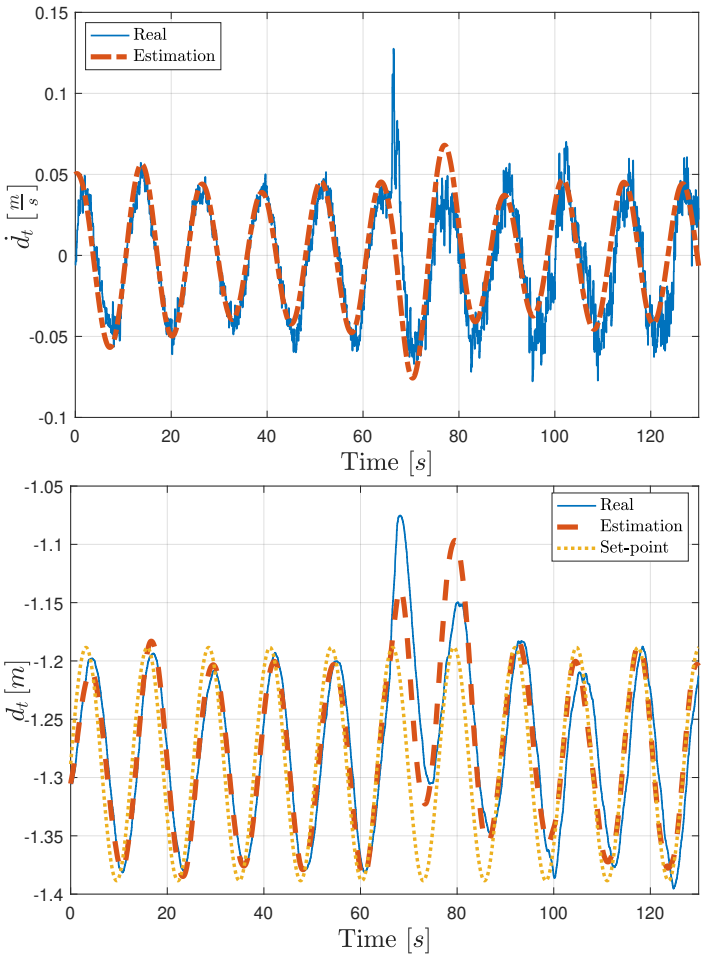


Figure 7.17: *Target position and velocity estimation in undisturbed and disturbed conditions with initial guess A (Table 7.4).*

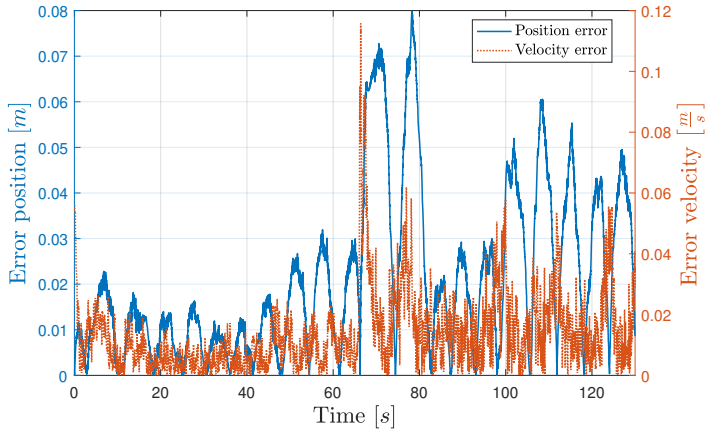


Figure 7.18: *Target position and velocity error for the estimate with initial guess A (Table 7.4).*

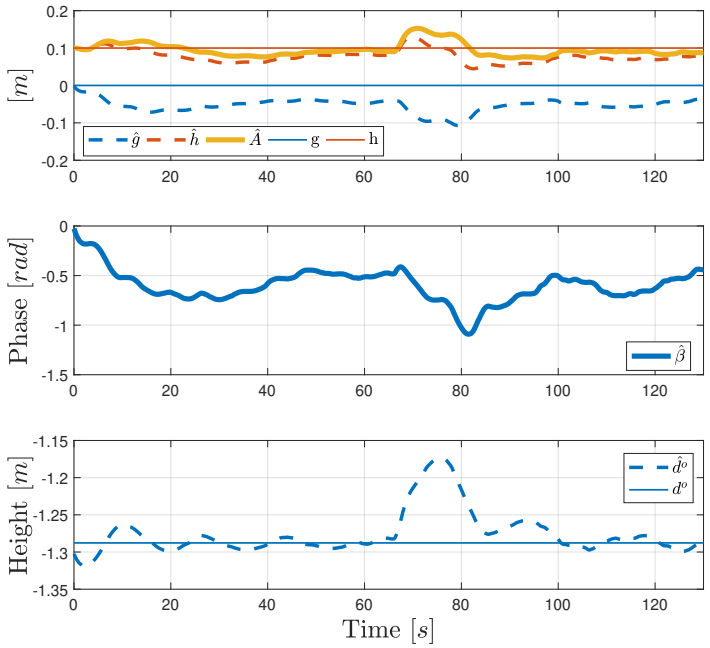


Figure 7.19: *Estimated parameters with initial guess A (Table 7.4).*

experiments; this has been done so as to reduce the relative landing velocity (touch down) that might be higher in this case because the target is moving vertically. In the LND A and LND B cases (Table 7.5) the land-

Parameter	Measurement unit	Value
Integration time step t_{int}	$[s]$	0.02
a_{max}	$[m/s^2]$	1
v_{max}	$[m/s]$	2
$v_{d_{max}}$	$[m/s]$	0.2
$a_{d_{min}}$	$[m/s^2]$	-0.1
$a_{d_{max}}$	$[m/s^2]$	0.1
ε_d	$[m]$	0.02
Target initial position	$[m]$	$[0\ -2\ -1.3]^T$
Follower initial position	$[m]$	$[0\ -3.5\ -2.8]^T$
Threshold on velocity estimation error	$[m/s]$	0.06

Table 7.6: *Parameters for sinusoidal landing experiments.*

ing is performed with success, the estimation of position and velocity are good and shown for the LND A case in Figure 7.20. The LND C case, at higher oscillating frequency, shows a particular behaviour. This because the safety check on the quality of the velocity estimate is activated; in fact when the error on the velocity estimate exceeds the threshold (chosen empirically looking at simulation, Section 7.4.4) the landing is paused. This can be seen in Figure 7.21 where the landing trajectory computed by the bang-zero-bang algorithm presents an interval about 18 *s* in which the position set-point is constant.

The trajectory results in the time histories in Figure 7.22 and Figure 7.23.

The position and velocity estimates are shown in Figure 7.24 and it can be seen that the quality of the estimate is not satisfactory.

A comparison among the three cases is shown in Table 7.7. In terms of horizontal position error all cases presents similar values and similar behaviour (Figure 7.25); the main differences are in the touch down velocity and in the time to land. The velocity has a strong dependency on the time at which the touch down occurs because it depends on the target motion. The longest time to land of the last case is due to the low quality estimate: in fact in Figure 7.26 it can be seen that case LND C is the one

