



# Modelling, Analysis, and Control of OmniMorph: an Omnidirectional Morphing Multi-rotor UAV

Youssef Aboudorra<sup>1</sup> · Chiara Gabellieri<sup>1</sup> · Ralph Brantjes<sup>1</sup> · Quentin Sablé<sup>1</sup> · Antonio Franchi<sup>1,2</sup>

Received: 31 July 2023 / Accepted: 31 December 2023  
© The Author(s) 2024

## Abstract

This paper introduces for the first time the design, modelling, and control of a novel morphing multi-rotor Unmanned Aerial Vehicle (UAV) that we call the OmniMorph. The morphing ability allows the selection of the configuration that optimizes energy consumption while ensuring the needed maneuverability for the required task. The most energy-efficient *uni-directional thrust* (UDT) configuration can be used, e.g., during standard point-to-point displacements. *Fully-actuated* (FA) and *omnidirectional* (OD) configurations can be instead used for full pose tracking, such as, e.g., constant attitude horizontal motions and full rotations on the spot, and for full wrench 6D interaction control and 6D disturbance rejection. Morphing is obtained using a single servomotor, allowing possible minimization of weight, costs, and maintenance complexity. The actuation properties are studied, and an optimal controller that compromises between performance and control effort is proposed and validated in realistic simulations. Preliminary tests on the prototype are presented to assess the propellers' mutual aerodynamic interference.

**Keywords** Unmanned aerial vehicles · Omnidirectional UAVs · Robot modelling · Robot control · Multi-rotor aerial vehicles · Aerial robotics

## 1 Introduction

Over the last decade, the design, perception, and control of Unmanned Aerial Vehicles (UAVs) have substantially evolved in terms of functionalities and capabilities indoors or outdoors. Research now considers the use of UAVs as Aerial

Robotic Manipulators that can interact physically with the environment [1] and humans [2]; contact-based inspection [3], object manipulation [4], and assisting human workers [5] are some of the typical tasks.

This led to the development of new multirotor aerial vehicles (MRAVs) and especially to the exploration of designs different from the popular underactuated multirotors that use fixed coplanar/collinear propellers (i.e., quadrotors, hexarotors, and their variants). The new designs rely on choosing the number, position, and orientation of the propellers to provide the multirotor with different actuation properties and abilities to achieve a specific task [6]. Morphing designs with varying arm length and consequent proper controller tuning have also been proposed in the literature [7–10].

Recently, several works addressed the design of *omni-directional* multirotors. Omnidirectionality is the property of the robot to sustain its weight in any orientation. In the literature, different actuation strategies to achieve omnidirectionality have been proposed. In [11], a novel metric to assess the omnidirectional property of a generic multi-rotor aerial vehicle was presented. This metric is calculated numerically from the force set of the platform, and as such it relies on the platform geometry and thrusters; moreover, it allows

✉ Youssef Aboudorra  
y.a.l.a.aboudorra@utwente.nl

✉ Chiara Gabellieri  
c.gabellieri@utwente.nl

Ralph Brantjes  
r.j.brantjes@student.utwente.nl

Quentin Sablé  
q.l.g.sable@utwente.nl

Antonio Franchi  
a.franchi@utwente.nl

<sup>1</sup> Robotics and Mechatronics group, Faculty of Electrical Engineering, Mathematics & Computer Science, University of Twente, Enschede, The Netherlands

<sup>2</sup> Department of Computer, Control and Management Engineering, Sapienza University of Rome, Rome 00185, Italy

a direct assessment of a platform's omnidirectional property given its weight. [11] also showed the use of this metric in the process of upgrading the design presented in [12], by placing the propellers on a virtual sphere centered around the platform's Center of Mass (CoM). In the following, we categorize omnidirectional designs into three main classes.

The first class uses fixedly tilted *bi*-directional propellers (which generate thrust in both directions by inverting their sense of rotation). The authors in [13] presented one of the first omnidirectional prototypes, where they placed eight bi-directional propellers on the edges of a cube. The novelty of their design is the optimization-based placement and orientation of the propellers, where the cube shape was chosen to ensure a rotation-invariant inertia tensor. Another omnidirectional design with fixed propellers was presented in [14] and [15], achieving omnidirectional wrench generation with six and eight bi-directional propellers, respectively.

The second class, like the design proposed in [16], optimizes the orientation of fixed *uni*-directional propellers (i.e., able to produce thrust in only one direction, or, in other words, to rotate in only one sense) to achieve omnidirectionality while equally distributing the thrust produced by each propeller and aiming for an isotropic (sphere-like) shape of the corresponding force set. In fact, [16] showed that the minimal number of 7 (seven) uni-directional propellers is necessary to achieve omnidirectional position and orientation tracking. Later, [12] presented a working prototype of the design with experiments.

