UNIVERSITÀ DEGLI STUDI DELLA BASILICATA
SCUOLA DI INGEGNERIA

DOTTORATO DI RICERCA IN INGEGNERIA DELL'INNOVAZIONE E LO SVILUPPO SOSTENIBILE,
XXXIV CICLO
SETTORE SCIENTIFICO DISCIPLINARE: AUTOMATICA ING-INF/04

PH.D. THESIS

An Omnidirectional Aerial Platform for Multi-Robot Manipulation

Michelangelo Nigro

ADVISOR

Prof. Ing. Fabrizio Caccavale
Dott. Ing. Francesco Pierri

Knowledge, Power to Fly.

Abstract

The objectives of this work were the modeling, control and prototyping of a new fully-actuated aerial platform. Commonly, the multirotor aerial platforms are under-actuated vehicles, since the total propellers thrust can not be directed in every direction without inferring a vehicle body rotation. The most common fully-actuated aerial platforms have tilted or tilting rotors that amplify the aerodynamic perturbations between the propellers, reducing the efficiency and the provided thrust. In order to overcome this limitation a novel platform, the ODQuad (OmniDirectional Quadrotor), has been proposed, which is composed by three main parts, the platform, the mobile and rotor frames, that are linked by means of two rotational joints, namely the roll and pitch joints. The ODQuad is able to orient the total thrust by moving only the propellers frame by means of the roll and pitch joints.

Kinematic and dynamic models of the proposed multirotor have been derived using the *Euler-Lagrange* approach and a model-based controller has been designed. The latter is based on two control loops: an outer loop for vehicle position control and an inner one for vehicle orientation and roll-pitch joint control. The effectiveness of the controller has been tested by means of numerical simulations in the MATLAB© SimMechanics environment. In particular, tests in free motion and in object transportation tasks have been carried out. In the transportation task simulation, a momentum based observer is used to estimate the wrenches exchanged between the vehicle and the transported object.

The ODQuad concept has been tested also in cooperative manipulation tasks. To this aim, a simulation model was considered, in which multiple ODQuads perform the manipulation of a bulky object with unknown inertial parameters which are identified in the first phase of the simulation. In order to reduce the mechanical stresses due to the manipulation and enhance the system robustness to the environment interactions, two admittance filters have been implemented: an external filter on the object motion and an internal one local for each multirotor.

Finally, the prototyping process has been illustrated step by step. In particular, three CAD models have been designed. The ODQuad.01 has been used in the simulations and in a preliminary static analysis that investigated the torque values for a rough sizing of the roll-pitch joint actuators. Since in the ODQuad.01 the components specifications and the related manufacturing techniques have not been taken into account, a successive model, the ODQuad.02, has been designed. The ODQuad.02 design can be developed with aluminum or carbon fiber profiles and 3D printed parts, but each component must be custom manufactured. Finally, in order to shorten the prototype development time, the ODQuad.03 has been created, which includes some components of the off-the-shelf quadrotor *Holybro X500* into a novel custom-built mechanical frame.

Acknowledgments

This thesis collects the scientific developments obtained in the PhD course in Engineering of Innovation and Sustainable Development. Therefore, first and foremost I want to thank my advisor Professor Fabrizio Caccavale. I appreciate all his contributions of time, ideas, and funding to make my Ph.D. experience productive and stimulating.

I am deeply grateful to Professor Francesco Pierri for teaching me method and rigor in scientific research and for his help and advice in academic pursuits. We worked together on many activities giving me a valuable contribution in the study of controller stability and in the exposition of the results obtained in these years.

I would like to offer my special thanks to Professor Katia Genovese, we spent together a lot of time in the study of vision systems calibration techniques, in particular on the hand-eye calibration procedure. She allowed me to 3D print the ODQuad components and taught me the tricks to get high quality 3D prints.

I would also like to thank Monica Sileo for the software support she gave me many times. I am grateful to my friends Sergio, Antonio and Marco for the interesting daily discussions we had, in particular, I thank Sergio for the support he gave me in the thesis submission.

Lastly, I wish to extend my special thanks to my family for all their support and encouragement. For my loving, supportive, encouraging, and patient girlfriend Margherita faithful support during this Ph.D. is so appreciated. For my parents who believed in my choices, trusted me and transmitted me honesty, seriousness and the work dedication. For my sisters Donatella and Marina who helped me every time I asked for something. Thank you.

Contents

List of Figures

List of Tables

Introduction

I.1 Aerial manipulation

Manipulation is one of the most studied topics in industrial robotics. Originally, the robot manipulators involved in manufacturing plants were used in point-to-point object transfer. Later on, thanks to the continuous technical improvement, it has been possible to start using robotic arms even in very complex tasks, i.e., mechanical assembly, welding, painting, packaging and many other industrial operations. Industry 4.0 has led to the emergence of much more dynamic, versatile and easily reconfigurable production plants. Therefore, new and even more competitive challenges have arisen, such as the extension of the fixed manipulators workspace and cooperation with humans. In the first case, by using mobile platforms, it has been possible to extend the workspace of a fixed manipulator in two dimensions and, in order to perform robotic operations also in three-dimensional space, the scientific and industrial communities proposed the remotely controlled or autonomous aerial platforms, called Unmanned Aerial Vehicles (UAVs).

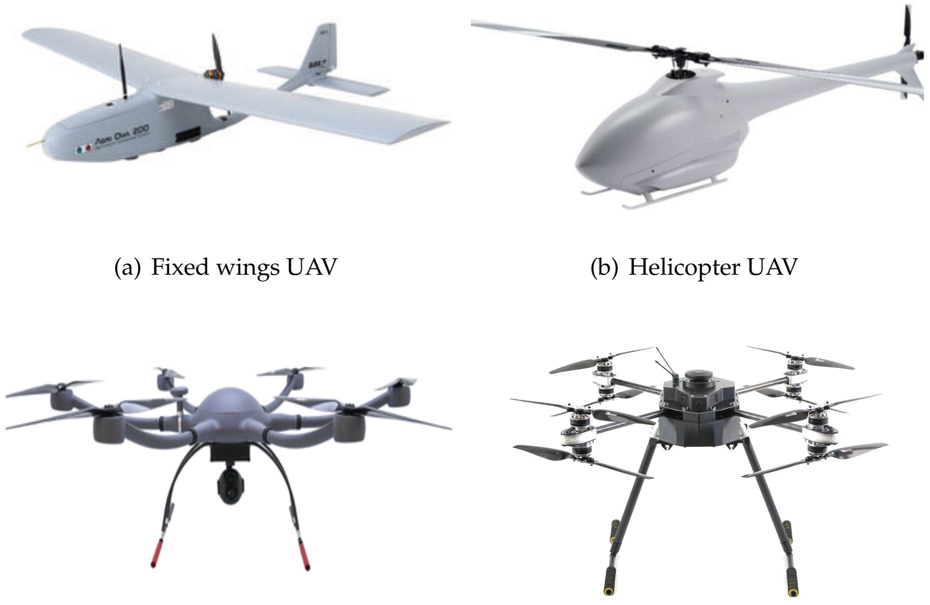
The term mobile manipulation was intoduced in the 80s, when research laboratories began to mount robot manipulators on mobile platforms and afterward when the aerial platforms were also equipped with a robotic arm the definition of aerial manipulation was coined. Aerial manipulation is the branch of robotics that studies the objects transportation by means UAVs, with particular attention to the complex problem of UAV-object and object-environment interaction [9]. Some examples of application are the laying of electrical cables on pylons and maintenance operations, intended as visual or tactile control in large production plants such as refineries [10]. During the most recent decades, significant efforts have been devoted to increasing the control performances, payload and flight endurance of UAVs, which are essential features for performing aerial manipulation operations. UAVs present various configurations with different sizes, endurance levels and capabilities. More in detail, UAV platforms typically fall into one of the following main categories [11]:

- Fixed-wing UAVs, which refer to unmanned airplanes (with wings) that require a runway to take-off and land, or catapult launching. These generally have long endurance and can fly at high cruising speeds, (Fig. I.1(a)).

- Rotary-wing UAVs, also called rotorcraft UAVs or vertical take-off and landing (VTOL) UAVs, which have the advantages of hovering capability and high maneuverability, that are useful for aerial robotic manipulation.

- A rotorcraft UAV may have different configurations, with main and tail rotors (Fig.I.1(b)), multi-rotors (Fig.I.1(c), coaxial rotors (Fig. I.1(d)).

Reference [12] reports a general taxonomy to characterize and describe multirotor aerial vehicles and their designs, in order to show their abilities in terms of tasks and system properties.



(a) Fixed wings UAV

(b) Helicopter UAV

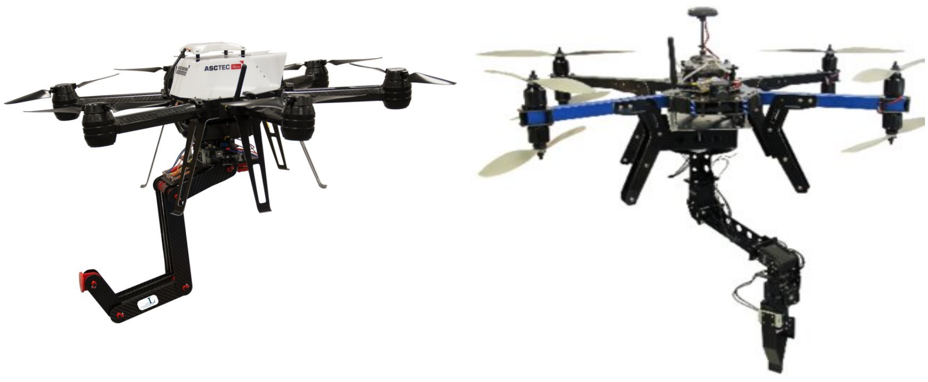
(c) Multirotor UAV

(d) Coaxial multirotor UAV

Figure I.1: Different typologies of rotorcraft UAVs.

With the purpose of performing aerial manipulation tasks, lightweight robotic arms have been mounted on aerial multirotor platform (Fig.I.2(a), I.2(b)).

The control of multirotor UAVs is very challenging, since i) the feedback data on the robot motion are often rough and affected by considerable noise, ii) the propeller thrust



(a) Multirotor UAV (b) Coaxial multirotor UAV

Figure I.2: Multirotors equipped with a robotic arm manipulator.

is highly variable, as it depends on the local air characteristics, i.e., air density and air velocity, and finally, iii) the common multirotor platforms are under-actuated aerial vehicles as only four of the six DoFs (Degrees of Freedom) are actuated. The under-actuation problem may become severe in some applications, e.g., manipulation tasks involving contact with the environment. For this reason, fully-actuated aerial multirotor platforms have been introduced that are actuated on the overall six DoFs. In [13] a literature review on the fully-actuated multirotors has been reported. Commonly adopted solutions involve the use of at least six tilted propellers or the addition of one or more actuators that permits to orient the propellers all together or individually, respectively.

I.2 Cooperative Manipulation

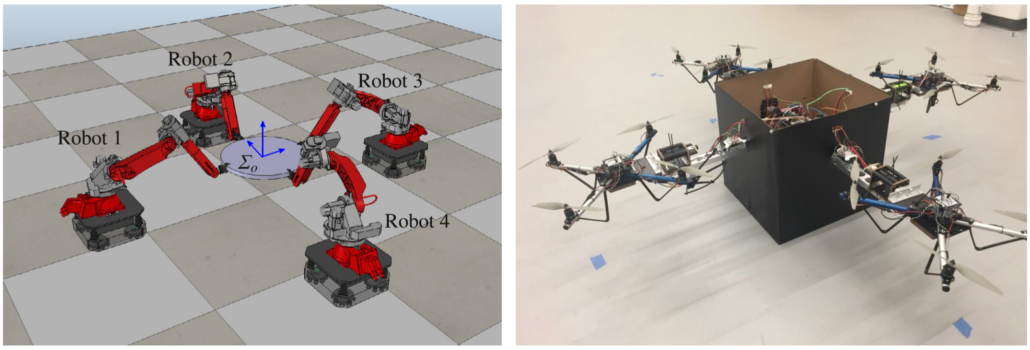
Making an analogy with the human body there are countless operations in which we engage both arms at the same time, such as grasping bulky objects or assembling two mechanical parts, hence analogously, in robotics, systems of multiple manipulators involved in cooperative tasks (i.e., cooperative systems) are used. Cooperative systems, generally refer to multiple robotic agents that share the workspace at the same time. On the other hand, the use of multi-arm systems for cooperative execution of manipulation tasks allows overcoming the limitations of single-arm systems, especially in the presence of large and/or heavy payloads [14]. Cooperative robotic systems are often classified in two classes:

- without physical interaction, when the robotic manipulators share only the workspace;

in this case the motion controller plans the end-effector so as to achieve coordination with the other team members;

- with physical interaction (often called tight cooperation), when a mechanical connection between the manipulators exists, that can be totally rigid or not.

In the first case, the robots of the cooperating work-cell are programmed such as to follow trajectories that avoid collisions between them. It is therefore necessary to have a supervision system that knows at each time step the kinematic status of all team members. In the second case, the interaction between the manipulators is based on the exchange of forces and torques given by the physical connection between the end-effectors. When the physical connection is totally rigid, i.e, it consists of an undeformable object rigidly grasped by the end-effectors, the resultant mechanical system can be considered as a single system of rigid bodies that can be modeled as a closed kinematic chain [15], [16]. Exploiting the closed-chain constraints, it is possible to compute the trajectory of all end-effectors starting from the object trajectory. When the manipulators are not rigidly connected, the system can still be modeled as a closed kinematic chain, but it is necessary to add other equations that take into account the additional DoFs at the grasp points [15], [17]. The mobile (Fig.I.3(a)) and aerial (Fig.I.3(b)) cooperative manipulation can be modeled in analogous way.



(a) Five mobile robot, image from [18].

(b) Transportation with four multirotors.

Figure I.3: Multi-robot object transportation.

In the case of cooperative manipulation with rigid connection between the robots and the object, a fundamental modeling framework is the symmetric kinetostatic formulation [19], [20] and [21]. In the symmetric formulation, the distinction between external and internal wrenches is introduced; the first one contributes to the object's motion, while, the second ones represents the mechanical stresses applied to the object. This formulation presents a few limitations: the first consists in the difficulty in the identification of the orientation variables associated with the external and internal angular velocities, and the second is related to the fact that the defined kinetostatic variables do not retain a clear meaning when pure coordination is at stake. An alternative formulation is the task-oriented formulation proposed in [22]. This latter formulation describes the coordinated motion via geometrically meaningful variables and it can be used in the case of non-rigid grasping and/or deformable object. In [23], the virtual linkage model has been proposed, that is a geometrical parameterization of the internal wrenches. In this model, the grasp points are connected by a closed non-intersecting set of virtual links, that a closed-chain, and the internal wrenches are characterized as force and torques acting on the actuators of the virtual links.

The first approaches to the control of cooperative systems are the master/slave control [24] and the leader/follower control [15]. In the master/slave approach the master is controlled in position and imposes the desired trajectory to the manipulated object; the remaining manipulators follow the object's trajectory via force/compliance control. A natural evolution of the above approach is the leader/follower, where the follower arms reference motion is computed via closed-chain constraints. However, such approaches suffered from implementation issues, mainly due to the fact that the compliance of the slave arms has to be very large, so as to follow the motion imposed by the master arm smoothly, thus, the perfectly tracking of the object's trajectory it is not guaranteed [25]. In order to overcome this drawback, controllers that consider the object's trajectory tracking and internal wrenches control at the same time have been introduced. These approaches are: object's space control [26], hybrid force/position control [27] and compliant control [28]. In the presence of tight cooperation and environment interaction, in order to keep bounded the internal wrenches, compliance control can be considered. A compliant behavior can be achieved in the following two ways [29]:

- passive interaction control, when the robot end-effector motion is modified by interaction forces thanks to the mechanical compliance of the robot. The compliance may be due to the structural compliance of the links, joint, and end-effector. A common mechanical device, that in the past has been used to confer passive compliance is the Remote Center of Compliance (RCC);

- active interaction control, which modifies the end-effector trajectory based on the contact wrenches measurements.

A further distinction must be made in the area of active interaction control: indirect force control and direct force control. The first category includes the impedance control and the admittance control, where a mass-spring-damper behavior is assigned to the manipulators end-effector, according to the internal wrenches feedback. More in detail, the robot behavior is an impedance if the control reacts to the motion deviation by generating wrenches, while it corresponds to an admittance if the control reacts to interaction wrenches by imposing a deviation from the desired motion. In the other case, i.e., with direct force control, an explicit model of the interaction task is required. More specifically, the user has to specify the desired motion and the desired contact wrench in a consistent way with respect to the constraints imposed by the environment. A commonly adopted strategy is hybrid force/motion control, which aims at controlling the motion and wrench along the unconstrained and constrained task directions, respectively.

I.3 Aerial cooperative manipulation

Manipulation tasks such as lifting, holding and moving an object are generally not realizable with a single robotic arm, but multiple arms may be required. In this way, it is possible to overcome the limitations of single-arm systems, especially in the presence of large and/or heavy payloads [14].

Such a paradigm has been successfully applied to mobile manipulators as well (see, e.g., [18, 30–33]). Among the first attempts to use multiple aerial robots for cooperative transportation, it is worth mentioning the work reported in [34], where multiple quadrotors manipulate and transport a payload via cables. In [35], a team of quadrotors is rigidly attached to a payload via suitably designed grippers and a partially decentralized control law is adopted to ensure coordinate transportation of the payload, without considering internal stresses acting on the grasped object and interaction with the environment. In [36], cooperative manipulation of a rigid object via VTOL aerial vehicles is investigated. Here, a sliding mode motion controller is adopted to achieve absolute motion of the object, while internal forces are controlled via a virtual linkage approach, however, the interaction with the environment is not considered. In [37], the impedance scheme reported in [38] has been applied to a team of cooperative VTOL aerial vehicles moving a rigid object. More recently, an effective master-slave approach has been devised for the cooperative transportation of a payload via multiple aerial VTOL vehicles where the master moves the payload along the desired path, while the slave actively guarantees compliance to the master motion, [39]. Among the various solutions, the one based on cables, proposed in [40], is the most suitable for the mentioned applications, thanks to the induced decoupling between the rotational dynamics of the vehicles and of the load.

In this thesis, VTOL Unmanned Aerial Vehicles are considered, since they offer several advantages over standard mobile robots operating on ground in many application scenarios (e.g., inspection and manteinance services [41], data collection and exploration operations [42], precision agriculture [43]). More recently, aerial manipulators (i.e., UAVs equipped with grippers or robotic arms) opened up a new stream of applications, such as assembly of structures in remote environments, maintenaince operations and payload transportation [44, 45].

I.4 Contribution

In recent years, the scientific community has made many efforts in the field of aerial manipulation by using the common multirotor platforms. With the diffusion and technical progress of fully-actuated multirotors, many scientific studies have highlighted the benefits that such vehicles bring to aerial manipulation, such as the increased ability to interact with the environment. This capability is very important and widely used in modern robotics, since it allows cooperation between multiple robotic agents and the human-robot collaboration. The development of new robots that easily interact with the surrounding environment is therefore of great interest and motivated the design of the novel fully-actuated quadrotor proposed in this work.

As mentioned in section I.1, the most common fully-actuated aerial platforms have tilted or tilting rotors that amplify the aerodynamic perturbations between the propellers, reducing the efficiency and the provided thrust. In order to overcome this limitations a novel platform, the ODQuad (OmniDirectional Quadrotor), has been proposed, which is composed by three main parts (the platform, the mobile and rotor frames) that are linked by means of two rotational joints, namely the roll and pitch joints. The ODQuad is able to orient the total thrust by moving only the propellers frame by means of the roll and pitch joints. In this way aerodynamic perturbation, due to the rotors' air flow crossing, and internal forces, due to non-parallel thrusts, are avoided. These two advantages increase the energy efficiency of the vehicle, hence the flight time between two consecutive recharges.

As explained in Section I.3, multi-robot aerial manipulation is often performed by using multirotor platforms, which are usually under-actuated, connected by cables to the transported object. The cables are required for the following purposes:

- To confer compliance to the manipulation system;
- To decouple the rotational dynamics of the vehicles and the one of the load, since the interaction wrench between the robots and the object is composed only by the force along the cable.

In this thesis, an aerial manipulation system in which the manipulators are rigidly connected to the transported object is presented. This can only be achieved by using fully- actuated multirotors, as the ODQuad presented in this work.

In conclusion, the major contributions that this thesis work leads to aerial manipulation are listed as follow.

- The design of a novel fully-actuated platform, the ODQuad, conceived for manipulation tasks.
- A custom model-based control for fully-actuated platform, such as the ODQuad.
- A cooperative aerial manipulation system in which the robots rigidly grasp the transported object.

I.5 Outline

The thesis is organized as follows:

Chapter 1 reports the main solution adopted in the realization of fully-actuated multirotors, with emphasis on the new ODQuad concept. The kinematic and dynamic models is derived using the *Euler-Lagrange* approach.

Chapter 2 illustrates the designed model-based controller, which is based on two control loops: an outer loop for vehicle position control and an inner one for vehicle orientation and roll-pitch joint control. The effectiveness of the controller is tested by means of numerical simulations both in free motion and in object transportation tasks. In the latter test, a momentum observer is used to estimate the wrenches exchanged between the vehicle and the transported object.

Chapter 3 presents a simulation model, in which multiple multirotors are involved to perform cooperative aerial manipulation tasks. More in detail, the manipulation of a bulky object is considered, whose unknown inertial parameters are identified in the first task phase. In order to reduce the mechanical stresses due to the manipulation and increase the system robustness to the environment interactions, the admittance filters are implemented to each multirotor and to the transported object, respectively.

Chapter 4 describes the prototyping steps. Three CAD models have been designed, in particular, the first model has been used in the simulations and the other two for the prototype development. The static analysis that investigates the torque values for a rough sizing of the roll-pitch joint actuators is presented. Finally, the prototype is presented, that it is developed by integrating some components of the quadcopter *Holybro X500* in a custom manufactured mechanical frame.

Chapter 1

Omnidirectional Multirotor Vehicles

1.1 Aerial multirotor vehicles

1.1.1 Overview on multirotor

Multirotor aerial vehicles are equipped with four or more co-planar propellers, able to take off and land vertically and for this reason they are often refered as VTOL (Vertical Take Off and Landing) vehicles. This capability allows such vehicles to move in very narrow spaces and to maintain a certain altitude without moving (i.e., hovering). Multirotors, compared to helicopters, do not have the complicated swashplates and the linkages found in conventional designs, as they use variable rotor speeds to maneuver [46]. Due to the reduction of mechanical complexity and wear, a well-designed multirotor is proven inherently more robust and reliable. On the other hand, multirotors are characterized by a lower aerodynamic efficiency, since each propeller is influenced by the turbulence caused by the motion of the other propellers [47].

Multirotors are classified, according to the rotor configurations, in tricopters, quadcopters, hexacopters, and octocopters. Reliability-wise, quadcopters have no chance of landing if a motor fails, while a hexacopter can land but with a limited yaw control, and octocopters can fly and land with a single motor failure. The main parts of a VTOL are the rotors, i.e. propellers with their motors, the mechanical frame and the electronic system. Propellers are usually two-bladed, because they deliver more thrust than three or four-blade propellers, which are more affected by turbulence. Three or up bladed propellers are used in competition multirotors since the two-bladed ones, to provide the same thrust, should rotate at higher speeds or must be larger [48].

The frame is composed by a central part, that hosts the electronic system, the battery and external devices, e.g., a camera and/or a gripper, and a number of arms equal to the number of rotors. The electronic system includes the IMU (Inertial Measurement Unit) board, the ESCs (Electronic Speed Controller), which control the speed of the motors, and the datalink-telemetry devices. As mentioned above, multirotor are controlled by varying the rotational speed of the rotors, i.e. the thrust of the propellers. Since the vehicle has 6 degrees of freedom, it is necessary to have at least six independent actuators to realize three translational and three rotational motions, such as the vehicle can be defined fully actuated. The previously mentioned multirotors, i.e. those with parallel axis rotors, are defined underactuated vehicles, since they are not able to realize the 6 motions independently. The underactuation problem will be addressed in detail in the following section.

1.1.2 Underactuation problem

A classical multirotor vehicle, with coplanar rotors, is an under-actuated system, since thrusts generated by the propellers are directed along parallel axes, orthogonal to the plane of the rotors [11] (see yellow arrows in Fig. 1.1). For this reason the resultant thrust can only be directed in the same direction of the propeller axis. Two coordinate frames are defined as follows:

- Inertial coordinate frame, $\boldsymbol{\Sigma}\{O, x, y, z\}$;
- Body coordinate frame attached to the aerial platform, $\boldsymbol{\Sigma}_b\{O_b, x_b, y_b, z_b\}$;

where the relative orientation between $\boldsymbol{\Sigma}$ and Σ_b is expressed by the following roll-pitch-yaw rotation matrix

$$\boldsymbol{R}_b(\boldsymbol{\phi}) = \begin{bmatrix} c_\psi c_\theta & -c_\theta s_\psi & s_\theta \\ c_\phi s_\psi + c_\psi s_\phi s_\theta & c_\phi c_\psi - s_\phi s_\psi s_\theta & -c_\theta s_\phi \\ s_\phi s_\psi - c_\phi c_\psi s_\theta & c_\psi s_\phi + c_\phi s_\psi s_\theta & c_\phi c_\theta \end{bmatrix}, \tag{1.1}$$

where $\boldsymbol{\phi} = [\phi \quad \theta \quad \psi]^{\mathrm{T}}$ is the roll, pitch and yaw angles vector, $c_\phi = cos(\phi)$, $s_\phi = sin(\phi)$, $c_\theta = cos(\theta)$, $s_\theta = sin(\theta)$ and $c_\psi = cos(\psi)$, $s_\psi = sin(\psi)$.

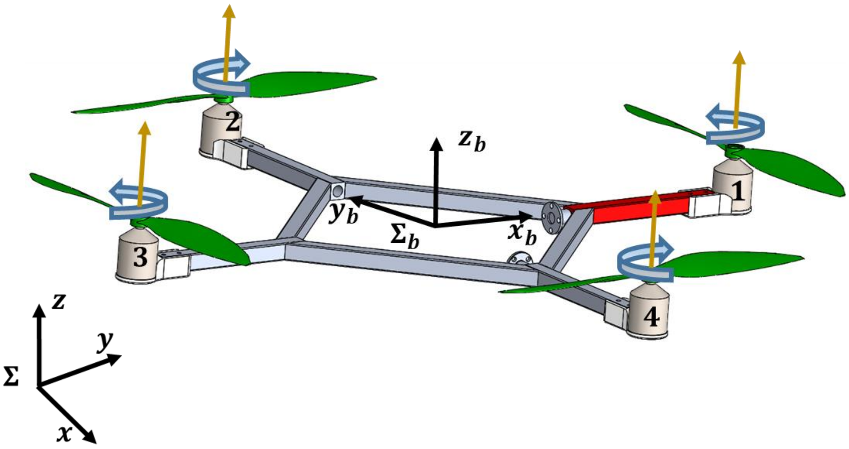


Figure 1.1: a) Schematic of a quadcopter with propeller direction, body reference frame and inertial reference frame.

In order to move the vehicle in the horizontal plane, it is necessary to project the total thrust in the $x - y$ plane by orienting the multirotor with specific roll and pitch angles. The total wrench that the propellers can exert on the vehicle, acting on O_b, expressed in $\boldsymbol{\Sigma}_b$ and defined by $[\boldsymbol{f}_b^{b\mathrm{T}} \;\; \boldsymbol{\mu}_b^{b\mathrm{T}}]^{\mathrm{T}}$, can be computed as follows

$$\begin{bmatrix} \|\boldsymbol{f}_b^b\| \\ \boldsymbol{\mu}_b^b \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ l & 0 & -l & 0 \\ 0 & -l & 0 & l \\ c & -c & c & -c \end{bmatrix} \begin{bmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \\ \sigma_4 \end{bmatrix} = \boldsymbol{N}\boldsymbol{\sigma}, \tag{1.2}$$

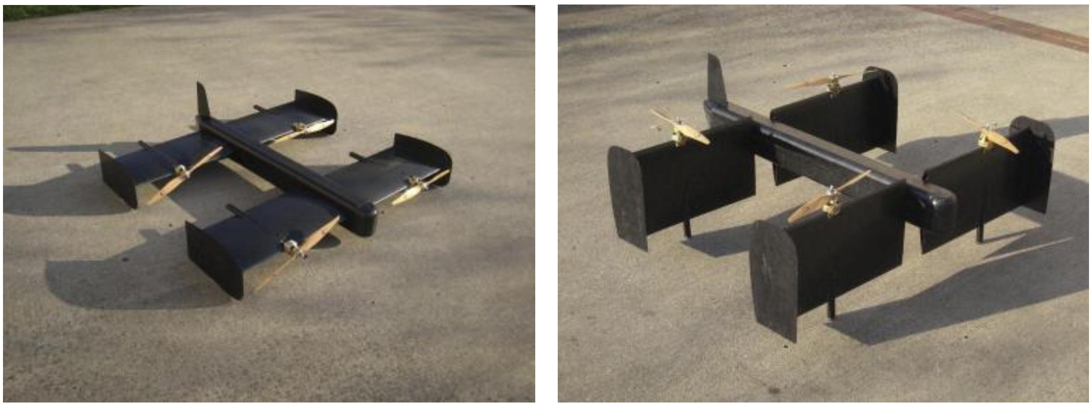
where $l > 0$ is the distance from each motor to the origin of $\boldsymbol{\Sigma}_b$, $c = \gamma_d/\gamma_t$, with γ_d and γ_t the drag and thrust coefficients, respectively, σ_*, $(* = 1, \dots, 4)$ is the propeller's thrust and the matrix $\boldsymbol{N}$ is named allocation matrix. It is worth noticing that the components of the forces in the $x_b - y_b$ plane, $f_{b,x}^b$ and $f_{b,y}^b$, are not generated by the thrust. Hence, when the vehicle is in hovering condition, i.e. $\boldsymbol{z}_b$ parallel with $\boldsymbol{z}$, also in the reference frame $\boldsymbol{\Sigma}$ there are not the f_x and f_y forces are non present. In fact, in order to change its position and/or counteract external disturbances, a rotation of vehicle body is required. The rotation of the vehicle body needed to obtain a given force vector $\boldsymbol{f}$ in $\boldsymbol{\Sigma}$ frame can be computed from the following relation

$$\boldsymbol{f}_b = \boldsymbol{R}_b(\boldsymbol{\phi})\boldsymbol{f}_b^b. \tag{1.3}$$

This limitation may become severe in some applications, e.g., manipulation tasks involving contact with the environment [8], and/or limit the disturbance rejection capabilities of the system.

1.2 Omnidirectional UAVs

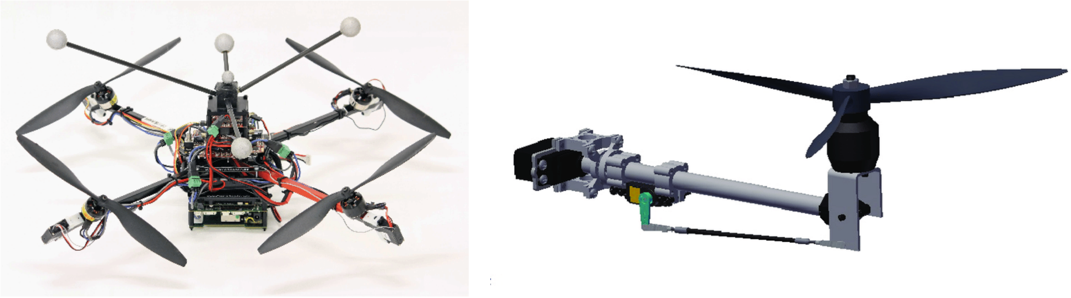
In order to overcome the underactuation problem, the interest of the research community has recently focused on omnidirectional platforms, which combine the advantages of existing multirotor systems with the agility and the maneuverability of fully actuated UAVs. One of the most common approaches is based on the use of tilted propellers. A tilt-wing mechanism has been proposed in [49] and [1] (Fig. 1.2), where a vehicle equipped with four wings, that can be rotated from vertical to horizontal position, is designed. Such a solution requires at least seven actuators, in order to decouple position and orientation control, thus resulting in an overactuated system. The main advantage of such vehicle is the cancellation of the vertical propellers' thrust component in a cruise flight (Fig.1.2(a)), since it is compensate by the aerodynamic lift generated by the wings.



(a) (b)

Figure 1.2: Rotating wing UAV, images from [1]. Cruise flight configuration (a), take-off and landing configuration (b).

Another overactuated quadrotor UAV with titling propellers has been proposed in [2], where 4 additional control inputs are adopted for tilting the propeller arms (Fig.1.3(a)), and in [3], where 4 motors are adopted for propellers rotation and 8 servomotors are used for tilting each propeller (Fig.1.3(b)).



(a) (b)

Figure 1.3: Overactuated prototype, image from [2] (a), Propeller tiltilg mechanism, image from [3] (b).

The adoption of overactuated platforms increases the system weight, due to additional motors, and the complexity of the control because of the need of handling both vertical and lateral air flows. To reduce the mechanical complexity and the required motors, hexarotors with tilted propellers have been proposed (Fig.1.4). In [4] each rotor is mounted in a fixed configuration rotated about two axes. The two angular parameters defining the rotor tilting are obtained via an optimization problem aimed at reducing the control effort magnitude. Adoption of an hexacopter in lieu of the quadrotor configuration allows to increase the system payload and the robustness to faults, at the expense of a slightly more complex mechanical structure. An eight-rotor configuration is adopted in [5] (Fig.1.5), where the rotors are mounted on a 3D mechanical frame.

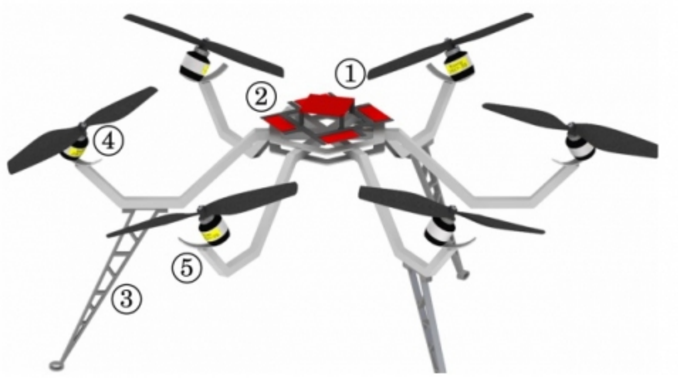


Figure 1.4: CAD model of the hexarotor with tilted propellers, image from [4]

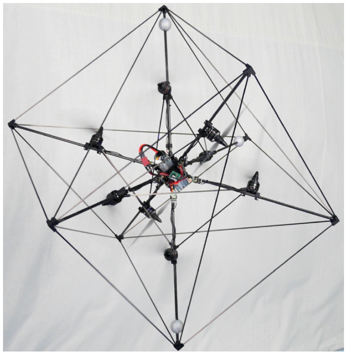


Figure 1.5: Prototype of the eigth-rotor UAV, image from [5]

To increase the efficiency, in [6], a new UAV concept is proposed (Fig. 1.6), with an additional motor aimed at tilting all propellers at the same time, where the tilting angle is computed by an high-level slow-rate controller. The use of a single motor reduces the energy consumption and the total mass, allowing to drive the platform from a configuration energetically efficient, but underactuated, to a fully actuated, but less efficient one.

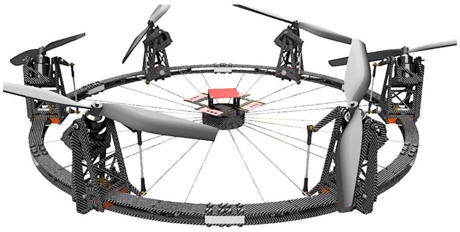


Figure 1.6: FAST-Hex concept, image from [6].

More recently, in [7], an hexarotor with independently tiltable rotors has been designed and experimentally tested (Fig. 1.7), featurig the tilting angle of the rotors is computed on line via a suitable control allocation scheme. This solution, however, is not able to avoid internal wrenches caused by counteracting actuators and uses redundant control inputs. By exploiting its bi-directional thrusters, this platform can apply forces in any direction, while, for the other platforms using uni-directional thrusters, the forces are usually limited in the upper hemi-sphere. An interesting analysis conducted in [50] proves that for uni-directional thrusters the minimum number of rotors that allow achieving omnidirectional forces is seven. The above analysis led to the platform in [51], which is equipped with seven propellers, whose tilting angles are computed according to the optimization procedure in [50], and guarantees the omnidirectional property of the platform, while enforcing equal sharing of the desired forces between the propellers.