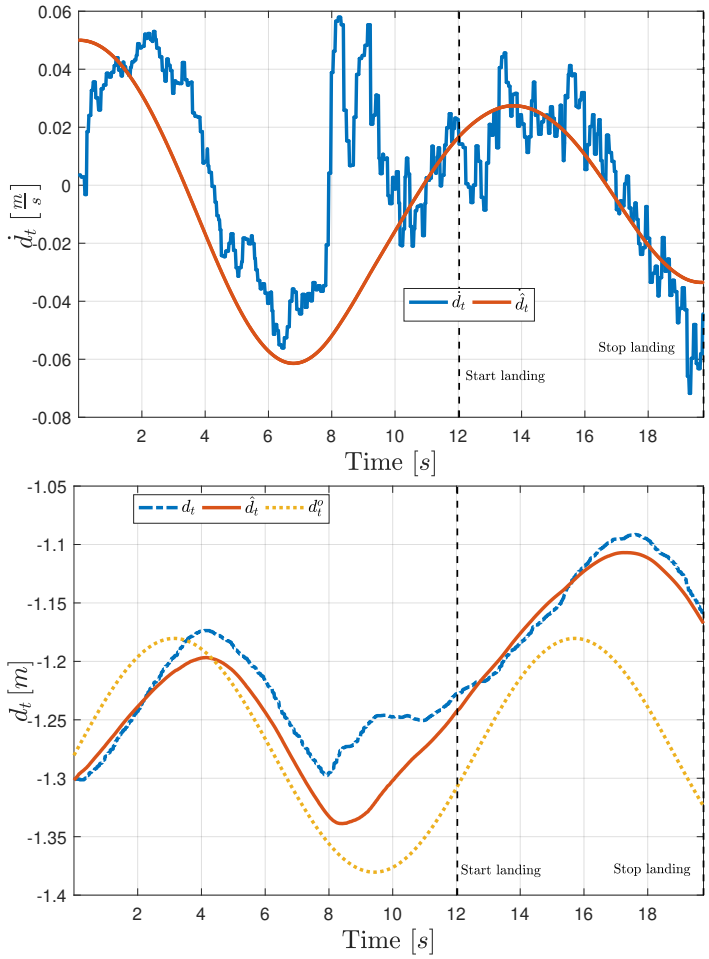


Figure 7.20: *Velocity and position estimation of the target motion in LND A case (Table 7.5).*

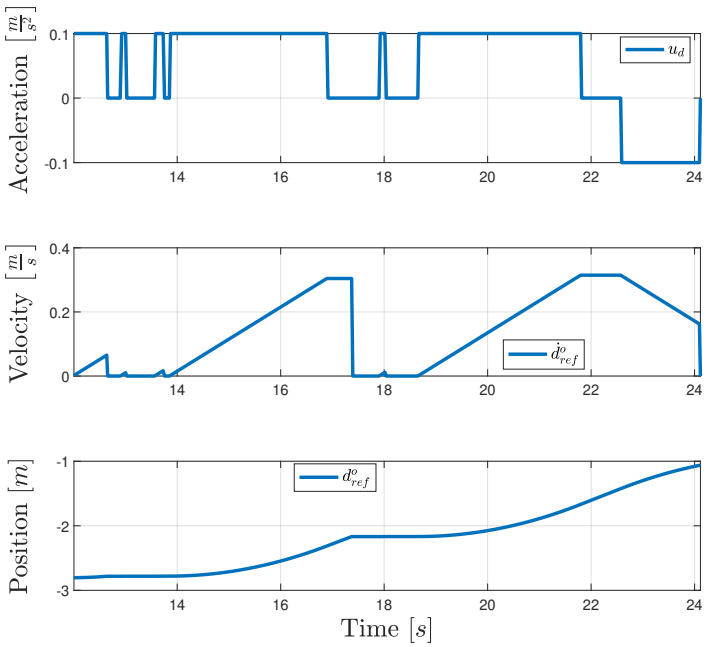


Figure 7.21: *Landing trajectory computed by the algorithm in LND C case (Table 7.5).*

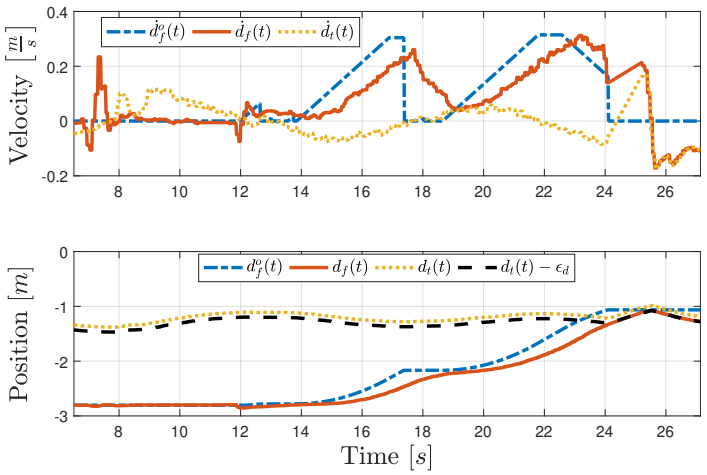


Figure 7.22: *Down velocity and position for the entire duration of the experiment LND C (Table 7.5).*

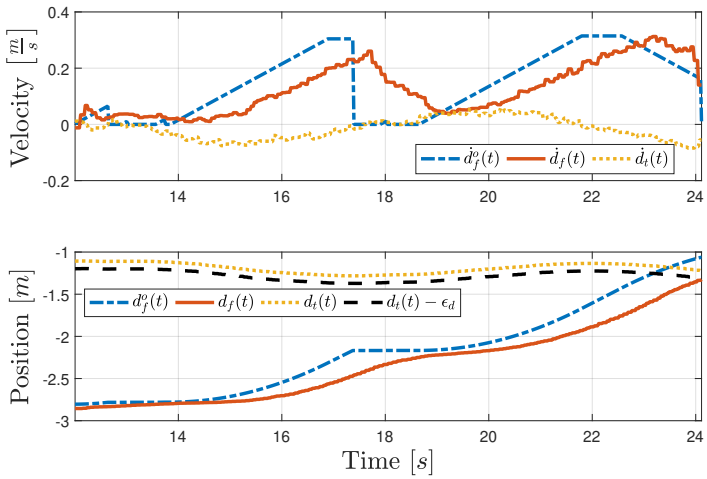


Figure 7.23: *Down velocity and position during landing phase of experiment LND C (Table 7.5).*

that exceeds the threshold also after the transient phase in which the target loses height because of the presence of the follower on it.

Test	Final error position $[m]$	$\dot{d}_r$ touch down $[m/s]$	Time to land $[s]$
LND A	0.0480	-0.2733	7.7230
LND B	0.0276	-0.2424	8.0956
LND C	0.0488	-0.1953	12.1084

Table 7.7: *Results for the landing with target sinusoidal motion for each test.*

7.5 Final considerations

In this chapter, the air-to-air autonomous landing manoeuvre for multirotor UAVs has been studied. The problem of the interaction between more aircraft during flight is of great interest in UAV operations, such as search and rescue, surveillance and air-to-air automatic refuelling (AAAR) and landing over the LHD of moving ships.

In the case of target moving in-plane, two different conditions have been investigated: the first one in a collaborative case in which the exact

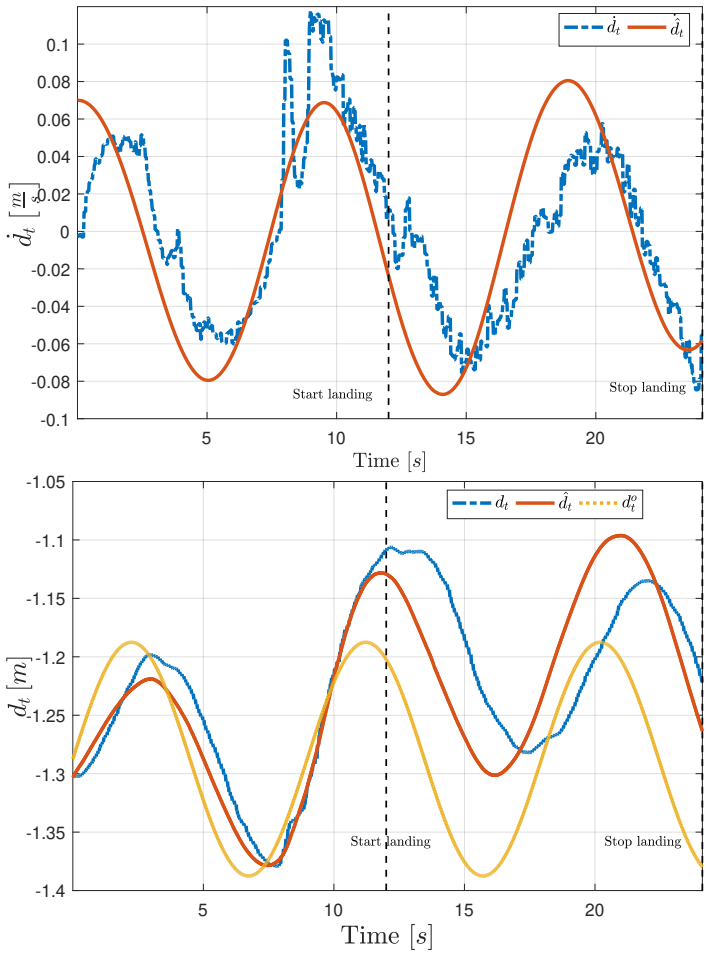


Figure 7.24: *Velocity and position estimation of the target motion in LND C case (Table 7.5).*