The third class of omnidirectional platforms relies on a morphing design. Uni- or bi-directional propellers *actively tilted* by servo motors are used. [17] proposes a quadrotor with propellers individually tilting about their radial axis. The authors of [18] use the same idea (independently controlled radially tilting thrusters) for an hexarotor. Another version of their design was presented in [19], in which double propeller groups were used, two propellers rotating in opposite directions on each rotation axis to increase the platform efficiency and payload capacity in addition to increased force bounds along different directions.

Design aimed at optimizing control effort is an established approach in aerial platforms [20] and it is even more critical for multirotor aerial robots, as they are typically characterized by limited flight endurance. The advantage of the third design approach relies on the possibility of optimizing the energy consumed for a given desired orientation, which is obtained by aligning together as much as possible the propeller spinning directions for a given orientation, in a way that each propeller tends to be as much counter aligned as possible to the gravity force. This also allows for balancing dexterity and energy efficiency based on the needs imposed by specific tasks [21, 22]. Indeed, the configuration in which all the directions are aligned is more energetically efficient. In that configuration, the propeller forces are all parallel to

each other but the thrust is uni-directional; when this is not the case, there are propeller forces that cancel each other without actively contributing to the motion. That is the price paid to achieve multi/omnidirectionality.

From a control allocation point of view, fixed-propeller multirotors in [12–14] are controlled by (pseudo)inverting the wrench map between the propellers' speeds and the total body wrench. Those systems are affine in the control inputs and the tilting angles of the propellers are constant design parameters. A similar control approach is used in [15], where a different rotor input allocation is proposed, based on the infinity-norm minimization; in this way, propellers' speed constraints are implicitly taken into account, even though with no guarantees.

For the second type of omnidirectional platforms, in [12] a dynamic inversion plus input allocation strategy is proposed. The input allocation modifies the minimum norm solution found by inverting the allocation matrix to ensure that the rotational speeds are positive. For a fixed-propeller hexarotor that is fully actuated but not omnidirectional, [23] proposes a Nonlinear Model-Predictive Controller.

The control of the third class of platforms is challenging, as the tilting angle must also be determined, which appears nonlinearly in the dynamics. Feedback linearization at the jerk level has been proposed [17], thanks to the fact that the time-differentiation the system is affine in the (new) control inputs. In [18], a change of variables is used to obtain a static allocation matrix not dependent on the tilting angle; that is pseudo-inverted, and the initial inputs are then computed. Both these methods require the full allocation matrix to be non-singular and may result in unfeasible inputs due to neglected input bounds. [19] proposes a jerk-level control in which the optimal control inputs are obtained by weighted pseudo-inversion of the full allocation matrix. As input constraints are not considered, the inputs are saturated before being integrated and sent to the actuators. The control method avoids singularities by adding a bias to the desired tilting angle in case of a large condition number of the wrench map matrix.

In this work, we overcome some of the limitations of existing concepts and controllers of omnidirectional platforms. Specifically, the contributions of this work are as follows.

- A novel omnidirectional morphing platform design concept, OmniMorph, is proposed. The platform belongs to the third class described in this section and exploits the synchronized drive of all propellers by only one servo motor. This choice allows keeping at the bare minimum the additional payload used to carry additional servo motors in the other designs seen before.
- The actuation properties of OmniMorph are formally studied.

- A novel control method is proposed for multi-rotors with actively and synchronously tilting propellers. Compared to the state of the art, the proposed controller allows automatic transitions between underactuated and omnidirectional configurations while explicitly accounting for input constraints, only indirectly handled through redundancy resolution in the literature [17].
- The proposed design and controller are tested in a realistic simulation environment.
- A preliminary prototype is built and tested to assess the mutual aerodynamic interference among the propellers. Hence, simulations under degraded and unknown propeller performance are carried out to assess the robustness of the proposed controller.

## 2 OmniMorph Design and Model

### 2.1 Concept and Design

The OmniMorph platform consists of 8 bi-directional propellers attached to the main body frame, placed on the vertices of a cube centered at the CoM, as in Fig. 1. The idea of this work is to design a morphing platform, i.e., capable of tilting its propellers, that can span while flying, a variety of configurations in between two extreme cases: a dexterous but less energetically efficient omnidirectional configuration and an efficient but underactuated unilateral thrust configuration. More than that, we ask that all propellers tilt in a synchronized way, meaning they can be all attached to a single extra actuation unit (a tilting mechanism) driven by only one servo motor. That reduces the complexity and the weight of the prototype, allowing for a higher payload.