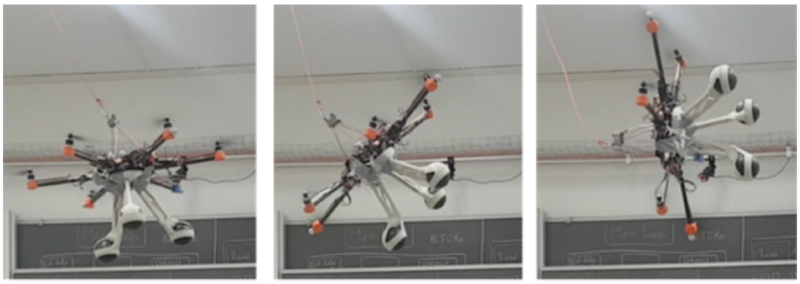


Figure 1.7: Voliro hexarotor, images from [7].

The previous solutions, do not avoid internal wrenches caused by counteracting actuators and require redundant control inputs. In order to overcome this problem, in [52], a quadrotor vehicle is designed, with the tilting angle of the propellers simultaneously controlled by two additional actuators through a parallelogram mechanism. In this way, only six inputs are adopted to control six DOFs and the thrust generated by all propellers is always aligned along the same direction, thus avoiding energy dissipation due to internal wrenches. A similar configuration, is proposed in this thesis, where a new concept of omnidirectional aerial vehicle, named ODQuad (Fig.1.8) is designed. A preliminary design has been presented in [52].

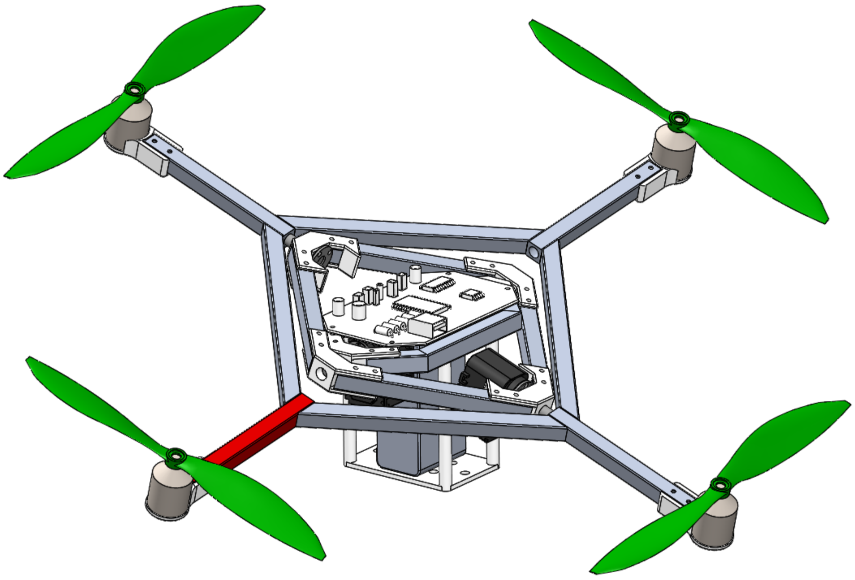


Figure 1.8: CAD model of the ODQuad.

1.3 ODQuad

The first part of this section focuses on the detailed description the omnidirectional vehicle previously introduced. Then, the mathematical model of the vehicle is developed, in particular, the kinematic and dynamic model. The ODQuad is able to fly in any position with any orientation, having the same main features of the solution presented in [52], i.e., the capability to avoid internal wrenches and redundant inputs. To this purpose, a novel mechanical design is adopted, where the rotors mounting frame is oriented via two rotational joints realizing a two-axis (roll and pitch) gimbal mechanism. In this way, the adoption of complex closed-chain mechanisms is avoided. Although the approach is developed for a quadrotor vehicle, it might be easily adapted to different multirotor configurations.

1.3.1 Mechanical design

The ODQuad vehicle is based on three mechanical frames:

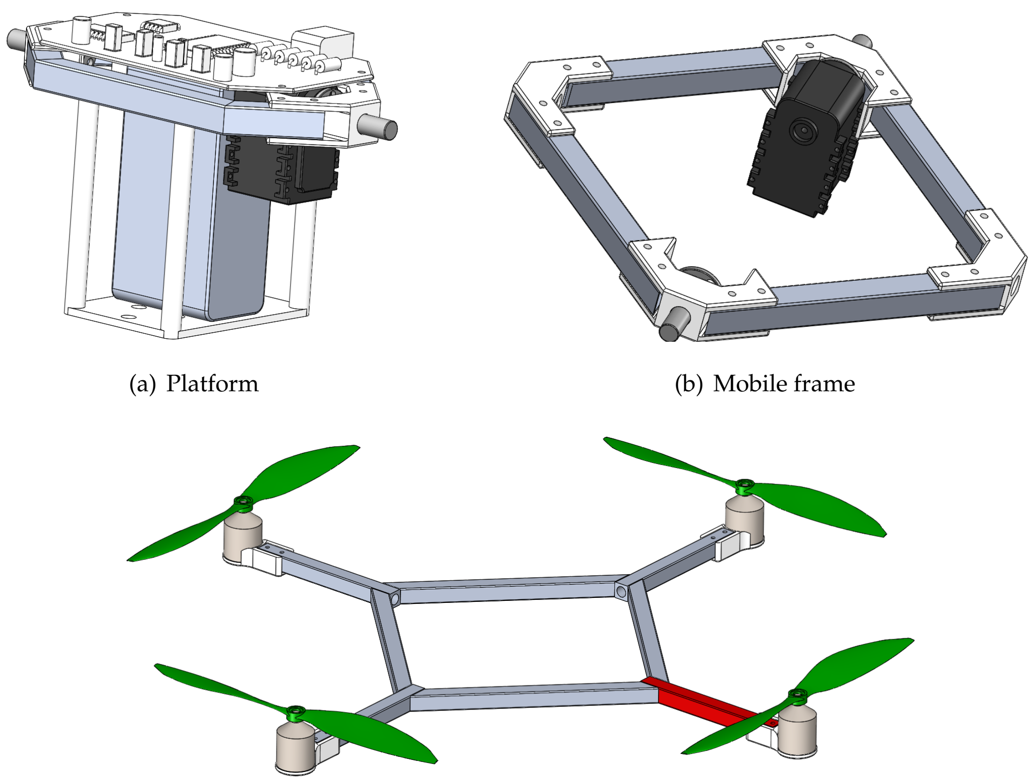
- A *platform*, Fig. 1.9(a), hosting the computing hardware, electronics, batteries and sensors;
- A *rotor frame*, Fig. 1.9(b), that supports the propellers;
- A *mobile frame*, Fig. 1.9(c), that connects the platform and the rotor frames.

The three components are connected by two rotational joint in a gimbal configuration. The *roll joint*, Fig.1.10(a)), connects the platform to the mobile frame, while the mobile frame and the rotor frame are connected via another rotational joint (*pitch joint*, Fig.1.10(b)), with axis orthogonal to the roll joint. Each joint is actuated by a servomotor.

During the motion, the vehicle modifies the orientation of the rotor frame to generate a thrust component along an arbitrary direction that complies with the mechanical limits imposed by the frames. In this way, the platform can assume a roll-pitch configuration independently from that of the rotor frame.

1.3.2 Kinematic modeling

In order to write the mathematical model of the ODQuad , in terms of kinematics and dynamics, the following relevant coordinate frames are defined:



(a) Platform

(b) Mobile frame

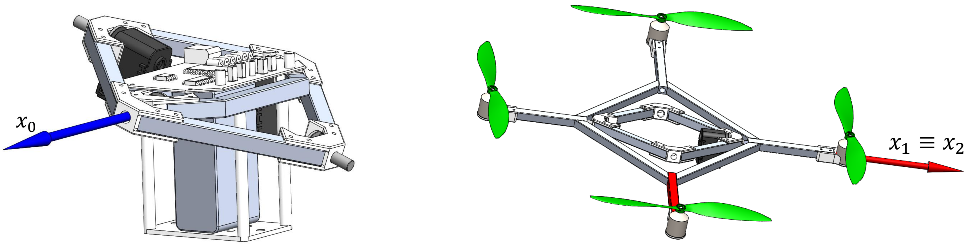
(c) Propellers' frame

Figure 1.9: ODQuad overall assembly (a), roll joint axis (b), pitch joint axis (c).

- The coordinate frame $\boldsymbol{\Sigma}_0\{O_0, x_0, y_0, z_0\}$, attached to the platform in such a way that the axis x_0 coincides with the axis of the roll joint; (Fig.1.10(a));
- The coordinate frame $\boldsymbol{\Sigma}_1\{O_1, x_1, y_1, z_1\}$, attached to mobile frame in such a way that the axis x_1 coincides with the axis of the pitch joint; (Fig.1.10(b));
- The coordinate frame $\boldsymbol{\Sigma}_2\{O_2, x_2, y_2, z_2\}$, attached to the rotor frame in such a way that the axis x_2 coincides with x_1.

$\boldsymbol{\Sigma}_0$, $\boldsymbol{\Sigma}_1$ and $\boldsymbol{\Sigma}_2$ are chosen such that $O_0 \equiv O_1 \equiv O_2$. The relative orientation between the coordinate frames is given by the following rotation matrices

$$\boldsymbol{R}_1^0(\alpha) = \begin{bmatrix} 0 & -1 & 0 \\ c_\alpha & 0 & -s_\alpha \\ s_\alpha & 0 & c_\alpha \end{bmatrix}, \tag{1.4}$$



(a) Roll joint

(b) Pitch joint

Figure 1.10: Roll joint axis (a), pitch joint axis (b).

$$\boldsymbol{R}_2^1(\beta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_\beta & -s_\beta \\ 0 & s_\beta & c_\beta \end{bmatrix}, \tag{1.5}$$

where α is the roll joint angle, β is the pitch joint angle, $c_\alpha = \cos\alpha$, $s_\alpha = \sin\alpha$, $c_\beta = \cos\beta$, $s_\beta = \sin\beta$ and $\boldsymbol{R}_i^j$ denotes the rotation matrix which describes the orientation of $\boldsymbol{\Sigma}_i$ with respect to $\boldsymbol{\Sigma}_j$.

The kinematic relations will be written considering the CoM (Center of Mass) of each part, denoted as C_* $(* = 0, 1, 2)$, and will be expressed in the inertial reference frame $\boldsymbol{\Sigma}$, by using the following generalized coordinates:

$$\boldsymbol{q} = \left[\boldsymbol{p}_0^{\mathrm{T}}\ \boldsymbol{\phi}_0^{\mathrm{T}}\ \boldsymbol{\delta}^{\mathrm{T}}\right]^{\mathrm{T}} \in \mathbb{R}^8, \tag{1.6}$$

where $\boldsymbol{p}_0$ is the position of the origin O_0 of the platform frame, $\boldsymbol{\Sigma}_0$, expressed in the inertial frame, $\boldsymbol{\Sigma}$, $\boldsymbol{\phi}_0 = [\varphi\ \vartheta\ \psi]^{\mathrm{T}}$ is the vector of roll-pitch-yaw angles representing the platform orientation with respect to the inertial frame, and $\boldsymbol{\delta} = [\alpha\ \beta]^{\mathrm{T}}$ collects the roll and pitch joint angles. The position of the center of mass C_0 of the platform can be expressed in the inertial frame as

$$\boldsymbol{p}_{C_0} = \boldsymbol{p}_0 + \boldsymbol{R}_0 \boldsymbol{r}_{0,C_0}^0, \tag{1.7}$$

where $\boldsymbol{R}_0$ is the rotation matrix expressing the orientation of $\boldsymbol{\Sigma}_0$ with respect to $\boldsymbol{\Sigma}$, $\boldsymbol{r}_{0,C_0}^0$ is the position vector of C_0 with respect to $\boldsymbol{\Sigma}_0$, expressed in $\boldsymbol{\Sigma}_0$. The angular and linear velocities of C_0 are given by

$$\begin{cases} \boldsymbol{\omega}_{C_0} = \boldsymbol{\omega}_0 \\ \dot{\boldsymbol{p}}_{C_0} = \dot{\boldsymbol{p}}_0 + \boldsymbol{S}(\boldsymbol{\omega}_0)\boldsymbol{R}_0\boldsymbol{r}_{0,C_0}^0 = \dot{\boldsymbol{p}}_0 - \boldsymbol{S}(\boldsymbol{r}_{0,C_0})\boldsymbol{\omega}_0, \end{cases} \tag{1.8}$$

where $\boldsymbol{\omega}_0$ is the angular velocity of frame $\boldsymbol{\Sigma}_0$ with respect to $\boldsymbol{\Sigma}$ and $\boldsymbol{S}(\cdot)$ is the skew symmetric operator performing the cross product [53]. Velocities in (1.8) can be expressed in terms of the generalized coordinates as

$$\begin{cases} \boldsymbol{\omega}_{C_0} & = & \boldsymbol{J}_{O_0}\dot{\boldsymbol{q}} \\ \dot{\boldsymbol{p}}_0 & = & \boldsymbol{J}_{P_0}\dot{\boldsymbol{q}} \end{cases} \tag{1.9}$$

where $\boldsymbol{J}_{P_0} \in \mathbb{R}^{3\times 8}$ and $\boldsymbol{J}_{O_0} \in \mathbb{R}^{3\times 8}$ are the position and orientation Jacobian matrices, respectively, given by

$$\boldsymbol{J}_{O_0} = [\boldsymbol{O}_3 \;\; \boldsymbol{T}(\boldsymbol{\phi}_0) \;\; \mathbf{0}_3 \;\; \mathbf{0}_3], \tag{1.10}$$

$$\boldsymbol{J}_{P_0} = [\boldsymbol{I}_3 \;\; -\boldsymbol{S}(\boldsymbol{r}_{0,C_0})\boldsymbol{T}(\boldsymbol{\phi}_0) \;\; \mathbf{0}_3 \;\; \mathbf{0}_3], \tag{1.11}$$

where $\boldsymbol{I}_n$ and $\boldsymbol{O}_n$ are the $(n \times n)$ identity and null matrices, respectively, $\mathbf{0}_n$ is $(n \times 1)$ null vector, while the matrix $\boldsymbol{T}(\phi_0)$, defined as

$$\boldsymbol{T}(\boldsymbol{\phi}_0) = \begin{bmatrix} 1 & 0 & s_\theta \\ 0 & c_\phi & -c_\theta s_\phi \\ 0 & s_\phi & c_\phi c_\theta \end{bmatrix}, \tag{1.12}$$

characterizes the relation between the angular velocity $\boldsymbol{\omega}_0$ and the derivative of the roll-pitch-yaw angles $\dot{\boldsymbol{\phi}}_0$, $c_\phi = \cos\phi$, $s_\phi = \sin\phi$, $c_\theta = \cos\theta$ and $s_\theta = \sin\theta$. Regarding the mobile frame, the position of its center of mass C_1 is given by

$$\boldsymbol{p}_{C_1} = \boldsymbol{p}_0 + \boldsymbol{R}_1\boldsymbol{r}^1_{1,C_1}, \tag{1.13}$$

where it has been considered $\boldsymbol{p}_1 = \boldsymbol{p}_0$, since $O_0 \equiv O_1$ and $\boldsymbol{r}^1_{1,C_1}$ is the position of C_1 with respect to $\boldsymbol{\Sigma}_1$ expressed in frame $\boldsymbol{\Sigma}_1$, while the rotation matrix $\boldsymbol{R}_1 = \boldsymbol{R}_0\boldsymbol{R}^0_1$ expresses the orientation of $\boldsymbol{\Sigma}_1$ with respect to $\boldsymbol{\Sigma}$. Also in this case, linear and angular velocities of the center of mass can be expressed in terms of generalized coordinates as

$$\begin{cases} \boldsymbol{\omega}_{C_1} = \boldsymbol{\omega}_1 = \boldsymbol{\omega}_0 + \dot{\alpha}\boldsymbol{x}_0 = \boldsymbol{J}_{O_1}\dot{\boldsymbol{q}} \\ \dot{\boldsymbol{p}}_{C_1} = \dot{\boldsymbol{p}}_0 - \boldsymbol{S}(\boldsymbol{r}_{1,C_1})\boldsymbol{\omega}_1 = \dot{\boldsymbol{p}}_0 - \boldsymbol{S}(\boldsymbol{r}_{1,C_1})\boldsymbol{\omega}_{C_1} = (\boldsymbol{J}_{P_0} - \boldsymbol{S}(\boldsymbol{r}_{1,C_1})\boldsymbol{J}_{O_1})\,\dot{\boldsymbol{q}} = \boldsymbol{J}_{P_1}\dot{\boldsymbol{q}} \end{cases} \tag{1.14}$$

where $\boldsymbol{x}_0$ is the first column of $\boldsymbol{R}_0$, $\boldsymbol{J}_{P_1} \in \mathbb{R}^{3\times 8}$ and $\boldsymbol{J}_{O_1} \in \mathbb{R}^{3\times 8}$ are the position and orientation Jacobian matrices of the mobile frame, respectively, given by

$$\boldsymbol{J}_{O_1} = [\boldsymbol{O}_3 \;\; \boldsymbol{T}(\boldsymbol{\phi}_0) \;\; \boldsymbol{x}_0 \;\; \mathbf{0}_3] \tag{1.15}$$

$$\boldsymbol{J}_{P_1} = [\boldsymbol{I}_3 \;\; -\boldsymbol{S}(\boldsymbol{r}_{1,C_1})\boldsymbol{T}(\boldsymbol{\phi}_0) \;\; -\boldsymbol{S}(\boldsymbol{r}_{1,C_1})\boldsymbol{x}_0 \;\; \mathbf{0}_3]. \tag{1.16}$$

As regards the orientable rotor frame, the position of its center of mass, C_2, can be expressed in the inertial frame as

$$\boldsymbol{p}_{C_2} = \boldsymbol{p}_1 + \boldsymbol{R}_2\boldsymbol{r}^2_{2,C_2}, \tag{1.17}$$

where $\boldsymbol{r}^2_{2,C_2}$ is the position of C_2 with respect to $\boldsymbol{\Sigma}_2$ expressed in $\boldsymbol{\Sigma}_2$, while $\boldsymbol{R}_2 = \boldsymbol{R}_1\boldsymbol{R}^1_2$ is the rotation matrix of the rotor frame. Also in this case it has been taken into account that $O_1 \equiv O_2$ and thus $\boldsymbol{p}_2 = \boldsymbol{p}_1$. The linear and angular velocities of the center of mass, after some straightforward mathematical steps, can be written as

$$\begin{cases} \boldsymbol{\omega}_{C_2} = \boldsymbol{\omega}_2 = \boldsymbol{\omega}_1 + \dot{\beta}\boldsymbol{x}_1 = \boldsymbol{\omega}_0 + \dot{\alpha}\boldsymbol{x}_0 + \dot{\beta}\boldsymbol{x}_1 = \boldsymbol{J}_{O_2}\dot{\boldsymbol{q}} & (1.18) \\ \dot{\boldsymbol{p}}_{C_2} = \dot{\boldsymbol{p}}_1 - \boldsymbol{S}(\boldsymbol{r}_{2,C_2})\boldsymbol{\omega}_2 = \dot{\boldsymbol{p}}_1 - \boldsymbol{S}(\boldsymbol{r}_{2,C_2})\boldsymbol{\omega}_{C_2} = \left(\boldsymbol{J}_{P_1} - \boldsymbol{S}(\boldsymbol{r}_{2,C_2})\boldsymbol{J}_{O_2}\right)\dot{\boldsymbol{q}} = \boldsymbol{J}_{P_2}\dot{\boldsymbol{q}} \end{cases}$$

where $\boldsymbol{x}_1$ is the first column of $\boldsymbol{R}_1$, $\boldsymbol{J}_{P_2} \in \mathbb{R}^{3\times 8}$ and $\boldsymbol{J}_{O_2} \in \mathbb{R}^{3\times 8}$ are the position and orientation Jacobian matrices, respectively, of the rotor frame given by

$$\boldsymbol{J}_{P_2} = [\boldsymbol{I}_3 \; -\boldsymbol{S}(\boldsymbol{r}_{2,C_2})\boldsymbol{T}(\boldsymbol{\phi}_0) \;\; -\boldsymbol{S}(\boldsymbol{r}_{2,C_2})\boldsymbol{x}_0 \;\; -\boldsymbol{S}(\boldsymbol{r}_{2,C_2})\boldsymbol{x}_1] \tag{1.19}$$

$$\boldsymbol{J}_{O_2} = [\boldsymbol{O}_3 \; \boldsymbol{T}(\boldsymbol{\phi}_0) \; \boldsymbol{x}_0 \; \boldsymbol{x}_1]\,. \tag{1.20}$$

1.3.3 Dynamic modeling

The dynamic model of the system can be derived by considering the *Euler-Lagrange formulation* [53], in which the mechanical system is characterized by the function

$$\mathcal{L} = \mathcal{T} - \mathcal{U}, \tag{1.21}$$

where $\mathcal{T}$ and $\mathcal{U}$ denote the system total kinetic and potential energy, respectively. The Lagrange equations are given by ($i = 1, \dots, 8$)

$$\frac{d}{dt}\frac{\partial\mathcal{L}}{\partial\dot{q}_i} - \frac{\partial\mathcal{L}}{\partial q_i} = \zeta_i, \tag{1.22}$$

where q_i is the i-th generalized Lagrange coordinate of the system and ζ_i is the i-th associated generalized force. The Lagrange coordinates are chosen equal to the generalized coordinates used in the kinematic formulation $\boldsymbol{q} = \left[\boldsymbol{p}_0^{\mathrm{T}} \; \boldsymbol{\phi}_0^{\mathrm{T}} \; \boldsymbol{\delta}^{\mathrm{T}}\right]^{\mathrm{T}} \in \mathbb{R}^8$.

Kinetic energy

The kinetic energy $\mathcal{T}$ for the ith mechanical frame ($i = 0, 1, 2$) can be derived as

$$\mathcal{T}_i = \frac{1}{2}m_i\dot{\boldsymbol{q}}^{\mathrm{T}}\boldsymbol{J}^{\mathrm{T}}_{P_i}\boldsymbol{J}_{P_i}\dot{\boldsymbol{q}} + \frac{1}{2}\dot{\boldsymbol{q}}^{\mathrm{T}}\boldsymbol{J}^{\mathrm{T}}_{O_i}\boldsymbol{M}_i\boldsymbol{J}_{O_i}\dot{\boldsymbol{q}}, \tag{1.23}$$

where m_i is the mass of the component, $\boldsymbol{M}_i = \boldsymbol{R}_i\boldsymbol{M}^i_i\boldsymbol{R}^{\mathrm{T}}_i$ is the inertia tensor expressed in the inertial frame, while $\boldsymbol{M}^i_i$ is the inertia tensor expressed in the attached frame $\boldsymbol{\Sigma}_i$.

Potential energy

The potential energy $\mathcal{U}$ for the ith mechanical frame ($i = 0, 1, 2$) can be derived as

$$\mathcal{U}_i = m_i \boldsymbol{g}^{\mathrm{T}} \boldsymbol{p}_{C_i}, \tag{1.24}$$

where $\boldsymbol{g}$ is the gravitational acceleration vector.

Dynamic model

By virtue of (1.21), (1.23) and (1.24) the dynamic model can be written in compact form as

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{F}\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{\zeta}, \tag{1.25}$$

where $\boldsymbol{M}(\boldsymbol{q}) \in \mathbb{R}^{8\times 8}$ is the inertia matrix of the whole system given by

$$\boldsymbol{M}(\boldsymbol{q}) = \sum_{i=0}^{2} (m_i \boldsymbol{J}_{P_i}^{\mathrm{T}} \boldsymbol{J}_{P_i} + \boldsymbol{J}_{O_i}^{\mathrm{T}} \boldsymbol{M}_i \boldsymbol{J}_{O_i}), \tag{1.26}$$

$\boldsymbol{C}(\boldsymbol{q}) \in \mathbb{R}^{8\times 8}$ represents the Coriolis and centrifugal term and is given by

$$\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} = \dot{\boldsymbol{M}}(\boldsymbol{q})\dot{\boldsymbol{q}} - \left[\frac{\partial}{\partial \boldsymbol{q}}\left(\frac{1}{2}\dot{\boldsymbol{q}}^{\mathrm{T}} \boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{q}}\right)\right]^{\mathrm{T}}, \tag{1.27}$$

$\boldsymbol{g}(\boldsymbol{q}) \in \mathbb{R}^{8}$ contains the generalized gravitational forces and can be written as

$$\boldsymbol{g}(\boldsymbol{q}) = \left(\frac{\partial \mathcal{U}(\boldsymbol{q})}{\partial \boldsymbol{q}}\right)^{\mathrm{T}} = -\sum_{i=0}^{2} m_i \boldsymbol{g}^{\mathrm{T}} \boldsymbol{J}_{P_i}. \tag{1.28}$$

$\boldsymbol{\zeta} \in \mathbb{R}^{8}$ collects the forces associated to the generalized coordinates $\boldsymbol{q}$, finally, the term $\boldsymbol{F}\dot{\boldsymbol{q}} \in \mathbb{R}^{8}$ models friction forces, as the roll-pitch joint friction and aerodynamics disturbances at low velocity (otherwise it would not be modeled with a simple linear relation).

Remark 1. Friction terms have been reported for completeness in the vehicle dynamic, however, they will be neglected thereafter, since these terms are considerably smaller than the other dynamic terms in (1.25) and they are difficult to model. Usually, the friction terms must be estimated from experimental data that are not available yet.

The matrices introduced in (1.25) can be detailed as bloc matrices

$$\boldsymbol{M}(\boldsymbol{q}) = \begin{bmatrix} \boldsymbol{M}_{pp} & \boldsymbol{M}_{p\phi} & \boldsymbol{M}_{p\delta} \\ \boldsymbol{M}_{p\phi}{}^{\mathrm{T}} & \boldsymbol{M}_{\phi\phi} & \boldsymbol{M}_{\phi\delta} \\ \boldsymbol{M}_{p\delta}{}^{\mathrm{T}} & \boldsymbol{M}_{\phi\delta}{}^{\mathrm{T}} & \boldsymbol{M}_{\delta\delta} \end{bmatrix},$$

where $\boldsymbol{M}_{pp} \in \mathbb{R}^{3\times3}$, $\boldsymbol{M}_{p\phi} \in \mathbb{R}^{3\times3}$, $\boldsymbol{M}_{p\delta} \in \mathbb{R}^{3\times2}$, $\boldsymbol{M}_{\phi\phi} \in \mathbb{R}^{3\times3}$, $\boldsymbol{M}_{\phi\delta} \in \mathbb{R}^{3\times2}$ and $\boldsymbol{M}_{\delta\delta} \in \mathbb{R}^{2\times2}$, and

$$\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}}) = \begin{bmatrix} \boldsymbol{C}_{\boldsymbol{p}} \\ \boldsymbol{C}_{\boldsymbol{\phi}} \\ \boldsymbol{C}_{\boldsymbol{\delta}} \end{bmatrix}, \boldsymbol{G}(\boldsymbol{q}) = \begin{bmatrix} \boldsymbol{g}_{\boldsymbol{p}} \\ \boldsymbol{g}_{\boldsymbol{\phi}} \\ \boldsymbol{g}_{\boldsymbol{\delta}} \end{bmatrix}, \boldsymbol{\zeta} = \begin{bmatrix} \boldsymbol{f}_0 \\ \boldsymbol{\mu}_0 \\ \boldsymbol{\tau}_\delta \end{bmatrix},$$

where $\boldsymbol{C}_{\boldsymbol{p}} \in \mathbb{R}^{3\times8}$, $\boldsymbol{C}_{\boldsymbol{\phi}} \in \mathbb{R}^{3\times8}$, $\boldsymbol{C}_{\boldsymbol{\delta}} \in \mathbb{R}^{2\times8}$, $\boldsymbol{g}_{\boldsymbol{p}} \in \mathbb{R}^{3}$, $\boldsymbol{g}_{\boldsymbol{\phi}} \in \mathbb{R}^{3}$, $\boldsymbol{g}_{\boldsymbol{\delta}} \in \mathbb{R}^{2}$,$\boldsymbol{f}_0 \in \mathbb{R}^{3}$, $\boldsymbol{\mu}_0 \in \mathbb{R}^{3}$, $\boldsymbol{\tau}_\delta \in \mathbb{R}^{2}$. The block decomposition allows to rearrange the dynamic model (1.25) as

$$\boldsymbol{M}_{pp}\ddot{\boldsymbol{p}}_0 + \boldsymbol{M}_{p\phi}\ddot{\boldsymbol{\phi}}_0 + \boldsymbol{M}_{p\delta}\ddot{\boldsymbol{\delta}} + \boldsymbol{C}_p\dot{\boldsymbol{p}}_0 + \boldsymbol{g}_p = \boldsymbol{f}_0 \tag{1.29}$$

$$\boldsymbol{M}_{p\phi}^{\mathrm{T}}\ddot{\boldsymbol{p}}_0 + \boldsymbol{M}_{\phi\phi}\ddot{\boldsymbol{\phi}}_0 + \boldsymbol{M}_{\phi\delta}\ddot{\boldsymbol{\delta}} + \boldsymbol{C}_\phi\dot{\boldsymbol{\phi}}_0 + \boldsymbol{g}_\phi = \boldsymbol{\mu}_0 \tag{1.30}$$

$$\boldsymbol{M}_{p\delta}^{\mathrm{T}}\ddot{\boldsymbol{p}}_0 + \boldsymbol{M}_{\phi\delta}^{\mathrm{T}}\ddot{\boldsymbol{\phi}}_0 + \boldsymbol{M}_{\delta\delta}\ddot{\boldsymbol{\delta}} + \boldsymbol{C}_\delta\dot{\boldsymbol{\delta}} + \boldsymbol{g}_\delta = \boldsymbol{\tau}_\delta, \tag{1.31}$$

where the matrix dependencies have been dropped for notation compactness, $\boldsymbol{f}_0$ and $\boldsymbol{\mu}_0$ are the force and moment, respectively, acting on the platform and $\boldsymbol{\tau}_\delta \in \mathbb{R}^2$ are the torques acting on the roll-pitch joint. The generalized force $\boldsymbol{\zeta}$ is given by the sum of the actuation force $\boldsymbol{\zeta}_c$ and the generalized force due to the propeller thrust

$$\boldsymbol{\zeta} = \boldsymbol{\zeta}_c + \boldsymbol{J}_{2,0}^{\mathrm{T}}\boldsymbol{h}_b = \boldsymbol{\zeta}_c + \boldsymbol{\zeta}_{b,2}, \tag{1.32}$$

where $\boldsymbol{J}_{2,0} = \begin{bmatrix} \boldsymbol{J}_{P_{2,0}}^{\mathrm{T}} & \boldsymbol{J}_{O_{2,0}}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}$ is the Jacobian matrix that relates the velocities and wrench between $\boldsymbol{\Sigma}_2$ and $\boldsymbol{\Sigma}_0$ and $\boldsymbol{h}_b = [\boldsymbol{f}_b^{\mathrm{T}} \ \boldsymbol{\mu}_b^{\mathrm{T}}]^{\mathrm{T}}$ are the wrench due to the propeller thrusts, expressed in $\boldsymbol{\Sigma}$. More in detail, $\boldsymbol{J}_{2,0}$ is given by

$$\boldsymbol{J}_{P_{2,0}} = [\boldsymbol{I}_3 \; -\boldsymbol{S}(\boldsymbol{r}_{2,0})\boldsymbol{T}(\boldsymbol{\phi}_0) \; -\boldsymbol{S}(\boldsymbol{r}_{2,0})\boldsymbol{x}_0 \; -\boldsymbol{S}(\boldsymbol{r}_{2,0})\boldsymbol{x}_1], \tag{1.33}$$

$$\boldsymbol{J}_{O_{2,0}} = [\boldsymbol{O}_3 \; \boldsymbol{T}(\boldsymbol{\phi}_0) \; \boldsymbol{x}_0 \; \boldsymbol{x}_1], \tag{1.34}$$

where the vector $\boldsymbol{r}_{20}$ represents the position of O_2 with respect to $\boldsymbol{\Sigma}_0$ expressed in $\boldsymbol{\Sigma}$. Since it has been assumed that $O_0 \equiv O_2$, $\boldsymbol{r}_{20}$ is a null vector, thus the following relation can be written

$$\boldsymbol{J}_{2,0} = \begin{bmatrix} \boldsymbol{I}_3 & \boldsymbol{O}_3 & \boldsymbol{0}_3 & \boldsymbol{0}_3 \\ \boldsymbol{O}_3 & \boldsymbol{T}(\boldsymbol{\phi}_0) & \boldsymbol{x}_0 & \boldsymbol{x}_1 \end{bmatrix}. \tag{1.35}$$

By exploiting the (1.35), $\boldsymbol{\zeta}_{b,2}$ can be written in a explicit form as

$$\boldsymbol{\zeta}_{b,2} = \begin{bmatrix} \boldsymbol{f}_{b,2} \\ \boldsymbol{\mu}_{b,2} \\ \boldsymbol{\tau}_{b,2} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}_b \\ \boldsymbol{T}^{\mathrm{T}}(\phi_0)\boldsymbol{\mu}_b \\ \begin{bmatrix} \boldsymbol{x}_0^{\mathrm{T}} \\ \boldsymbol{x}_1^{\mathrm{T}} \end{bmatrix} \boldsymbol{\mu}_b \end{bmatrix}.$$

Regarding the actuation force it is given by $\boldsymbol{\zeta}_c = \left[\mathbf{0}_3^{\mathrm{T}}\ \mathbf{0}_3^{\mathrm{T}}\ \boldsymbol{\tau}_c{}^{\mathrm{T}}\right]$, where $\boldsymbol{\tau}_c$ is the vector that collects the roll-pitch joints actuation torques. The torque vector $\boldsymbol{\tau}_\delta$ acting at the joints, according to (1.32), is given by the sum of the actuation torque vector and the effect of the thrusts on the joints

$$\boldsymbol{\tau}_\delta = \boldsymbol{\tau}_c + \boldsymbol{\tau}_{b,2}. \tag{1.36}$$

Finally, the propellers thrust can be computed via (1.2), which can be rewritten as

$$\begin{bmatrix} \|f_{b,2}\| \\ \boldsymbol{\mu}_{b,2}^2 \end{bmatrix} = \boldsymbol{N}\boldsymbol{\sigma}\,, \tag{1.37}$$

where $\boldsymbol{\mu}_{b,2}^2$ is the vector $\boldsymbol{\mu}_{b,2}$ expressed in the frame $\boldsymbol{\Sigma}_2$.

1.4 Conclusion

In this Chapter the under actuation problem of the common multirotors aerial platform has been explained. The main aerial platforms that overcame this problem, namely the fully-actuated multitoros, have been presented, and, in particular, the novel ODQuad concept has been introduced. The proposed mechanism is composed with three main parts arranged in a gimbal configuration, its kinematic and the dynamic models have been derived, latter model has been written by using Euler-Lagrange approach. It is worth remarking that the designed system is conceived to provide a fully controllable platform for aerial manipulators, i.e., UAVs carrying a robotic arm to execute manipulation tasks, even in contact with the external environment. Therefore, the ODQuad is not designed to execute aggressive maneuvers and/or reach extreme orientations (e.g., putting the aerial platform in a vertical configuration, as in [7]).

Chapter 2

Motion control

For classical underactuated multi-rotors, several controllers have been proposed, mostly based on linear approaches [54, 55], often requiring model linearization about a set of equilibrium conditions. Such controllers allow to achieve satisfactory performance near the design conditions and in hovering conditions, but performance degradation is usually observed when the aircraft moves away from these conditions. To overcome these limitations, nonlinear controllers have been designed, including model predictive control [56], backstepping and sliding mode techniques [57,58], adaptive control [59], impedance control [60, 61] and geometric control [62]. More recently, data-based control approaches have been designed for achieving high-performance trajectory tracking with UAVs: in detail an innovative Iterative Learning Control algorithm is proposed in [63], based on an estimation of repetitive disturbances to improve the learning performance.

Regarding the omnidirectional aerial platforms, the most common control approach is the feedback linearization, aimed at decoupling the six DOFs, that can be controlled separately. Such approaches are model-based, the nonlinear dynamical effects are compensated and then the thrust is derived by the inversion of the control allocation matrix. Some examples of feedback linearization approaches can be found in [4,52,64]. One of the main issues occurring in feedback linearization is that the actuator saturation can destabilize the system. Solutions to this limitations have been proposed in [65], [66] and [67]. In [68], a general geometric tracking controller for time-varying references is proposed, that takes into account also the bounds on the lateral control force.