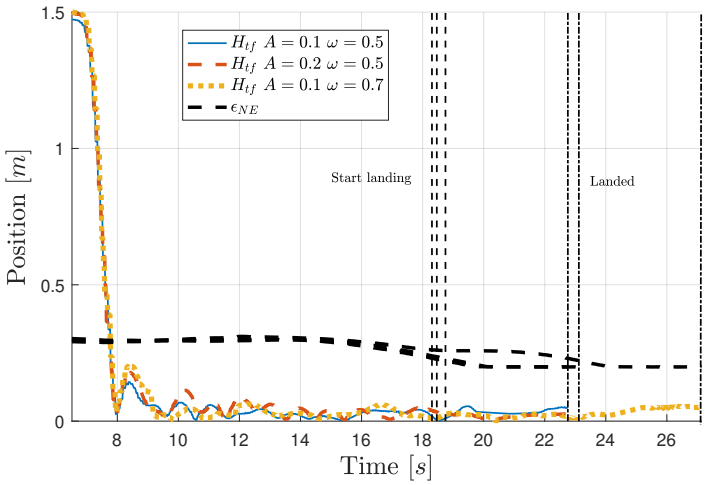


Figure 7.25: *Time history of the horizontal position error during all the sinusoidal landings.*

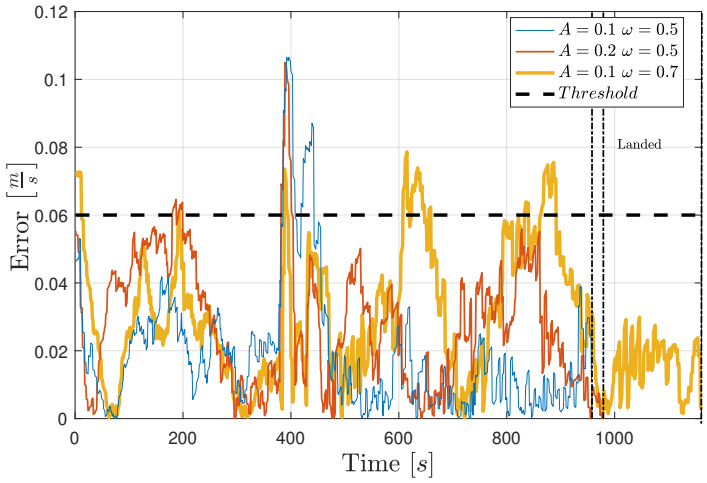


Figure 7.26: *Error in the velocity estimates in the three sinusoidal landing conditions analysed.*

target state is known by the follower, the second one in a non-collaborative case, in which the follower has to track the target based only on relative position and velocity. The problem of landing a multirotor UAV over another one involves a number of non-trivial problems to be solved: because of aerodynamic interactions between the two drones, *e.g.*, the target is disturbed by the wake of the follower that flies above it; in sinusoidal landing an estimation of the target oscillatory motion in terms of position and velocity has been required. Finally experimental activity has been conducted to validate the proposed algorithm: the landing algorithms with target moving in-plane and oscillating vertically were successfully tested.

Conclusions

The main results and contributions of the dissertation will be described hereafter.

This dissertation focuses on different aspects of the multirotor aerial vehicles. The first result is the improvement of the multirotor modelling both for conventional and non-conventional platforms. The described modelling exploits two different approaches: the control oriented and the simulation-oriented ones. In particular, the simulation-oriented modelling takes into account different effects which are usually neglected on the control design, as the dynamical aerodynamic acting on the propeller. Moreover, a particular attention is also given to the ground effect and it has been presented in the dissertation an extensive campaign aimed to identify the model of the ground effect for the multirotor platforms.

The second main contribution is related to the development of two different methods aimed to the design of multirotor UAVs. The first presented method is the Inverse Design, which can provide an estimation of the performance capabilities given the multirotor's components, while the second one (Forward Design) can provide a component's parameters list given the mission requirements.

Furthermore, since an important element in the design process of the multirotor platforms is the control architecture, in this thesis some inno-

vative results in the advanced control architectures for multirotor UAVs have been presented. Such control architectures have been specifically designed for conventional multirotors (*i.e.*, quadrotors) and for thrust vectoring multirotors (*i.e.*, tiltrotors). For what concern the conventional quadrotors, the presented architectures range from the classical hierarchical position and attitude control to the more innovative inversion-based methods. In addition, to overcome the limitations of linear controllers, an adaptive augmentation of the attitude control system for multirotor UAVs have been presented in order to better face the external disturbances and model uncertainties. The other control techniques presented in the thesis have been specifically designed for thrust vectoring multirotors. This platforms being able to deliver both force and torque in any direction are very attractive for the aerial manipulation tasks. For this reason two full-pose control algorithms have been presented and tested in this thesis.

Finally, the air-to-air automatic landing has been presented. Such innovative application results interesting when surveillance, reconnaissance and search-and-rescue missions are considered since the small-scale UAVs suffer of low endurance. In particular, in this thesis the guidance laws aimed at providing a reference descend trajectory for the small multirotor landing on the bigger one are presented. The proposed guidance laws separate the problem of vertical landing from the horizontal synchronisation of the two multirotor UAVs and they can provide a descend trajectory even in the case of a moving target (horizontally and vertically).

Bibliography

[1] E. S. Latorre, "Propulsion system optimization for an unmanned lightweight quadrotor," Master's thesis, Univ. Politec. de Catalunya, Master in Aerospace Science and Technology, 2011.

[2] D. Bershadsky, S. Haviland, and E. N. Johnson, "Electric multirotor UAV propulsion system sizing for performance prediction and design optimization," in *57th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, 2016.

[3] O. Gur and A. Rosen, "Optimizing electric propulsion systems for UAVs," in *12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, 2008.

[4] C. Ampatis and E. Papadopoulos, "Parametric design and optimization of multi-rotor aerial vehicles," in *Applications of Mathematics and Informatics in Science and Engineering*, Springer, 2014.

[5] U. Saetti, S. Lakhmani, T. Berger, J. F. Horn, and C. Lagoa, "Design of Dynamic Inversion and Explicit Model Following control laws for quadrotor inner and outer loops," *AHS Forum 74, At Phoenix, AZ*, 2018.

[6] M. Ryll, D. Bicego, and A. Franchi, "Modeling and control of FAST-Hex: a fully-actuated by synchronized-tilting hexarotor," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2016.

[7] B. L. Stevens, F. L. Lewis, and E. N. Johnson, *Aircraft control and simulation: dynamics, controls design, and autonomous systems*. John Wiley & Sons, 2015.

[8] M. E. Dreier, *Introduction to helicopter and tiltrotor flight simulation*. American Institute of Aeronautics and Astronautics, 2007.

[9] W. R. Hamilton, "On quaternions; or on a new system of imaginaries in algebra," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 1844.

[10] F. L. Markley, "Attitude error representations for kalman filtering," *Journal of guidance, control, and dynamics*, 2003.

[11] F. L. Markley, "Unit quaternion from rotation matrix," *Journal of guidance, control, and dynamics*, 2008.

[12] M. Ryll, H. H. Bülthoff, and P. R. Giordano, "A novel overactuated quadrotor unmanned aerial vehicle: Modeling, control, and experimental validation," *IEEE Transactions on Control Systems Technology*, 2014.

[13] S. Rajappa, M. Ryll, H. H. Bülthoff, and A. Franchi, "Modeling, control and design optimization for a fully-actuated hexarotor aerial vehicle with tilted propellers," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2015.