Note that each propeller tilts about a different axis of the same amount, which we call the *tilting angle*  $\alpha$ . The change in the orientation of the propellers (morphing), changes (i.e., it morphs) the force and moment feasible sets. When all rotors are parallel to each other (i.e. tilting angle  $\alpha$  equal to zero)

the force set is a thin line, i.e., the generated total thrust can be produced only along one axis, making the vehicle underactuated (UA)— see Fig. 1 left. In this case, the platform is, from the actuation point of view, equivalent to the standard underactuated quadrotor, hexarotor, and so on: only 4 Degrees of Freedom (DoF) are independently controlled, the 3D position and the heading angle. This configuration is the most energetically efficient and it is recommended in tasks in which the control of the full orientation or full wrench is not essential.

When the tilting angle  $\alpha$  assumes positive values, the feasible force set morphs into a polyhedron with non-zero volume. The larger the tilting angle the larger the volume. The total thrust can be generated in all three orthogonal directions, to a certain extent, depending on the tilting angle value. Ultimately, the vehicle can assume an OD configuration similar to the vehicle developed in [13]— see Fig. 1, right. Such a configuration allows controlling the 6D full pose of the vehicle sustaining its weight in any possible orientation. A representation of the polytopes of feasible forces of OmniMorph for different values of  $\alpha$  computed as in [6] is given in Fig. 2.

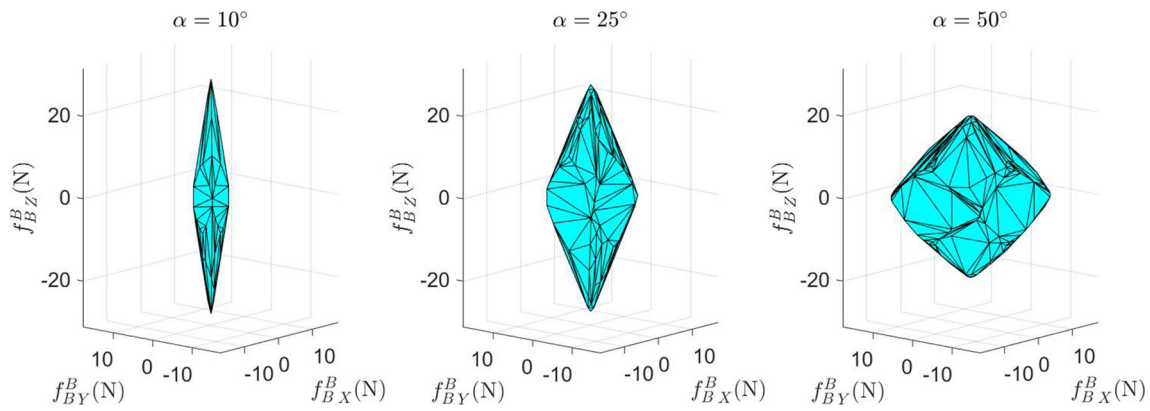
In the following, the design concept of OmniMorph is detailed. We define a reference frame attached to the robot as a set of an origin and three orthogonal axes:  $\mathcal{F}_B : \{O_B, \mathbf{x}_B, \mathbf{y}_B, \mathbf{z}_B\}$ ; the origin,  $O_B$ , is coincident with the robot CoM and the three axes,  $\mathbf{x}_B$ ,  $\mathbf{y}_B$ , and  $\mathbf{z}_B$ , are aligned with the edges of a cube of side length  $L$ . Let us define a point at the  $i$ -th propeller’s center as  $O_{p_i}$ , and its position in the body-fixed frame as  $\mathbf{p}_{p_i}^B$ . Then, we define, as done in [24], the matrix  $\mathbf{P} = [\mathbf{p}_{p_1}^B, \dots, \mathbf{p}_{p_8}^B]$ . Particularly, we have that

$$\mathbf{P}_p = \frac{L}{\sqrt{3}} \begin{bmatrix} 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \end{bmatrix}, \tag{1}$$

where the propellers are numbered as in Fig. 3.