In this thesis, a motion controller for the ODQuad is developed, based on a model-based feedback linearization approach. Such a controller is based on a hierarchical two-loop scheme, where the outer loop is in charge of computing the vehicle total thrust and the reference signals for the roll and pitch joints, while the inner loop computes the moments acting on the ODQuad platform and the joint torques. The controller stability and the effectiveness of the proposed control scheme, tested both in free-space motion and in the presence of external disturbances, such as those arising when the robot comes in contact with an object during a task of manipulation and transportation, will be presented in the following sections.

2.1 Motion Control

The proposed motion control is inherited from the approach developed in [45] for aerial manipulators. It is an inverse dynamic controller based on a two-loop approach: the outer loop, by means of the *position controller*, tracks the platform position, while the inner loop controls the platforms orientation and the roll-pitch joint position through the *Attitude* and *Roll-pitch controller*, respectively. A sketch of the controller block scheme is reported in Fig. 2.3.

More in details, the *position controller* is in charge of computing the force, $\boldsymbol{f}_2^2$, to be exerted on the vehicle platform in order to achieve the desired vehicle position trajectory, and the reference angles, α_{des} and β_{des}, for the roll-pitch joints. The reference angles are fed to the *inner loop*, that computes, on the basis of the platform desired orientation, both the torques $\boldsymbol{\tau}_c$ for the roll-pitch joint actuators and the moment $\boldsymbol{\mu}_2^2$. Then, the rotor thrusts are obtained by the inversion of equation (1.37). An offline planner must provide the desired trajectory for position, $\boldsymbol{p}_{0,d}(t)$, and the orientation of the frame $\boldsymbol{\Sigma}_0$, $\boldsymbol{\phi}_{0,d}(t)$, together with the corresponding desired linear and angular velocities, $\dot{\boldsymbol{p}}_{0,d}(t)$ and $\dot{\boldsymbol{\phi}}_{0,d}(t)$ and acceleration, $\ddot{\boldsymbol{p}}_{0,d}(t)$ and $\ddot{\boldsymbol{\phi}}_{0,d}(t)$.

Remark 2. It is worth noticing that the ODQuad, differently from usual underactuated quadrotors, allows the control of both position and orientation, provided that the assigned orientation, $\phi_{0,d}$, can be reached without violation of the limit of roll and pitch angles of the gimbal mechanism (i.e. $\alpha, \beta \in [-25,\ +25]$ deg for the current design).

2.1.1 Outer loop: position controller

The outer control loop is designed to track the assigned position by introducing a virtual control vector aimed at linearizing the position dynamics. In order to design the control

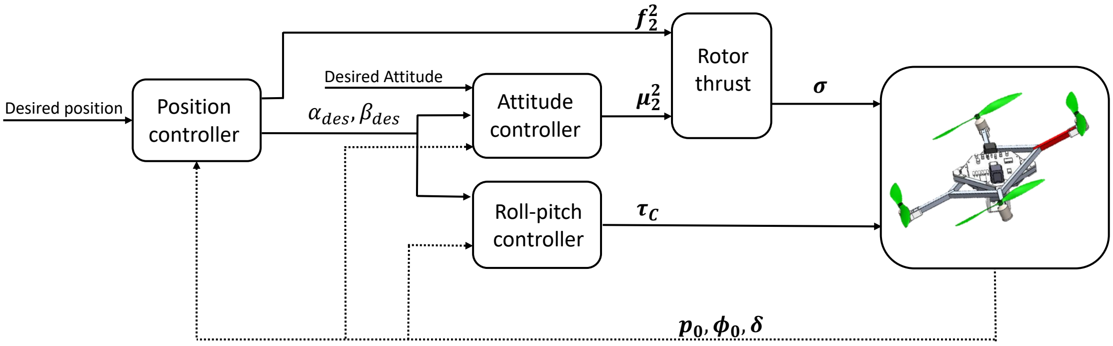


Figure 2.1: Block scheme of the overall control architecture, including the two loop controller and the propeller thrust mapping.

law, it is worth rewriting equation (1.29) as

$$\boldsymbol{M}_{pp}\ddot{\boldsymbol{p}}_0 + \boldsymbol{M}_{p\phi}\ddot{\boldsymbol{\phi}}_0 + \boldsymbol{C}_p\dot{\boldsymbol{p}}_0 + \boldsymbol{g}_p = \boldsymbol{f}_0, \tag{2.1}$$

where it has been assumed $\boldsymbol{M}_{p\delta} \simeq \boldsymbol{O}_{3\times 2}$. This assumption allows to decouple the inner and the outer loops. Hence, on the basis of (2.1), the force to be exerted on the platform is computed as

$$\boldsymbol{f}_0 = \boldsymbol{M}_{pp}\boldsymbol{a}_p + \boldsymbol{M}_{p\phi}\boldsymbol{a}_\phi + \boldsymbol{C}_p\dot{\boldsymbol{p}}_0 + \boldsymbol{g}_p, \tag{2.2}$$

where the auxiliary inputs $\boldsymbol{a}_*$ $(* = p, \phi)$ are defined as

$$\boldsymbol{a}_p = \ddot{\boldsymbol{p}}_{0,d} + \boldsymbol{K}_{D,p}\left(\dot{\boldsymbol{p}}_{0,d} - \dot{\boldsymbol{p}}_0\right) + \boldsymbol{K}_{P,p}\left(\boldsymbol{p}_{0,d} - \boldsymbol{p}_0\right) + \boldsymbol{K}_{I,p}\int_0^t \left(\boldsymbol{p}_{0,d} - \boldsymbol{p}_0\right) d\rho, \tag{2.3}$$

$$\boldsymbol{a}_\phi = \ddot{\boldsymbol{\phi}}_{0,d} + \boldsymbol{K}_{D,\phi}\left(\dot{\boldsymbol{\phi}}_{0,d} - \dot{\boldsymbol{\phi}}_0\right) + \boldsymbol{K}_{P,\phi}\left(\boldsymbol{\phi}_{0,d} - \boldsymbol{\phi}_0\right) + \boldsymbol{K}_{I,\phi}\int_0^t \left(\boldsymbol{\phi}_{0,d} - \boldsymbol{\phi}_0\right) d\rho, \tag{2.4}$$

and $\boldsymbol{K}_{D,*}$, $\boldsymbol{K}_{P,*}$, $\boldsymbol{K}_{I,*}$ $(* = p, \phi)$ are symmetric and positive definite gain matrices.

Remark 3. The position control law is designed on the basis of (2.1) in lieu of (1.29), since (1.29) depends on $\boldsymbol{\delta}$, and, thus, computing of $\boldsymbol{f}_0$ would have required the reference values for $\boldsymbol{\delta}$, not available at this stage.

Once the force $\boldsymbol{f}_0$ is determined, it is possible to compute the reference values for the roll-pitch angles by exploiting (1.3) and the expression of $\boldsymbol{R}_1^0$ and $\boldsymbol{R}_2^1$. More in detail, $\boldsymbol{f}_0$ can be expressed in the frame $\boldsymbol{\Sigma}_0$,

$$\boldsymbol{f}_0^0 = \boldsymbol{R}_0^{\mathrm{T}}\boldsymbol{f}_0. \tag{2.5}$$

Then, in view of (1.4) and (1.5) it results

$$\boldsymbol{f}_0^0 = \boldsymbol{R}_0^{\mathrm{T}}\boldsymbol{R}_2\boldsymbol{f}_2^2 = \boldsymbol{R}_2^0\boldsymbol{f}_2^2 = \boldsymbol{R}_1^0\boldsymbol{R}_2^1\boldsymbol{f}_2^2. \tag{2.6}$$

By expanding (2.6), the following relation can be derived

$$\begin{bmatrix} f^0_{0,x} \\ f^0_{0,y} \\ f^0_{0,z} \end{bmatrix} = \begin{bmatrix} 0 & -c_\beta & s_\beta \\ c_\alpha & -s_\alpha s_\beta & -c_\beta s_\alpha \\ s_\alpha & c_\alpha s_\beta & c_\alpha c_\beta \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ f^2_z \end{bmatrix}. \tag{2.7}$$

Therefore, the desired values of the joints angles and the total thrust to be provided by the propellers are given by

$$f^2_z = \|\boldsymbol{f}^0_0\| = \|\boldsymbol{f}_0\|, \tag{2.8}$$

$$\alpha_{des} = \arctan\left(-\frac{f^0_{0,y}}{f^0_{0,z}}\right), \tag{2.9}$$

$$\beta_{des} = \arcsin\left(\frac{f^0_{0,x}}{\|\boldsymbol{f}^0_0\|}\right). \tag{2.10}$$

The desired velocity and acceleration of the roll-pitch joints can be derived by numerical differentiation; in practice, since α_{des} and β_{des} are affected by noise, it is recommended to use suitable robust-to-noise filters. An effective approach is detailed in [45], where the time-varying filter proposed in [69] is adopted in order to compute both the first and second time-derivatives of the reference values.

Remark 4. In view of (2.7), it could be noted that (2.9)–(2.10) are not well defined if

- $\|\boldsymbol{f}^0_0\|$ vanishes: it can happen only if $f_z = 0$, namely in the presence of a null total thrust;
- $f^0_{0,z}$ is equal to zero, but, in this case, the function arctan is still well defined and returns

$$\begin{cases} \alpha_{des} = 0 & \text{if } f^0_{0,y} = 0 \qquad (2.11) \\ \alpha_{des} = \pm 90 \text{ deg} & \text{in the other cases.} \qquad (2.12) \end{cases}$$

The assumption of total thrust always non-null is a reasonable assumption for vehicles not involved in acrobatic maneuvers, while $f^0_{0,z} = 0$ arises when the platform $\boldsymbol{z}_0$ axis lies in the $x - y$ inertial plane, but, for the limits of the mechanical frame, this situation cannot happen, otherwise the ODQuad becomes uncontrollable.

2.1.2 Inner loop: attitude and roll-pitch controller

Once the reference values for roll-pitch angles, $\boldsymbol{\delta}_d = [\alpha_{des},\ \beta_{des}]^{\mathrm{T}}$, and their derivatives, $\dot{\boldsymbol{\delta}}_d$ and $\ddot{\boldsymbol{\delta}}_d$, have been determined using robust-to-noise filters, the auxiliary input $\boldsymbol{a}_\delta$ is computed as

$$\boldsymbol{a}_\delta = \ddot{\boldsymbol{\delta}}_d + \boldsymbol{K}_{D,\delta}\left(\dot{\boldsymbol{\delta}}_d - \dot{\boldsymbol{\delta}}\right) + \boldsymbol{K}_{P,\delta}\left(\boldsymbol{\delta}_d - \boldsymbol{\delta}\right) + \boldsymbol{K}_{I,\delta}\int_0^t \left(\boldsymbol{\delta}_d - \boldsymbol{\delta}\right) d\rho, \tag{2.13}$$

where $\boldsymbol{K}_{D,\delta}$, $\boldsymbol{K}_{P,\delta}$ and $\boldsymbol{K}_{I,\delta}$ are symmetric and positive definite gain matrices.

Therefore, the moment to be applied to the platform and the joint torques can be computed as

$$\boldsymbol{\mu}_0 = \boldsymbol{M}_{p\phi}^{\mathrm{T}}\boldsymbol{a}_p + \boldsymbol{M}_{\phi\phi}\boldsymbol{a}_\phi + \boldsymbol{M}_{\phi\delta}\boldsymbol{a}_\delta + \boldsymbol{C}_\phi\dot{\boldsymbol{\phi}}_0 + \boldsymbol{g}_\phi \tag{2.14}$$

$$\boldsymbol{\tau}_c = \boldsymbol{M}_{p\delta}^{\mathrm{T}}\boldsymbol{a}_p + \boldsymbol{M}_{\phi\delta}^{\mathrm{T}}\boldsymbol{a}_\phi + \boldsymbol{M}_{\delta\delta}\boldsymbol{a}_\delta + \boldsymbol{C}_\delta\dot{\boldsymbol{\delta}} + \boldsymbol{g}_\delta - \boldsymbol{J}_2^{\mathrm{T}}\boldsymbol{h}_2. \tag{2.15}$$

Finally, in order to allocate the total thrust $\|f_{b,2}\|$ and total moment $\boldsymbol{\mu}_{b,2}^2$ onto the propellers, the (1.37) must be inverted, i.e.,

$$\boldsymbol{\sigma} = \boldsymbol{N}^{-1}\begin{bmatrix}\|f_{b,2}\| \\ \boldsymbol{\mu}_{b,2}^2\end{bmatrix}. \tag{2.16}$$

2.2 Stability Analysis

By folding the control outputs (2.2), (2.14), and (2.15) into the ODQuad dynamics (1.29)–(1.31), the following set of equations can be written

$$\boldsymbol{M}_{pp}(\ddot{\boldsymbol{p}}_0 - \boldsymbol{a}_p) + \boldsymbol{M}_{p\phi}(\ddot{\boldsymbol{\phi}}_0 - \boldsymbol{a}_\phi) + \boldsymbol{M}_{p\delta}\ddot{\boldsymbol{\delta}} = \boldsymbol{0}_3 \tag{2.17}$$

$$\boldsymbol{M}_{p\phi}^{\mathrm{T}}(\ddot{\boldsymbol{p}}_0 - \boldsymbol{a}_p) + \boldsymbol{M}_{\phi\phi}(\ddot{\boldsymbol{\phi}}_0 - \boldsymbol{a}_\phi) + \boldsymbol{M}_{\phi\delta}(\ddot{\boldsymbol{\delta}} - \boldsymbol{a}_\delta) = \boldsymbol{0}_3 \tag{2.18}$$

$$\boldsymbol{M}_{p\delta}^{\mathrm{T}}(\ddot{\boldsymbol{p}}_0 - \boldsymbol{a}_p) + \boldsymbol{M}_{\phi\delta}^{\mathrm{T}}(\ddot{\boldsymbol{\phi}}_0 - \boldsymbol{a}_\phi) + \boldsymbol{M}_{\delta\delta}(\ddot{\boldsymbol{\delta}} - \boldsymbol{a}_\delta) = \boldsymbol{0}_2. \tag{2.19}$$

By adding and subtracting $\boldsymbol{M}_{p\delta}\boldsymbol{a}_\delta$, (2.17) can be rearranged in the same form of (2.18) and (2.19) as

$$\boldsymbol{M}_{pp}(\ddot{\boldsymbol{p}}_0 - \boldsymbol{a}_p) + \boldsymbol{M}_{p\phi}(\ddot{\boldsymbol{\phi}}_0 - \boldsymbol{a}_\phi) + \boldsymbol{M}_{p\delta}(\ddot{\boldsymbol{\delta}} - \boldsymbol{a}_\delta) + \boldsymbol{M}_{p\delta}\boldsymbol{a}_\delta = \boldsymbol{0}_3. \tag{2.20}$$

By defining the vector $\boldsymbol{a}$ and the block matrix $\widetilde{\boldsymbol{M}}$ as

$$\boldsymbol{a} = \begin{bmatrix}\boldsymbol{a}_p \\ \boldsymbol{a}_\phi \\ \boldsymbol{a}_\delta\end{bmatrix} \in \mathbb{R}^8, \qquad \widetilde{\boldsymbol{M}} = \begin{bmatrix}\boldsymbol{O}_3 & \boldsymbol{O}_3 & \boldsymbol{M}_{p\delta} \\ \boldsymbol{O}_3 & \boldsymbol{O}_3 & \boldsymbol{O}_{3\times2} \\ \boldsymbol{O}_{2\times3} & \boldsymbol{O}_{2\times3} & \boldsymbol{O}_2\end{bmatrix} \in \mathbb{R}^{8\times8},$$

(2.17)–(2.20) can be rewritten in the following compact form:

$$\boldsymbol{M}(\ddot{\boldsymbol{q}} - \boldsymbol{a}) + \widetilde{\boldsymbol{M}}\boldsymbol{a} = \boldsymbol{0}_8, \tag{2.21}$$

where $\mathbf{0}_8$ is the (8×1) vector of zeros. By exploiting (2.3), (2.4), and (2.13), the following error dynamics is obtained:

$$\ddot{\boldsymbol{e}} + \boldsymbol{K}_D\dot{\boldsymbol{e}} + \boldsymbol{K}_P\boldsymbol{e} + \boldsymbol{K}_I\int \boldsymbol{e} + \boldsymbol{M}^{-1}\widetilde{\boldsymbol{M}}\boldsymbol{a} = \mathbf{0}_8, \tag{2.22}$$

where $\boldsymbol{e} = \boldsymbol{q}_d - \boldsymbol{q}$, with $\boldsymbol{q}_d = [\boldsymbol{p}_{0,d}^{\mathrm{T}},\ \boldsymbol{\phi}_{0,d}^{\mathrm{T}},\ \boldsymbol{\delta}_d^{\mathrm{T}}]^{\mathrm{T}}$, and the matrices $\boldsymbol{K}_\star$ ($\star = \{I, P, D\}$) are

$$\boldsymbol{K}_\star = \mathrm{blkdiag}\left\{\boldsymbol{K}_{\star,p}, \boldsymbol{K}_{\star,\phi}, \boldsymbol{K}_{\star,\delta}\right\}.$$

Define the state vector $\boldsymbol{z} = \left[\boldsymbol{z}_1^{\mathrm{T}},\ \boldsymbol{z}_2^{\mathrm{T}},\ \boldsymbol{z}_3^{\mathrm{T}}\right]^{\mathrm{T}} = \left[\int \boldsymbol{e}^{\mathrm{T}},\ \boldsymbol{e}^{\mathrm{T}},\ \dot{\boldsymbol{e}}^{\mathrm{T}}\right]^{\mathrm{T}}$ and rewrite (2.22) in the state space form as

$$\begin{aligned}\dot{\boldsymbol{z}}_1 &= \boldsymbol{z}_2\\ \dot{\boldsymbol{z}}_2 &= \boldsymbol{z}_3\\ \dot{\boldsymbol{z}}_3 &= -(\boldsymbol{K}_D\boldsymbol{z}_3 + \boldsymbol{K}_P\boldsymbol{z}_2 + \boldsymbol{K}_I\boldsymbol{z}_1) - \boldsymbol{M}^{-1}\widetilde{\boldsymbol{M}}\boldsymbol{a},\end{aligned} \tag{2.23}$$

which can be rewritten in matrix form as

$$\dot{\boldsymbol{z}} = \begin{bmatrix} \boldsymbol{O}_8 & \boldsymbol{I}_8 & \boldsymbol{O}_8\\ \boldsymbol{O}_8 & \boldsymbol{O}_8 & \boldsymbol{I}_8\\ -\boldsymbol{K}_I & -\boldsymbol{K}_P & -\boldsymbol{K}_D \end{bmatrix}\boldsymbol{z} + \begin{bmatrix} \mathbf{0}_8\\ \mathbf{0}_8\\ -\boldsymbol{M}^{-1}\widetilde{\boldsymbol{M}}\boldsymbol{a} \end{bmatrix} = \boldsymbol{K}\boldsymbol{z} + \widetilde{\boldsymbol{a}}. \tag{2.24}$$

In other words, the system (2.24) can be seen as a *nominal* linear system ($\dot{\boldsymbol{z}} = \boldsymbol{K}\boldsymbol{z}$) with a perturbation term $\widetilde{\boldsymbol{a}}$. Since matrix $\boldsymbol{K}$ is in block companion form and can be proven to be Hurwitz [70], the nominal system is exponentially stable.

As for the perturbation term, the following property of the inertia matrix can be exploited [53,71]:

Property 1. *$\boldsymbol{M}$ is symmetric and positive definite; therefore, if $\lambda_{min}(\cdot)$ ($\lambda_{max}(\cdot)$) denotes the minimum (maximum) eigenvalue, it is*

$$0 < \lambda_{min}(\boldsymbol{M})\boldsymbol{I}_8 \leq \boldsymbol{M}(\boldsymbol{q}) \leq \lambda_{max}(\boldsymbol{M})\boldsymbol{I}_8. \tag{2.25}$$

Thus, based on Property 1, the following holds:

$$\lambda_{max}^{-1}(\boldsymbol{M}) < \left\|\boldsymbol{M}^{-1}\right\| < \lambda_{min}^{-1}(\boldsymbol{M}). \tag{2.26}$$

Regarding matrix $\widetilde{\boldsymbol{M}}$, it is worth considering the explicit expression of its sole non-null block (1.26), i.e.,

$$\boldsymbol{M}_{p\delta} = \begin{bmatrix} -\left(m_1\boldsymbol{S}(\boldsymbol{r}_{1,C_1}) + m_2\boldsymbol{S}(\boldsymbol{r}_{2,C_2})\right)\boldsymbol{x}_0\,, & -m_2\boldsymbol{S}(\boldsymbol{r}_{2,C_2})\boldsymbol{x}_1 \end{bmatrix}, \tag{2.27}$$

where $\boldsymbol{x}_0$ and $\boldsymbol{x}_1$ are unit vectors and all the other elements are constant and depend on the system's geometrical and inertial properties. Thus, its norm is constant

$$\left\|\widetilde{\boldsymbol{M}}\right\| = \|\boldsymbol{M}_{p\delta}\| = \Lambda. \tag{2.28}$$

Finally, the term $\boldsymbol{a}$ can be written in the explicit form as

$$\boldsymbol{a} = \ddot{\boldsymbol{q}}_d + \boldsymbol{K}_D\dot{\boldsymbol{e}} + \boldsymbol{K}_P\boldsymbol{e} + \boldsymbol{K}_I\int\boldsymbol{e}.$$

The following assumption on the planned trajectory is made:

Assumption 1. *The planned acceleration $\ddot{\boldsymbol{q}}_d$ is bounded $\forall\ t \in \mathcal{T}$, i.e., $\|\ddot{\boldsymbol{q}}_d(t)\| \leq \bar{\ddot{q}}$.*

Therefore, the non-null block of the perturbation term can be rewritten as

$$\widetilde{\boldsymbol{a}}_3 = -\left(\boldsymbol{M}^{-1}\widetilde{\boldsymbol{M}}\ddot{\boldsymbol{q}}_d + \boldsymbol{M}^{-1}\widetilde{\boldsymbol{M}}\left(\boldsymbol{K}_D\dot{\boldsymbol{e}} + \boldsymbol{K}_P\boldsymbol{e} + \boldsymbol{K}_I\int\boldsymbol{e}\right)\right). \tag{2.29}$$

The first term, in view of (2.26), (2.28) and of Assumption 1, is norm-bounded $\forall\ t$

$$\left\|\boldsymbol{M}^{-1}\widetilde{\boldsymbol{M}}\ddot{\boldsymbol{q}}_d\right\| \leq \frac{1}{\lambda_{min}(\boldsymbol{M})}\Lambda\bar{\ddot{q}}. \tag{2.30}$$

The second term in (2.29) is a vanishing perturbation (in the sense of [72]), since the following chain of inequalities holds:

$$\begin{aligned}\left\|\boldsymbol{M}^{-1}\widetilde{\boldsymbol{M}}\left(\boldsymbol{K}_D\dot{\boldsymbol{e}} + \boldsymbol{K}_P\boldsymbol{e} + \boldsymbol{K}_I\int\boldsymbol{e}\right)\right\| &\leq \frac{1}{\lambda_{min}(\boldsymbol{M})}\Lambda\left\|\boldsymbol{K}_D\dot{\boldsymbol{e}} + \boldsymbol{K}_P\boldsymbol{e} + \boldsymbol{K}_I\int\boldsymbol{e}\right\| \\ &\leq \frac{1}{\lambda_{min}(\boldsymbol{M})}\Lambda k_M\|\boldsymbol{z}\|,\end{aligned} \tag{2.31}$$

where $k_M = \max\{\|\boldsymbol{K}_P\|, \|\boldsymbol{K}_D\|, \|\boldsymbol{K}_I\|\}$.

Thus, the effects of the two perturbation terms can be summarized as follows:

- by virtue of Lemma 9.1 in [72], the origin of the system (2.24), under the vanishing perturbation is an exponentially stable equilibrium;
- by virtue of Lemma 9.2 in [72], the origin of the system (2.24), under the non-vanishing bounded perturbation is uniformly bounded.

By exploiting the linearity of the nominal system (2.24), it can be stated that, under Assumption 1, the error $\boldsymbol{z}$ is uniformly bounded. Moreover, if the planned acceleration vanishes, the error converges to zero exponentially.

2.3 Interaction wrench compensation

In task involving contact with the environment, e.g., manipulation or cooperative transportation, the capability of exerting both forces and moments without modifying the orientation of the load, makes the omnidirectional platforms preferable to standard multirotors [8,73].

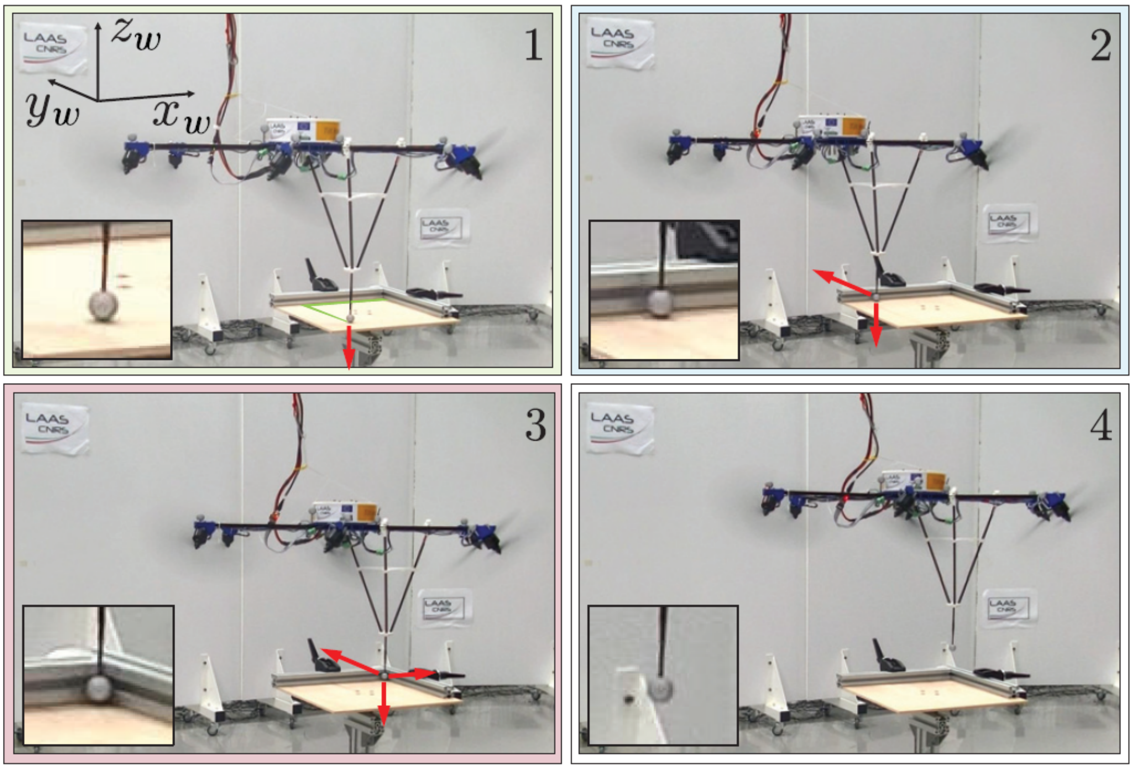


Figure 2.2: Omnidirectional platform in contact with a surface, images from [8].

Moreover, even in simple transportation tasks, it is of the utmost importance to react to external disturbances due to the contact with the object, by compensating the wrench exerted by the object. Namely, an estimation of the wrench exerted by the object on the vehicle is fed back to the controller as an additional term in (2.2), (2.14) and (2.15). By defining the coordinate frame $\boldsymbol{\Sigma}_e\{O_e, x_e, y_e, z_e\}$, attached to the platform and with origin at the contact point, in the presence of an interaction with an object in O_e, the dynamics (1.25) of the ODQuad becomes

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{F}\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{\zeta} + \boldsymbol{J}_e^{\mathrm{T}}\boldsymbol{h}_e, \tag{2.32}$$

where $\boldsymbol{J}_e$ is the Jacobian matrix that relates the velocity $\boldsymbol{v}_e$, defined as the vector of the linear and angular velocities of the contact point O_e, to the velocity of the generalized coordinates $\dot{\boldsymbol{q}}$ as

$$\boldsymbol{v}_e = \boldsymbol{J}_e \dot{\boldsymbol{q}}, \tag{2.33}$$

and $\boldsymbol{h}_e$ is the wrench exerted by the object on the contact point of the ODQuad. More in detail $\boldsymbol{J}_e$ can be defined in the following way

$$\boldsymbol{J}_e = \begin{bmatrix} \boldsymbol{I}_3 & -\boldsymbol{S}(\boldsymbol{r}_e) \\ \boldsymbol{O}_3 & \boldsymbol{I}_3 \end{bmatrix} \boldsymbol{J}_0, \tag{2.34}$$

where $\boldsymbol{r}_e = \boldsymbol{p}_e - \boldsymbol{p}_0$ is the constant vector from the contact point with respect to $\boldsymbol{\Sigma}_0$ expressed in $\boldsymbol{\Sigma}$ and $\boldsymbol{J}_0 = \begin{bmatrix} \boldsymbol{J}_{P_0}^{\mathrm{T}} & \boldsymbol{J}_{O_0}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}$ is the (6×8) Jacobian matrix. Given an estimate, $\hat{\boldsymbol{h}}_e$, of $\boldsymbol{h}_e$, the control outputs (2.2), (2.14) and (2.15) become

$$\boldsymbol{f}_0 = \boldsymbol{M}_{pp}\boldsymbol{a}_p + \boldsymbol{M}_{p\phi}\boldsymbol{a}_\phi + \boldsymbol{C}_p\dot{\boldsymbol{p}}_0 + \boldsymbol{g}_p - \boldsymbol{J}_{e,f}^{\mathrm{T}}\hat{\boldsymbol{h}}_e, \tag{2.35}$$

$$\boldsymbol{\mu}_0 = \boldsymbol{M}_{p\phi}^{\mathrm{T}}\boldsymbol{a}_p + \boldsymbol{M}_{\phi\phi}\boldsymbol{a}_\phi + \boldsymbol{M}_{\phi\delta}\boldsymbol{a}_\delta + \boldsymbol{C}_\phi\dot{\boldsymbol{\phi}}_0 + \boldsymbol{g}_\phi - \boldsymbol{J}_{e,\mu}^{\mathrm{T}}\hat{\boldsymbol{h}}_e \tag{2.36}$$

$$\boldsymbol{\tau}_c = \boldsymbol{M}_{p\delta}^{\mathrm{T}}\boldsymbol{a}_p + \boldsymbol{M}_{\phi\delta}^{\mathrm{T}}\boldsymbol{a}_\phi + \boldsymbol{M}_{\delta\delta}\boldsymbol{a}_\delta + \boldsymbol{C}_\delta\dot{\boldsymbol{\delta}} + \boldsymbol{g}_\delta - \boldsymbol{J}_2^{\mathrm{T}}\boldsymbol{h}_2 - \boldsymbol{J}_{e,\tau}^{\mathrm{T}}\hat{\boldsymbol{h}}_e, \tag{2.37}$$

where $\boldsymbol{J}_{e,f}$, $\boldsymbol{J}_{e,\mu}$ and $\boldsymbol{J}_{e,\tau}$ are obtained from $\boldsymbol{J}_e$ by selecting, respectively, the first three columns, the forth to sixth column and the last two columns.

2.3.1 Wrench estimation

In order to measure the wrench exerted by the object on the ODQuad a force/torque sensor could be mounted on the contact point robot-object [74]. This is usually capable to provide reliable measures, but increases both the cost and the weight of the robot. In the aerial robotics field, a more viable solution is the adoption of a wrench estimator, that can provide a sufficiently accurate estimation [8].

In this work, the interaction wrench, exerted by the manipulated object on the vehicle has been estimated via a momentum-based observer [75]. With reference to the system dynamics (2.32), the angular momentum $\boldsymbol{\nu}$ can be computed as

$$\boldsymbol{\nu} = \boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{q}}\,. \tag{2.38}$$

The momentum dynamics can be computed by keeping its time-derivative as

$$\dot{\boldsymbol{\nu}} = \dot{\boldsymbol{M}}(\boldsymbol{q})\dot{\boldsymbol{q}} + \boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}}, \tag{2.39}$$

Let recall the following property of the dynamic model (1.25) [53,71].

Property 2. *There always exists a choice of $\boldsymbol{C}$ such that*

$$\dot{\boldsymbol{M}}(\boldsymbol{q}) = \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{C}^{\mathrm{T}}(\boldsymbol{q}, \dot{\boldsymbol{q}}). \tag{2.40}$$

By exploiting the Property 2 and the model (1.25), the time-derivative of (2.38) can be expressed as

$$\dot{\boldsymbol{\nu}} = (\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{C}^{\mathrm{T}}(\boldsymbol{q}, \dot{\boldsymbol{q}}))\dot{\boldsymbol{q}} - \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} - \boldsymbol{g}(\boldsymbol{q}) + \boldsymbol{\zeta} + \boldsymbol{J}_e^{\mathrm{T}}\boldsymbol{h}_e, \tag{2.41}$$

by eliminating the opposite terms the following expression can be written as

$$\dot{\boldsymbol{\nu}} = \boldsymbol{C}^{\mathrm{T}}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} - \boldsymbol{g}(\boldsymbol{q}) + \boldsymbol{\zeta} + \boldsymbol{J}_e^{\mathrm{T}}\boldsymbol{h}_e. \tag{2.42}$$

An estimate of $\boldsymbol{h}_e$ is, then, computed as a difference between the momentum (2.38), only due to vehicle dynamics, and the integral of (2.42), that represents the vehicle's momentum in the presence of an external interaction wrench, according the following expression

$$\hat{\boldsymbol{h}}_e = \boldsymbol{J}_e^{\dagger\mathrm{T}}\boldsymbol{K}_o\left[(\boldsymbol{\nu}(t) - \boldsymbol{\nu}(t_0)) - \int_{t_0}^{t}(\boldsymbol{C}^{\mathrm{T}}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} - \boldsymbol{g}(\boldsymbol{q}) + \boldsymbol{\zeta} + \boldsymbol{J}_e^{\mathrm{T}}\hat{\boldsymbol{h}}_e)d\varsigma\right], \tag{2.43}$$

where t and t_0 are the current and initial time instant, respectively, $\boldsymbol{K}_o$ is a positive definite gain matrix and $\boldsymbol{J}_e^{\dagger}$ is the right pseudo-inverse of the Jacobian matrix.

By exploiting (2.42) the estimation dynamics, i.e. the time derivative of (2.43), is

$$\dot{\hat{\boldsymbol{h}}}_e = \boldsymbol{J}_e^{\dagger\mathrm{T}}\boldsymbol{K}_o\left[(\dot{\boldsymbol{\nu}}(t) - (\boldsymbol{C}^{\mathrm{T}}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} - \boldsymbol{g}(\boldsymbol{q}) + \boldsymbol{\zeta} + \boldsymbol{J}_e^{\mathrm{T}}\hat{\boldsymbol{h}}_e)\right] = \boldsymbol{J}_e^{\dagger\mathrm{T}}\boldsymbol{K}_o\left[\boldsymbol{J}_e^{\mathrm{T}}\boldsymbol{h}_e - \boldsymbol{J}_e^{\mathrm{T}}\hat{\boldsymbol{h}}_e\right],$$

which can be rearranged as

$$\dot{\hat{\boldsymbol{h}}}_e + \boldsymbol{K}_o\hat{\boldsymbol{h}}_e = \boldsymbol{K}_o\boldsymbol{h}_e, \tag{2.44}$$

that is a first-order low-pass dynamic system. Under the assumption of constant or slowly varying interaction wrench, $\hat{\boldsymbol{h}}_e$ converges to $\boldsymbol{h}_e$ when $t \to \infty$ for any positive definite gain matrix $\boldsymbol{K}_o$. The matrix $\boldsymbol{K}_o$ is chosen as a trade-off between the convergence rate and the filtering properties of the observer: larger values of the gains allow for faster convergence while smaller values allow filtering the high-frequency noise.

2.3.2 Control loop

The motion controller introduced in the Section 2.1 is able to control the ODQuad in free motion, i.e. when the multirotor does not interact with the environment. When an external wrench, as a transported load or a wind thrust, perturbs the system a simple motion control is not able to control the ODQuad with the expected performance or could even lead to the destabilization of the vehicle. For this reason external wrench compensation is introduced in the control loop, according to the scheme reported in Fig.2.3, where the feedback of the estimated interaction wrench, $\hat{\boldsymbol{h}}_e$, is provided to the motion controller.