[14] D. Brescianini and R. D'Andrea, "Design, modeling and control of an omni-directional aerial vehicle," in *2016 IEEE international conference on robotics and automation (ICRA)*, IEEE, 2016.

[15] G. Michieletto, M. Ryll, and A. Franchi, "Fundamental actuation properties of multirotors: Force-moment decoupling and fail-safe robustness," *IEEE Transactions on Robotics*, 2018.

[16] B. Theys, G. Dimitriadis, P. Hendrick, and J. D. Schutter, "Influence of propeller configuration on propulsion system efficiency of multi-rotor unmanned aerial vehicles," in *2016 international conference on unmanned aircraft systems (ICUAS)*, IEEE, 2016.

[17] A. Oosedo, S. Abiko, S. Narasaki, A. Kuno, A. Konno, and M. Uchiyama, "Flight control systems of a quad tilt rotor unmanned aerial vehicle for a large attitude change," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2015.

[18] D. Invernizzi, M. Giurato, P. Gattazzo, and M. Lovera, "Full pose tracking for a tilt-arm quadrotor UAV," in *2018 IEEE Conference on Control Technology and Applications (CCTA)*, IEEE, 2018.

[19] M. Kamel, S. Verling, O. Elkhatib, C. Sprecher, P. Wulkop, Z. Taylor, R. Siegwart, and I. Gilitschenski, "Voliro: An omnidirectional hexacopter with tiltable rotors," *arXiv preprint arXiv:1801.04581*, 2018.

[20] D. Invernizzi and M. Lovera, "Geometric tracking control of a quadcopter tiltrotor UAV," *IFAC-PapersOnLine*, 2017.

[21] D. M. Sima, *Regularization techniques in model fitting and parameter estimation (regularisatietechnieken in modellering en parameterschatting)*. PhD thesis, TU Delft, 2006.

[22] W. Johnson, *Rotorcraft aeromechanics*. Cambridge University Press, 2013.

[23] G. J. Leishman, *Principles of helicopter aerodynamics*. Cambridge university press, 2006.

[24] A. R. S. Bramwell, D. Balmford, and G. Done, *Bramwell's helicopter dynamics*. Elsevier, 2001.

[25] D. M. Pitt and D. A. Peters, "Theoretical prediction of dynamic-inflow derivatives," *9th European Rotorcraft Forum*, 1980.

[26] D. A. Peters and N. HaQuang, "Dynamic inflow for practical applications," *American Helicopter Society*, 1988.

[27] R. Gill and R. D'Andrea, "Propeller thrust and drag in forward flight," in *2017 IEEE Conference on Control Technology and Applications (CCTA)*, 2017.

[28] M. Bangura, M. Melega, R. Naldi, and R. Mahony, "Aerodynamics of rotor blades for quadrotors," *arXiv preprint arXiv:1601.00733*, 2016.

[29] H. Huang, G. M. Hoffmann, S. L. Waslander, and C. J. Tomlin, "Aerodynamics and control of autonomous quadrotor helicopters in aggressive maneuvering," in *2009 IEEE international conference on robotics and automation*, 2009.

[30] J. Brandt and M. Selig, "Propeller performance data at low reynolds numbers," in *49th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, 2011.

[31] R. E. Sheldahl and P. C. Klimas, "Aerodynamic characteristics of seven symmetrical airfoil sections through 180-degree angle of attack for use in aerodynamic analysis of vertical axis wind turbines," tech. rep., Sandia National Labs., Albuquerque, NM (USA), 1981.

[32] "Airfoil tools." `http://www.airfoiltools.com/airfoil/details?airfoil=naca0015-il`. Accessed: 14-10-2019.

[33] G. D. Padfield, *Helicopter flight dynamics*. Wiley Online Library, 2008.

[34] S. Aich, C. Ahuja, T. Gupta, and P. Arulmozhivarman, "Analysis of ground effect on multi-rotors," in *2014 International Conference on Electronics, Communication and Computational Engineering (ICECCE)*, IEEE, 2014.

[35] M. Bangura and R. Mahony, "Nonlinear dynamic modeling for high performance control of a quadrotor," in *Australasian Conference on Robotics and Automation*, Australian Robotics and Automation Association, 2012.

[36] L. Danjun, Z. Yan, S. Zongying, and L. Geng, "Autonomous landing of quadrotor based on ground effect modelling," in *2015 34th Chinese Control Conference (CCC)*, IEEE, 2015.

[37] E. Davis, J. Spollard, and P. Pounds, "Passive height stability and trajectory repeatability of a quadrotor maneuvering in ground effect with regulated voltage bus," in *Australasian conference on robotics and automation (ACRA 2015)*, 2015.

[38] I. Sharf, M. Nahon, A. Harmat, W. Khan, M. Michini, N. Speal, M. Trentini, T. Tsadok, and T. Wang, "Ground effect experiments and model validation with Draganflyer X8 rotorcraft," in *2014 International Conference on Unmanned Aircraft Systems (ICUAS)*, IEEE, 2014.

[39] P. Sanchez-Cuevas, G. Heredia, and A. Ollero, "Characterization of the aerodynamic ground effect and its influence in multirotor control," *International Journal of Aerospace Engineering*, vol. 2017, pp. 1–17, 2017.

[40] X. Kan, J. Thomas, H. Teng, H. G. Tanner, V. Kumar, and K. Karydis, "Analysis of ground effect for small-scale UAVs in forward flight," *IEEE Robotics and Automation Letters*, 2019.

[41] X. He, G. Kou, M. Calaf, and K. K. Leang, "In-ground-effect modeling and nonlinear-dsturbance observer for multirotor unmanned aerial vehicle control," *Journal of Dynamic Systems, Measurement, and Control*, 2019.

[42] A. Matus-Vargas, G. Rodríguez-Gómez, and J. Martínez-Carranza, "Aerodynamic disturbance rejection acting on a quadcopter near ground," in *2019 6th International Conference on Control, Decision and Information Technologies (CoDIT)*, IEEE, 2019.

[43] P. Wei, S. N. Chan, S. Lee, and Z. Kong, "Mitigating ground effect on mini quadcopters with model reference adaptive control," *International Journal of Intelligent Robotics and Applications*, 2019.

[44] M. Saedan and P. Puangmali, "Characterization of motor and propeller sets for a small radio controlled aircraft," in *2015 10th Asian Control Conference (ASCC)*, IEEE, 2015.

[45] M. Yoon, "On driving signal of electronic speed controller for small multi-rotor helicopter," *International Journal of Engineering Research and Technology*, 2015.

[46] M. Yoon, "Experimental identification of thrust dynamics for a multirotor helicopter," *International Journal of Engineering Research and Technology*, 2015.

[47] F. Riccardi and M. Lovera, "Robust attitude control for a variable-pitch quadrotor," in *IEEE Multi-Conference on Systems and Control, Antibes, France*, IEEE, 2014.

[48] A. Radhakrishnan and F. Schmitz, "An experimental investigation of a quad tilt rotor in ground effect," in *21st AIAA Applied Aerodynamics Conference*, 2003.

[49] V. Gupta and J. D. Baeder, "Investigation of quad tiltrotor aerodynamics in forward flight using CFD," in *20th AIAA Applied Aerodynamics Conference*, 2002.

[50] V. Gupta and J. D. Baeder, "Quad tiltrotor aerodynamics in ground effect," in *ANNUAL FORUM PROCEEDINGS-AMERICAN HELICOPTER SOCIETY*, AMERICAN HELICOPTER SOCIETY, INC, 2002.

[51] C. Powers, D. Mellinger, A. Kushleyev, B. Kothmann, and V. Kumar, "Influence of aerodynamics and proximity effects in quadrotor flight," in *Experimental robotics*, Springer, 2013.

[52] M. Giurato, "Design, integration and control of a multirotor UAV platform," Master's thesis, Politecnico di Milano, Italy, 2015.

[53] L. Ljung, *System identification: theory for the user*. Prentice-Hall, 1999.

[54] A. Chiuso, "The role of autoregressive modeling in predictor-based subspace identification," *Automatica*, 2007.