**Fig. 1** OmniMorph: UA configuration (left) and OD configuration (right)





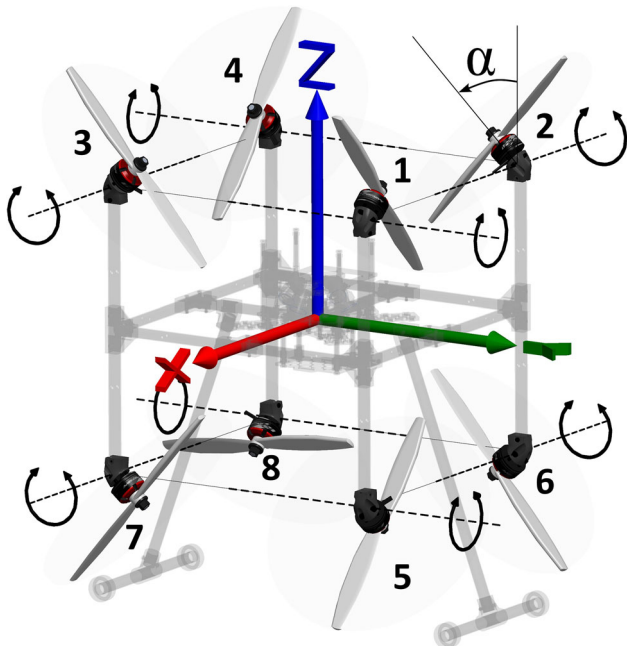
**Fig. 2** Sets of feasible forces for different values of  $\alpha$ . The three components of the thrust in the body frame are displayed. OmniMorph is omnidirectional when a sphere with a radius equal to the robot’s weight is inscribed inside the polytope of feasible forces

In [24], an optimal *fixed* orientation of the propellers to attain omnidirectionality is found. Especially, in [24], the authors define the coordinates of the spinning axis of their *fixedly tilted* propellers, let us call them  $\tilde{z}_i^B$ , and they collect them into the matrix  $\mathbf{B} = [\tilde{z}_1^B \dots \tilde{z}_8^B]$  given by

$$\mathbf{B} = \begin{bmatrix} -a & b & -b & a & a & -b & b & -a \\ b & a & -a & -b & -b & -a & a & b \\ c & -c & -c & c & c & -c & -c & c \end{bmatrix}, \quad (2)$$

where  $a = \frac{1}{2} + \frac{1}{\sqrt{12}}$ ,  $b = \frac{1}{2} - \frac{1}{\sqrt{12}}$ , and  $c = \frac{1}{\sqrt{3}}$ .

Let us now define as  $\tilde{b}_i$  the coordinates of the unit vector, centered on  $O_i$ , that describes the axis of rotation of propeller



**Fig. 3** Representation of OmniMorph in which the axes of rotation of the propellers, lying along the cube edges, are highlighted

$i$ –th to bring its spinning axis from the direction described in Eq. 2 to a configuration in which it is parallel to  $\tilde{z}_i$ , thus bringing the robot from the configuration described in [24] into the unilateral-thrust configuration. We have that  $\tilde{b}_i$  is given by the cross product between  $\tilde{z}_i$  and  $e_3$ , where  $e_i$  is the  $i$ –th column of the identity matrix  $\mathbf{I}_3 \in \mathbb{R}^{3 \times 3}$ .

Very interestingly,  $\tilde{b}_i$  is orthogonal to  $\tilde{b}_j \forall i \in \{1, 4, 5, 8\}, \forall j \in \{2, 3, 6, 7\}$ , similarly to Fig. 3, where the dotted lines representing the rotational axes of the propellers are orthogonal to each other. More than that, we note that by rotating the cube generated by those axes of  $\frac{\pi}{12}$  radians (i.e., 15 deg) around  $\mathbf{z}_B$ , one obtains that the propellers’ tilting axes lie along the edged of the cube formed by the propellers. Let us call these new propeller tilting axes  $\mathbf{b}_i$ . Such a symmetric configuration greatly simplifies the mechanical realization of the tilting mechanism, and, as it will be more clear in the following, it allows for transitioning between UDT and omnidirectional configurations. Eventually, that configuration is conveniently chosen for OmniMorph and is reported in Fig. 3.