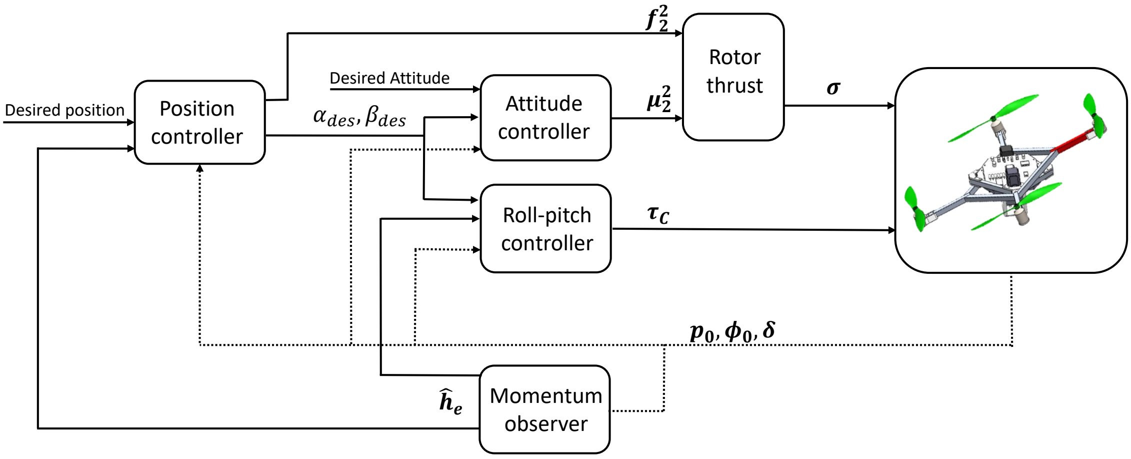


Figure 2.3: Block scheme of the overall control architecture, including the two loop controller, the propeller's thrust mapping and the wrench estimator.

Remark 5. This is not a force control for robot-environment interactions, but a simple motion controller which allows to compensate limited external perturbations, acting on the multirotor.

2.3.3 Stability Analysis

By folding the new control outputs (2.35), (2.36), and (2.37) into the ODQuad dynamics (2.32), the following set of equations can be written

$$\boldsymbol{M}_{pp}(\ddot{\boldsymbol{p}}_0 - \boldsymbol{a}_p) + \boldsymbol{M}_{p\phi}(\ddot{\boldsymbol{\phi}}_0 - \boldsymbol{a}_\phi) + \boldsymbol{M}_{p\delta}\ddot{\boldsymbol{\delta}} - \boldsymbol{J}_{e,f}^{\mathrm{T}}(\boldsymbol{h}_e - \hat{\boldsymbol{h}}_e) = \boldsymbol{0}_3 \tag{2.45}$$

$$\boldsymbol{M}_{p\phi}^{\mathrm{T}}(\ddot{\boldsymbol{p}}_0 - \boldsymbol{a}_p) + \boldsymbol{M}_{\phi\phi}(\ddot{\boldsymbol{\phi}}_0 - \boldsymbol{a}_\phi) + \boldsymbol{M}_{\phi\delta}(\ddot{\boldsymbol{\delta}} - \boldsymbol{a}_\delta) - \boldsymbol{J}_{e,\mu}^{\mathrm{T}}(\boldsymbol{h}_e - \hat{\boldsymbol{h}}_e) = \boldsymbol{0}_3 \tag{2.46}$$

$$\boldsymbol{M}_{p\delta}^{\mathrm{T}}(\ddot{\boldsymbol{p}}_0 - \boldsymbol{a}_p) + \boldsymbol{M}_{\phi\delta}^{\mathrm{T}}(\ddot{\boldsymbol{\phi}}_0 - \boldsymbol{a}_\phi) + \boldsymbol{M}_{\delta\delta}(\ddot{\boldsymbol{\delta}} - \boldsymbol{a}_\delta) - \boldsymbol{J}_{e,\tau}^{\mathrm{T}}(\boldsymbol{h}_e - \hat{\boldsymbol{h}}_e) = \boldsymbol{0}_2. \tag{2.47}$$

By manipulating (2.45) in the same way of (2.20), the dynamics (2.21), in presence of interaction wrench, can be rewritten as

$$\boldsymbol{M}(\ddot{\boldsymbol{q}} - \boldsymbol{a}) + \widetilde{\boldsymbol{M}}\boldsymbol{a} - \boldsymbol{J}_e^{\mathrm{T}}(\boldsymbol{h}_e - \hat{\boldsymbol{h}}_e) = \boldsymbol{0}_8, \tag{2.48}$$

by exploiting (2.44) the above equation can be rearranged in the following way

$$\boldsymbol{M}(\ddot{\boldsymbol{q}} - \boldsymbol{a}) + \widetilde{\boldsymbol{M}}\boldsymbol{a} - \boldsymbol{J}_e^{\mathrm{T}}\boldsymbol{K}_o^{-1}\dot{\hat{\boldsymbol{h}}}_e = \boldsymbol{0}_8, \tag{2.49}$$

that, in view of the stability analysis presented in Section 2.2, the error is still bounded, if the term $\boldsymbol{J}_e^{\mathrm{T}}\boldsymbol{K}_o^{-1}\dot{\hat{\boldsymbol{h}}}_e$ is also bounded, i.e. under the assumption of slowly varying, non-impulsive and non-high frequency interaction wrenches.

2.4 Simulation Results: Motion Control

Until a few decades ago, any novel robot concept was tested directly as a prototype since computers with enough computing power to guarantee numerical simulation tests in reasonable amount of time were not available. Fortunately, nowadays is very easy and less time consuming to perform a dynamic simulation of a new robotic system concept, thus it is possible to shorten and accelerate the prototyping phase. For this reason, also the proposed ODQuad concept has been tested via numerical simulation in the MATLAB© SimMechanics environment. The assembly CAD model has been created using the software SolidWorks and then imported, with the plug-in Simscape Multibody Link, in the simulator mentioned above.

In order to verify the effectiveness and the robustness of the controller, simulations in free-space motion and in the presence of contact with an object have been conducted. In the first one, in order to show the capability of controlling both position and orientation at same time, a trajectory in free-space has been commanded to the vehicle platform. In a further simulation the behavior in the presence of a task requiring grasp and transportation of an object has been tested. The control gains are reported in Table 2.1.

It is assumed that reliable measurements of the position and the orientation of the platform are available. This assumption can be hardly fulfilled in many practical cases; however, since the aim of this section is to validate the feasibility of the proposed controller, it looks appropriate to tackle details on sensing, i.e., the meausurement noise, later in the Section 2.5. Moreover, it is assumed that measurements of the angular positions of the roll-pitch joint are available as well, while velocities are obtained via numerical filtering of the corresponding variables. In this section, measurement noise was not considered, to better highlight the intrinsic tracking performance of the proposed controller. In the presence of the measurement noise, however, the time-varying filter proposed in [69] can be adopted in order to compute both the first and second time-derivatives of $\boldsymbol{\delta}$, wich has been proven to be effective (see, e.g., the results in [45]).

Table 2.1: Gain matrices for the auxiliary inputs (2.3), (2.4) and (2.13).

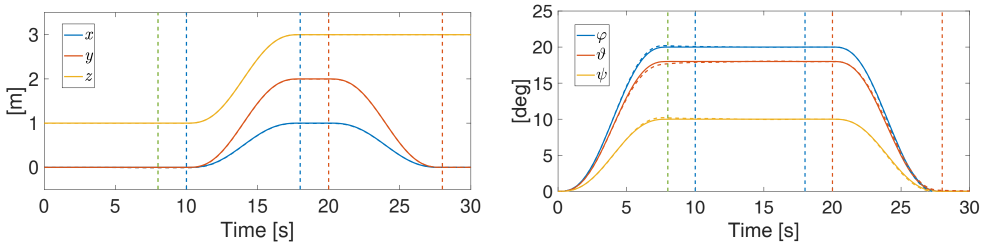
Gain	value	Gain	Value	Gain	Value
$\boldsymbol{K}_{P,p}$	$\begin{bmatrix}10 & 0 & 0\\ 0 & 10 & 0\\ 0 & 0 & 10\end{bmatrix}$	$\boldsymbol{K}_{D,p}$	$\begin{bmatrix}1.8 & 0 & 0\\ 0 & 1.8 & 0\\ 0 & 0 & 1.8\end{bmatrix}$	$\boldsymbol{K}_{I,p}$	$\begin{bmatrix}0.01 & 0 & 0\\ 0 & 0.01 & 0\\ 0 & 0 & 0.01\end{bmatrix}$
$\boldsymbol{K}_{P,\phi}$	$\begin{bmatrix}325 & 0 & 0\\ 0 & 325 & 0\\ 0 & 0 & 125\end{bmatrix}$	$\boldsymbol{K}_{D,\phi}$	$\begin{bmatrix}25.5 & 0 & 0\\ 0 & 25.5 & 0\\ 0 & 0 & 8.5\end{bmatrix}$	$\boldsymbol{K}_{I,\phi}$	$\begin{bmatrix}150 & 0 & 0\\ 0 & 150 & 0\\ 0 & 0 & 50\end{bmatrix}$
$\boldsymbol{K}_{P,\delta}$	$\begin{bmatrix}325 & 0\\ 0 & 325\end{bmatrix}$	$\boldsymbol{K}_{D,\delta}$	$\begin{bmatrix}25.5 & 0\\ 0 & 25.5\end{bmatrix}$	$\boldsymbol{K}_{I,\delta}$	$\begin{bmatrix}150 & 0\\ 0 & 150\end{bmatrix}$

In order to make a first test of the control law robustness, the following uncertainties have been intentionally introduced:

- centrifugal and Coriolis terms have been neglected in control laws (2.2), (2.14) and (2.15);
- viscous friction has been considered in the simulation model but not considered in the control design.

The assigned desired trajectory is composed by three phases (at the end of each phase, the desired variables are kept constant for 2 s):

1. Phase 1 ($0 \leq t \leq 8$ s): Rotations around the axes x, y and z (respectively, 20 deg, 18 deg and 10 deg) are commanded, while the vehicle is in hovering.
2. Phase 2 ($10 \leq t \leq 18$ s): Position displacements along the axes x, y and z (1 m, 2 m and 2 m, respectively) are commanded, while the orientation is kept constant.
3. Phase 3 ($20 \leq t \leq 28$ s): Position displacements along the axes x and y (-1 m and -2 m, respectively) and orientation displacements around axes x, y and z (-20 deg, -18 deg and -10 deg, respectively) are commanded.

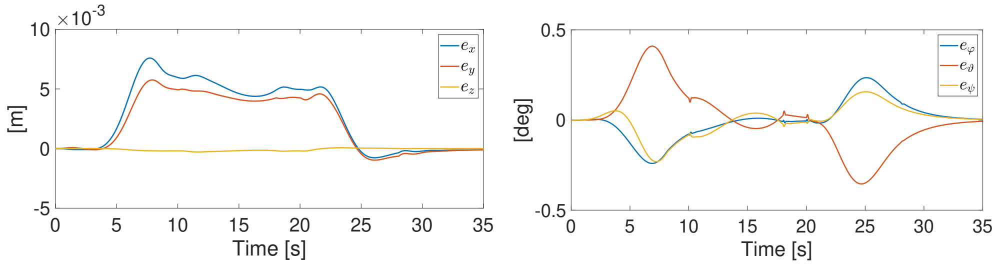


(a) Position Trajectory

(b) Orientation Trajectory

Figure 2.4: Desired (continuous line) and actual (dashed lines) position and orientation trajectories for the platform. The vertical dashed lines represent the steady state phases.

The resulting trajectory can be seen in Figs. 2.4(a)–2.4(b). Tracking errors are reported in Figs. 2.5(a)–2.5(b), which show that the adopted control law ensures accurate tracking of the desired trajectories; indeed, the maximum error is less than 8 mm for the position and 0.4 deg for the orientation, and asymptotic convergence to zero is achieved.

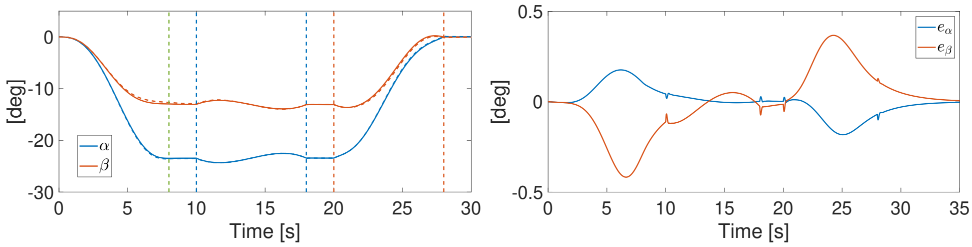


(a) Position Error

(b) Orientation Error

Figure 2.5: Position and orientation tracking errors.

The desired and the actual position of the gimbal joint angles are reported in Fig. 2.6(a), while the tracking errors are reported in Fig. 2.6(b). Good tracking performance are obtained as well, with a maximum error of about 0.2 and 0.4 deg.

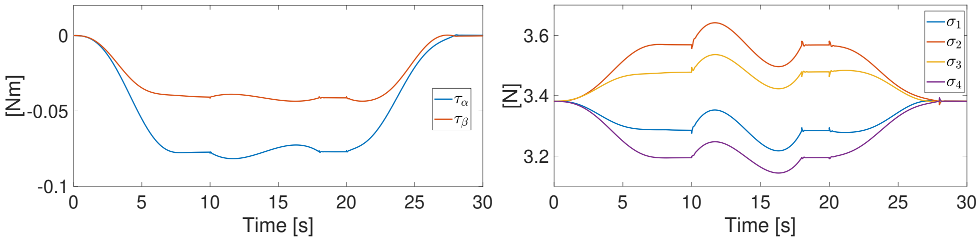


(a) Roll and pitch joint positions

(b) Roll and pitch joint errors

Figure 2.6: Performance of roll and pitch joints: desired (continuous line) and actual (dashed line) positions, steady state phases (vertical dashed lines) (a) and tracking errors (b).

Finally, Figs. 2.7(a)–2.7(b) report the control inputs, i.e. the joint torques and vehicle thrusts. It is worth pointing out that both the vehicle thrusts and the joint torques are in such a range that they can be obtained via off-the-shelf motors.



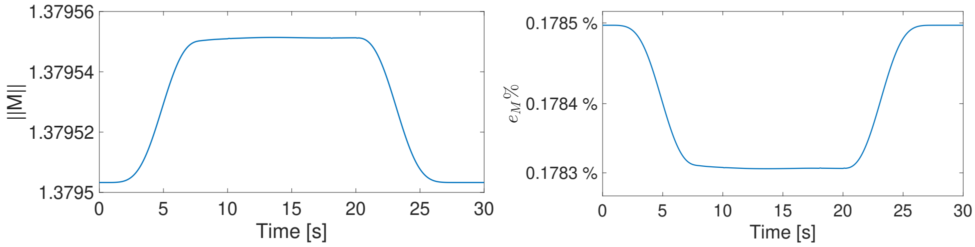
(a) Roll and pitch joint torques (b) Propeller thrusts

Figure 2.7: Control inputs.

To highlight the effect of the approximation introduced in (2.2), Fig. 2.8(a) reports the norm of the inertia matrix along the trajectory, while Fig. 2.8(b) reports the percentage error between the norm of $\boldsymbol{M}$ and the approximated matrix, $\widetilde{\boldsymbol{M}}$, obtained by nullifying the therm $\boldsymbol{M}_{p\delta}$, i.e.

$$e_M\% = 100\frac{\left\|\boldsymbol{M} - \widetilde{\boldsymbol{M}}\right\|}{\|\boldsymbol{M}\|}. \tag{2.50}$$

It can be noticed that the difference is almost constant and very close to zero (about 0.17%), thus confirming that the proposed approximation is reasonable. Similar results have been obtained along a number of different simulated trajectories.



(a) Norm of $\boldsymbol{M}$ (b) Norm error

Figure 2.8: Comparison of inertial matrix with the approximated matrix used in the control inputs.

2.4.1 Comparison with an under-actuated quadrotor

In order to show the effectiveness of the ODQuad with respect to an under-actuated quadrotor UAV, a simulation in the presence of a disturbance force in the x_0–y_0 plane has been carried out. The under-actuated UAV has been modeled in SimMechanics starting from the ODQuad and removing the gimbal joints and the corresponding motors; the controller in [76] has been considered and tuned in such a way to achieve the same step response achieved by the ODQuad controller (with the gains in Table2.1). During the simulation, a force disturbance along x_0 axis with magnitude

$$f(t) = (5 + \cos(t/5))\sin(t) \ \ \mathrm{N}, \tag{2.51}$$

has been applied to the origin of $\boldsymbol{\Sigma}_0$ (see Fig. 2.9), while the UAV is in hovering condition.

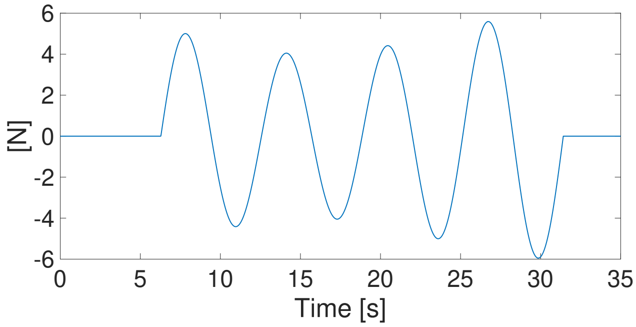
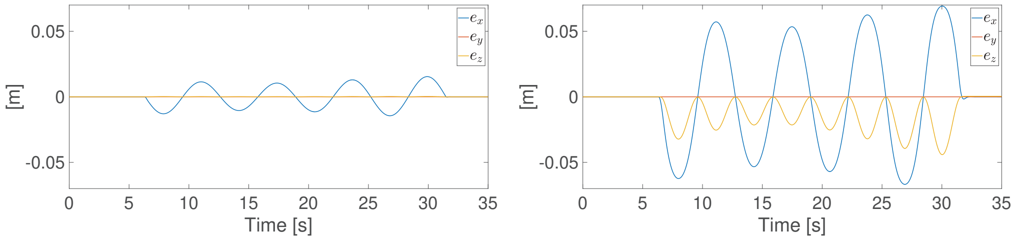


Figure 2.9: Disturbance force.

Figure 2.10 shows the position error of the ODQuad (Fig.2.10(a)) and the under-actuated UAV (Fig.2.10(b)). It can be noticed that the ODQuad presents an error along the x axis below 2 cm while the error of the under-actuated reaches a peak of 7 cm. Moreover, the orientation of the ODQuad platform is kept below 0.5 deg; this cannot be achieved by the under-actuated UAV, since the roll and pitch angles are computed by the controller to generate the horizontal motion needed to compensate the disturbance.

The capability to counteract disturbances on the horizontal plane makes the ODQuad particularly suitable for applications requiring the UAV to carry a gripper or a robotic arm in order to execute manipulation tasks.



(a) ODquad position error.

(b) Under-actuated UAV position error.

Figure 2.10: Position error in the presence of disturbance force.

2.5 Simulation Results: Interaction Wrench Compensation

In the simulation results presented in Section 2.4 the tracking performances in free motion and the ODQuad capability to counteract forces in the horizontal plane have been shown. In order to verify the effectiveness and the robustness of the controller in a realistic scenario, simulations in free motion and in the presence of interaction with the environment have been performed. In the former, in order to show the capability of controlling both position and orientation at same time, a trajectory in free-space has been commanded to the vehicle platform; in the latter simulation the behavior in the presence of a task requiring grasp and transportation of an object has been tested.

To simulate a realistic scenario, measurement noise has been introduced and the following assumptions have been considered:

- reliable but noisy measurements of the platform position and orientation are available, e.g. provided by a motion capture system and an IMU;
- white noise has been added to the measures of the platform position (with standard deviation 10^{-2} m) and orientation (with standard deviation 10^{-2} rad);
- the angular positions of the roll and pitch joints are available via sensors, usually integrated into the servomotors, while the angular velocities are obtained via numerical filtering. Thus, the orientation of the rotor frame relative to the platform frame can be computed via (1.4).

The observer gain is reported in Table 2.2 and it has been set via a trial-and-error procedure, by looking for the best trade off between the convergence rate and the robustness to measurement noise. The control gains, that are reported in Table 2.3, have been tuned via a trial-and-error procedure, by taking into account that the internal loop must be faster than the outer one and with the goal of minimizing the tracking error.

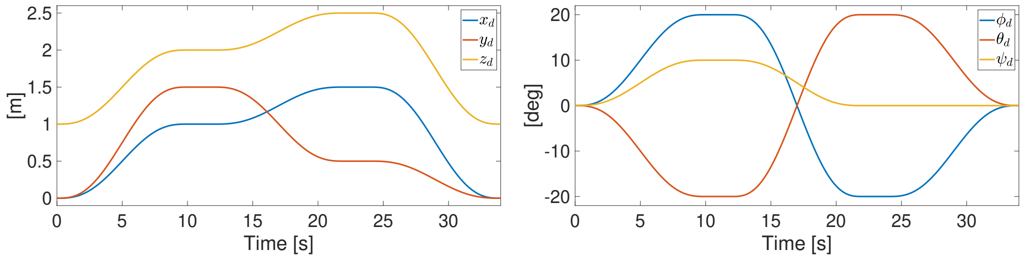
Table 2.2: Momentum observer gain.

Gain	**value**
$\boldsymbol{K}_o$	$\text{diag}\begin{bmatrix}1000 & 1000 & 1000 & 400 & 400 & 35 & 200 & 200\end{bmatrix};$

Table 2.3: Gain matrices for the auxiliary inputs (2.3), (2.4) and (2.13).

Gain	value	Gain	Value	Gain	Value
$\boldsymbol{K}_{P,p}$	$\begin{bmatrix}25 & 0 & 0\\ 0 & 25 & 0\\ 0 & 0 & 25\end{bmatrix}$	$\boldsymbol{K}_{D,p}$	$\begin{bmatrix}6.3 & 0 & 0\\ 0 & 6.3 & 0\\ 0 & 0 & 6.3\end{bmatrix}$	$\boldsymbol{K}_{I,p}$	$\begin{bmatrix}0.025 & 0 & 0\\ 0 & 0.025 & 0\\ 0 & 0 & 0.025\end{bmatrix}$
$\boldsymbol{K}_{P,\phi}$	$\begin{bmatrix}375 & 0 & 0\\ 0 & 375 & 0\\ 0 & 0 & 125\end{bmatrix}$	$\boldsymbol{K}_{D,\phi}$	$\begin{bmatrix}25.5 & 0 & 0\\ 0 & 25.5 & 0\\ 0 & 0 & 8.5\end{bmatrix}$	$\boldsymbol{K}_{I,\phi}$	$\begin{bmatrix}150 & 0 & 0\\ 0 & 150 & 0\\ 0 & 0 & 50\end{bmatrix}$
$\boldsymbol{K}_{P,\delta}$	$\begin{bmatrix}375 & 0\\ 0 & 375\end{bmatrix}$	$\boldsymbol{K}_{D,\delta}$	$\begin{bmatrix}25.5 & 0\\ 0 & 25.5\end{bmatrix}$	$\boldsymbol{K}_{I,\delta}$	$\begin{bmatrix}150 & 0\\ 0 & 150\end{bmatrix}$

In order to verify the trajectory tracking capabilities, in the first case simulation study, time-varying desired position (see. Fig.2.11(a)) and orientation (see Fig.2.11(b)) are comanded to the ODQuad platform. As previously remarked, it is worth noticing that, differently from common quadrotors, motion involving all the 6 degrees of freedoms can assigned.

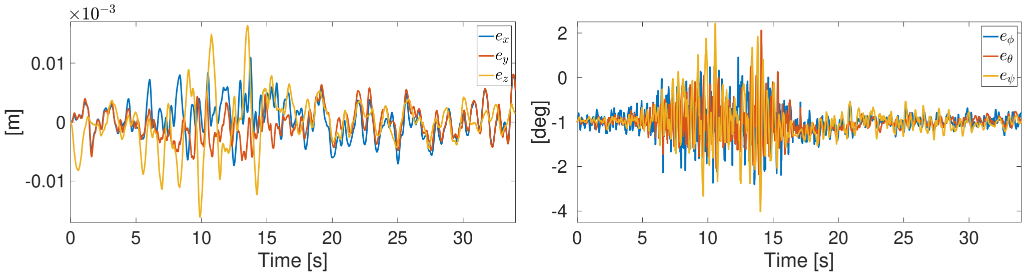


(a) Position Trajectory

(b) Orientation Trajectory

Figure 2.11: First case study: desired position and orientation trajectories for the platform.

The obtained results are reported in Fig. 2.12(a) and Fig. 2.12(b) in terms of position ($\boldsymbol{p}_{0,d} - \boldsymbol{p}_0$) and orientation ($\boldsymbol{\phi}_{0,d} - \boldsymbol{\phi}_0$) errors. In both cases, satisfactory performance is achieved, since the maximum position error is about 5 mm along the axes x and y and 15 mm along the z axis, while the orientation error oscillates around the zero, with some peaks below 5 deg.



(a) Position tracking error

(b) Orientation tracking error

Figure 2.12: First case study: position and orientation tracking errors.

The performance of the internal control loop is measured by the roll-pitch joint tracking errors (($\alpha_{des} - \alpha$) and ($\beta_{des} - \beta$)), reported in Fig. 2.13. Also in this case, the peak error is below 5 deg, and the oscillation magnitude is about 2 degrees.

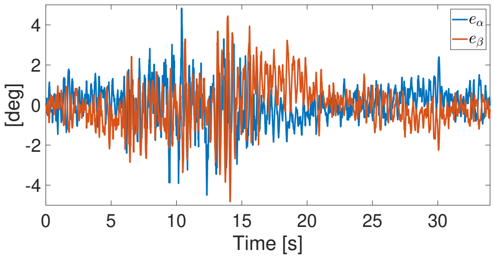


Figure 2.13: First case study: roll-pitch joint errors.

The second simulation case study is aimed at proving the controller robustness in the presence of an external disturbances. In this case, the control outputs (2.2), (2.14) and (2.15) are replaced by (2.35). A task involving grasping and transportation of an object has been considered: the ODQuad moves to the grasping point, grasps the object and then rigidly holds it during its motion. Fig. 2.14 shows some snapshots of the simulation.

The position tracking error, reported in Fig. 2.15(a), increases along the z axis during the grasping phase (at about $t = 10$ s), as expected, and comes back to the initial value after 3 seconds. In other words, after an initial transition due to the first contact with the object, the wrench estimator allows to correctly compensate the effect of the additional load attached to the vehicle. Then, during the transportation, an error is experienced along the y axis and in the platform orientation (with a peak of about 6 deg), due to the dynamics of the gimbal mechanism.

Fig. 2.16(a) reports, the roll-pitch joint tracking errors: the roll and pitch errors during the transportation oscillate between -5 and 5 deg. Fig. 2.16(b) reports the actuation

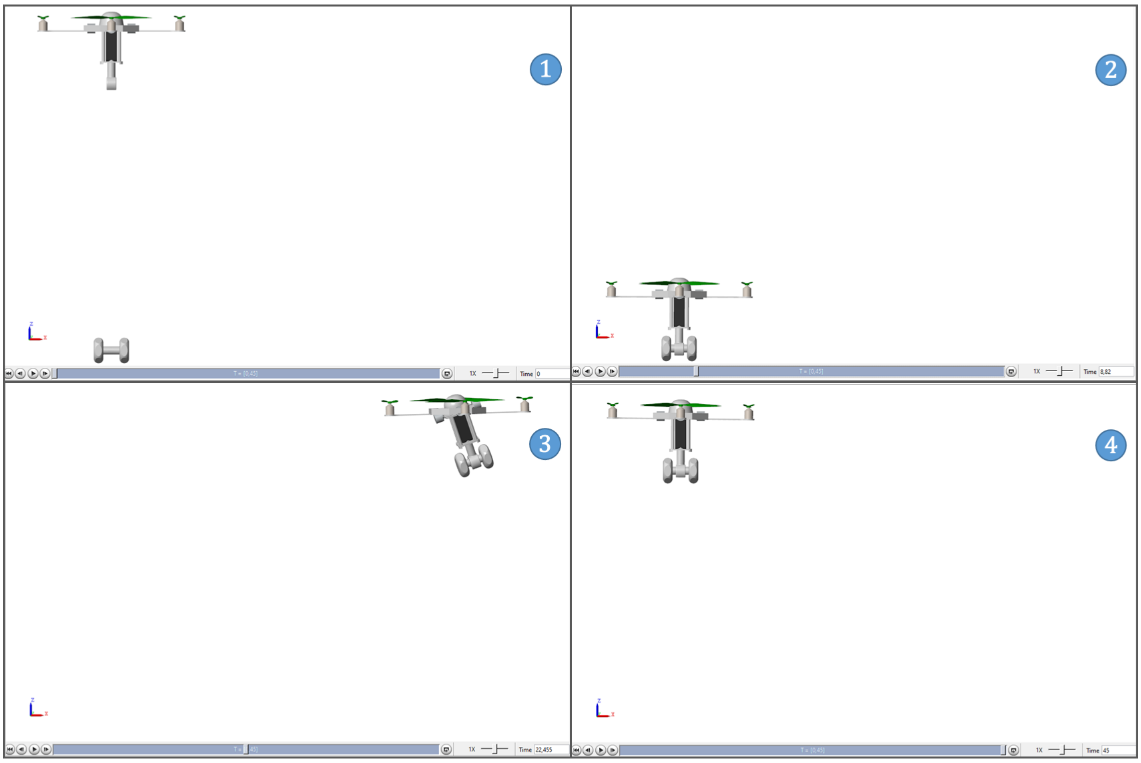
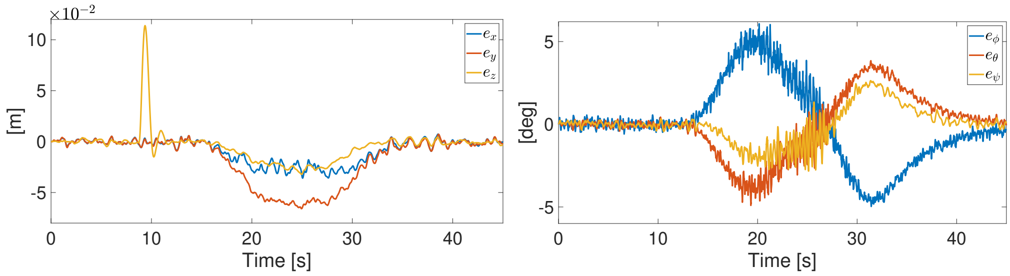


Figure 2.14: Second case study: screenshots of Matlab Mechanics Explore. 1) ODQuad stays in hovering position. 2) ODQuad grasps the object, 3) ODQuad performs a position and orientation trajectory. 4) ODQuad comes back to the initial configuration moving the grasped object.

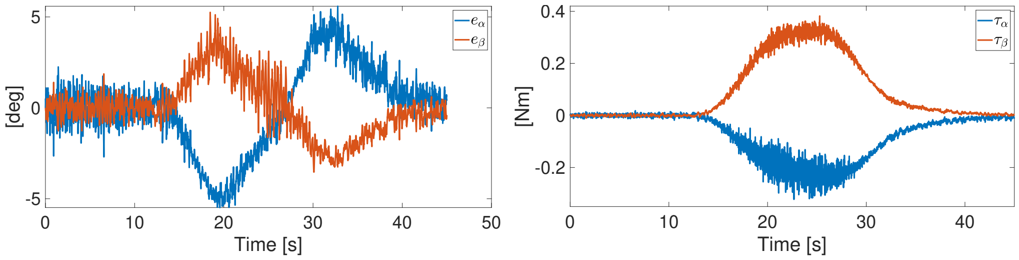
torques of the roll-pitch joints: it can be noticed that, even in the presence of an external load, the required control effort can be easily delivered by off-the-shelf motors.



(a) Position tracking error

(b) Orientation tracking error

Figure 2.15: Second case study: platform position and orientation tracking errors.

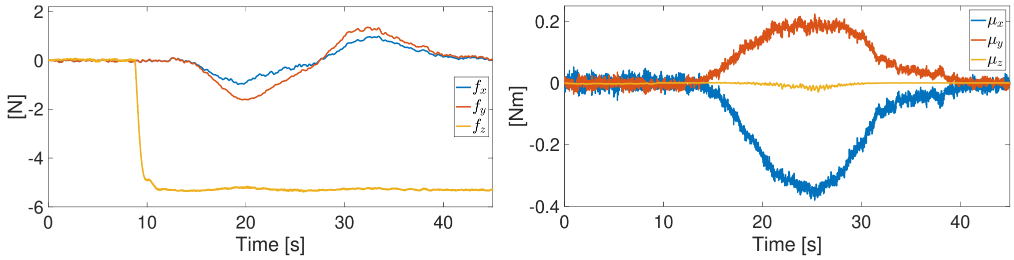


(a) Roll-pitch joint tracking error

(b) Roll-pitch joint torques

Figure 2.16: Second case study: roll-pitch joint tracking error (a) and torques (b).

Finally, the estimated interaction forces and torques, computed via the observer (2.43), are reported in Figs. 2.17(a) and 2.17(b), respectively. In detail, the force along the z axis correctly converges to the load weight (Fig. 2.17(a)), while small forces along x and y axes are experienced during the transportation, due to the object inertia and the unmodeled ODQuad's dynamics (Coriolis and friction terms have been neglected both in controller and in observer). Regarding the estimated torques (Fig. 2.17(b)), small torques are experienced around the x and y axes, due to the misalignment during the motion between the object and the platform CoM.



(a) Estimated force

(b) Estimated torque

Figure 2.17: Second case study: estimated wrench.

2.6 Conclusion

This chapter has reported the design and verification of a model-based motion control scheme for an omnidirectional UAV, the ODQuad, enriched with a wrench estimation approach to manage situations in which disturbances, due to external forces, arise. The control validation has been carried out via simulations in free motion and in task involving object transportation. The effectiveness of the control scheme has been tested both in an ideal and a real scenario that includes several real-world effects (e.g., measurement noise, unmodeled dynamics), in this way is possible to have a results that are closer to the real prototype behavior. The results in Section 2.4 shown the ODQuad capability to perform a 6-DoF trajectory, since it is able to decouple linear motion from angular motion by exploiting the roll-pitch joint. The comparison with a common quadrotor shows the ODQuad superior disturbance rejection capabilities when an horizontal force acts on the vehicle's body, which is due to the omnidirectional thrust that the multirotor can exert in the upper hemisphere of the platform. The results shown in Section 2.5 confirm that the adopted control scheme achieves satisfactory performance in terms of tracking capabilities of free-space motion as well as of correct estimation and compensation of the contact wrench.

Chapter 3

Aerial cooperative manipulation

3.1 Introduction

Recent developments in the field of Unmanned Aerial Vehicles (UAVs) have been focusing on application scenarios requiring physical interaction with the external environment and/or other robotic systems. Thus, Aerial Manipulators, i.e., UAVs equipped with robotic manipulators or grippers to confer the required grasping and manipulation capabilities, have aroused interest among roboticists [77]. Noticeably, omnidirectional aerial platforms [4,7], combining the advantages of Vertical Take Off and Landing (VTOL) aerial vehicles with the capability to control position and orientation independently and to exert force and torque on the environment, allow the improvement of the robustness and reliability of the interaction [7,78].

On the other hand, the use of multi-arm systems for cooperative execution of manipulation tasks overcomes the limitations of single-arm systems, especially in the presence of large and/or heavy payloads [14].

Such a paradigm has been successfully applied to mobile manipulators as well (see, e.g., [18,30–33]). Among the first attempts to use multiple aerial robots for cooperative transportation, it is worth mentioning the work in [34], where multiple quadrotors manipulate and transport a payload via cables. In [35], a team of quadrotors is rigidly attached to a payload via suitably designed grippers and a partially decentralized control law is adopted to ensure coordinate transportation of the payload, without considering internal stresses acting on the grasped object and interaction with the environment. In [36], a cooperative manipulation of a rigid object via VTOL aerial vehicles is investigated, where a sliding mode motion controller is adopted to achieve absolute motion of the object and internal forces are controlled via a virtual linkage approach; however, interaction with the environment is not considered. In [37], the impedance scheme in [38] has been applied to a team of cooperative VTOL aerial vehicles moving a rigid object. More recently, an effective master-slave approach for the cooperative transportation of a payload via multiple aerial VTOL vehicles has been devised in [39], where the master moves the payload in the desired direction while the slave actively guarantees compliance to the master motion.