[55] F. Riccardi and M. Lovera, "Dynamic inflow and ground effect in multirotor UAV attitude dynamics," in *43rd European Rotorcraft Forum (ERF 2017)*, 2017.

[56] J. M. Tarascon and M. Armand, "Issues and challenges facing rechargeable lithium batteries," in *Materials for Sustainable Energy: A Collection of Peer-Reviewed Research and Review Articles from Nature Publishing Group*, World Scientific, 2011.

[57] A. M. Harrington, "Optimal propulsion system design for a micro quad rotor," Master's thesis, Department of Aerospace Engineering, University of Maryland, College Park, MD, 2011.

[58] "Georgia institute of technology uav research facility." `http://uavrf.gatech.edu/`. Accessed: 17-10-2019.

[59] "ecalc - drive calculator." `https://ecalc.ch/`. Accessed: 17-10-2019.

[60] J. A. Benito, G. G. de Rivera, J. Garrido, and R. Ponticelli, "Design considerations of a small UAV platform carrying medium payloads," in *Conference on Design of Circuits and Integrated Circuits (DCIS), 2014*, IEEE, 2014.

[61] A. Roza and M. Maggiore, "A class of position controllers for underactuated VTOL vehicles," *IEEE Transactions on Automatic Control*, 2014.

[62] D. Invernizzi and M. Lovera, "Trajectory tracking control of thrust-vectoring UAVs," *Automatica*, 2018.

[63] D. Invernizzi, M. Lovera, and L. Zaccarian, "Geometric trajectory tracking with attitude planner for vectored-thrust VTOL UAVs," in *2018 Annual American Control Conference (ACC)*, pp. 3609–3614, IEEE, 2018.

[64] J. F. Horn, "Non-linear dynamic inversion control design for rotorcraft," *Aerospace*, 2019.

[65] U. Saetti and J. F. Horn, "Use of harmonic decomposition models in rotorcraft flight control design with alleviation of vibratory loads," *European Rotorcraft Forum, Milan, Italy*, 2017.

[66] U. Saetti, J. F. Horn, T. Berger, and M. B. Tischler, *Rotorcraft flight control design with alleviation of unsteady rotor loads*. PhD thesis, The Pennsylvania State University, 2019.

[67] K. K. Cheung, J. A. Wagster, M. B. Tischler, C. M. Ivler, M. G. Berrios, T. Berger, O. Juhasz, E. L. Tobias, C. L. Goerzen, P. S. Barone, F. C. Sanders, M. J. S. Lopez, and R. M. Lehman, "An overview of the U.S. Army Aviation development directorate quadrotor Guidance, Navigation, and Control project," *AHS 73rd Annual Forum*, 2017.

[68] S. Formentin and M. Lovera, "Flatness-based control of a quadrotor helicopter via feedforward linearization," in *2011 50th IEEE Conference on Decision and Control and European Control Conference*, IEEE, 2011.

[69] D. Migliore, "Model identification and inversion based control for multirotor UAV," Master's thesis, Politecnico di Milano, Italy, 2019.

[70] A. A. Ghaffar and T. Richardson, "Model reference adaptive control and LQR control for quadrotor with parametric uncertainties," *International Journal of Mechanical, Aerospace, Industrial, Mechatronic and Manufacturing Engineering*, 2015.

[71] Z. T. Dydek, A. M. Annaswamy, and E. Lavretsky, "Adaptive control of quadrotor UAVs in the presence of actuator uncertainties," *AIAA Infotech Aerospace*, 2010.

[72] E. Lavretsky and K. Wise, *Robust and Adaptive Control With Aerospace Applications*. Springer London, 2013.

[73] E. Lavretsky, "Combined/composite model reference adaptive control," *IEEE Transactions on Automatic Control*, 2009.

[74] Z. T. Dydek, A. M. Annaswamy, and E. Lavretsky, "Adaptive control of quadrotor UAVs: A design trade study with flight evaluations," *IEEE Transactions on Control Systems Technology*, 2013.

[75] T. E. Gibson, A. M. Annaswamy, and E. Lavretsky, "On adaptive control with closed-loop reference models: transients, oscillations, and peaking," *IEEE Access*, 2013.

[76] R. W. Beard, N. Knoebel, C. Cao, N. Hovakimyan, and J. Matthews, "An L_1 adaptive pitch controller for miniature air vehicles," in *AIAA Guidance, Navigation, and Control Conference, Keystone, CO*, 2006.

[77] J. Wang, V. Patel, C. A. Woolsey, N. Hovakimyan, and D. Schmale, "L_1 adaptive control of a UAV for aerobiological sampling," in *2007 American Control Conference*, IEEE, 2007.

[78] B. Michini and J. How, "L_1 adaptive control for indoor autonomous vehicles: design process and flight testing," in *Proceeding of AIAA Guidance, Navigation, and Control Conference*, pp. 5754–5768, 2009.

[79] S. Mallikarjunan, B. Nesbitt, E. Kharisov, E. Xargay, N. Hovakimyan, and C. Cao, "L_1 adaptive controller for attitude control of multirotors," in *AIAA Guidance, Navigation and Control Conference, Minneapolis, AIAA-2012-48312012*, 2012.

[80] G. V. Chowdhary, T. Wu, M. Cutler, N. M. Ure, and J. How, "Experimental results of concurrent learning adaptive controllers," in *AIAA Guidance, Navigation, and Control Conference (GNC),(Minneapolis, MN), AIAA*, 2012.

[81] G. V. Chowdharyand and E. N. Johnson, "Theory and flight-test validation of a concurrent-learning adaptive controller," *Journal of Guidance, Control, and Dynamics*, 2011.

[82] T. Madani and A. Benallegue, "Adaptive control via backstepping technique and neural networks of a quadrotor helicopter," *IFAC Proceedings Volumes*, 2008.

[83] E. N. Johnson and S. K. Kannan, "Adaptive trajectory control for autonomous helicopters," *Journal of Guidance, Control, and Dynamics*, 2005.

[84] A. Russo, D. Invernizzi, M. Giurato, and M. Lovera, "Adaptive augmentation of the attitude control system for a multirotor UAV," in *7th European Conference for Aeronautics and Space Sciences, 2017*, 2017.

[85] A. Russo, "Adaptive control of multirotor UAVs," Master's thesis, Politecnico di Milano, Italy, 2017.

[86] S. Kim and Y. Kim, "Mismatch-observer based model reference adaptive control for transient performance improvement of aircraft," in *AIAA guidance, navigation, and control conference*, 2017.

[87] N. Hovakimyan and C. Cao, L_1 *adaptive control theory: guaranteed robustness with fast adaptation*. Siam, 2010.

[88] D. Kastelan, M. Konz, and J. Rudolph, "Fully actuated tricopter with pilot-supporting control," *IFAC-PapersOnLine*, 2015.

[89] B. Crowther, A. Lanzon, M. Maya-Gonzalez, and D. Langkamp, "Kinematic analysis and control design for a nonplanar multirotor vehicle," *Journal of Guidance, Control, and Dynamics*, 2011.

[90] A. Franchi, R. Carli, D. Bicego, and M. Ryll, "Full-pose tracking control for aerial robotic systems with laterally bounded input force," *IEEE Transactions on Robotics*, 2018.

[91] F. Goodarzi, D. Lee, and T. Lee, "Geometric nonlinear PID control of a quadrotor UAV on SE(3)," in *Proc. European Control Conf. (ECC)*, 2013.

[92] F. Forni, S. Galeani, and L. Zaccarian, "A family of global stabilizers for quasi-optimal control of planar linear saturated systems," *IEEE Transactions on Automatic Control*, 2010.

[93] M. D. Hua and C. Samson, "Time sub-optimal nonlinear PI and PID controllers applied to longitudinal headway car control," *International Journal of Control*, 2011.

[94] "PX4." `https://github.com/PX4/Firmware`. Accessed: 02-11-2019.

[95] D. Brescianini, M. Hehn, and R. D'Andrea, "Nonlinear quadrocopter attitude control," *Department of Mechanical and Process Engineering, ETHZ, Tech. Rep*, 2013.

[96] C. G. Mayhew, R. G. Sanfelice, and A. R. Teel, "Quaternion-based hybrid control for robust global attitude tracking," *IEEE Transactions on Automatic Control*, 2011.

[97] E. Kaufman, K. Caldwell, D. Lee, and T. Lee, "Design and development of a free-floating hexrotor UAV for 6-dof maneuvers," in *2014 IEEE Aerospace Conference*, IEEE, 2014.

[98] T. Paul, T. R. Krogstad, and J. T. Gravdahl, "Modelling of UAV formation flight using 3d potential field," *Simulation Modelling Practice and Theory*, 2008.

[99] F. Borrelli, T. Keviczky, and G. J. Balas, "Collision-free UAV formation flight using decentralized optimization and invariant sets," in *2004 43rd IEEE Conference on Decision and Control (CDC) (IEEE Cat. No.04CH37601)*, IEEE, 2004.

[100] Z. Chao, S. L. Zhou, L. Ming, and W. G. Zhang, "UAV formation flight based on nonlinear model predictive control," *Mathematical Problems in Engineering*, 2012.

[101] P. R. Thomas, U. Bhandari, S. Bullock, T. S. Richardson, and J. L. du Bois, "Advances in air to air refuelling," *Progress in Aerospace Sciences*, 2014.

[102] C. Martinez, T. Richardson, and P. Campoy, "Towards autonomous air-to-air refuelling for UAVs using visual information," in *2013 IEEE International Conference on Robotics and Automation*, IEEE, 2013.

[103] J. Valasek, K. Gunnam, J. Kimmett, J. L. Junkins, D. Hughes, and M. D. Tandale, "Vision-based sensor and navigation system for au-

tonomous air refueling," *Journal of Guidance, Control, and Dynamics*, 2005.

[104] B. Herissé, T. Hamel, R. Mahony, and F. X. Russotto, "Landing a VTOL unmanned aerial vehicle on a moving platform using optical flow," *IEEE Transactions on Robotics*, 2012.

[105] D. Lee, T. Ryan, and H. J. Kim, "Autonomous landing of a VTOL UAV on a moving platform using image-based visual servoing," in *2012 IEEE International Conference on Robotics and Automation*, IEEE, 2012.

[106] J. W. Kim, Y. Jung, D. Lee, and D. H. Shim, "Outdoor autonomous landing on a moving platform for quadrotors using an omnidirectional camera," in *2014 International Conference on Unmanned Aircraft Systems (ICUAS)*, IEEE, 2014.

[107] P. Vlantis, P. Marantos, C. P. Bechlioulis, and K. J. Kyriakopoulos, "Quadrotor landing on an inclined platform of a moving ground vehicle," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2015.

[108] B. Hu, L. Lu, and S. Mishra, "Fast, safe and precise landing of a quadrotor on an oscillating platform," in *2015 American Control Conference (ACC)*, IEEE, 2015.

[109] B. Hu, L. Lu, and S. Mishra, "A control architecture for fast and precise autonomous landing of a VTOL UAV onto an oscillating platform," in *American Helicopter Society 71st Annual Forum*, 2015.

[110] A. Monneau, N. Msirdi, S. Mavromatis, G. Varra, M. Salesse, and J. Sequeira, "Adaptive prediction for ship motion in rotorcraft maritime operations," in *5th CEAS Specialist Conference on Guidance, Navigation & Control-EUROGNC*, 2019.

[111] B. Hu, L. Lu, and S. Mishra, "A control architecture for time-optimal landing of a quadrotor onto a moving platform," *Asian Journal of Control*, 2018.

[112] L. Lu and B. Yao, "A performance oriented multi-loop constrained adaptive robust tracking control of one-degree-of-freedom mechanical systems: Theory and experiments," *Automatica*, 2014.

[113] A. Borowczyk, D. T. Nguyen, A. P.-V. Nguyen, D. Nguyen, D. Saussié, and J. L. Ny, "Autonomous landing of a multirotor micro air vehicle on a high velocity ground vehicle," *IFAC-PapersOnLine*, 2017.

[114] G. Gozzini, "Uav autonomous landing on moving aerial vehicle," Master's thesis, Politecnico di Milano, Italy, 2019.

[115] "Aerospace System and Control Laboratory - ASCL." `http://ascl.daer.polimi.it/`. Accessed: 02-11-2019.

[116] "Department of AERospace science and technology - DAER." https://www.aero.polimi.it/.

[117] "Optitrack motion capture system." https://optitrack.com/.

[118] "Motive." https://optitrack.com/products/motive/.

[119] "Windows 10 Pro." `https://www.microsoftstore.com/store/mseea/it_IT/pdp/Windows-10-Pro/productID.320433200`. Accessed: 02-11-2019.

[120] "Ubuntu 16.04 LTS." `https://www.ubuntu.com`. Accessed: 02-11-2019.

[121] "ROS." `http://www.ros.org`. Accessed: 02-11-2019.

[122] "MATLAB." `https://www.mathworks.com/products/matlab.html`. Accessed: 02-11-2019.

[123] "Pixhawk Mini." `https://docs.px4.io/v1.9.0/en/flight_controller/pixhawk_mini.html`. Accessed: 02-11-2019.

[124] "MAVLink." `https://mavlink.io/en/`. Accessed: 02-11-2019.

[125] "QGroundControl." http://qgroundcontrol.com/. Accessed: 02-11-2019.

[126] "NanoPi NEO Air." http://wiki.friendlyarm.com/wiki/index.php/NanoPi_NEO_Air. Accessed: 02-11-2019.

[127] "Mavros." https://github.com/mavlink/mavros. Accessed: 02-11-2019.

APPENDIX $\mathcal{A}$

The PBSID algorithm

Consider the finite dimensional LTI state space model

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) + w(t), \\ y(k) &= Cx(k) + Du(k) + v(k), \end{aligned} \tag{A.1}$$

where $x(k) \in \mathbb{R}^n$, $u(k) \in \mathbb{R}^m$, $y(k) \in \mathbb{R}^p$ and $\{w(k) \in \mathbb{R}^n, v(k) \in \mathbb{R}^p\}$ are ergodic sequences of finite variance

$$E\left[\begin{bmatrix} w(t) \\ v(t) \end{bmatrix} \begin{bmatrix} w(t)^\top & v(t)^\top \end{bmatrix}\right] = \begin{bmatrix} Q & S \\ S^\top & R \end{bmatrix} \delta_{s,t} \tag{A.2}$$

with $\delta_{s,t}$ denoting the Kronecker delta function, possibly correlated with the input $u(t)$. Let now

$$z(k) = \begin{bmatrix} u^\top(k) & y^\top(k) \end{bmatrix}^\top \tag{A.3}$$

and

$$\bar{A} = A - KC, \quad \bar{B} = B - KD, \quad \widetilde{B} = \begin{bmatrix} \bar{B} & K \end{bmatrix} \tag{A.4}$$

where K is the Kalman gain associated with the system (A.1), and note that the system (A.1) can be written as

$$\begin{aligned} x(k+1) &= \bar{A}x(k) + \widetilde{B}z(k) \\ y(k) &= Cx(k) + Du(k) + e(k), \end{aligned} \tag{A.5}$$

where $e(k) \in \mathbb{R}^p$ is the innovation vector. The data equations for the PBSID algorithm can be then derived by noting that propagating $p-1$ steps forward the first of equations (A.5), where p is the so-called past window length, one gets

$$\begin{aligned} x(k+2) &= \bar{A}^2 x(k) + \begin{bmatrix} \bar{A}\tilde{B} & \tilde{B} \end{bmatrix} \begin{bmatrix} z(k) \\ z(k+1) \end{bmatrix} \\ &\vdots \\ x(k+p) &= \bar{A}^p x(k) \mathscr{K}^p Z^{0,p-1} \end{aligned} \tag{A.6}$$

where

$$\mathscr{K}^p = \begin{bmatrix} \bar{A}^{p-1}\widetilde{B}_0 & \dots & \widetilde{B} \end{bmatrix}, \tag{A.7}$$

is the extended controllability matrix of the system and

$$Z^{0,p-1} = \begin{bmatrix} z(k) \\ \vdots \\ z(k+p-1) \end{bmatrix}. \tag{A.8}$$