This Chapter is aimed at developing a novel framework for cooperative manipulation of large and/or heavy objects via omnidirectional aerial vehicles acting as aerial manipulators [77]. The system is composed by multiple ODQuads that tightly grasp a rigid object by means of a gripper. Moreover, physical interaction of the held object with the external environment is taken into account; object/environment interaction might be due to either unforeseen contact with humans/obstacles or planned actions (e.g., in parts mating tasks).

When moving in free space, two systems of wrenches act on the object: the external wrenches, causing object motion, and the internal wrenches, i.e., internal loading of the object, which should be limited to avoid damaging due to excessive mechanical stresses. Moreover, when the object comes in contact with the external environment, interaction wrenches due to contact arise: again, such contact wrenches are to be properly limited, so as to avoid damages of the parts in contact. In order to limit both internal and contact wrenches, the impedance control approach is adopted, based on the two-loop general control framework developed in [38]: an external impedance is in charge of limiting the interaction wrenches between the object and the environment while an internal impedance is aimed at avoiding large internal wrenches. Such approach has been extended to aerial manipulators moving a rigid object in [37]. An alternative approach for free-flying robots is the multiple impedance control (MIC) paradigm [79], which is in charge of enforcing an impedance behavior both at the manipulator end-effectors and at the grasped object level. A comprehensive comparison of such solution with the object-based impedance approach can be found in recent literatire [80].

However, differently from [38] and [37], in this Chapter the dynamic parameters of the manipulated object are assumed unknown. In order to achieve estimation of the object parameters, a centralized version of the two-stage approach developed in [18] is adopted. In particular, in a first stage, the system performs simple motions to collect the data needed to estimate the geometric and dynamic parameters. Then, the estimated parameters are used in the control laws to execute the assigned cooperative manipulation task.

In order to provide further insights on the proposed approach, a simulation case study is developed, where the effects of measurement noise, aerodynamic disturbances and model uncertainties are considered.

3.2 Problem Statement

An unknown object rigidly grasped by N omnidirectional UAVs is considered. The use of omnidirectional platforms in lieu of standard quadrotors is suggested for their capability of exerting both forces and moments at the contact points, without modifying the orientation of the load, and by reacting to external disturbances in a more effective way. The effectiveness of omnidirectional platforms in performing contact tasks has been experimentally verified in [78].

More in detail, the ODQuad aerial vehicle is equipped with a grasping tool attached to its platform as illustrated in Fig.3.1.

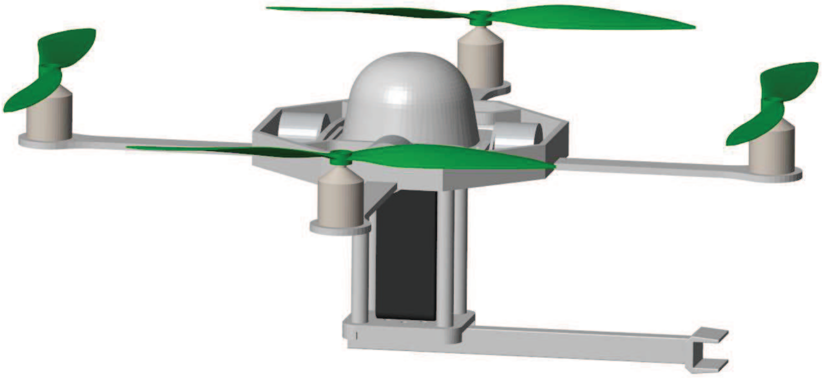


Figure 3.1: ODQuad simulation model with a rigid grasping tool.

In order to grasp and move the object, the grasping tool is assumed to rigidly connect the platform to the object and a force-torque sensor is mounted on the tool, in order to measure the contact wrench between the object and the ODQuad.

The following coordinate frames are defined (see Fig 3.2):

- The coordinate frame $\mathbf{\Sigma}_{0_k}\{O_{0_k}, x_{0_k}, y_{0_k}, z_{0_k}\}$, attached to the kth robot platform ($k = 1, 2, \ldots, N$);

- The coordinate frame $\boldsymbol{\Sigma}_{e_k}\{O_{e_k}, x_{e_k}, y_{e_k}, z_{e_k}\}$, attached to the kth tool and with origin at the contact point;
- The coordinate frame $\boldsymbol{\Sigma}_o\{O_o, x_o, y_o, z_o\}$ attached to the object center of mass.

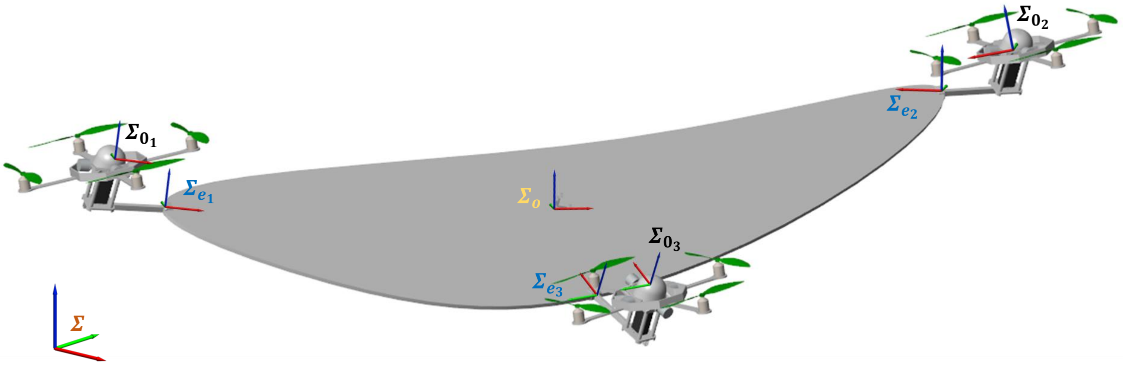


Figure 3.2: Overall system with main reference frames

3.3 Modeling of the aerial vehicles

The model of the ODQuad aerial vehicle has been derived in Chapter 1 and is briefly reviewed here for the sake of clarity. The configuration of the kth vehicle in the cooperative system ($k = 1, \ldots, N$) can be described by the vector $\boldsymbol{q}_k \in \mathbb{R}^8$ of generalized coordinates

$$\boldsymbol{q}_k = \begin{bmatrix} \boldsymbol{p}_{0_k} \\ \boldsymbol{\phi}_{0_k} \\ \boldsymbol{\delta}_k \end{bmatrix}, \tag{3.1}$$

where $\boldsymbol{p}_{0_k} \in \mathbb{R}^3$ is the position of O_{0_k} with respect to the inertial frame, $\boldsymbol{\phi}_{0_k} = [\varphi_k \ \vartheta_k \ \psi_k]^{\mathrm{T}} \in \mathbb{R}^3$ is the vector of roll-pitch-yaw angles representing the platform orientation with respect to the inertial frame, and $\boldsymbol{\delta}_k = [\alpha_k \ \beta_k]^{\mathrm{T}} \in \mathbb{R}^2$, with α_k and β_k the position of roll and pitch joints, respectively.

By exploiting (2.33), the following differential kinematic equation can be written

$$\boldsymbol{v}_{e_k} = \boldsymbol{J}_{e_k}\dot{\boldsymbol{q}}_k, \tag{3.2}$$

where $\boldsymbol{v}_{e_k}$ is the vector of the linear and angular velocities of the frames $\boldsymbol{\Sigma}_{e_k}$, while $\boldsymbol{J}_{e_k}$ is the Jacobian matrix defined in (2.34), which can be rewritten as

$$\boldsymbol{J}_{e_k} = \begin{bmatrix} \boldsymbol{I}_3 & -\boldsymbol{S}(\boldsymbol{r}_{e_k}) \\ \boldsymbol{O}_3 & \boldsymbol{I}_3 \end{bmatrix} \boldsymbol{J}_{0_k}, \tag{3.3}$$

where $\boldsymbol{r}_{e_k} = \boldsymbol{p}_{e_k} - \boldsymbol{p}_{0_k}$ is the vector from the kth grasping point to the origin of $\boldsymbol{\Sigma}_{0_k}$, $\boldsymbol{J}_{0_k} = \begin{bmatrix}\boldsymbol{J}_{P_{0_k}}^{\mathrm{T}} & \boldsymbol{J}_{O_{0_k}}^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}}$ is the kth (6 × 8) Jacobian matrix, while $\boldsymbol{J}_{P_{0_k}}$ and $\boldsymbol{J}_{O_{0_k}}$ have been defined in (1.11) and (1.10), respectively.

The dynamics (1.25) of the kth aerial vehicle can be written in compact form as

$$\boldsymbol{M}_k(\boldsymbol{q}_k)\ddot{\boldsymbol{q}}_k + \boldsymbol{C}_k(\boldsymbol{q}_k, \dot{\boldsymbol{q}}_k)\dot{\boldsymbol{q}}_k + \boldsymbol{F}_k\dot{\boldsymbol{q}}_k + \boldsymbol{g}_k(\boldsymbol{q}_k) = \boldsymbol{\zeta}_k + \boldsymbol{\zeta}_{k,e}, \tag{3.4}$$

where $\boldsymbol{M}_k(\boldsymbol{q}_k) \in \mathbb{R}^{8\times 8}$ is the symmetric and positive definite inertia matrix, $\boldsymbol{C}_k(\boldsymbol{q}_k, \dot{\boldsymbol{q}}_k) \in \mathbb{R}^{8\times 8}$ collects the Coriolis and centrifugal terms, $\boldsymbol{F}_k \in \mathbb{R}^{8\times 8}$ is a matrix collecting friction forces, i.e. joint damping and aerodynamics disturbances at low velocities, and $\boldsymbol{G}_k(\boldsymbol{q}_k) \in \mathbb{R}^{8}$ is the vector of gravity terms.

The input vector $\boldsymbol{\zeta}_k$ collects the actuation forces and torques acting on the system, i.e.,

$$\boldsymbol{\zeta}_k = \begin{bmatrix}\boldsymbol{f}_{0_k} \\ \boldsymbol{\mu}_{0_k} \\ \boldsymbol{\tau}_{\delta_k}\end{bmatrix}, \tag{3.5}$$

where $\begin{bmatrix}\boldsymbol{f}_{0_k}^{\mathrm{T}}, \boldsymbol{\mu}_{0_k}^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}}$ is the wrench acting on the kth aerial vehicle while $\boldsymbol{\tau}_{\delta_k} \in \mathbb{R}^2$ represents the torques generated by the roll and pitch joint actuators and the effect of the propeller thrusts on the joints [81]. Finally, $\boldsymbol{\zeta}_{k,e} \in \mathbb{R}^8$ represents the effect of the contact wrench $\boldsymbol{h}_{e_k} = \begin{bmatrix}\boldsymbol{f}_{e_k}^{\mathrm{T}} & \boldsymbol{\mu}_{e_k}^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^6$ exerted by the grasped object on the robot

$$\boldsymbol{\zeta}_{k,e} = \boldsymbol{J}_{e_k}^{\mathrm{T}}\boldsymbol{h}_{e_k}. \tag{3.6}$$

For control design purposes, it is worth rewriting the model (3.4) in terms of block matrices

$$\boldsymbol{M}_k = \begin{bmatrix}\boldsymbol{M}_{k,pp} & \boldsymbol{M}_{k,p\phi} & \boldsymbol{M}_{k,p\delta} \\ \boldsymbol{M}_{k,p\phi}{}^{\mathrm{T}} & \boldsymbol{M}_{k,\phi\phi} & \boldsymbol{M}_{k,\phi\delta} \\ \boldsymbol{M}_{k,p\delta}{}^{\mathrm{T}} & \boldsymbol{M}_{k,\phi\delta}{}^{\mathrm{T}} & \boldsymbol{M}_{k,\delta\delta}\end{bmatrix},$$

where $\boldsymbol{M}_{pp} \in \mathbb{R}^{3\times 3}$, $\boldsymbol{M}_{p\phi} \in \mathbb{R}^{3\times 3}$, $\boldsymbol{M}_{p\delta} \in \mathbb{R}^{3\times 2}$, $\boldsymbol{M}_{\phi\phi} \in \mathbb{R}^{3\times 3}$, $\boldsymbol{M}_{\phi\delta} \in \mathbb{R}^{3\times 2}$ and $\boldsymbol{M}_{\delta\delta} \in \mathbb{R}^{2\times 2}$; and

$$\boldsymbol{C}_k = \begin{bmatrix}\boldsymbol{C}_{k,p} \\ \boldsymbol{C}_{k,\phi} \\ \boldsymbol{C}_{k,\delta}\end{bmatrix}, \quad \boldsymbol{F}_k = \begin{bmatrix}\boldsymbol{F}_{k,p} \\ \boldsymbol{F}_{k,\phi} \\ \boldsymbol{F}_{k,\delta}\end{bmatrix}, \quad \boldsymbol{g}_k = \begin{bmatrix}\boldsymbol{g}_{k,p} \\ \boldsymbol{g}_{k,\phi} \\ \boldsymbol{g}_{k,\delta}\end{bmatrix}, \quad \boldsymbol{\zeta}_{k,e} = \begin{bmatrix}\boldsymbol{\zeta}_{k,e,p} \\ \boldsymbol{\zeta}_{k,e,\phi} \\ \boldsymbol{\zeta}_{k,e,\delta}\end{bmatrix},$$

where $\boldsymbol{C}_{k,p} \in \mathbb{R}^{3\times 8}$, $\boldsymbol{C}_{k,\phi} \in \mathbb{R}^{3\times 8}$, $\boldsymbol{C}_{k,\delta} \in \mathbb{R}^{2\times 8}$,$\boldsymbol{F}_{k,p} \in \mathbb{R}^{3\times 8}$, $\boldsymbol{F}_{k,\phi} \in \mathbb{R}^{3\times 8}$, $\boldsymbol{F}_{k,\delta} \in \mathbb{R}^{2\times 8}$, $\boldsymbol{g}_{k,p} \in \mathbb{R}^{3}$, $\boldsymbol{g}_{k,\phi} \in \mathbb{R}^{3}$, $\boldsymbol{g}_{k,\delta} \in \mathbb{R}^{2}$, $\boldsymbol{\zeta}_{k,e,p} \in \mathbb{R}^{3}$, $\boldsymbol{\zeta}_{k,e,\phi} \in \mathbb{R}^{3}$, $\boldsymbol{\zeta}_{k,e,\delta} \in \mathbb{R}^{2}$.

3.4 Object Dynamics

When an object is grasped by some manipulators and moved along a planned trajectory by assigning a compliant behavior to the robotic arms, the tracking performance depends on the dynamic interaction that the manipulated object transfer to the robots. Therefore, in order to improve the trajectory tracking performance, the object dynamic model is required, which can be written as

$$\boldsymbol{M}_o(\boldsymbol{x}_o)\ddot{\boldsymbol{x}}_o + \boldsymbol{C}_o(\boldsymbol{x}_o, \dot{\boldsymbol{x}}_o)\dot{\boldsymbol{x}}_o + \boldsymbol{g}_o = \boldsymbol{\gamma}_o - \boldsymbol{\gamma}_{o,env} \tag{3.7}$$

where

$$\boldsymbol{x}_o = \begin{bmatrix} \boldsymbol{p}_o \\ \boldsymbol{\phi}_o \end{bmatrix}, \quad \boldsymbol{M}_o = \begin{bmatrix} m_o \boldsymbol{I}_3 & \boldsymbol{O}_3 \\ \boldsymbol{O}_3 & \boldsymbol{T}^{\mathrm{T}}(\boldsymbol{\phi}_o)\boldsymbol{I}_o\boldsymbol{T}(\boldsymbol{\phi}_o) \end{bmatrix},$$

$$\boldsymbol{C}_o = \begin{bmatrix} \boldsymbol{O}_3 & \boldsymbol{O}_3 \\ \boldsymbol{O}_3 & \boldsymbol{T}^{\mathrm{T}}(\boldsymbol{\phi}_o)\boldsymbol{I}_o\dot{\boldsymbol{T}}(\boldsymbol{\phi}_o) + \boldsymbol{T}^{\mathrm{T}}(\boldsymbol{\phi}_o)\boldsymbol{S}\left(\boldsymbol{T}(\boldsymbol{\phi}_o)\dot{\boldsymbol{\phi}}_o\right)\boldsymbol{I}_o\boldsymbol{T}(\boldsymbol{\phi}_o) \end{bmatrix}, \quad \boldsymbol{g}_o = \begin{bmatrix} m_o g \boldsymbol{e}_3 \\ \boldsymbol{0}_3 \end{bmatrix},$$

$\boldsymbol{p}_o$ represents the position of the object center of mass (CoM), $\boldsymbol{\phi}_o$ describes the orientation of Σ_o in the inertial frame, m_o is the object mass, g is the gravity acceleration, $\boldsymbol{e}_3 = [0\ 0\ 1]^{\mathrm{T}}$, $\boldsymbol{I}_o$ is the unknown object inertia matrix, finally, the right-hand side of (3.7) is given by

$$\boldsymbol{\gamma}_o = \begin{bmatrix} \boldsymbol{f}_o \\ \boldsymbol{\mu}_o \end{bmatrix} = \boldsymbol{T}_A^{\mathrm{T}}(\boldsymbol{\phi}_o)\boldsymbol{h}_o, \quad \boldsymbol{\gamma}_{o,env} = \begin{bmatrix} \boldsymbol{f}_{o,env} \\ \boldsymbol{\mu}_{o,env} \end{bmatrix} = \boldsymbol{T}_A^{\mathrm{T}}(\boldsymbol{\phi}_o)\boldsymbol{h}_{o,env},$$

where the matrix $\boldsymbol{T}_A \in \mathbb{R}^{6\times 6}$ has the following form

$$\boldsymbol{T}_A(\boldsymbol{\phi}_o) = \begin{bmatrix} \boldsymbol{I}_3 & \boldsymbol{O}_3 \\ \boldsymbol{O}_3 & \boldsymbol{T}(\boldsymbol{\phi}_o) \end{bmatrix},$$

while $\boldsymbol{h}_o$ and $\boldsymbol{h}_{o,env}$ are the wrenches exerted, respectively, by the aerial vehicles on the object and by the object on the environment, both acting at the CoM.

The object is assumed to be large enough to neglect aerodynamic interactions among quadrotors and/or with the object. For an object characterized by a large surface, however, aerodynamic drag cannot be considered *a priori* negligible, although its effect might be limited if the assigned task imposes relatively low velocities, as usual for interaction tasks. Aerodynamic drag is considered as unknown disturbance, included in the simulation model (see Section 3.7), but not in the dynamic model (3.7) used by the estimation and control systems, since it has been modeled as an external force acting on the object.

The wrench exerted on the object CoM by the robots, can be expressed as

$$\boldsymbol{h}_o = \boldsymbol{G}\boldsymbol{h}_e, \tag{3.8}$$

where $\boldsymbol{G}$ is the *grasp matrix* [14, 38]

$$\boldsymbol{G} = \begin{bmatrix} \boldsymbol{G}_1 & \boldsymbol{G}_2 & \dots & \boldsymbol{G}_N \end{bmatrix}, \quad \boldsymbol{G}_k = \begin{bmatrix} \boldsymbol{I}_3 & \boldsymbol{O}_3 \\ \boldsymbol{S}(\boldsymbol{r}_k) & \boldsymbol{I}_3 \end{bmatrix}, k = 1, 2, \dots, N, \tag{3.9}$$

$\boldsymbol{r}_k = \boldsymbol{p}_{e_k} - \boldsymbol{p}_o$ is the vector from the object CoM to the kth contact point, while $\boldsymbol{h}_e$ collects the wrenches exerted by the vehicles at their grasp locations, i.e.

$$\boldsymbol{h}_e = \begin{bmatrix} -\boldsymbol{h}_{e_1} \\ -\boldsymbol{h}_{e_2} \\ \vdots \\ -\boldsymbol{h}_{e_N} \end{bmatrix}, \tag{3.10}$$

since $\boldsymbol{h}_{e_k}$ has been defined as the wrench exerted by the object on the robot, the minus signs have been added. $\boldsymbol{G}$ is a full row rank matrix hence, for a given $\boldsymbol{h}_o$, the inverse solution to (3.8) is given by [14]

$$\boldsymbol{h}_e = \boldsymbol{G}^{\dagger}\boldsymbol{h}_o + \boldsymbol{V}\boldsymbol{h}_i = \boldsymbol{h}_E + \boldsymbol{h}_I \tag{3.11}$$

where $\boldsymbol{G}^{\dagger}$ denotes a suitable right pseudo-inverse of $\boldsymbol{G}$, $\boldsymbol{V} \in \mathbb{R}^{6N \times (6N-6)}$ is a full column rank matrix spanning the null space of $\boldsymbol{G}$, and $\boldsymbol{h}_i \in \mathbb{R}^{6N-6}$ collects the wrench that gives no contribution to the object motion [14] (i.e., the *internal wrench*). A possible expression of $\boldsymbol{V}$ is [82]

$$\boldsymbol{V} = \begin{bmatrix} \boldsymbol{V}_1 & \boldsymbol{O}_6 & \dots & \boldsymbol{O}_6 \\ \boldsymbol{O}_6 & -\boldsymbol{V}_2 & \dots & \boldsymbol{O}_6 \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{O}_6 & \boldsymbol{O}_6 & \dots & -\boldsymbol{V}_{N-1} \\ \boldsymbol{V}_N & \boldsymbol{V}_N & \dots & \boldsymbol{V}_N \end{bmatrix}, \quad \boldsymbol{V}_k = \begin{bmatrix} \boldsymbol{I}_3 & \boldsymbol{O}_3 \\ -\boldsymbol{S}(\boldsymbol{r}_k) & \boldsymbol{I}_3 \end{bmatrix}. \tag{3.12}$$

The vector

$$\boldsymbol{h}_E = \begin{bmatrix} \boldsymbol{h}_{E_1} \\ \boldsymbol{h}_{E_2} \\ \vdots \\ \boldsymbol{h}_{E_N} \end{bmatrix} = \boldsymbol{G}^{\dagger}\boldsymbol{G}\boldsymbol{h}_e \tag{3.13}$$

represents the wrench balancing the object dynamics and the interaction with the environment, while

$$\boldsymbol{h}_I = \begin{bmatrix} \boldsymbol{h}_{I_1} \\ \boldsymbol{h}_{I_2} \\ \vdots \\ \boldsymbol{h}_{I_N} \end{bmatrix} = \boldsymbol{V}\boldsymbol{V}^{\dagger}\boldsymbol{h}_e = \boldsymbol{V}\boldsymbol{h}_i \tag{3.14}$$

represents the vector of the contributes of each robot to the sole internal wrench.

3.4.1 Closed-chain constraints

The following equations describe the kinematic relationship between the object motion and that of the object/vehicles contact points O_{e_k} $(k = 1, \ldots, N)$

$$\begin{cases} \boldsymbol{p}_{e_k} &= \boldsymbol{p}_o + \boldsymbol{R}_o\boldsymbol{r}_k^o\,, \\ \dot{\boldsymbol{p}}_{e_k} &= \dot{\boldsymbol{p}}_o - \boldsymbol{S}(\boldsymbol{R}_o\boldsymbol{r}_k^o)\boldsymbol{T}(\boldsymbol{\phi}_o)\dot{\boldsymbol{\phi}}_o\,, \\ \ddot{\boldsymbol{p}}_{e_k} &= \ddot{\boldsymbol{p}}_o - \boldsymbol{S}(\boldsymbol{R}_o\boldsymbol{r}_k^o)\dot{\boldsymbol{T}}(\boldsymbol{\phi}_o)\dot{\boldsymbol{\phi}}_o + \\ & - \boldsymbol{S}(\boldsymbol{R}_o\boldsymbol{r}_k^o)\boldsymbol{T}(\boldsymbol{\phi}_o)\ddot{\boldsymbol{\phi}}_o - \boldsymbol{S}(\boldsymbol{T}(\boldsymbol{\phi}_o)\dot{\boldsymbol{\phi}}_o)\boldsymbol{S}(\boldsymbol{R}_o\boldsymbol{r}_k^o)\boldsymbol{T}(\boldsymbol{\phi}_o)\dot{\boldsymbol{\phi}}_o \\ \boldsymbol{\phi}_{e_k} &= \boldsymbol{\phi}(\boldsymbol{R}_o\boldsymbol{R}_{e_k}^o)\,, \\ \dot{\boldsymbol{\phi}}_{e_k} &= \boldsymbol{T}^{-1}(\boldsymbol{\phi}_{e_k})\boldsymbol{T}(\boldsymbol{\phi}_o)\dot{\boldsymbol{\phi}}_o \\ \ddot{\boldsymbol{\phi}}_{e_k} &= \boldsymbol{T}^{-1}(\boldsymbol{\phi}_{e_k})\dot{\boldsymbol{T}}(\boldsymbol{\phi}_o)\dot{\boldsymbol{\phi}}_o + \boldsymbol{T}^{-1}(\boldsymbol{\phi}_{e_k})\boldsymbol{T}(\boldsymbol{\phi}_o)\ddot{\boldsymbol{\phi}}_o - \boldsymbol{T}^{-1}(\boldsymbol{\phi}_{e_k})\dot{\boldsymbol{T}}(\boldsymbol{\phi}_{e_k})\dot{\boldsymbol{\phi}}_{e_k}\,, \end{cases} \tag{3.15}$$

where $\boldsymbol{R}_o$ is the rotation matrix representing the orientation of $\boldsymbol{\Sigma}_o$, $\boldsymbol{r}_k^o = \boldsymbol{R}_o^{\mathrm{T}}\boldsymbol{r}_k$, $\boldsymbol{R}_{e_k}^o = \boldsymbol{R}_o^{\mathrm{T}}\boldsymbol{R}_{e,k}$ represents the relative orientation of the frame $\boldsymbol{\Sigma}_{e_k}$ with respect to the frame $\boldsymbol{\Sigma}_o$ and $\boldsymbol{\phi}(\boldsymbol{R})$ is the triplet of Euler angles extracted from the rotation matrix $\boldsymbol{R}$. It is worth remarking that, assuming that a rigid object is tightly grasped by the N vehicles, $\boldsymbol{r}_k^o$ and $\boldsymbol{R}_{e_k}^o$ are constant quantities describing the kinematics of the grasp. Thus, the above equations define a kinematic model of the grasp, i.e., a set of mechanical constraints (closed-chain constraints) on the variables describing the motion of each end effector $\mathcal{T}_k = \{\boldsymbol{x}_{e_k}, \dot{\boldsymbol{x}}_{e_k}, \ddot{\boldsymbol{x}}_{e_k}\}$ $(k = 1, 2, \ldots, N)$ and those describing the object's motion $\mathcal{T}_o = \{\boldsymbol{x}_o, \dot{\boldsymbol{x}}_o, \ddot{\boldsymbol{x}}_o\}$.

3.5 Estimation of the object's parameters

It is assumed that the dynamic parameters of the grasped object are not known by the aerial vehicles: thus, a procedure to cooperatively estimate them needs to be devised. By referring to Fig. 3.3, the parameters to be estimated are:

- the mass m_o;
- the position of the center of mass, O_o, with respect to the contact points;
- the inertia matrix $\boldsymbol{I}_o$.

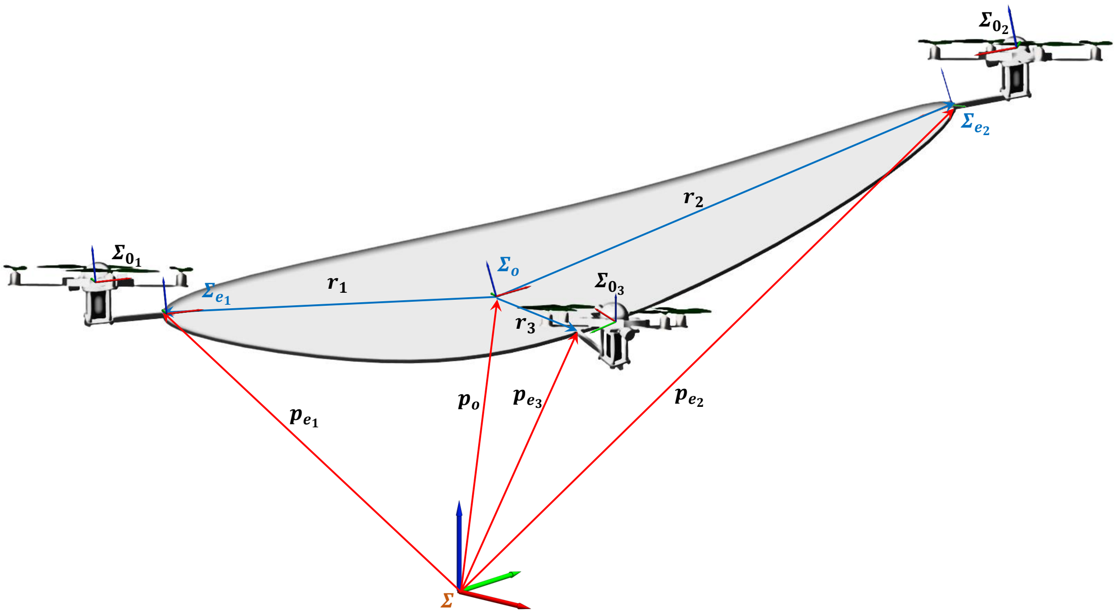


Figure 3.3: Geometric parameter of the object

Following the approach proposed in [18], the estimation of the unknown parameters will be achieved on the basis of the data collected during a preliminary identification stage, in which the aerial vehicles perform simple motions before starting the execution of the assigned task. It is assumed that the parameter identification stage occurs in the free space, i.e., in the absence of any contact between the external environment and the cooperative system. By virtue of (3.15), planning the motion of each robot requires the knowledge of the position of at least one point attached to the object and its relative position with respect to the contact points $O_{e_k}(k = 1, \dots, N)$. Therefore, during this phase, since the object center of mass is not known, another known triplet, for example $\boldsymbol{\Sigma}_{e_k}$ (for $k \in \{1, \dots, N\}$), is exploited. It is worth noticing that, at this stage, no strategies for limiting the object internal wrenches can be adopted, thus small and slow motions must be planned.

3.5.1 Estimation of the object mass

In order to estimate the mass m_o, the robots are commanded to move along the z axis, in such a way that the object is not in contact with the environment; then, they are commanded to keep their pose in hovering condition. In this way, each robot measures the external wrench due to the object

$$\boldsymbol{h}_{e_k} = \begin{bmatrix} 0 \; 0 \; f_{e_k,z} \; \boldsymbol{\mu}_{e_k}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \tag{3.16}$$

from which the object mass can be easily computed as

$$\widehat{m}_o = \frac{\sum\limits_{k=1}^{N} f_{e_k,z}}{g}. \tag{3.17}$$

3.5.2 Estimation of object center of mass

As mentioned previously, at this stage the ODQuad motion is planned by considering one of the contact points O_{e_l} ($l \in \{1, \dots, N\}$) and the position of the CoM in $\boldsymbol{\Sigma}_{e_l}$ is estimated. The main idea is to exploit the equilibrium of the moments in static conditions (hovering) and in the absence of contact wrenches, i.e.

$$\sum_{k=1}^{N} \left((\boldsymbol{p}_{e_k}^{e_l} - \boldsymbol{p}_o^{e_l}) \times \boldsymbol{f}_{e_k}^{e_l} \right) + \sum_{k=1}^{N} \boldsymbol{\mu}_{e_k}^{e_l} = \boldsymbol{0}_3 \,, \tag{3.18}$$

where the superscript $(\cdot)^{e_l}$ denotes that the variables are expressed in the reference frame $\boldsymbol{\Sigma}_{e_l}$ and $\boldsymbol{p}_o^{e_l}$ is constant, since the grasp and the object are assumed to be rigid. Equation (3.18) can be viewed as a system of 3 equation in 3 unknown variables, i.e. the components of $\boldsymbol{p}_o^{e_l}$, thus the estimate of $\boldsymbol{p}_o^{e_l}$ can be easily obtained. In practice, since $\boldsymbol{f}_{e_k}$ and $\boldsymbol{\mu}_{e_k}$ are likely to be affected by measurement noise, it is worth resorting to a least-squares approach. To the aim, it is necessary to consider the equilibrium (3.18) for ν different poses. Therefore, a linear system of 3ν equation in 3 unknown variables can be obtained as

$$\boldsymbol{\Lambda} \boldsymbol{p}_o^{e_l} = \boldsymbol{\Upsilon}, \tag{3.19}$$

where

$$\boldsymbol{\Lambda} = \begin{bmatrix} \sum_{k=1}^{N} \boldsymbol{S}({}^1\boldsymbol{f}_{e_k}^{e_l}) \\ \vdots \\ \sum_{k=1}^{N} \boldsymbol{S}({}^\nu\boldsymbol{f}_{e_k}^{e_l}) \end{bmatrix}, \quad \boldsymbol{\Upsilon} = \begin{bmatrix} \sum_{k=1}^{N} \left(\boldsymbol{S}({}^1\boldsymbol{f}_{e_k}^{e_l}) {}^1\boldsymbol{p}_{e_k}^{e_l} + {}^1\boldsymbol{\mu}_{e_k}^{e_l} \right) \\ \vdots \\ \sum_{k=1}^{N} \left(\boldsymbol{S}({}^\nu\boldsymbol{f}_{e_k}^{e_l}) {}^\nu\boldsymbol{p}_{e_k}^{e_l} + {}^\nu\boldsymbol{\mu}_{e_k}^{e_l} \right) \end{bmatrix},$$

and the superscript $i = 1, \ldots, \nu$ denotes the ith equilibrium pose. Then, the least-squares estimate of $\boldsymbol{p}_o^{e_l}$ can be computed as follows

$$\widehat{\boldsymbol{p}}_o^{e_l} = (\boldsymbol{\Lambda}^{\mathrm{T}}\boldsymbol{\Lambda})^{-1}\boldsymbol{\Lambda}\boldsymbol{\Upsilon}\,, \tag{3.20}$$

and, thus, the estimates $\widehat{\boldsymbol{r}}_k^o = \boldsymbol{R}_o^{\mathrm{T}}\left(\boldsymbol{p}_{e_k} - \boldsymbol{R}_{e_l}\widehat{\boldsymbol{p}}_o^{e_l}\right)$ $(k = 1, \ldots, N)$ can be derived.

Once the object CoM has been estimated, the reference frame $\boldsymbol{\Sigma}_{\hat{o}}\{\hat{O}_o, x_o, y_o, z_o\}$, oriented as $\boldsymbol{\Sigma}_o$ and attached to the estimated CoM can be defined and the object motion can be planned.

3.5.3 Estimation of the object inertia tensor

For the sake of simplicity, consider the estimation of the inertia tensor $\boldsymbol{I}_o$ in the object frame (i.e., $\boldsymbol{I}_o^o = \boldsymbol{R}_o^{\mathrm{T}}\boldsymbol{I}_o\boldsymbol{R}_o$), since it is constant. It can be estimated by exploiting the previously estimated object CoM and the dynamics (3.7). Since $\boldsymbol{I}_o^o$ is a symmetric matrix its estimation is equivalent to the estimation of the vector

$$\boldsymbol{\xi} = \begin{bmatrix} I_{xx}^o, & I_{xy}^o, & I_{xz}^o, & I_{yy}^o, & I_{yz}^o, & I_{zz}^o \end{bmatrix}^{\mathrm{T}} \in \mathrm{I\!R}^6. \tag{3.21}$$

The strategy devised to estimate $\boldsymbol{\xi}$ consists in keeping the position of the origin of $\boldsymbol{\Sigma}_o$ constant, while its orientation is suitably modified by imposing a constant angular velocity. In principle, the assumption of constant angular velocity it is not required, but the angular acceleration should be computed via numerical differentiation and this, usually, leads to excessive noise. Under the assumption of constant velocity, the dynamics in (3.7) written in the frame $\boldsymbol{\Sigma}_o$ becomes

$$\boldsymbol{T}^{\mathrm{T}}(\boldsymbol{\phi}_o)\Big(\boldsymbol{I}_o^o\dot{\boldsymbol{T}}(\boldsymbol{\phi}_o^o)+\boldsymbol{S}\left(\boldsymbol{T}(\boldsymbol{\phi}_o^o)\dot{\boldsymbol{\phi}}_o^o\right)\boldsymbol{I}_o^o\boldsymbol{T}(\boldsymbol{\phi}_o^o)\Big)\dot{\boldsymbol{\phi}}_o^o=\boldsymbol{T}^{\mathrm{T}}(\boldsymbol{\phi}_o)\left(\sum_{k=1}^{N}\hat{\boldsymbol{r}}_k^o\times\boldsymbol{f}_{e_k}^o+\sum_{k=1}^{N}\boldsymbol{\mu}_{e_k}^o\right). \tag{3.22}$$

Therefore, being the left-hand side of (3.22) linear in $\boldsymbol{\xi}$, it can be rearranged such to isolate the unknown parameters as

$$\boldsymbol{\Lambda}_{I_o}(\boldsymbol{\phi}_o^o, \dot{\boldsymbol{\phi}}_o^o)\boldsymbol{\xi} = \boldsymbol{T}^{\mathrm{T}}(\boldsymbol{\phi}_o)\left(\sum_{k=1}^{N}\hat{\boldsymbol{r}}_k^o \times \boldsymbol{f}_{e_k}^o + \sum_{k=1}^{N}\boldsymbol{\mu}_{e_k}^o\right). \tag{3.23}$$

From (3.23), it is possible to estimate the unknown parameters by resorting to a least square solution as in Section 3.5.2.