Under the considered assumptions, $\bar{A}$ represents the dynamics of the optimal one-step ahead predictor for the system and therefore it has all the eigenvalues inside the open unit circle, so the term $\bar{A}^p x(k)$ is negligible for sufficiently large values of p and one have that

$$x(k+p) \approx \mathscr{K}^p Z^{0,p-1}. \tag{A.9}$$

As a consequence, the input-output behaviour of the system is approximately given by

$$\begin{aligned} y(k+p) &\approx C\mathscr{K}^p Z^{0,p-1} + Du(k+p) + e(k+p) \\ &\vdots \\ y(k+p+f) &\approx C\mathscr{K}^p Z^{f,p+f-1} + Du(k+p+f) + e(k+p+f). \end{aligned} \tag{A.10}$$

Introducing the matrix notation defined as:

$$\begin{aligned} \bar{Z}^{p,f} &= \begin{bmatrix} Z^{0,p-1} & Z^{0,p} & \dots & Z^{0,p+f-1} \end{bmatrix}^\top \\ X^{p,f} &= \begin{bmatrix} x(k+p) & x(k+p+1) & \dots & x(k+p+f) \end{bmatrix}^\top \\ Y^{p,f} &= \begin{bmatrix} y(k+p) & y(k+p+1) & \dots & y(k+p+f) \end{bmatrix}^\top \\ U^{p,f} &= \begin{bmatrix} u(k+p) & u(k+p+1) & \dots & u(k+p+f) \end{bmatrix}^\top \\ E^{p,f} &= \begin{bmatrix} e(k+p) & e(k+p+1) & \dots & e(k+p+f) \end{bmatrix}^\top \end{aligned}$$

the data equations are given by

$$\begin{aligned} X^{p,f} &\simeq \mathscr{K}^p \bar{Z}^{p,f}, \\ Y^{p,f} &\simeq C\mathscr{K}^p \bar{Z}^{p,f} + DU^{p,f} + E^{p,f}. \end{aligned} \tag{A.11}$$

Considering $p = f$ (where f is the so-called future window length), an estimate for the matrices $C\mathscr{K}^p$ and D is obtained by solving the least-squares problem

$$\min_{C\mathscr{K}^p, D} \|Y^{p,p} - C\mathscr{K}^p \bar{Z}^{p,p} - DU^{p,p}\|_F. \tag{A.12}$$

Defining now the extended observability matrix Γ^p as

$$\Gamma^p = \begin{bmatrix} C \\ C\bar{A} \\ \vdots \\ C\bar{A}^{p-1} \end{bmatrix}, \tag{A.13}$$

and noting that the product of Γ^p and $\mathscr{K}^p$ can be written as

$$\Gamma^p \mathscr{K}^p \approx \begin{bmatrix} C\bar{A}^{p-1}\tilde{B} & \dots & C\tilde{B} \\ 0 & \dots & C\bar{A}\tilde{B} \\ \vdots & \ddots & \vdots \\ 0 & \dots & C\bar{A}^{p-1}\tilde{B} \end{bmatrix}, \tag{A.14}$$

such product can be computed using the estimate $\widehat{C\mathscr{K}^p}$ of $C\mathscr{K}^p$ obtained by solving the least squares problem (A.12). Recalling now that

$$X^{p,p} \approx \mathscr{K}^p \bar{Z}^{p,p}, \tag{A.15}$$

it also holds that

$$\Gamma^p X^{p,p} \approx \Gamma^p \mathscr{K}^p \bar{Z}^{p,p}. \tag{A.16}$$

Therefore, computing the Singular Value Decomposition (SVD)

$$\Gamma^p \mathscr{K}^p \bar{Z}^{p,p} = U\Sigma V^\top, \tag{A.17}$$

an estimate of the state sequence can be obtained as

$$\widehat{X}^{p,p} = \Sigma_n^{1/2} V_n^\top = \Sigma_n^{-1/2} U_n^\top \Gamma^p \mathscr{K}^p \bar{Z}^{p,p}, \tag{A.18}$$

from which an estimate of matrix C can be obtained solving the least squares problem

$$\min_C \|Y^{p,p} - \widehat{D}U^{p,p} - C\widehat{X}^{p,p}\|_F. \tag{A.19}$$

The final steps consist of the estimation of the innovation data matrix $E_N^{p,f}$

$$E_N^{p,f} = Y^{p,p} - \widehat{C}\widehat{X}^{p,p} - \widehat{D}U^{p,p}, \tag{A.20}$$

and of the entire set of the state space matrices for the system, which can be obtained by solving the least squares problem

$$\min_{A,B,K} \|\widehat{X}^{p+1,p} - A\widehat{X}^{p,p-1} - BU^{p,p-1} - KE^{p,p-1}\|_F. \tag{A.21}$$

APPENDIX B

Additional UAV platforms

The aim of this appendix is to describe some additional UAV platform which have been used during the experimental activities of this thesis but which have not been included in the platform description of Chapter 1.

B.1 ANT-1

The ANT-1 is a lightweight and small scale quadrotor built with commercial off-the-shelf components. Its main features are:

- Take Off Weight (TOW): 0.23 kg;
- Frame size: 160 mm;
- Motors: 4× QAV1306-3100kV;
- Propellers: 4× three-bladed propellers 3055;

- Rotor configuration: *X* configuration;
- Battery: LiPo 950 mA h 3*S*;
- Flight time (hovering): 8 min.

A detailed view of the quadcopter is shown in Figure B.1.

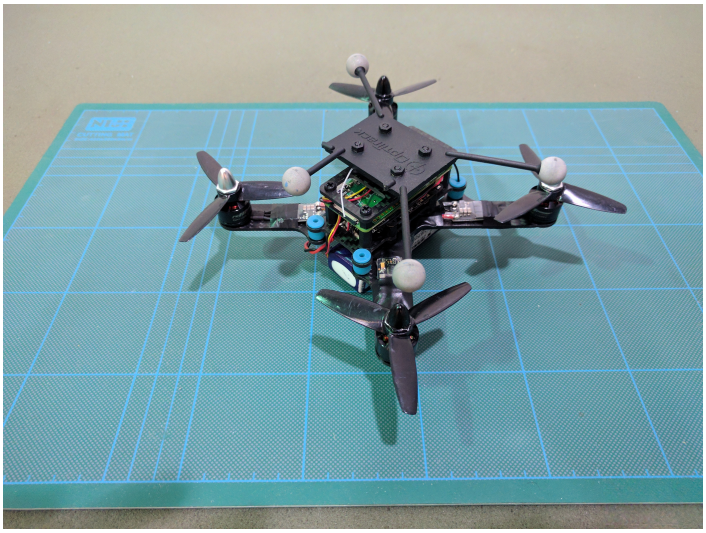


Figure B.1: *ANT-1 quadcopter.*

B.2 ANT-R

The ANT-R has been assembled with commercial off-the-shelf components, with the aim of obtaining a light racer quadcopter with high performance and optimized for forward flight. Its main features are:

- Take Off Weight (TOW): 0.73 kg;
- Frame size: 250 mm;
- Motors: 4× Emax RS2205-2300KV;
- Propellers: 4× three-bladed propellers 5045;

- Rotor configuration: *H* configuration;
- Battery: LiPo 2650 mA h 4*S*;
- Flight time (hovering): 13 min.

A detailed view of the quadcopter is shown in Figure B.2. As it can be seen from the figure, the *H* rotors configuration, which can be considered equal to an *X* configuration in terms of rotors numbering, makes the platform asymmetric, determining different inertia moments about the *x* and *y* body axes.

Figure B.2: *ANT-R quadcopter.*

B.3 CARRIER-1

As the ANT-R, the CARRIER-1 has been assembled with off-the-shelf components as well but with the purpose of obtaining a UAV which could provide a high flight time and a high thrust-to-weight ratio. Another interesting feature of this platform is that its frame has been specifically designed with the purpose to work as a flying landing pad for smaller UAVs as the ANT-R. Its features in numbers are:

- Take Off Weight (TOW): 2.9 kg;
- Frame size: 0.4×0.4 m;

- Motors: 8× T-motor F40 PRO II 1600KV;
- Propellers: 8× two-bladed propellers 6535;
- Rotor configuration: octocopter configuration;
- Battery: LiPo 8000 mA h 6*S* + 2200 mA h 3*S*;
- Flight time (hovering): 16 min.