It is worth noticing that, since the actual CoM is not known but only its estimate $\hat{O}$ is available, the constant inertia tensor $\boldsymbol{\xi}_{\hat{o}} = \begin{bmatrix} I_{xx}^{\hat{o}}, & I_{xy}^{\hat{o}}, & I_{xz}^{\hat{o}}, & I_{yy}^{\hat{o}}, & I_{yz}^{\hat{o}}, & I_{zz}^{\hat{o}} \end{bmatrix}^{\mathrm{T}}$ with respect to the frame $\boldsymbol{\Sigma}_{\hat{o}}$ is estimated in lieu of $\boldsymbol{\xi}$.

3.6 Control architecture

The control objectives are: (i) ensuring tracking of an assigned desired trajectory for the grasped object (i.e., for position and orientation of $\Sigma_{\hat{o}}$), (ii) limiting the mechanical stresses at the held object (internal wrench), (iii) ensuring a compliant behavior of the object in the presence of contact with the environment. The proposed control architecture, inherited by [37], and depicted in Fig. 3.4, is based on two admittance filters aimed at ensuring an impedance behavior with respect to both the internal wrenches exerted by the aerial robots and the object/environment contact wrench. The external admittance filter outputs a reference trajectory for $\Sigma_{\hat{o}}$ that can be mapped into the corresponding desired trajectories for the grasp locations of the aerial robots via the closed-chain constraints (3.15). The desired trajectories of grasp locations are the input of the internal admittance filters, one for each robot, together with the contribution of the robot to the object internal wrench: their outputs are the reference trajectories for the grasp locations, that are mapped into trajectories for the aerial vehicles position and orientation. A low-level motion controller is in charge of tracking them.

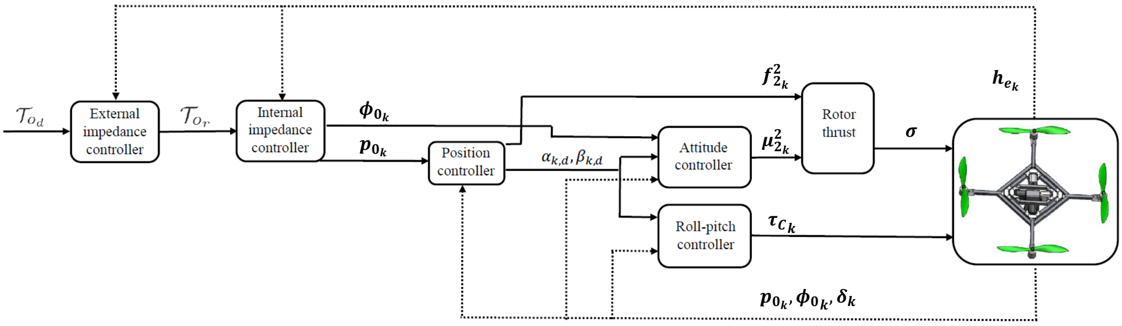

$\mathcal{T}_{o_d}$
External impedance controller
$\mathcal{T}_{o_r}$
Internal impedance controller
$\boldsymbol{\phi}_{0_k}$
$\boldsymbol{p}_{0_k}$
Position controller
$\alpha_{k,d}, \beta_{k,d}$
$\boldsymbol{f}^2_{2_k}$
Attitude controller
$\boldsymbol{\mu}^2_{2_k}$
Rotor thrust
$\boldsymbol{\sigma}$
$\boldsymbol{h}_{e_k}$
Roll-pitch controller
$\boldsymbol{\tau}_{C_k}$
$\boldsymbol{p}_{0_k}, \boldsymbol{\phi}_{0_k}, \boldsymbol{\delta}_k$

Figure 3.4: Control architecture block scheme

3.6.1 External Impedance

The external impedance is in charge of achieving bounded wrenches in the presence of contact with the environment. More in detail, a compliant behavior is enforced between the object pose and the wrench exerted by the object on the environment [37]. Under the assumption of perfect object parameter estimation, the desired trajectory of the object can be assigned in terms of a set of motion variables for the center of mass $\mathcal{T}_{o,d} = \{\boldsymbol{x}_{o,d}, \dot{\boldsymbol{x}}_{o,d}, \ddot{\boldsymbol{x}}_{o,d}\}$ while the corresponding reference variables, $\mathcal{T}_{o,r} = \{\boldsymbol{x}_{o,r}, \dot{\boldsymbol{x}}_{o,r}, \ddot{\boldsymbol{x}}_{o,r}\}$, can be obtained via the differential equation

$$(\boldsymbol{A}-\boldsymbol{I}_6)\boldsymbol{M}_o\ddot{\boldsymbol{x}}_{o,r}+\boldsymbol{D}_o\dot{\boldsymbol{x}}_{o,r}+\boldsymbol{K}_o\boldsymbol{x}_{o,r}+\boldsymbol{\gamma}_o=\boldsymbol{C}_o\dot{\boldsymbol{x}}_o+\boldsymbol{g}_o+\boldsymbol{A}\boldsymbol{M}_o\ddot{\boldsymbol{x}}_{o_d}+\boldsymbol{D}_o\dot{\boldsymbol{x}}_{o_d}+\boldsymbol{K}_o\boldsymbol{x}_{o_d}, \quad (3.24)$$

where $\boldsymbol{D}_o$ and $\boldsymbol{K}_o$ are symmetric and positive definite matrices while $\boldsymbol{A}$ is a positive definite diagonal matrix of weights. Equation (3.24) has been chosen in such a way to include only known variables, such as the object dynamic model, and measured quantities such as the object pose and twist, that can be computed starting from the measured ODQuad pose and velocity.

Under the assumption of perfect motion tracking of the commanded trajectories (i.e., $\mathcal{T}_o \simeq \mathcal{T}_{o,r}$), (3.24) leads to the following impedance law

$$\boldsymbol{A}\boldsymbol{M}_o\boldsymbol{\Delta}\ddot{\boldsymbol{x}}_o+\boldsymbol{D}_o\boldsymbol{\Delta}\dot{\boldsymbol{x}}_o+\boldsymbol{K}_o\boldsymbol{\Delta}\boldsymbol{x}_o=\boldsymbol{\gamma}_{o,env}, \quad (3.25)$$

where $\boldsymbol{\Delta x}_o = \boldsymbol{x}_{o,d} - \boldsymbol{x}_{o,r}$. Equation (3.25) represents an exponentially stable system provided that the eigenvalues of $\boldsymbol{A}$ are chosen greater or equal to 1 [38].

Since only an estimation of the CoM is available, it can be convenient to consider the motion equation of $\hat{O}$, i.e.,

$$\boldsymbol{M}_{\hat{o}}(\hat{\boldsymbol{x}}_o)\ddot{\hat{\boldsymbol{x}}}_o+\boldsymbol{C}_{\hat{o}}(\hat{\boldsymbol{x}}_o,\dot{\hat{\boldsymbol{x}}}_o)\dot{\hat{\boldsymbol{x}}}_o+\boldsymbol{g}_{\hat{o}}=\boldsymbol{\gamma}_{\hat{o}}-\boldsymbol{\gamma}_{\hat{o},env}, \quad (3.26)$$

where $\boldsymbol{M}_{\hat{o}}$ and $\boldsymbol{C}_{\hat{o}}$ denote the matrices in (3.8) computed by considering the estimated inertia matrix expressed with respect to the reference frame $\boldsymbol{\Sigma}_{\hat{o}}$ instead of $\boldsymbol{\Sigma}_o$ as in (3.7), $\boldsymbol{g}_{\hat{o}}$ represents the gravity terms written with respect to the estimated CoM, while $\boldsymbol{\gamma}_{\hat{o}} = \boldsymbol{T}_A^{\mathrm{T}}(\boldsymbol{\phi}_o)\boldsymbol{G}(\hat{\boldsymbol{r}}_1,\hat{\boldsymbol{r}}_2,\ldots,\hat{\boldsymbol{r}}_N)\boldsymbol{h}_e$ and $\boldsymbol{\gamma}_{\hat{o},env}$ represent, respectively, the wrench exerted by the aerial vehicles on the estimated object CoM and the effect on the estimated object CoM of the external wrenches exerted by the object on the environment.

In the presence of parameter estimation errors, (3.24) must be rewritten by considering $\boldsymbol{M}_{\hat{o}}$, $\boldsymbol{C}_{\hat{o}}$ and $\boldsymbol{g}_{\hat{o}}$ that are, respectively, the terms $\hat{\boldsymbol{M}}_{\hat{o}}$, $\hat{\boldsymbol{C}}_{\hat{o}}$ and $\hat{\boldsymbol{g}}_{\hat{o}}$ in (3.26), computed by using the parameters estimated in Sections 3.5.1–3.5.3.

Therefore, the dynamics in (3.25) becomes

$$\boldsymbol{A}\hat{\boldsymbol{M}}_{\hat{o}}\boldsymbol{\Delta}\ddot{\hat{\boldsymbol{x}}}_o+\boldsymbol{D}_o\boldsymbol{\Delta}\dot{\hat{\boldsymbol{x}}}_o+\boldsymbol{K}_o\boldsymbol{\Delta}\hat{\boldsymbol{x}}_o=\hat{\boldsymbol{\gamma}}_{\hat{o},env}+\boldsymbol{\epsilon}(\hat{\boldsymbol{x}}_o,\dot{\hat{\boldsymbol{x}}}_o), \quad (3.27)$$

where $\boldsymbol{\epsilon}(\hat{\boldsymbol{x}}_o,\dot{\hat{\boldsymbol{x}}}_o)$ collects the terms due to the estimation errors

$$\boldsymbol{\epsilon}(\hat{\boldsymbol{x}}_o,\dot{\hat{\boldsymbol{x}}}_o) \;=\; (\hat{\boldsymbol{C}}_{\hat{o}}-\boldsymbol{\Gamma}_o\boldsymbol{C}_{\hat{o}})\dot{\hat{\boldsymbol{x}}}_o+(\hat{\boldsymbol{g}}_{\hat{o}}-\boldsymbol{\Gamma}_o\boldsymbol{g}_{\hat{o}})+(\boldsymbol{\Gamma}_o-\boldsymbol{I}_6)\left(\boldsymbol{\gamma}_{\hat{o}}-\boldsymbol{\gamma}_{\hat{o},env}\right), \quad (3.28)$$

where $\boldsymbol{\Gamma}_o = \hat{\boldsymbol{M}}_{\hat{o}}\boldsymbol{M}_{\hat{o}}^{-1}$ can be seen as a measure of the accuracy of inertia matrix estimation. If the uncertainties $\boldsymbol{\epsilon}(\hat{\boldsymbol{x}}_o, \dot{\hat{\boldsymbol{x}}}_o)$ are negligible, the dynamics of (3.27) is close to the ideal dynamics (3.25). The simulation results in Section 3.7 prove that neglecting such term does not compromise the compliant behavior of the system.

Once the object's reference motion variables, $\mathcal{T}_{o,r}$, have been computed, the desired motion, $\mathcal{T}_{e_k,d} = \{\boldsymbol{x}_{e_k,d}, \dot{\boldsymbol{x}}_{e_k,d}, \ddot{\boldsymbol{x}}_{e_k,d}\}$ $(k = 1, 2, \ldots, N)$, can be easily obtained via the closed-chain constraints (3.15).

3.6.2 Internal Impedance

The internal impedance loop is aimed at limiting the internal wrenches on the object. To this aim, a compliant behavior of each aerial vehicle with respect to internal wrenches is enforced. The desired trajectory for each grasp location, $\mathcal{T}_{e_k,d}$, computed via the closed-chain constraints (3.15), is given as input of an internal admittance filter together with the contributes of the kth robot to the object internal wrench, $\boldsymbol{h}_{I,k}$ computed as in (3.14).

The reference trajectories for the grasp locations, $\mathcal{T}_{e_k,r} = \{\boldsymbol{x}_{e_k,r}, \dot{\boldsymbol{x}}_{e_k,r}, \ddot{\boldsymbol{x}}_{e_k,r}\}$, are computed by integrating the following differential equation $(k = 1, \ldots, N)$

$$\boldsymbol{M}_{e_k}\boldsymbol{\Delta}\ddot{\boldsymbol{x}}_{e_k} + \boldsymbol{D}_{e_k}\boldsymbol{\Delta}\dot{\boldsymbol{x}}_{e_k} + \boldsymbol{K}_{e_k}\boldsymbol{\Delta}\boldsymbol{x}_{e_k} = \boldsymbol{\gamma}_{I_k}, \tag{3.29}$$

where $\boldsymbol{M}_{e_k}$, $\boldsymbol{D}_{e_k}$ and $\boldsymbol{K}_{e_k}$ are, respectively, the virtual inertia, damping and stiffness imposed to the aerial vehicle, $\boldsymbol{\Delta}\boldsymbol{x}_{e_k} = \boldsymbol{x}_{e_k,d} - \boldsymbol{x}_{e_k,r}$ and $\boldsymbol{\gamma}_{I_k} = \boldsymbol{T}_A^{\mathrm{T}}(\boldsymbol{\phi}_o)\boldsymbol{h}_{I_k} = \boldsymbol{T}_A^{\mathrm{T}}(\boldsymbol{\phi}_o)\left[\hat{\boldsymbol{V}}\hat{\boldsymbol{V}}^{\dagger}\boldsymbol{h}_e\right]_k$, where $[\cdot]$ extracts the kth (6 × 1) block of the argument (see (3.14)) and $\hat{\boldsymbol{V}}$ is computed as in (3.12) with $\hat{\boldsymbol{r}}_k$ in lieu of $\boldsymbol{r}_k$.

3.6.3 Motion control

Once the reference poses of the grasp locations are known, it is necessary to derive the reference motion in terms of reference position and orientation of the coordinate frame $\boldsymbol{\Sigma}_{0_k}$ $(k = 1, 2 \ldots, N)$ attached to the robot platform. For the kth robot, the position and orientation of the platform frame can be written in terms of the contact frame $\boldsymbol{\Sigma}_{e_k}$ in the following way

$$\boldsymbol{p}_{0_k} = \boldsymbol{p}_{e_k} - \boldsymbol{R}_{0_k}\boldsymbol{r}_{e_k}^{0_k} \tag{3.30}$$

$$\boldsymbol{\phi}_{0_k} = \boldsymbol{\phi}(\boldsymbol{R}_{e_k}\boldsymbol{R}_{0_k}^{e_k}), \tag{3.31}$$

where $\boldsymbol{R}_{0_k}$, $\boldsymbol{R}_{e_k}$ and $\boldsymbol{R}_{0_k}^{e_k}$ are, respectively, the rotation matrices expressing the orientation of $\boldsymbol{\Sigma}_{0_k}$ and $\boldsymbol{\Sigma}_{e_k}$ with respect to the inertial frame, and $\boldsymbol{\Sigma}_{0_k}$ with respect to $\boldsymbol{\Sigma}_{e_k}$; $\boldsymbol{r}_{e_k}^{0_k}$ denotes the position of O_{e_k} with respect to O_{0_k}, expressed in $\boldsymbol{\Sigma}_{0_k}$. By differentiating (3.30)-(3.31), the linear and angular velocities and accelerations of $\boldsymbol{\Sigma}_{0_k}$ can be computed.

Finally, a motion control in charge of tracking such reference values need to be considered. As an example, and for the sake of completeness, in the following a model-based control scheme based on that designed in Chapter 2 is described. However, it is worth noticing that the adoption of other (even non model-based) control schemes is possible, since the devised strategy is, in theory, compatible with any motion controller (e.g., the sliding mode controller in [36]), provided that it ensures accurate and fast tracking of the references computed by the admittance filters [83]. The proposed approach is based on a two-loop scheme: the outer loop includes a position controller, aimed at computing the force to be applied to the vehicle's platform and the reference angles for the roll and pitch joints; the inner loop, on the basis of the vehicle's desired orientation and the reference joint angles, outputs the moment to be applied to the vehicle's platform and the torques for the roll and pitch joints. A sketch of the control block scheme is reported in Fig. 3.4.

Position controller

The force to be applied to the platform is derived by the inverse dynamics controller presented in the Chapter 2. More in detail, it is given by

$$\boldsymbol{f}_{0_k} = \boldsymbol{M}_{k,pp}\boldsymbol{a}_{k,p} + \boldsymbol{M}_{k,p\phi}\boldsymbol{a}_{k,\phi} + \boldsymbol{C}_{k,p}\dot{\boldsymbol{p}}_k + \boldsymbol{g}_{k,p} + \boldsymbol{\zeta}_{k,e,p}, \tag{3.32}$$

where $\boldsymbol{M}_{k,p\delta}$ has been neglected as in (2.1). The effectiveness of the assumption on $\boldsymbol{M}_{k,p\delta}$ has been tested in Chapter 2 via a simulation study, while, as regards the friction term $\boldsymbol{F}_{k,p}$, as highlighted in Remark 1, it can be neglected. The auxiliary inputs $\boldsymbol{a}_{k,*}$ $(* = p, \phi)$ are computed according to the PID control law shown in Chapter 2.

Once the force $\boldsymbol{f}_{0_k}$ is known, it can be adopted to compute the desired position of roll and pitch angles. To the aim, define a reference frame, $\boldsymbol{\Sigma}_{2_k}$, attached to kth rotor frame. The relative rotation of such reference frame with respect to $\boldsymbol{\Sigma}_{0_k}$ depends on roll and pitch angles α_k and β_k via the rotation matrix $\boldsymbol{R}_{2_k}^{0_k}$ obtained by using (1.4) and (1.5). By expressing $\boldsymbol{f}_{0_k}$ in $\boldsymbol{\Sigma}_{0_k}$ as

$$\boldsymbol{f}_{0_k}^{0_k} = \boldsymbol{R}_{2_k}^{0_k} \begin{bmatrix} 0 \\ 0 \\ \|\boldsymbol{f}_{0_k}\| \end{bmatrix}, \tag{3.33}$$

where $\boldsymbol{f}_{0_k}^{k} = \boldsymbol{R}_{0_k}^{\mathrm{T}}\boldsymbol{f}_{0_k}$, the desired values, $\alpha_{k,d}$ and $\beta_{k,d}$, of the joint angles can be obtained as in Section 2.1.1. The desired velocity and acceleration of the joints can be computed via numerical differentiation as in Chapter 2.

Attitude and roll-pitch controller

The inner loop motion controller is in charge of computing the moment to be applied to the platform and the joint torques. Similarly to (2.2) they can be obtained as

$$\begin{aligned} \boldsymbol{\mu}_{0_k} &= \boldsymbol{M}^{\mathrm{T}}_{k,p\phi}\boldsymbol{a}_{k,p}+\boldsymbol{M}_{k,\phi\phi}\boldsymbol{a}_{k,\phi}+\boldsymbol{M}_{k,\phi\delta}\boldsymbol{a}_{k,\delta}+\boldsymbol{C}_{k,\phi}\dot{\boldsymbol{\phi}}_k+\boldsymbol{g}_{k,\phi}+\boldsymbol{\zeta}_{k,e,\phi} \\ \boldsymbol{\tau}_{\delta_k} &= \boldsymbol{M}^{\mathrm{T}}_{k,p\delta}\boldsymbol{a}_{k,p}+\boldsymbol{M}^{\mathrm{T}}_{k,\phi\delta}\boldsymbol{a}_{k,\phi}+\boldsymbol{M}_{k,\delta\delta}\boldsymbol{a}_{k,\delta}+\boldsymbol{C}_{k,\delta}\dot{\boldsymbol{\delta}}_k+\boldsymbol{g}_{k,\delta}+\boldsymbol{J}^{\mathrm{T}}_{2_k}\boldsymbol{h}_{2_k}+\boldsymbol{\zeta}_{k,e,\delta}, \end{aligned} \tag{3.34}$$

where $\boldsymbol{J}^{\mathrm{T}}_{2_k}\boldsymbol{h}_{2_k}$ is the effect of the propeller thrust on the roll and pitch joints as shown in (1.36) and the auxiliary input $\boldsymbol{a}_{k,\delta}$ can be computed according to a PID control law. It is worth noticing that $\boldsymbol{F}_{k,\phi}$ and $\boldsymbol{F}_{k,\delta}$ have been neglected for the same reasons reported in Remark 1.

The mapping from the generalize force and moment, expressed in the rotor frame, to the propellers thrusts $\boldsymbol{\sigma}_k$, according to (2.16), can be computed as

$$\boldsymbol{\sigma}_k = \boldsymbol{N}^{-1}\begin{bmatrix} \|f_{b,2_k}\| \\ \boldsymbol{\mu}^{2_k}_{b,2_k} \end{bmatrix}. \tag{3.35}$$

3.7 Simulation case study

The proposed cooperative control scheme has been tested via realistic simulations by using the MATLAB© SimMechanics environment.

A team of three robots is commanded to move the object depicted in Fig. 3.3, whose dynamic parameters are supposed to be unknown to the robots, along a planned path.

The control gains are reported in Table 3.1.

	$\boldsymbol{K}_{D,*}$	$\boldsymbol{K}_{P,*}$	$\boldsymbol{K}_{I,*}$
Position	$12\,\boldsymbol{I}_3$	$30\,\boldsymbol{I}_3$	$3\,\boldsymbol{I}_3$
Orientation	$\begin{bmatrix} 2500 & 0 & 0 \\ 0 & 2500 & 0 \\ 0 & 0 & 25 \end{bmatrix}$	$\begin{bmatrix} 240 & 0 & 0 \\ 0 & 240 & 0 \\ 0 & 0 & 75 \end{bmatrix}$	$\begin{bmatrix} 640 & 0 & 0 \\ 0 & 640 & 0 \\ 0 & 0 & 200 \end{bmatrix}$
Joints	$1500\,\boldsymbol{I}_2$	$240\,\boldsymbol{I}_2$	$600\,\boldsymbol{I}_2$

Table 3.1: Motion control gains

In order to simulate a realistic scenario the following assumptions have been done:

- Measurements of position and orientation of the aerial robots are affected by noise with mean $1 \cdot 10^{-2}$ m for position and 0 rad for the orientation, and a standard deviation of 10^{-3} m for position and 10^{-3} rad for the orientation.
- The platform linear velocity are obtained via derivative filter, while angular velocity is assumed measurable with a 0 mean measurement noise and standard deviation 10^{-3} rad/s.
- A force-torque sensor is mounted at the grasping point.
- Centrifugal and Coriolis terms have been neglected in control laws (2.2) and (2.15).
- Viscous friction has been modeled but not considered in the control design.
- All the data are available with a frequency rate of 500 Hz, which is commonly achievable in an aerial robotics setup (see, e.g., [78]).

Moreover, the robustness of the proposed algorithm to disturbances has been tested by considering the aerodynamic drag, acting on the object, modeled as

$$\boldsymbol{F}_{\text{ad}} = \frac{1}{2}\rho \boldsymbol{R}_o \boldsymbol{S}_{\text{sgn}} \begin{bmatrix} (w_x^o - \dot{p}_x^o)^2 \\ (w_y^o - \dot{p}_y^o)^2 \\ (w_z^o - \dot{p}_z^o)^2 \end{bmatrix}, \tag{3.36}$$

where $\rho \in \mathbb{R}$ is the air density, $\boldsymbol{R}_o \in \mathbb{R}^3$ is the object rotation matrix, $w_\star^o$ $(\star = x, y, z)$ are the wind velocity components expressed in the object reference frame and $\dot{p}_\star^o$ are the components of the object center of mass velocity. The term $\boldsymbol{S}_{\text{sgn}} \in \mathbb{R}^{3\times 3}$ is the diagonal matrix

$$\boldsymbol{S}_{\text{sgn}} = \begin{bmatrix} \text{sgn}(w_x^o - \dot{p}_x^o)s_x & 0 & 0 \\ 0 & \text{sgn}(w_y^o - \dot{p}_y^o)s_y & 0 \\ 0 & 0 & \text{sgn}(w_z^o - \dot{p}_z^o)s_z \end{bmatrix}, \tag{3.37}$$

where $s_\star$ $(\star = x, y, z)$ is the maximum section normal to the $\star$ axes of the object reference frame, $\text{sgn}(\cdot)$ is the signum function. The wind velocity has been modeled via the following expression [84]

$$\boldsymbol{w} = \sin\left(\frac{2\pi}{T_w}(t - t_w)\right)\boldsymbol{w}_b, \tag{3.38}$$

where $T_w \in \mathbb{R}$ is the period of the sine, $t_w \in \mathbb{R}$ is the initial time in which the wind gust starts and $\boldsymbol{w}_b \in \mathbb{R}^3$ is the vector of the maximum amplitude of the wind components along the three axes of the inertial base frame.

The parameters of the aerodynamic drag used for the simulations are reported in Table 3.2.

Parameter	Value
ρ	1.225 kg/m^3
T_w	20 s
t_w	2 s
$\boldsymbol{w}_b^{\mathrm{T}}$	[5 5 2] m/s
$\begin{bmatrix} s_x & s_y & s_z \end{bmatrix}$	[0.007 0.010 0.940] m^2

Table 3.2: Parameters of the aerodynamic disturbance.

3.7.1 Object parameter estimation

In the first stage, the system is moved in order to estimate the object parameters. The estimation performance of a generic vector of parameters $\boldsymbol{v}$ are evaluated via the relative error

$$\varepsilon_{\%} = \frac{\|\boldsymbol{v} - \hat{\boldsymbol{v}}\|}{\|\boldsymbol{v}\|} \cdot 100. \tag{3.39}$$

In order to estimate the object mass, m_o, the system is kept in hovering position for 10 s and the corresponding contact wrenches are measured. Since the force/torque measurements are affected by white noise, the mass estimation value, obtained via (3.17), has been set as the mean value of the estimates obtained during the hovering phase. The results are reported in Table 3.3: it can be noticed that the estimated mass is very close to the actual value and the relative estimation error is below the 0.5%.

Parameter	m_o
Actual [kg]	1.2
Estimated [kg]	1.2059
Std. deviation [kg]	0.1099
Avg. error $\varepsilon_{\%}$	0.49%

Table 3.3: Performance of the estimation of the object inertia matrix.

In order to estimate the object CoM, the reference frame $\boldsymbol{\Sigma}_{e_l}$ has been considered coincident with $\boldsymbol{\Sigma}_{e_1}$ and six different poses have been selected (i.e, $\nu = 6$ in (3.19)-(3.20)). In order to consider statistically significant values, the estimation procedure has been replicated 10 times, in which the object's poses have been randomly selected. By referring to the relative position $\boldsymbol{r}_k^o$ ($k = 1, 2, 3$) of the contact point of each robot with respect to the object CoM, the performance are evaluated by comparing the actual and the estimated values. The obtained results, in terms of average and maximum error, are reported in Table 3.4, where the estimated values are computed as the mean value of the 10 replications.

Parameter	$\boldsymbol{r}_1^o$	$\boldsymbol{r}_2^o$	$\boldsymbol{r}_3^o$
Actual [mm]	$[1011 \quad 738 \quad 0]$	$[62 \quad -365 \quad 0]$	$[-989 \quad -265 \quad 0]$
Estimated [mm]	$[1004 \quad 735 \quad 0]$	$[50 \quad -372 \quad -14]$	$[-971 \quad -256 \quad 13]$
Std. deviation [mm]	$[7 \quad 6 \quad 7]$	$[6 \quad 7 \quad 9]$	$[10 \quad 2 \quad 11]$
Avg. error $\varepsilon_{\%}$	1.31%	2.14%	1.47%
Max. error $\varepsilon_{\%}$	2.16%	3.14%	2.67%

Table 3.4: Performance of the estimation of the object center of mass.

Regarding the estimation of the inertia tensor, a trajectory, with duration of 20 s, involving only orientation changes, has been commanded to the object. During the execution, 905 samples have been evenly extracted for the least-squares estimation. As in the previous case, the estimation has been replicated 10 times, with different trajectories randomly planned. As shown in Table 3.5, the error magnitude is about 10^{-5} $kg \cdot m^2$, that is acceptable and comparable with usual uncertainty on inertial parameters.

Parameter	$\boldsymbol{\zeta}$
Actual [kg · m$^{2]}$	$[0.1019 \quad -0.1360 \quad 0.0000 \quad 0.2813 \quad 0.0000 \quad 0.3832]$
Estimated [kg · m$^{2]}$	$[0.1019 \quad -0.1360 \quad -0.0000 \quad 0.2812 \quad -0.0000 \quad 0.3832]$
Std. deviation [kg · m$^{2]}$	$10^{-4}\,[0.2104 \quad 0.1040 \quad 0.0988 \quad 0.1388 \quad 0.0965]$
Avg. error $\varepsilon_{\%}$	0.24%
Max. error $\varepsilon_{\%}$	0.92%

Table 3.5: Performance of the estimation of the object inertia properties.

3.7.2 Impedance control

In order to test the proposed impedance control approach, a linear trajectory for the position of the object CoM in the inertial frame, from the point $[0\ 0\ 1]^{\mathrm{T}}$ m to the point $[2\ 3\ 0]^{\mathrm{T}}$ m, has been commanded. The commanded trajectory for the orientation of $\boldsymbol{\Sigma}_o$ starts from the initial value of the Euler angles $[0\ \ 0\ \ 0]^{\mathrm{T}}$ deg and a displacement of $[18\ \ 18\ \ 0]^{\mathrm{T}}$ deg is imposed. Along the path, the object comes in contact with the external environment, assumed to be visco-elastic, i.e., when the object is in contact with the environment, an elastic wrench is applied to its CoM, which has been modeled as

$$\boldsymbol{\gamma}_{env} = \boldsymbol{K}_{env}\boldsymbol{\Delta x}_{env} + \boldsymbol{D}_{env}\boldsymbol{\Delta}\dot{\boldsymbol{x}}_{env}, \tag{3.40}$$

where

$$\boldsymbol{K}_{env} = \begin{bmatrix} \boldsymbol{K}_{env,p} & \boldsymbol{O}_3 \\ \boldsymbol{O}_3 & \boldsymbol{K}_{env,o} \end{bmatrix}, \quad \boldsymbol{D}_{env} = \begin{bmatrix} \boldsymbol{D}_{env,p} & \boldsymbol{O}_3 \\ \boldsymbol{O}_3 & \boldsymbol{D}_{env,o} \end{bmatrix}, \tag{3.41}$$

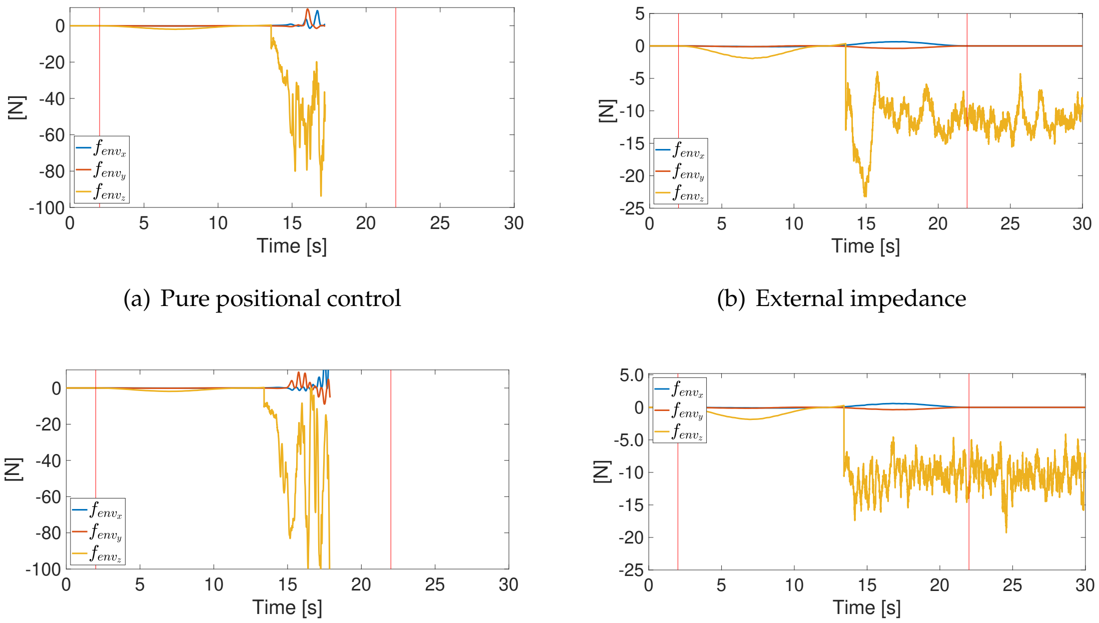
$\boldsymbol{\Delta x}_{env}$ and $\boldsymbol{\Delta}\dot{\boldsymbol{x}}_{env}$ are the environment deformation and its time derivative, respectively. More in detail, the stiffness of the environment has been set as $\boldsymbol{K}_{env,p} = \mathrm{diag}(1000\ 1000\ 1000)$ N/m for the linear part and $\boldsymbol{K}_{env,o} = \mathrm{diag}(2.5\ \ 4\ \ 1)$ Nm/rad, for the torsional part, while the damping coefficients are $\boldsymbol{D}_{env,p} = \mathrm{diag}(100\ \ 100\ \ 100)$ Ns/m and $\boldsymbol{D}_{env,o} = \mathrm{diag}(0.01\ 0.01\ 0.01)$ Nms/rad. The duration of the simulation is 30 s: for the first 5 s the object keeps the initial pose, then, for the next 10 s, it moves along the reference trajectory; finally, there is an hovering phase of 15 s. The impedance filter gains are summarized in Table 3.6.

Gain	Value
$\boldsymbol{A}$	$29 \cdot \boldsymbol{I}_6$
$\boldsymbol{D}_o$	$\mathrm{diag}[200 \cdot \boldsymbol{I}_3, 100 \cdot \boldsymbol{I}_3]$
$\boldsymbol{K}_o$	$\mathrm{diag}[150 \cdot \boldsymbol{I}_3, 15 \cdot \boldsymbol{I}_3]$
$\boldsymbol{M}_e$	$2.9 \cdot \boldsymbol{I}_6$
$\boldsymbol{D}_e$	$\mathrm{diag}[25, 25, 50, 25, 50, 50]$
$\boldsymbol{K}_e$	$\mathrm{diag}[10, 10, 5, 10, 10, 10]$

Table 3.6: Impedance filter gains.

Four different simulations have been carried out: in the first one, a pure positional control is adopted, i.e., both the impedance filters have been disabled; in the second and third simulations, only the internal and external impedance, respectively, are activated; finally, in the last simulation both the impedance filters are activated.

In Figs. 3.5 and 3.6 the comparison of the interaction forces and moments, respectively, exerted by the object on the environment is reported. It is worth noticing that, in the absence of the external impedance (see Figs. 3.5(a), 3.5(c) and Figs. 3.6(a), 3.6(c)),



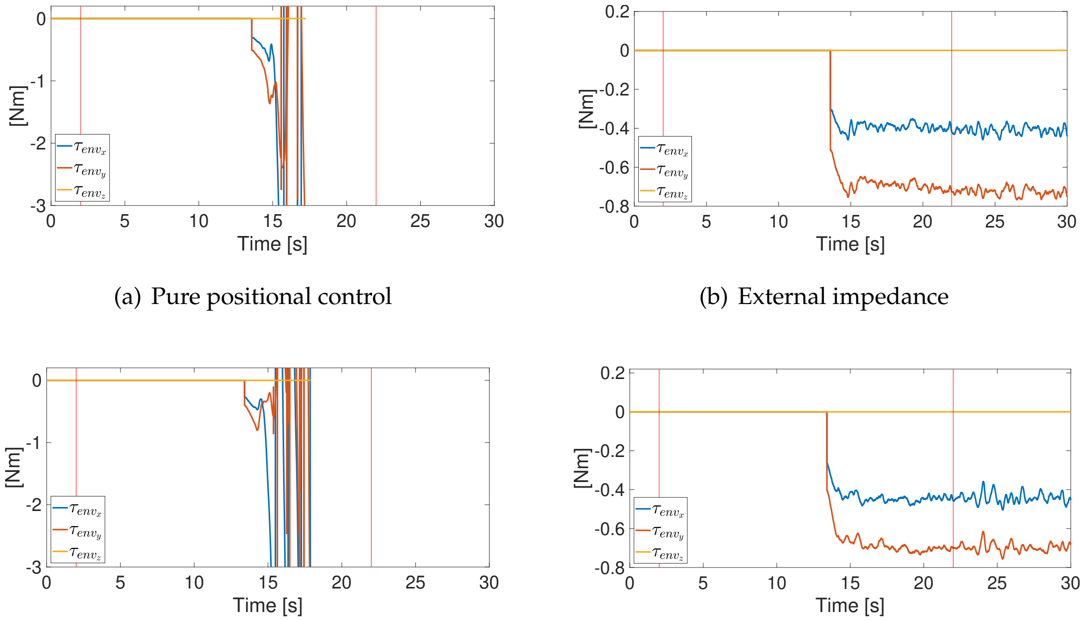
(a) Pure positional control

(b) External impedance

(c) Internal impedance

(d) External & internal impedance

Figure 3.5: External forces: pure positional control (a), only external impedance (b), only internal impedance (c), external+internal impedance (d).



(a) Pure positional control

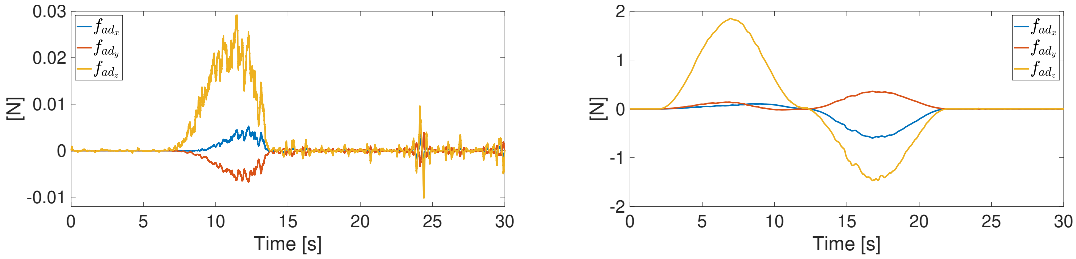
(b) External impedance

(c) Internal impedance

(d) External & internal impedance

Figure 3.6: External torques: pure positional control (a), only external impedance (b), only internal impedance (c), external+internal impedance (d).

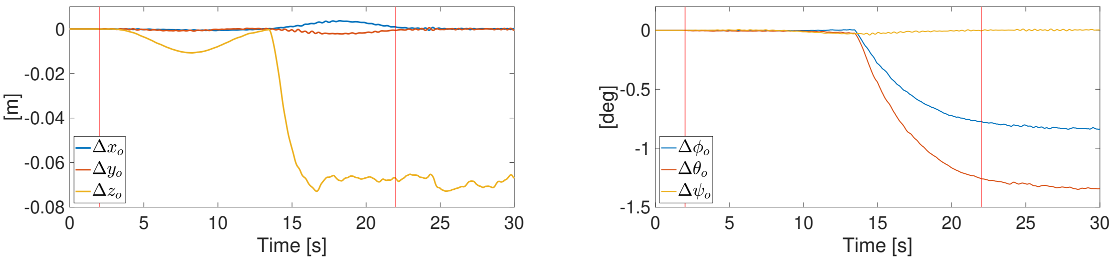
when the contact arises, i.e. after about 13 s, the interaction wrenches lead the system to instability and the simulation failes at about 18 s. Such an effect is coherent with the physical behaviour of a real aerial vehicle experienced, e.g., in [85]. On the other side, the external impedance (see Figs. 3.5(b), 3.5(d) and Figs. 3.6(b), 3.6(d)) allows to limit the external contact force below 10 N and the maximum external moment at around 0.6 Nm. The vertical red lines in Figs. 3.5 and 3.6 correspond to the time interval of wind action (between 2 and 22 s). As expected (see Fig. 3.5), a corresponding sinusoidal disturbance affects the forces acting on the object, due to the aerodynamic drag. The time history of the aerodynamic drag is reported in Fig. 3.7; in detail, Fig. 3.7(a) reports the drag in the absence of wind, while Fig. 3.7(b) reports the combined effect of the wind and object motion on the aerodynamic drag. It is worth noticing that the aerodynamic drag in the absence of wind is almost negligible, since the system's velocity is relatively low.



(a) Aerodynamic drag no wind

(b) Aerodynamic drag wind

Figure 3.7: Aerodynamic drag: aerodynamic effect due to the object motion (a), aerodynamic effect due to the object motion and wind velocity (b).



(a) Position

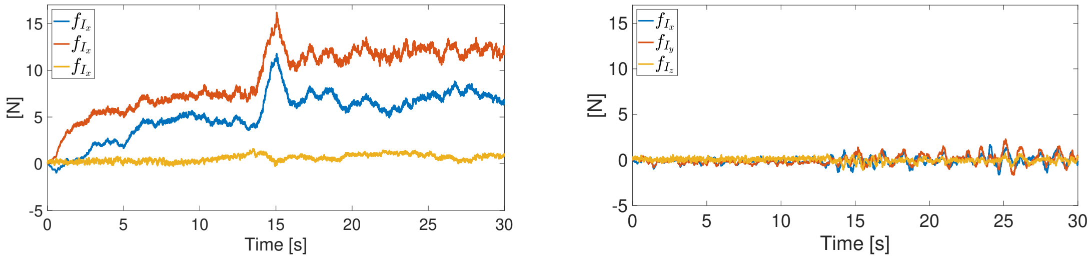
(b) Orientation

Figure 3.8: External impedance error (external+internal impedance control).

Of course, the wrench limitation is obtained at the expense of the tracking performance, as shown in Fig. 3.8, where the difference between the planned desired object position (Fig.3.8(a)) and orientation (Fig.3.8(b)) and the reference position and orientation computed by the external impedance (3.24) are reported. It is worth noticing that, in the presence of the disturbance, the reference trajectory deviates from the planned one, since the aerodynamic drag is perceived as an external force by the impedance. In order to avoid this effect, especially in the presence of high wind velocity, a dead zone for the measured wrench in (3.25) can be used, as e.g. in [78].

In Figs. 3.9 and 3.10 the contribution of one of the vehicles to the internal force and moments, respectively, ($\boldsymbol{h}_{I_k}$ in (3.14)) are shown; for the sake of brevity only the results of the simulations with the external impedance activated are reported. It can be noticed that the internal impedance strongly reduced the internal stresses.

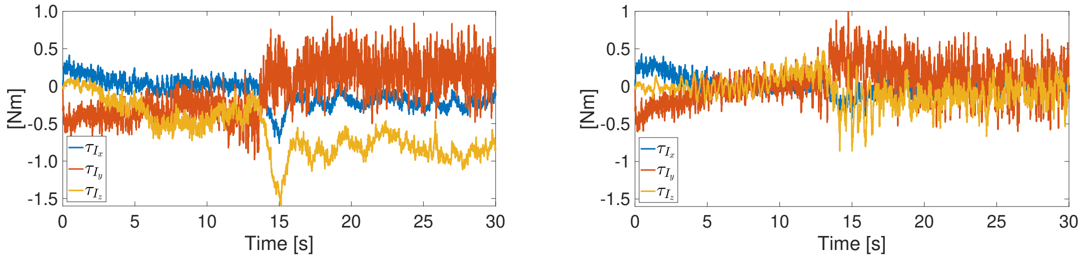
Finally, the object tracking errors, $\boldsymbol{x}_{o_r} - \boldsymbol{x}_o$, are shown in Figs. 3.11 and 3.12.



(a) External impedance

(b) External+internal impedance

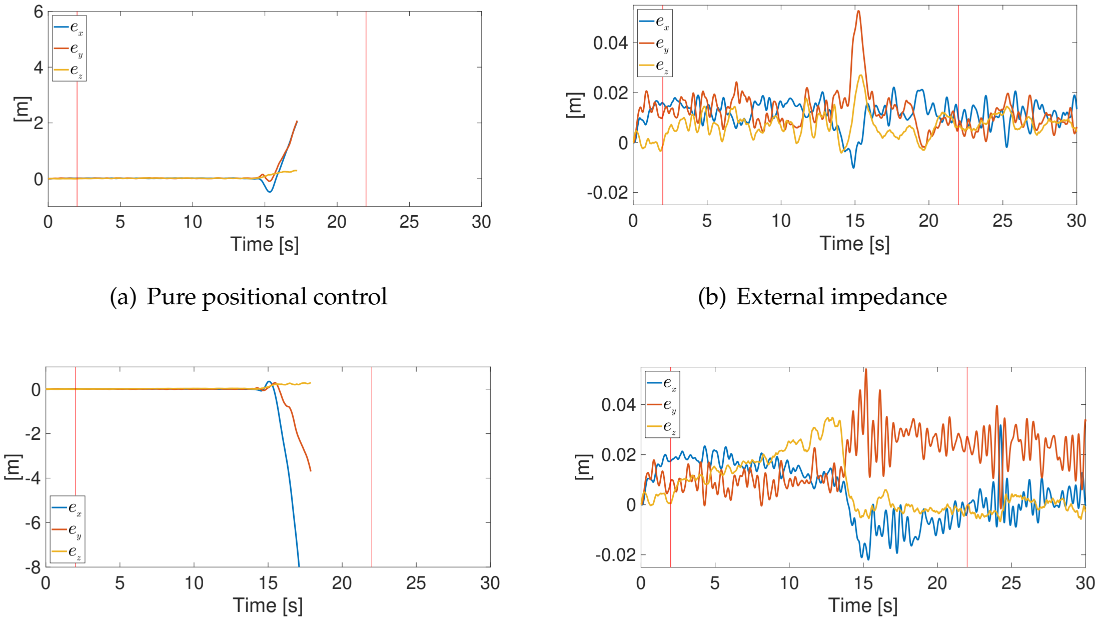
Figure 3.9: Internal forces: only external impedance (a), external+internal impedance (b).



(a) External impedance

(b) External+internal impedance

Figure 3.10: Internal momenta: only external impedance (a), external+internal impedance (b).



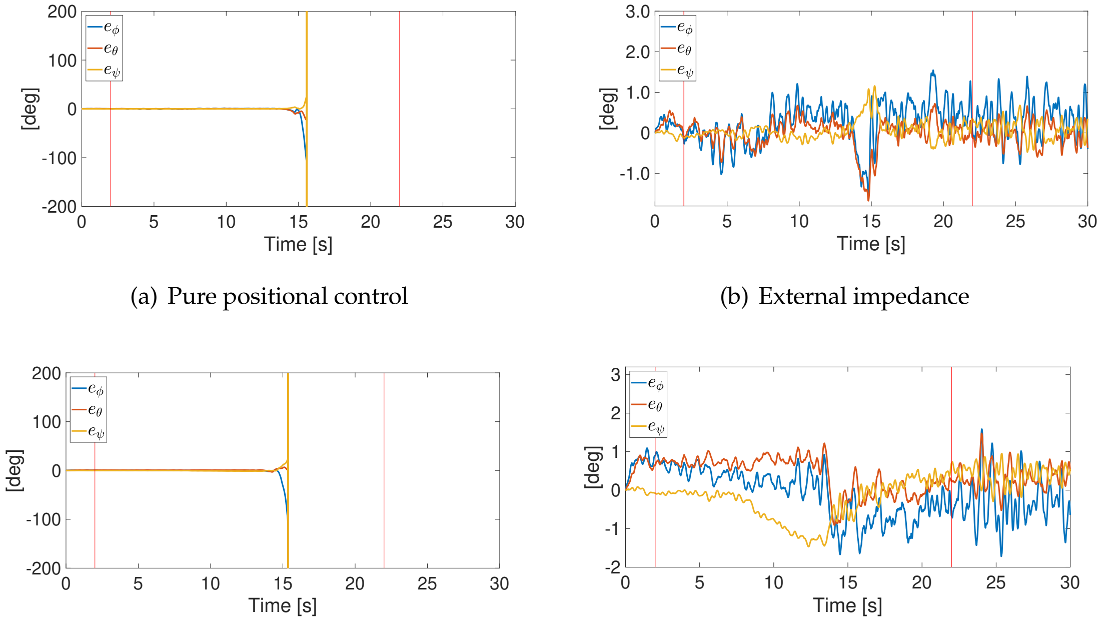
(a) Pure positional control

(b) External impedance

(c) Internal impedance

(d) External+internal impedance

Figure 3.11: Object position tracking errors: pure positional control (a), only external impedance (b), only internal impedance (c), external+internal impedance (d).



(a) Pure positional control

(b) External impedance

(c) Internal impedance

(d) External+internal impedance

Figure 3.12: Object orientation tracking errors: pure positional control (a), only external impedance (b), only internal impedance (c), external+internal impedance (d).

In the absence of the external impedance, also the tracking errors progressively increase before the simulation fails. In the other cases, the tracking performance are quite good and, as expected, the errors slightly increase only in the first seconds after the contact.

3.8 Conclusion

In this chapter, an impedance control scheme for a team of omnidirectional flying robots grasping an unknown object is proposed. The approach aims at reducing both interaction wrenches and mechanical stresses at the object. The adoption of omnidirectional aerial robots in lieu of standard quadrotors, allows the exertion of both forces and moments at the contact points, without modifying the orientation of the load. Thus, an increased robustness to external disturbances is expected, as well as a general improvement of manipulation capabilities. Although omnidirectional aerial platforms are characterized by an increased complexity, in terms of both hardware and control software, which could hinder their practical applicability, recent advances and experimental demonstrations provide solid foundations for the development of more complex applications, such as the one proposed in this chapter. The approach is based on a two-loop control scheme. A major challenge for the design and the implementation of the inner motion loop is to guarantee accurate and fast tracking of the references computed by the outer interaction control loop. On the other hand, the outer loop needs wrench measurement provided by a dedicated sensor, which might be an additional source of costs, complexity and failures. The use of wrench observers in lieu of sensors has been successfully demonstrated in the execution of less demanding tasks (e.g., interaction of a single aerial vehicle with the environment), but it could be worth investigating their use in cooperative aerial systems as well. Thus, future work on experimental validation of the proposed scheme on a real setup is needed to address the above mentioned challenges. Even though the ODQuad is still under development, the proposed approach is planned to be applied and experimentally tested by using VTOL vehicles with tiltable rotors, by considering physical interaction with humans as well.

Chapter 4

Prototyping

Introduction

In Chapter 1 the concept of the ODQuad has been introduced by showing a CAD model of the multirotor and its three main components. The first ODQuad validation has been performed by using a CAD software, the vehicle dimensions and weights have been defined and then the presence of possible collisions between the three parts has been verified. In Chapter 2, the kinematic and dynamic models derived in Chapter 1, have been used for designing the motion controller and its effectiveness has been tested by means of numerical simulations, which prove the feasibility of the vehicle and the control system. With this certainty, in order to have a first ODQuad prototype for further validation by experimental tests, a prototyping phase has been started. Once the CAD model is available, it is possible to define a bill of materials in order to discriminate the components off-the-shelf and those that must be manufactured. Typically, in common multirotor development, all the mechanical and electronic components are available on the market, but for new concepts, such as the ODQuad, many components must be custom fabricated. Nowadays, rapid prototyping tools, such as 3D printers, are affordable and allow the fabrication of custom components without the need of sophisticated CNC machining. The ODQuad prototype has been assembled starting from the *Holybro X500 kit*, from which the electronic components, the four propellers and the carbon fiber arms have been taken. The other parts, i.e. the frames, the rotation shafts and the junctions have been built, respectively, in aluminum, steel and 3D printed. The Chapter is organized into three main parts and each reports a different phase of the prototype development. More in detail, the first part reports the CAD model used in the simulation and a static analysis, which investigates the torque values for a rough sizing of the roll-pitch joint actuators. In the second part is detailed the second prototype model conceived to be realized. Finally, the phases of the prototype realization are described, including the assembly of the mechanical parts, the electrical wiring and the communication scheme.

4.1 Prototype Model: ODQuad.01

As well known the CAD modeling of a mechanical assembly and its components is an iterative process. Indeed, even for the ODQuad the third version has been reached. The first one, named ODQuad.01, (shown in Fig.4.1), is a simplified version for numerical simulations, which has only the basic parts, for reducing the computational load. For this reason, the junctions between the assembled parts are considered rigid, i.e. without bolts and screws, and the propellers are not actuated.

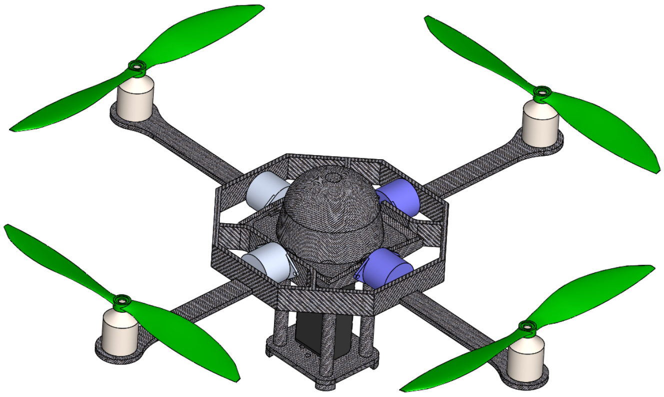
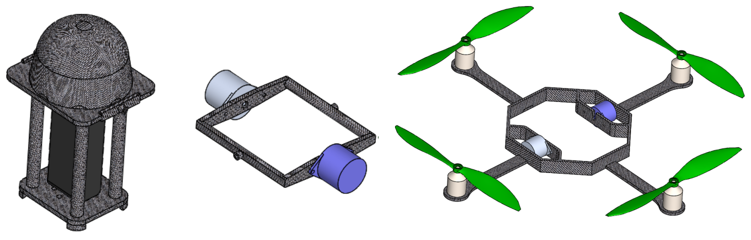


Figure 4.1: First CAD model of the ODQuad.

4.1.1 CAD Modeling

All the components of the multirotor have been designed in Solidworks and, with the same software, they have been assembled in three main sub-assemblies, i.e. the platform, the mobile frame and the rotor frame (Fig. 4.2). These three components represent the three main parts of the vehicle, which, once connected by the roll and the pitch joints, compose the ODQuad. In the first version, all the mechanical frames have been modeled in carbon fiber. To make the vehicle geometrically symmetric, two counterweights with the same inertial properties of the roll and pitch joint actuators, have been added to the mobile and rotor frames (blue components in Figs. 4.2(b) and 4.2(c)).

In Table 4.1 the size of the vehicle parts and the propeller properties are reported,



(a) (b) (c)

Figure 4.2: Platform (a), Mobile frame (b), Rotor frame (c).

while Table 4.2 shows the components inertial parameters expressed in their attached reference frame.

Table 4.1: Main model parameters

Parameter	Value
Vehicle total mass	1.378 kg
Vehicle arm length l	0.212 m
Propellers diameter	0.2 m
$c = \gamma_d/\gamma_t$	0.1

4.1.2 Roll-pitch joint actuation torque

The roll-pitch actuators of the ODQuad.01 have been designed with a overall dimensions such to guarantee a torque of a few Nm, which has been determined with a static analysis. This study included two tests: in the first one, the static torque to keep the platform in a given position has been computed by constraining the rotor frame (Fig. 4.3), while in the second one, with the platform constrained (Fig. 4.4), the roll-pitch joint torques that balance the maximum resultant moment generated by the propellers have been evaluated.

Table 4.2: Inertia parameters of main ODQuad.01 components

Component	Mass [kg]	COM [mm]	Inertia [kg/mm^2]
Platform	0.699	$\begin{bmatrix} 0 \\ 0 \\ -28.97 \end{bmatrix}$	$\begin{bmatrix} 2057338.56 & 0.78 & 0 \\ 0.78 & 2115794.25 & 0 \\ 0 & 0 & 296984.98 \end{bmatrix}$
Mobile frame	0.128	$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 435289.96 & 0 & 0 \\ 0 & 53069.66 & 0 \\ 0 & 0 & 477071.65 \end{bmatrix}$
Rotor frame	0.551	$\begin{bmatrix} 0 \\ 0 \\ 4.46 \end{bmatrix}$	$\begin{bmatrix} 7245233.16 & 0 & 0 \\ 0 & 7710954.68 & 0 \\ 0 & 0 & 14761136.92 \end{bmatrix}$

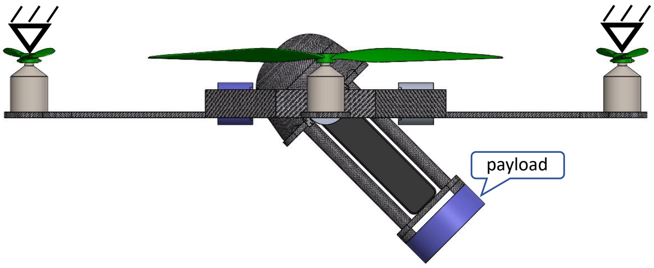


Figure 4.3: Schematic of the rotor frame constrained.

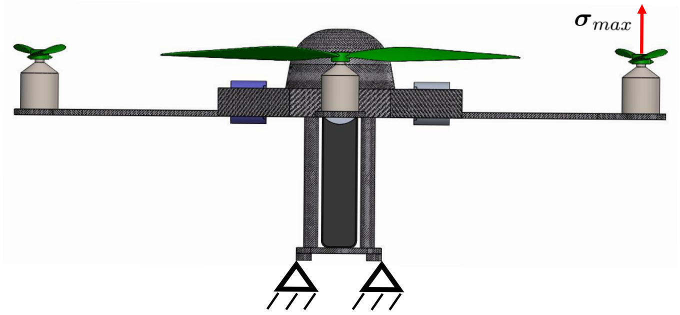


Figure 4.4: Schematic of the platform constrained..

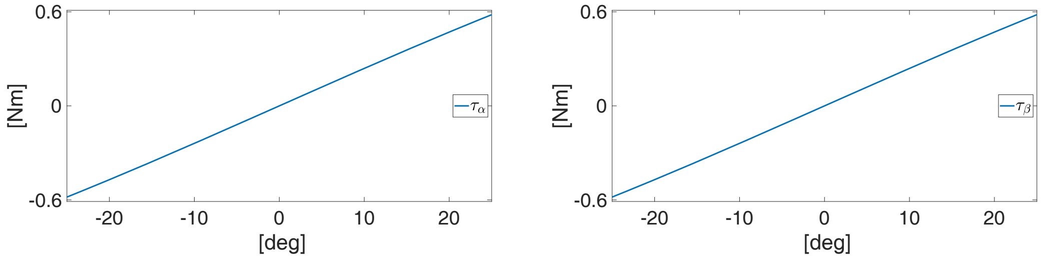
In particular, the first analysis is carried out by considering the rotor frame constrained, was performed by varying the roll and pitch joints from -25 deg to 25 deg and, to test a real scenario, a payload, with mass $m_p = 1$ kg (Fig.4.3), was connected to the platform. In this case, the ODQuad dynamic model (2.32) can be rewritten as

$$\boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{\zeta} + \boldsymbol{J}_e^{\mathrm{T}} \boldsymbol{h}_e, \tag{4.1}$$

where $\boldsymbol{J}_e$, as in (2.33), relates the COM velocity of the attached payload to the velocity of the generalized coordinates, $\dot{\boldsymbol{q}}$, and $\boldsymbol{h}_e = [0\ 0\ -m_p g\ 0\ 0\ 0]^{\mathrm{T}}$ represents the payload gravity force. By using (4.1), the term $\boldsymbol{\zeta}$ that contains the actuation torques $\boldsymbol{\tau}_\delta$ can be derived and, by exploiting (1.36), the actuation torque $\boldsymbol{\tau}_c$ can be computed as

$$\boldsymbol{\tau}_c = \boldsymbol{\tau}_\delta - \boldsymbol{J}_2^{\mathrm{T}} \boldsymbol{h}_2, \tag{4.2}$$

where the term $\boldsymbol{J}_2^{\mathrm{T}} \boldsymbol{h}_2$ is equal to zero, since the rotor frame was constrained. The results of such analysis are reported in Fig.(4.5) where it can be noticed that the magnitude of the maximum required torque is about 0.6 Nm.



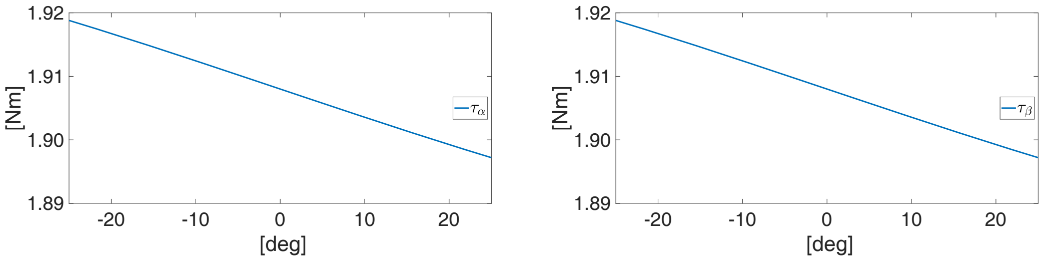
(a) (b)

Figure 4.5: Results of the static analysis of the model with the the rotor frame constrained (Fig. 4.3). Roll joint torque (a), Pitch joint torque (b).

In the second analysis, where the platform is considered constrained, it has been assumed that the resultant moment generated by the propellers reaches its maximum value. To this aim, only two propellers generate the maximum thrust while the other two are kept idle. The roll-pitch joint torques can be computed by using (4.2), where the term $\boldsymbol{J}_2^{\mathrm{T}} \boldsymbol{h}_2$ is not null and the moment generated by the propellers can be computed by using (1.37), rewritten as

$$\begin{bmatrix} \|\boldsymbol{f}_0\| \\ (\boldsymbol{J}_{O_2}^{\mathrm{T}})^{\dagger} \boldsymbol{\zeta}^2 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ l & 0 & -l & 0 \\ 0 & -l & 0 & l \\ c & -c & c & -c \end{bmatrix} \begin{bmatrix} \sigma_{max} \\ \sigma_{max} \\ 0 \\ 0 \end{bmatrix}, \tag{4.3}$$

where σ_{max} is the maximum thrust that the propellers can provide. The results have been reported in Fig. 4.6, and the largest torque value measured is about 1.9 Nm, which is quite higher than 0.6. However, this condition should not be considered in practice, since it could compromise the stability of the vehicle. In Section 4.2, by exploiting these results and those reported in Chapter 2, the roll and pitch joints actuators have been selected.



(a)

(b)

Figure 4.6: Results of static analysis with the platform constrained (Fig. 4.4). Roll joint torque (a), Pitch joint torque (b).

4.2 Prototype model: ODQuad.02

Since the first simplified version of the ODQuad prototype has been designed for simulations, hence it is not suitable to be practically realized, since the components specifications and the related manufacturing techniques have not been taken into account. For this reason, a preliminary choice of roll-pitch joint actuators was done.

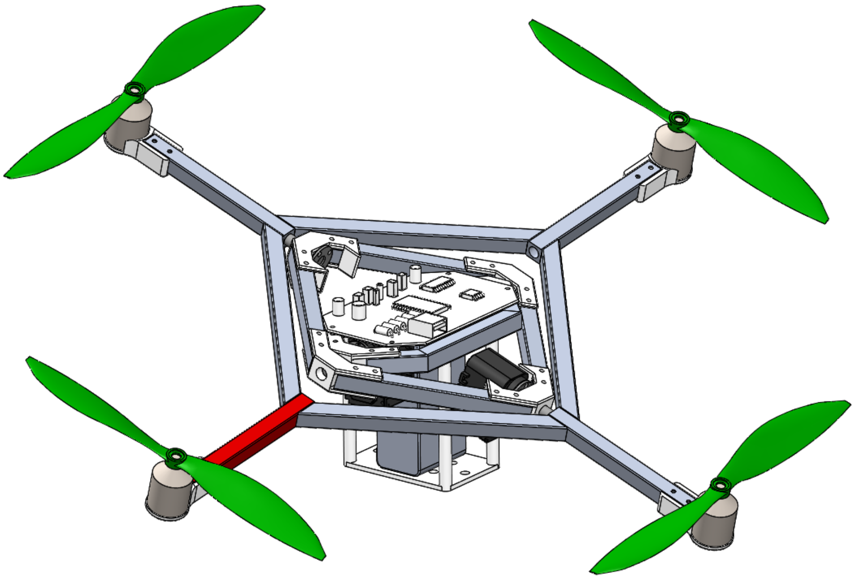


Figure 4.7: CAD model of the ODQuad.02

The CAD model has been developed by assuming the *Dynamixel AX 12A* motors, which is characterized by a stall torque of 1.5 Nm: this value is suitable for the normal use of the vehicle, such as for object transportation and for other manipulation tasks. Indeed, in the dynamic simulations reported in Section 2.5, the largest torque value reached leaving an object transportation task is less than 0.5 Nm. To develop the prototype, a new CAD model, named ODQuad.02 (Fig.4.7), was created by using aluminum profiles, 3D printed parts and steel for the roll and pitch joints shafts. The main components involved in the ODQuad.02 prototype are shown in Fig. 4.8.

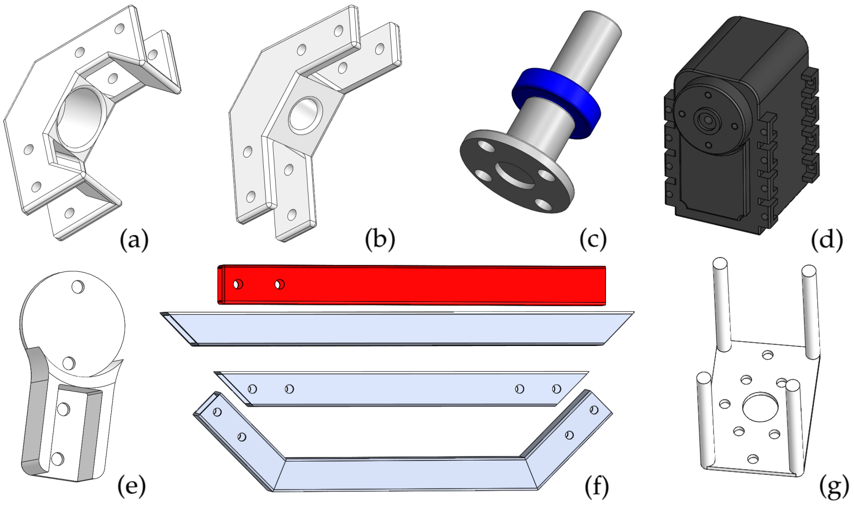


Figure 4.8: Junction and bearing support (a), junction (b), roll-pitch shaft and bearing (c), Dynamixel motor (d), propeller motor support (e), aluminum profiles (f), battery support (g).

In particular, the aluminum profiles (Fig. 4.8(f)) are assembled by using the junctions reported in Fig. 4.8(a) and Fig. 4.8(b), in such a way to build the three main frames (i.e., platform, mobile and rotor frames) of the vehicle. The junctions and the supports of the propellers motors (Fig.4.8(e)) and the battery support (Fig.4.8(g)) are designed to be 3D printed in ABS, while the aluminum profiles and the shaft have been manufactured by using conventional machining tools. The roll and picth joints are realized using 4 shafts and 4 bearings assembled via the mechanical coupling shown in Fig.4.9. In particular, the shafts of the roll joint are constrained to the mobile frame, while, the pitch shafts are locked to the rotor frame.

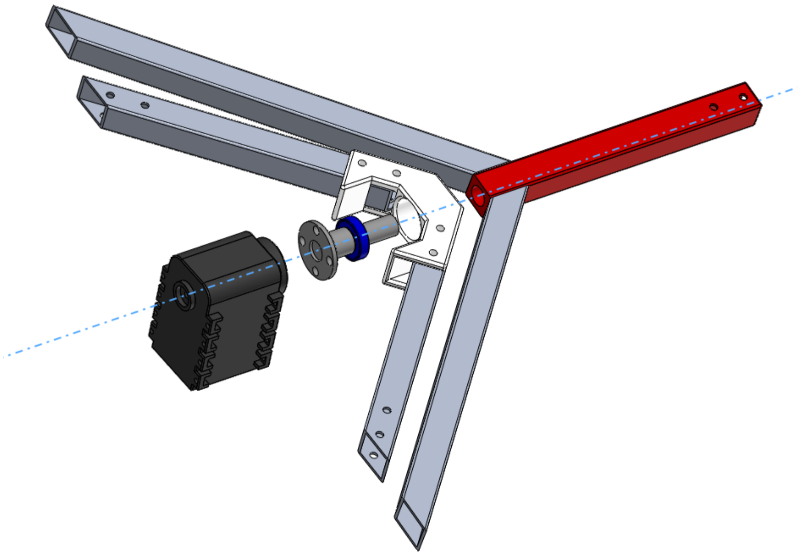


Figure 4.9: Detail of the pitch joint mechanical coupling.

4.3 ODQuad Prototype

The CAD models of the first two prototypes have been presented in Sections 4.1 and 4.2. The first one has been designed for the ODQuad concept validation via simulations and the second one for a preliminary mechanical design. On the basis of the previous analyses, the final version of the ODQuad prototype has been developed by using some parts of the quadcopter *Holybro X500* (Fig.4.10)(a), which were integrated in the new CAD model ODQuad.03 (Fig.4.10)(b). The frame has been manufactured with aluminum profiles with a $8 \times 12.5\ mm^2$ section and $1\ mm$ thickness, which have been assembled by using the 3D printed junctions shown in Fig. 4.11(a). In order to increase the available space on the platform, the *Dynamixel XC-430* motors have been selected, keeping the same performance than the *Dynamixel AX 12A* motors but with lower size, and the layout of the roll and pitch joints has been changed. More in detail, in ODQuad.02, the axes of the roll-pitch joints are aligned with the propeller arms (i.e. the joint to the zero position) while in the new prototype they are angled at 45 degrees with respect to the arms.

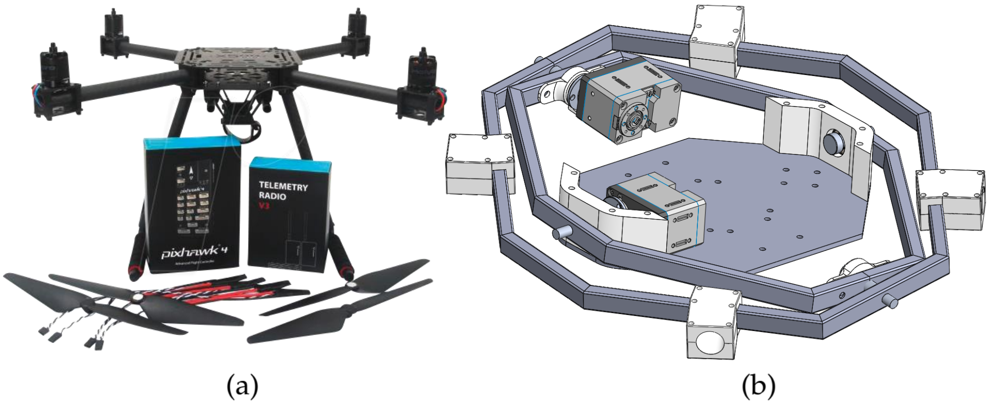


Figure 4.10: Holybro X500 kit (a), CAD model ODQuad.03 (b).

The platform is composed by a plywood sheet on which the *Holybro* support feet, the battery holder and the 3D printed roll joint bearing supports (Fig. 4.11(b)) have been mounted. The bearing holders of the pitch joint are of two types: the first one is shown in Fig. 4.11(c) while the second one is reported in Fig. 4.11(d), and unlike the first one, it hosts the motor bracket. The joint shafts have been fabricated from steel and can be idle, (Fig. 4.11(e) and Fig. 4.11(f)), or with motor coupling (Fig. 4.11(g) and Fig. 4.11(h)). The propellers arms and rotor supports have been taken from *Holybro* and have been connected to the propellers frame via the 3D printed junctions shown in Fig.4.11(i).