A detailed view of the octocopter is shown in Figure B.3.

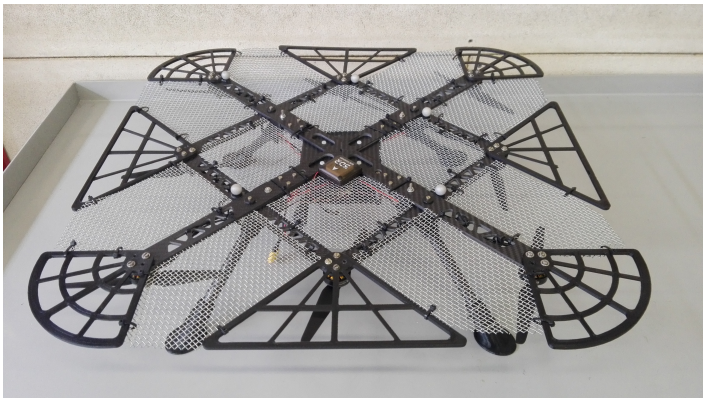


Figure B.3: *CARRIER-1 octocopter.*

APPENDIX C

The Flying Arena for Rotorcraft Technologies

The Flying Arena for Rotorcraft Technologies (FlyART) is an indoor facility designed and developed from the Aerospace System and Control Laboratory (ASCL, see [115]) which is the scientific laboratory of the Department of Aerospace Science and Technology of Politecnico di Milano (see [116]) devoted to systems and control research. This facility allows the indoor testing and prototyping of new small UAV concepts and new subsystems (*e.g.*, guidance, navigation and control systems) both for single and formation flight. The FlyART has a flight volume of $12 \times 6 \times 4\,\mathrm{m}$ equipped with a motion capture system (OptiTrack, see [117]). In Figure C.1 is reported a picture of the FlyART facility.

In the following sections will be described the systems which compose the facility.

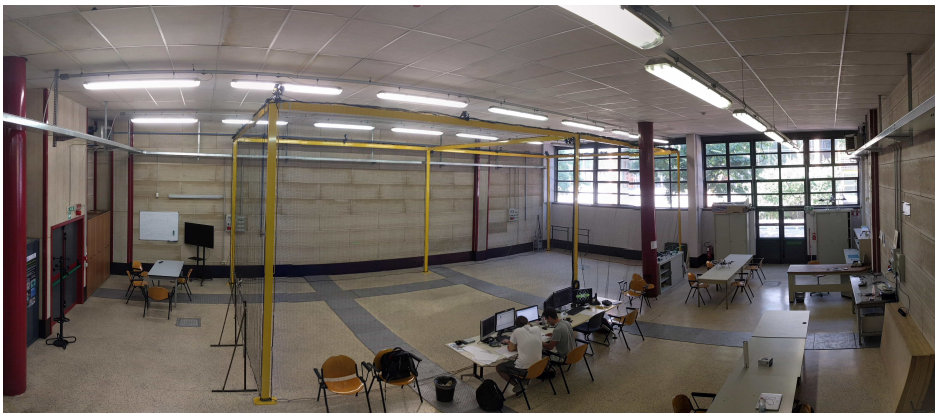


Figure C.1: *The FlyART facility.*

C.1 Motion Capture system

The Motion Capture system (Mo-Cap) is composed by 12 Infra-Red (IR) sensitive Optitrack [117] cameras (Figure C.2(a)) which incorporate IR flood lights. The cameras, mounted on the FlyART, are fixed at calibrated positions and orientation so that the measurement subject is into the field of view of multiple cameras. Through markers sensitive to infra-red light (Figure C.2(b)) mounted on top of the drones, it is possible to estimate and track their attitude and positions inside the flight volume. Each UAV mounts a different marker layout to be uniquely identified when the two drones fly at the same time.

To control the motion capture system, the Motive software [118] is installed on the ground station. It does not only allows the user to calibrate the system, but it also provides interfaces for capturing and processing 3D data, that can be recorded or live-streamed. The accuracy of the position estimated by the UAV depends on the frequency with which the position informations are sent to it (cameras rate), which can be selected in the range $30 - 240\,\text{Hz}$.

(a) Infrared camera (from [117]). (b) Infrared markers (from [117]).

Figure C.2: *Motion Capture System.*

C.2 Ground Control System

The Ground Control System the part of the system dedicated to send and retrieve information from the UAVs during a flight session. The main task of the GCS is to read the attitude and position information provided by motion capture system and send them to the drones (at a frequency of 100 Hz), besides it is also possible to define a trajectory to be followed using specific way points (usually sent with a frequency of 50 Hz) and view telemetry data in real time.

The GCS architecture is divided into two different computers: the first computer which runs Windows 10 [119] is used to execute Motive while the second computer, which runs Linux OS (more precisely Ubuntu 16.04 [120]), is used to execute ROS [121] and MATLAB [122]. The GCS architectural division was necessarily done because Motive is a Windows software while ROS integrates better in a Linux environment.

Figure C.3: *Pixhawk Mini FCU (from [123]).*

C.3 Drones HW/SW architecture

Each UAV which fly inside the indoor facility mounts two different electronic boards, the Flight Control Unit (FCU) and the Flight Computer Companion (FCC).

C.3.1 Flight Control Unit

The FCU is an electronic board which runs the control and the navigation algorithms. The adopted FCU is the electronic board Pixhawk Mini [123] which integrates the inertial sensors, such as 3-axes accelerometer, 3-axes gyroscope, magnetometer and barometer.

The main features of the Pixhawk Mini are:

- Processor: main STM32F427 Rev 3 based on 32 bit ARM Cortex® M4 core with 180 MHz CPU and an IO processor STM32F103 based on the Cortex® M3 core with 72 MHz CPU;
- Interfaces: UART serial port for GPS, spektrum DSM/DSM2/DSM-X® satellite compatible RC input, Futaba S BUS® compatible RC input, PPM sum signal RC input, I2C for digital sensors, CAN for digital motor control with compatible controllers, ADC for analog sensors and micro USB port;
- Weight: 15.8 g;
- Dimensions: 38 x 43 x 12 mm.

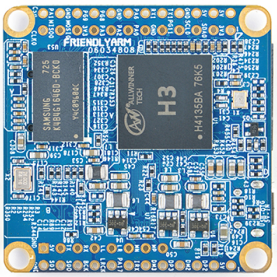


Figure C.4: *NanoPi NEO Air (from [126]).*

The MAVLInk [124] protocol is used for serial communication between FCU and FCC, PX4 firmware [94] is the one supported by Pixhawk Mini and QGroundcontrol [125] is the software used for Pixhawk Mini configuration and real time information.

C.3.2 Flight Computer Companion

The FCC computer (Figure C.4) is the part of the drone system used to interface and communicate with PX4 using the MAVLink protocol. It enables a variety of functionalities, such as the possibility to execute processes that require heavy CPU load. During the flight test in the flight arena, the companion computer is used to receive:

- the position and attitude from the Ground Control System, connected to the Mo-Cap system;
- the commands coming from the Ground Control System;

these informations are sent to the FCU through the serial communication. On the drones the NanoPi NEO Air [126] in Figure C.4, that has the characteristics reported in Table C.1, is used.

Name CPU	Quad-core Cortex-A7 1.2 GHz
RAM	512 MB
Wireless	2.4 GHz 802.11 $b/g/n$
Dimensions	40 x 40 mm
Weight	7.9 g
Power	5 V - 2 A

Table C.1: *NanoPi NEO Air features.*

From a software point of view the Robot Operating System (ROS) [121] is used to communicate with Ground Control Station through ROS messages. In particular Mavros [127], that is a ROS package, provides communication driver for Pixhawk Mini autopilot with MAVLink communication protocol. Additionally it provides UDP MAVLink bridge for Ground Control Station (*e.g.*, QGroundControl);