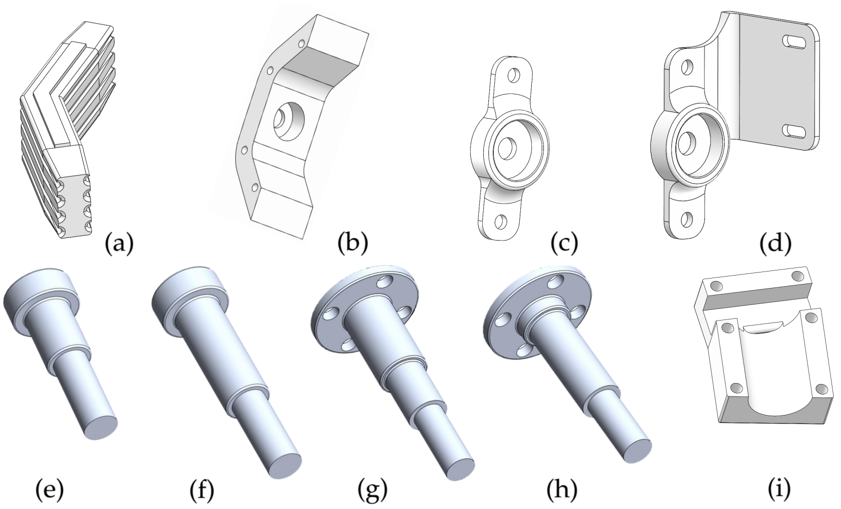


Figure 4.11: Aluminum profiles junction (a), roll bearing holder (b), pitch bearing holder (c), pitch bearing holder with motor bracket(d), roll idle shaft (e), pitch idle shaft (f), roll motor shaft (g), pitch motor shaft (h) and propellers arm junction (i).

The propulsion system was taken from the *Holybro*, that consists of three main components, namely, the electronic speed controllers (ESCs), the brushless DC motors and the propellers. The assembled prototype, without the electronic boards, is shown in Fig. 4.12.

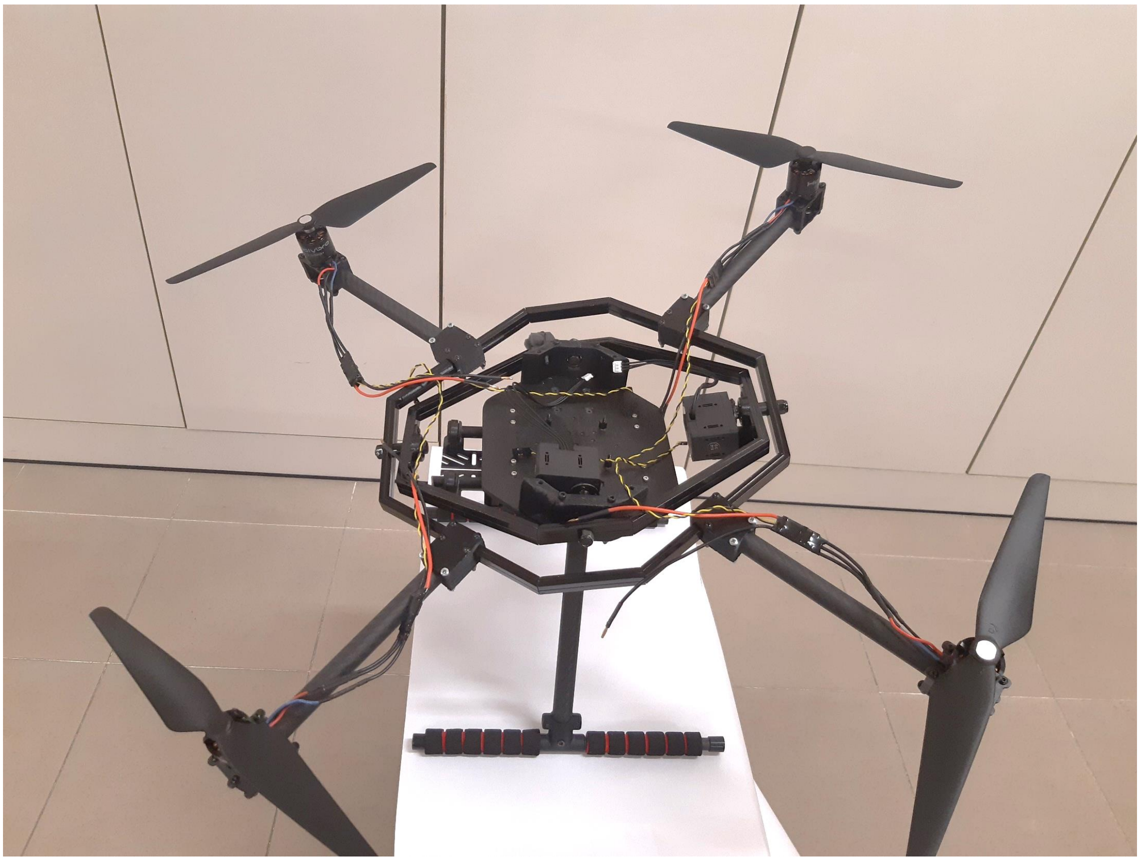


Figure 4.12: Assembled prototype.

The electronic components required for the vehicle control system are listed as follows:

- *Odroid XU4* board, on which the custom flight controller, described in Chapter 2 runs;
- *Pixhawk 4*®, an advanced autopilot device with an Inertial Measurement Unit (IMU);
- *PM07* Power Management Board;
- *UBEC* to adjust the power voltage;
- *Wifi radio* for the Odroid board,

- *Radio controller*.

The *Odroid XU4* is a mini-computer with a 2 GHz processor and an embedded Multi-Media chip (eMMc) flash storage ranging from 8 GB to 64 GB. It is connected to Pixhawk by using the FTDI to USB cable shown in Fig. 4.13, which links the Odroid USB port to the Pixhawk telemetry. In this way the Odroid board is able to communicate with a Pixhawk flight controller using the MAVLink protocol (Micro Aerial Vehicles Link, a very lightweight messaging protocol for communicating with aerial vehicles) over a serial connection. The power management board not only provides regulated power to the Pixhawk 4 and ESCs, but it also sends information to the autopilot about battery voltage and current supplied to the flight controller and motors. The UBEC device has been connected between the Odroid board and the battery to set the supply voltage to 5 V. Finally, a radio controller that communicates with the *Pixhawk* board to intervene in emergency situations is added.

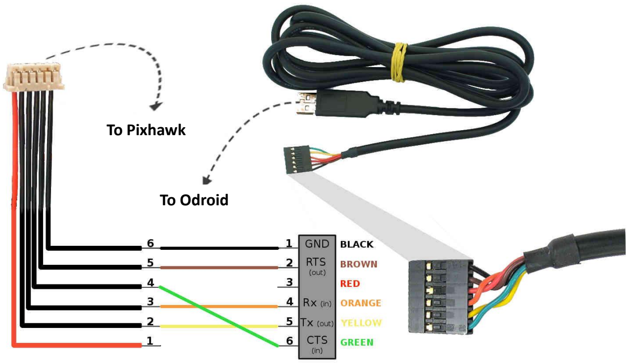


Figure 4.13: FTDI cable and wiring diagram.

The communication scheme between the components, using the same approach adopted in [86], is shown in Fig.4.14. This approach will be used in the early experimental tests, which will be performed indoor, thus a motion capture system for vehicle localization has been considered. More in detail, the Odroid board receives from the motion capture system information about the position and velocity of the vehicle, while from the Pixhawk board it gets the orientation and angular velocity (which could also be obtained from the motion capture system) and also the position and velocity of the roll-pitch joints. With this information and these received from the command station, it computes the reference values for the four rotor thrusts and the references for the roll-pitch joints. The actuators of the roll-pitch joint are not controlled in a torque mode, as in the control scheme shown in Section 2.3, but in position mode with the following PD controller

$$\boldsymbol{\delta} = \boldsymbol{K}_{\boldsymbol{\delta}_P}\boldsymbol{\delta}_{des} + \boldsymbol{K}_{\boldsymbol{\delta}_D}(\dot{\boldsymbol{\delta}}_{des} - \dot{\boldsymbol{\delta}}), \tag{4.4}$$

where $\boldsymbol{K}_{\boldsymbol{\delta}_P} \in \mathbb{R}^{2\times 2}$ and $\boldsymbol{K}_{\boldsymbol{\delta}_D} \in \mathbb{R}^{2\times 2}$ are positive definite gain matrices given by

$$\boldsymbol{K}_{\boldsymbol{\delta}_P} = \begin{bmatrix} k_{\alpha_P} & 0 \\ 0 & k_{\beta_P} \end{bmatrix}, \quad \boldsymbol{K}_{\boldsymbol{\delta}_D} = \begin{bmatrix} k_{\alpha_D} & 0 \\ 0 & k_{\beta_D} \end{bmatrix},$$

where k_{α_P}, k_{β_P},k_{α_D} and k_{β_D} are the gains that will be derived through experimental testing of motor response performance.

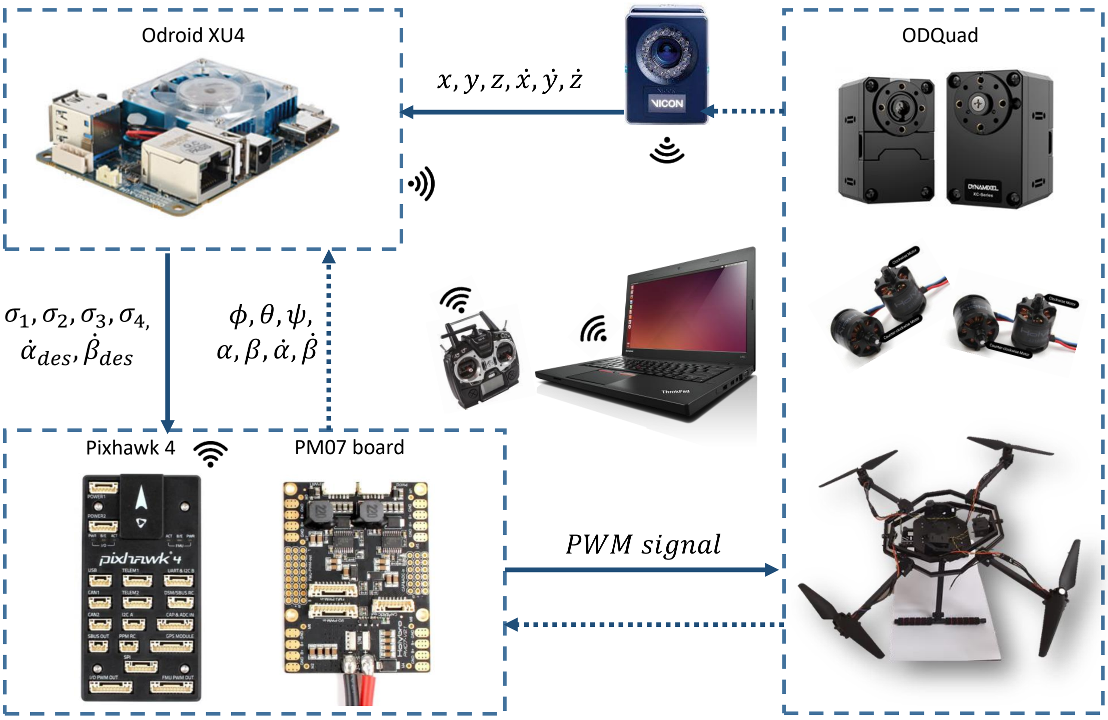

Odroid XU4
$x, y, z, \dot{x}, \dot{y}, \dot{z}$
VICON
ODQuad
$\sigma_1, \sigma_2, \sigma_3, \sigma_4,$
$\dot{\alpha}_{des}, \dot{\beta}_{des}$
$\phi, \theta, \psi,$
$\alpha, \beta, \dot{\alpha}, \dot{\beta}$
Pixhawk 4
PM07 board
$PWM\ signal$

Figure 4.14: Setup for Odroid, Pixhawk, and Vicon communication.

The vehicle assembly was completed only recently and currently the first experimental tests are being running. The goal of such tests is to verify the communication between the electronic boards and the correct running of the roll-pitch joint motors. The next step will be the implementation of the controller on the Odroid board, in which the experimental data obtained from a future dynamic identification of the prototype will be used.

4.4 Conclusion

In this Chapter the phases that led to the realization of the prototype have been illustrated. In the first part, the CAD model used both in the static and dynamic analyses were presented. The first analysis investigates the torque values for a rough sizing of the roll-pitch joint actuators, while the second analysis has been carried out by means of the numerical simulations presented in the Chapter 2. Since in the ODQuad.01 the components specifications and the particular construction techniques have not been taken into account, the model ODQuad.02 was designed. The ODQuad.02 is suited to be manufactured with aluminum or carbon fiber profiles and 3D printed parts, but each component must be custom manufactured. In order to shorten the prototype development, the ODQuad.03 has been created, which includes some components of the quadrotor *Holybro X500* into a newly designed mechanical frame. The main differences with respect to the ODQuad.02 are the adoption of the *Dynamixel XC-430* motors, which have the same performance than the *Dynamixel AX 12A* motors but are less bulky, and the roll and pitch joints orientation arrangement. More in detail, in ODQuad.02 the axes of the roll-pitch joints are aligned with the propeller arms (i.e. the joint to the zero position) while in the ODQuad.03 they are angled at 45 degrees with respect to the arms. The new arrangement of the roll and pitch joints resulted in more space on the platform, thus allowing for an easier mounting of the electronic components and of the roll joint actuator. The prototype has proven the feasibility of the ODQuad concept, therefore, in the next months, in order to confirm the simulation results, the first flight tests will be carried out.

Conclusions and future work

This thesis proposes a solution to the under-actuation problem, which is common to all aerial platforms with parallel-axis rotors. Nowadays, multirotors have a variety of applications in many fields, due to their vertical take-off and landing capability, their maneuverability and the possibility to be equipped with many devices, i.e., cameras, sensors and even robotic arms. In many scientific studies, it has been proven that in those operations where the robot is expected to interact with the environment, it is very difficult to control an under-actuated multirotor since it can be easily destabilize by contact dynamics. Hence, because of their better disturbances rejection capabilities, fully-actuated multirotors are preferred in those circumstances that involve vehicle interaction with the environment, such as object transportation, manipulation tasks and contact inspections. In literature there are many solutions that allow for solving the under-actuation problem by using at least six propellers or by adding one or more actuators that permit to orient the propellers all together or individually, respectively.

As many studies and experimental evidences have shown, the propulsive efficiency of a propeller is strongly influenced by the surrounding air motion field. Furthermore, while for parallel axis multirotors the aerodynamic perturbation between the propellers represents a significant reduction of the overall efficiency, for tilted rotor vehicles the aerodynamic dissipation is much more severe, since the rotor airflow crossing amplifies the air field perturbations. Energy efficiency plays a key role in aerial vehicles, since, for the same battery capacity, it permits to increase the flight time between two successive recharges. Therefore, solutions that possess the efficiency of a common multirotor and the maneuverability of a fully-actuated aerial vehicle should be adopted. To this aim the ODQuad (OmniDirectional Quadrotor) has been designed and developed.

Often, a distinction is made between *fully-actuated* aerial vehicles, in which the control of the 6 DoFs is achieved independently, and *omnidirectional* ones, which are multirotors able to project the total thrust in any direction, like a flying balloon. However, in common applications, where the total thrust describes a limited solid angle of the spherical space, the two concepts substantially coincide.

Chapter 1 reports the state of art of the solutions adopted in the realization of fully-actuated multirotors, including also the ODQuad concept. The ODQuad is composed by three main parts, i.e., the platform, the mobile and rotor frames, which are linked by means of two rotational joints, namely the roll and pitch joints. In the same chapter, the kinematic and dynamic models have been derived using the *Euler-Lagrange* approach. The dynamic model has been formulated in general terms and it is therefore easily adaptable to other aerial robots similar to the ODQuad.

By using the dynamic model of the vehicle, in Chapter 2, a model-based controller has been designed. It is based on two control loops, an outer one for vehicle position control and an inner one for vehicle orientation and roll-pitch joint control. The outer loop, based on the desired and actual ODQuad position, computes the references for the roll-pitch joints The inner loop, based on this information and using the actual and desired vehicle orientation, computes the roll-pitch joint actuation torques. The thrust allocation on the propellers is obtained by exploiting the force and moment calculated by the outer and inner loops, respectively. The effectiveness of the controller has been tested by means of numerical simulations both in free motion and in object transportation tasks. In the latter test, a momentum observer has been used to estimate the wrenches exchanged between the vehicle and the transported object, in order to compensate its effect.

After assessing that the ODQuad and the designed controller are also effective in manipulation tasks, in Chapter 3, a simulation model, in which multiple multirotors are involved to perform cooperative aerial manipulation tasks, has been implemented. In particular, the manipulation of a bulky object has been considered, whose unknown inertial parameters have been identified in the first task phase. In order to reduce the mechanical stresses due to the manipulation, an admittance filter (defined internal admittance) has been implemented on each multirotor. Moreover, in order to make the system robust to interactions with the environment, an external admittance filter has been considered which yields the manipulation system, i.e., the object and the ODQuads, compliant to external interaction wrenches.

The first three chapters of the thesis demonstrated the feasibility of the ODQquad and control systems. Thus, in Chapter 4 the prototyping process has been described. Most of the mechanical parts have been custom manufactured and the remaining components, such as propellers, motors, battery support and electronics, have been taken from the quadcopter *Holybro X500*. An *Odroid* board has also been introduced to implement the proposed controller.

In conclusion, the main contributions of this thesis are:

- Design of a new fully-actuated aerial vehicle concept, the ODQuad.
- Proposal of a model-based controller for omnidirectional multirotor vehicles.
- Presentation of simulation results highlighting the robustness of fully-actuated vehicle with the proposed controller to external disturbances.
- Presentation and validation via numerical tests of cooperative aerial manipulation of an object with unknown inertial parameters.
- The ODQuad development, with description of the design solutions and main components.

The next development of the work will consist in the first flight tests of the prototype, in order to validate the ODQuad concept also experimentally. Afterwards, more complex aerial manipulation operations, requiring the use of a robotic arm mounted on the aerial platform can be conducted first with numerical simulations and then experimentally. Special consideration could be given to payload capacity of the platform, by replacing simple rotors with coaxial double propeller rotors, and, moreover, new control schemes could be investigated to improve the energy efficiency of the vehicle. Finally, further studies could be carried out regarding the aerial transportation of flexible objects.

Bibliography

[1] E. Çetinsoy, S. Dikyar, C. Hançer, K. Oner, E. Sirimoglu, M. Unel, and M. Aksit, "Design and construction of a novel quad tilt-wing uav," *Mechatronics*, vol. 22, no. 6, pp. 723–745, 2012.

[2] 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*, vol. 23, no. 2, pp. 540–556, 2015.

[3] P. Segui-Gasco, Y. Al-Rihani, H.-S. Shin, and A. Savvaris, "A novel actuation concept for a multi rotor UAV," *Journal of Intelligent & Robotic Systems*, vol. 74, no. 1-2, pp. 173–191, 2014.

[4] 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 *EEE Int. Conf. on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 4006–4013.

[5] D. Brescianini and R. D'Andrea, "Design, modeling and control of an omni-directional aerial vehicle," in *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, 2016, pp. 3261–3266.

[6] M. Ryll, D. Bicego, and A. Franchi, "Modeling and control of fast-hex: a fully-actuated by synchronized-tilting hexarotor," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2016.

[7] M. Kamel, S. Verling, O. Elkhatib, C. Sprecher, P. Wulkop, and Z. Taylor, "The voliro omniorientational hexacopter: An agile and maneuverable tiltable-rotor aerial vehicle," *IEEE Robotics and Automation Magazine*, vol. 25, pp. 34–44, 2018.

[8] M. Ryll, G. Muscio, F. Pierri, E. Cataldi, G. Antonelli, F. Caccavale, D. Bicego, and A. Franchi, "6d interaction control with aerial robots: The flying end-effector paradigm," *The International Journal of Robotics Research*, vol. 38, no. 9, pp. 1045–1062, 2019.

[9] F. Ruggiero, V. Lippiello, and A. Ollero, "Aerial manipulation: A literature review," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1957–1964, 2018.

[10] A. Ollero, G. Heredia, A. Franchi, G. Antonelli, K. Kondak, A. Sanfeliu, A. Viguria, J. R. Martinez-de Dios, F. Pierri, J. Cortés *et al.*, "The aeroarms project: Aerial robots with advanced manipulation capabilities for inspection and maintenance," *IEEE Robotics & Automation Magazine*, vol. 25, no. 4, pp. 12–23, 2018.

[11] K. Nonami, F. Kendoul, S. Suzuki, and W. Wang, *Atonomous Flying Robots, Unmanned Aerial Vehicles and Micro Aerial Vehicles*. London, UK: Springer, 2010.

[12] M. Hamandi, F. Usai, Q. Sablé, N. Staub, M. Tognon, and A. Franchi, "Design of multirotor aerial vehicles: A taxonomy based on input allocation," *The International Journal of Robotics Research*, vol. 40, no. 8-9, pp. 1015–1044, 2021.

[13] R. Rashad, J. Goerres, R. Aarts, J. B. Engelen, and S. Stramigioli, "Fully actuated multirotor uavs: A literature review," *IEEE Robotics & Automation Magazine*, vol. 27, no. 3, pp. 97–107, 2020.

[14] F. Caccavale and M. Uchiyama, "Cooperative manipulation," in *Springer Handbook of Robotics*, B. Siciliano and O. Khatib, Eds., 2016.

[15] J. Luh and Y. Zheng, "Constrained relations between two coordinated industrial robots for motion control," *The International journal of robotics research*, vol. 6, no. 3, pp. 60–70, 1987.

[16] T. Tarn, A. Bejczy, and X. Yun, "Design of dynamic control of two cooperating robot arms: Closed chain formulation," vol. 4, pp. 7–13, 1987.

[17] Y. Nakamura, K. Nagai, and T. Yoshikawa, "Dynamics and stability in coordination of multiple robotic mechanisms," *The International Journal of Robotics Research*, vol. 8, no. 2, pp. 44–61, 1989.

[18] A. Marino and F. Pierri, "A two stage approach for distributed cooperative manipulation of an unknown object without explicit communication and unknown number of robots," *Robotics and Autonomous Systems*, vol. 103, pp. 122–133, 2018.

[19] M. Uchiyama and P. Dauchez, "A symmetric hybrid position/force control scheme for the coordination of two robots," in *Proceedings. 1988 IEEE international conference on robotics and automation*. IEEE, 1988, pp. 350–356.

[20] ——, "Symmetric kinematic formulation and non-master/slave coordinated control of two-arm robots," vol. 7, no. 4. Taylor & Francis, 1992, pp. 361–383.

[21] R. Bonitz and T. C. Hsia, "Internal force-based impedance control for cooperating manipulators," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 1, pp. 78–89, 1996.

[22] P. Chiacchio, S. Chiaverini, and B. Siciliano, "Direct and inverse kinematics for coordinated motion tasks of a two-manipulator system," 1996.

[23] D. Williams and O. Khatib, "The virtual linkage: A model for internal forces in multi-grasp manipulation," in *[1993] Proceedings IEEE International Conference on Robotics and Automation*. IEEE, 1993, pp. 1025–1030.

[24] E. Nakano, "Cooperational control of the anthropomorphous manipulator" melarm"," in *Proc. of 4th International Symposium on Industrial Robots*, 1974, pp. 251–260.

[25] F. Caccavale and M. Uchiyama, "Cooperative manipulation," in *Springer handbook of robotics*. Springer, 2016, pp. 989–1006.

[26] O. Khatib, "Inertial properties in robotic manipulation: An object-level framework," *The international journal of robotics research*, vol. 14, no. 1, pp. 19–36, 1995.

[27] M. UCHIYAMA and P. DAUCHEZ, "Symmetric kinematic formulation and non-master: slave coordinated control of two-arm robots," *Advanced robotics*, vol. 7, no. 4, pp. 361–383, 1993.

[28] S. Schneider and R. H. Cannon, "Object impedance control for cooperative manipulation: Theory and experimental results," in *Proceedings, 1989 International Conference on Robotics and Automation*. IEEE, 1989, pp. 1076–1083.

[29] L. Villani and J. De Schutter, "Force control," in *Springer handbook of robotics*. Springer, 2016, pp. 195–220.

[30] O. Khatib, K. Yokoi, K. Chang, D. Ruspini, R. Holmberg, and A. Casal, "Coordination and decentralized cooperation of multiple mobile manipulators," *Journal of Robotic Systems*, vol. 13, no. 11, pp. 755–764, 1996.

[31] T. Sugar and V. Kumar, "Control of cooperating mobile manipulators," *IEEE Transactions on Robotics and Automation*, vol. 18, pp. 94–103, 2002.

[32] H. Bai and J. Wen, "Cooperative load transport: A formation-control perspective," *IEEE Transactions on Robotics and Automation*, vol. 26, pp. 742–750, 2010.

[33] A. Petitti, A. Franchi, D. Di Paola, and A. Rizzo, "Decentralized motion control for cooperative manipulation with a team of networked mobile manipulators," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 441–446.

[34] J. Fink, N. Michael, S. Kim, and V. Kumar, "Planning and control for coopera- tive manipulation and transportation with aerial robots," *International Journal of Robotics Research*, vol. 30, pp. 324–334, 2011.

[35] D. Mellinger, M. Shomin, N. Michael, and V. Kumar, "Cooperative grasping and transport using multiple quadrotors," in *Distributed autonomous robotic systems*. Springer, 2013, pp. 545–558.

[36] V. Parra-Vega, A. Sanchez, C. Izaguirre, O. Garcia, and F. Ruiz-Sanchez, "Toward aerial grasping and manipulation with multiple uavs," *Journal of Intelligent and Robotic Systems*, vol. 70, pp. 575–593, 2013.

[37] F. Caccavale, G. Giglio, G. Muscio, and F. Pierri, "Cooperative impedance control for multiple uavs with a robotic arm," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 2366–2371.

[38] F. Caccavale, P. Chiacchio, A. Marino, and L. Villani, “Six-dof impedance control of dual-arm cooperative manipulators,” *Mechatronics, IEEE/ASME Transactions on*, vol. 13, no. 5, pp. 576–586, 2008.

[39] A. Tagliabue, M. Kamel, S. Verling, R. Siegwart, and J. Nieto, “Collaborative transportation using mavs via passive force control,” in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 5766–5773.

[40] D. Sanalitro, H. J. Savino, M. Tognon, J. Cortés, and A. Franchi, “Full-pose manipulation control of a cable-suspended load with multiple uavs under uncertainties,” *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2185–2191, 2020.

[41] S. Jordan, J. Moore, S. Hovet, J. Box, J. Perry, K. Kirsche, D. Lewis, and Z. T. H. Tse, “State-of-the-art technologies for UAV inspections,” *IET Radar, Sonar & Navigation*, vol. 12, no. 2, pp. 151–164, 2017.

[42] D. Alejo, J. A. Cobano, G. Heredia, J. R. Martínez-de Dios, and A. Ollero, “Efficient trajectory planning for wsn data collection with multiple uavs,” in *Cooperative Robots and Sensor Networks 2015*. Springer, 2015, pp. 53–75.

[43] P. Tokekar, J. Vander Hook, D. Mulla, and V. Isler, “Sensor planning for a symbiotic UAV and ugv system for precision agriculture,” *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1498–1511, 2016.

[44] A. Ollero, G. Heredia, A. Franchi, G. Antonelli, K. Kondak, A. Sanfeliu, A. Viguria, J. R. Martinez-de Dios, F. Pierri, J. Cortes, A. Santamaria, M. Angel Trujillo Soto, R. Balachandran, J. Andrade Cetto, and A. RodriÂguez, “The aeroarms project: Aerial robots with advanced manipulation capabilities for inspection and maintenance,” *IEEE Robotics and Automation Magazine*, vol. 25, no. 4, pp. 12–23, 2018.

[45] F. Pierri, G. Muscio, and F. Caccavale, “An adaptive hierarchical control for aerial manipulators,” *Robotica*, vol. 36, no. 10, pp. 1527–1550, 2018.

[46] R. Mahony, V. Kumar, and P. Corke, “Multirotor aerial vehicles: Modeling, estimation, and control of quadrotor,” *IEEE Robotics and Automation magazine*, vol. 19, no. 3, pp. 20–32, 2012.

[47] D. Shukla and N. Komerath, “Multirotor drone aerodynamic interaction investigation,” *Drones*, vol. 2, no. 4, p. 43, 2018.

[48] H. González-Jorge, J. Martínez-Sánchez, M. Bueno *et al.*, “Unmanned aerial systems for civil applications: A review,” *Drones*, vol. 1, no. 1, p. 2, 2017.

[49] K. T. Oner, E. Cetinsoy, M. Unel, M. F. Aksit, I. Kandemir, and K. Gulez, “Dynamic model and control of a new quadrotor unmanned aerial vehicle with tilt-wing mechanism,” *International Journal of Mechanical, Aerospace, Industrial, Mechatronic and Manufacturing Engineering*, vol. 2, no. 9, pp. 1008–1013, 2008.

[50] M. Tognon and A. Franchi, “Omnidirectional aerial vehicles with unidirectional thrusters: Theory, optimal design, and control,” *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 2277–2282, 2018.

[51] M. Hamandi, K. Sawant, M. Tognon, and A. Franchi, “Omni-plus-seven (o7+): An omnidirectional aerial prototype with a minimal number of unidirectional thrusters,” in *2020 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 2020, pp. 754–761.

[52] M. Odelga, P. Stegagno, and H. H. Bülthoff, “A fully actuated quadrotor UAV with a propeller tilting mechanism: Modeling and control,” in *Advanced Intelligent Mechatronics (AIM), 2016 IEEE International Conference on*, 2016, pp. 306–311.

[53] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics – Modelling, Planning and Control*. London, UK: Springer, 2009.

[54] G. Hoffmann, H. Huang, S. Waslander, and C. Tomlin, “Quadrotor helicopter flight dynamics and control: Theory and experiment,” in *Proc. of the AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2007.

[55] O. Araar and N. Aouf, “Full linear control of a quadrotor uav, lq vs $\mathcal{H}_\infty$,” in *2014 UKACC International Conference on Control (CONTROL)*. IEEE, 2014, pp. 133–138.

[56] M. Bangura and R. Mahony, “Real-time model predictive control for quadrotors,” *IFAC Proceedings Volumes*, vol. 47, no. 3, pp. 11 773–11 780, 2014.

[57] M. E. Antonio-Toledo, E. N. Sanchez, A. Y. Alanis, J. Flórez, and M. A. Perez-Cisneros, “Real-time integral backstepping with sliding mode control for a quadrotor uav,” *IFAC-PapersOnLine*, vol. 51, no. 13, pp. 549–554, 2018.

[58] S. Nadda and A. Swarup, “Development of backstepping based sliding mode control for a quadrotor,” in *2014 IEEE 10th International Colloquium on Signal Processing and its Applications*. IEEE, 2014, pp. 10–13.

[59] I. Palunko, P.Cruz, and R. Fierro, “Agile load transportation. safe and efficient load manipulation with aerial robots,” *IEEE Robotics & Automation Magazine*, vol. 19, no. 3, pp. 69–79, 2012.

[60] F. Ruggiero, J. Cacace, H. Sadeghian, and V. Lippiello, “Passivity-based control of vtol uavs with a momentum-based estimator of external wrench and unmodeled dynamics,” *Robotics and Autonomous Systems*, vol. 72, pp. 139–151, 2015.

[61] F. Forte, R. Naldi, A. Macchelli, and L. Marconi, “Impedance control of an aerial manipulator,” in *American Control Conference (ACC), 2012*. IEEE, 2012, pp. 3839–3844.

[62] M. Watterson, A. Zahra, and V. Kumar, “Geometric control and trajectory optimization for bidirectional thrust quadrotors,” in *International Symposium on Experimental Robotics*. Springer, 2018, pp. 165–176.

[63] S. Meraglia and M. Lovera, "Smoother-based iterative learning control for uav trajectory tracking," *IEEE Control Systems Letters*, vol. 6, pp. 1501–1506, 2022.

[64] S. Park, J. Her, J. Kim, and D. Lee, "Design, modeling and control of omni-directional aerial robot," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 1570–1575.

[65] G. Ducard and M.-D. Hua, "Discussion and practical aspects on control allocation for a multi-rotor helicopter," *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 38, no. 1/C22, 2011.

[66] B. Convens, K. Merckaert, M. M. Nicotra, R. Naldi, and E. Garone, "Control of fully actuated unmanned aerial vehicles with actuator saturation," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 12 715–12 720, 2017.

[67] M. M. Nicotra, R. Naldi, and E. Garone, "A robust explicit reference governor for constrained control of unmanned aerial vehicles," in *2016 American Control Conference (ACC)*. IEEE, 2016, pp. 6284–6289.

[68] 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*, vol. 34, no. 2, pp. 534–541, 2018.

[69] J. Reger, H. S. Ramírez, and M. Fliess, "On non-asymptotic observation of nonlinear systems," in *Proceedings of the 44th IEEE Conference on Decision and Control*. IEEE, 2005, pp. 4219–4224.

[70] F. M. Pereira, "Block matrices and stability theory," *Tatra Mt. Math. Publ*, vol. 38, pp. 147–162, 2007.

[71] B. Siciliano and L. Villani, "An adaptive force/position regulator for robot manipulators," *International journal of adaptive control and signal processing*, vol. 7, no. 5, pp. 389–403, 1993.

[72] H. Khalil, *Nonlinear Systems (2nd ed.)*. Upper Saddle River, NJ: Prentice Hall, 1996.

[73] E. Cataldi, G. Muscio, M. A. Trujillo, Y. Rodríguez, F. Pierri, G. Antonelli, F. Caccavale, A. Viguria, S. Chiaverini, and A. Ollero, "Impedance control of an aerial-manipulator: Preliminary results," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 3848–3853.

[74] F. Pierri, M. Nigro, G. Muscio, and F. Caccavale, "Cooperative manipulation of an unknown object via omnidirectional unmanned aerial vehicles," *Journal of Intelligent & Robotic Systems*, vol. 100, no. 3, pp. 1635–1649, 2020.

[75] A. De Luca and R. Mattone, "Sensorless robot collision detection and hybrid force/motion control," in *2005 IEEE Int.Conf. on Robotics and Automation*, 2005, pp. 999–1004.

[76] G. Arleo, F. Caccavale, G. Muscio, and F. Pierri, "Control of quadrotor aerial vehicles equipped with a robotic arm," in *Proc. of 21th Mediterranean Conference on Control and Automation*, 2013, pp. 1174–1180.

[77] A. Ollero and B. Siciliano, *Aerial Robotic Manipulation*. Springer International Publishing, 2019.

[78] M. Ryll, G. Muscio, F. Pierri, E. Cataldi, G. Antonelli, F. Caccavale, D. Bicego, and A. Franchi, "6d interaction control with aerial robots: The flying end-effector paradigm," *The International Journal of Robotics Research*, vol. 38, no. 9, pp. 1045–1062, 2019.

[79] S. A. A. Moosavian, R. Rastegari, and E. Papadopoulos, "Multiple impedance control for space free-flying robots," *Journal of Guidance Control and Dynamics*, vol. 28, no. 5, p. 939, 2005.

[80] P. Zarafshan, R. Larimi, S. A. A. Moosavian, and B. Siciliano, "Which impedance strategy is the most effective for cooperative object manipulation?" *Industrial Robot: An International Journal*, vol. 44, no. 2, pp. 198–209, 2017.

[81] M. Nigro, F. Pierri, and F. Caccavale, "Preliminary design, modeling and control of a fully actuated quadrotor UAV," in *2019 International Conference on Unmanned Aircraft Systems*, 2019, pp. 1108–1116.

[82] P. Chiacchio, S. Chiaverini, L. Sciavicco, and B. Siciliano, "Global task space manipulability ellipsoids for multiple-arm systems," *IEEE Transactions on Robotics and Automation*, vol. 7, no. 5, pp. 678–685, 1991.

[83] L. Villani and J. De Schutter, "Force control," in *Handbook of Robotics*, Springer-Verlag, B. Siciliano, and O. Khatib, Eds., 2008.

[84] P. Castaldi, N. Mimmo, R. Naldi, and L. Marconi, "Robust trajectory tracking for underactuated vtol aerial vehicles: Extended for adaptive disturbance compensation," *IFAC Proceedings Volumes*, vol. 47, no. 3, pp. 3184–3189, 2014.

[85] E. Cataldi, G. Muscio, M. A. Trujillo, Y. Rodríguez, F. Pierri, G. Antonelli, F. Caccavale, A. Viguria, S. Chiaverini, and A. Ollero, "Impedance control of an aerial-manipulator: Preliminary results," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 3848–3853.

[86] R. Anderson, "Robust adaptive control laws for tilt-rotor quadcopters subject to user-defined constraints," 2019.