To summarize formally, the positions of the centers of the propellers are given in Eq. 1, and the attitude is described as follows. Define a frame attached to each propeller  $\mathcal{F}'_i : \{O'_i, \mathbf{x}'_i, \mathbf{y}'_i, \mathbf{z}'_i\}$ , where  $\mathbf{x}'_i = \tilde{b}_i, \mathbf{z}'_i = \mathbf{z}'_B$ , and  $\mathbf{y}'_i$  is chosen orthogonal to the first two. Denote the rotation matrix between  $\mathcal{F}_B$  and  $\mathcal{F}'_i$  with  $\mathbf{R}'_i$ . Eventually, the rotation between  $\mathcal{F}_B$  and each propeller’s frame of OmniMorph,  $\mathcal{F}_{p_i} : \{O_{p_i}, \mathbf{x}_i, \mathbf{y}_i, \mathbf{z}_i\}$  is given by  ${}^B\mathbf{R}_i(\alpha) = \mathbf{R}'_i \mathbf{R}_Z(\frac{\pi}{12}) \mathbf{R}_X(\alpha)$ , where  $\mathbf{R}_X(\alpha) \mathbf{R}_Z(\alpha)$  are the elementary rotations about the  $X$  and  $Z$  axes. Each propeller exerts its thrust along the axis  $\mathbf{z}_i$  which is rotated by  $\alpha$  about an axis lying along the edges of the cube that connects the propellers’ centers.

### 2.2 Model of the OmniMorph

In this section, the dynamics equations of OmniMorph are given. Let us define an inertial frame  $\mathcal{F}_W : \{O_W, \mathbf{x}_W, \mathbf{y}_W,$

$\mathbf{z}_W$ . The position of  $O_B$  in  $\mathcal{F}_W$  is indicated as  $\mathbf{p}$  and its attitude is expressed compactly as  $\mathbf{R} := {}^W \mathbf{R}_B$ .<sup>1</sup>

The dynamic equations of the robot, expressed as Newton-Euler equations of an actuated rigid body, are as follows, where the translational dynamics is expressed in the inertial frame and the rotational dynamics in the body frame. We indicate as  $\boldsymbol{\omega}$  the angular velocity of  $\mathcal{F}_B$  w.r.t.  $\mathcal{F}_W$  expressed in  $\mathcal{F}_B$ , with  $m$  the robot mass, and with  $\mathbf{J} \in \mathbb{R}^{3 \times 3}$  its rotational inertia.

$$\begin{bmatrix} m\mathbf{I}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{J} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{p}} \\ \dot{\boldsymbol{\omega}} \end{bmatrix} = \begin{bmatrix} -mge_3 \\ -\boldsymbol{\omega} \times \mathbf{J}\boldsymbol{\omega} \end{bmatrix} + \begin{bmatrix} \mathbf{R} & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{I}_3 \end{bmatrix} \begin{bmatrix} \mathbf{f}_B^B \\ \boldsymbol{\tau}_B^B \end{bmatrix} \quad (3)$$

$\mathbf{f}_B^B$  and  $\boldsymbol{\tau}_B^B$  are the total force and torque applied to the CoM, expressed in  $\mathcal{F}_B$ .

The rotational kinematics of the robot is:

$$\dot{\mathbf{R}} = \mathbf{R}[\boldsymbol{\omega}]_{\times} \quad (4)$$

where  $[\boldsymbol{\omega}]_{\times}$  is the skew symmetric matrix associated to  $\boldsymbol{\omega}$ . Each rotor produces a thrust force and a drag moment which we model, similarly to [25], as follows.

$$\mathbf{f}_i = c_f w_i |w_i| \mathbf{z}_i, \quad \boldsymbol{\tau}_{d_i} = -k_i c_{\tau} w_i |w_i| \mathbf{z}_i, \quad (5)$$

where  $c_f, c_{\tau} > 0$  are the thrust and drag coefficient, respectively, and  $w_i$  is a scalar with a module equal to the norm of the propeller angular velocity and a sign defined such that  $w_i > 0$  when the thrust produced by the propeller has the same direction of  $\mathbf{z}_i$ . The drag moment is always opposite to the angular velocity of the propeller; as a consequence,  $k_i = -1$  for propellers with descending chord (i.e., producing thrust in the same direction of the angular velocity, also called ‘counterclockwise’) and  $k_i = +1$  for propellers with ascending chord (i.e., producing thrust in the opposite direction of the angular velocity, also called ‘clockwise’).

The applied body forces and torques are the results of all propellers and their orientation:

$$\mathbf{f}_B^B = \sum_{i=1}^n \mathbf{f}_i^B = \sum_{i=1}^n {}^B \mathbf{R}_{p_i} \mathbf{e}_3 f_i \quad (6)$$

$$\begin{aligned} \boldsymbol{\tau}_B^B &= \sum_{i=1}^n \boldsymbol{\tau}_{f_i}^B + \boldsymbol{\tau}_{d_i}^B = \sum_{i=1}^n \mathbf{p}_{p_i}^B \times \mathbf{f}_i^B - k_i c_{\tau}^{\tau} \mathbf{f}_i^B \\ &= \sum_{i=1}^n ([\mathbf{p}_{p_i}^B]_{\times} - k_i c_{\tau}^{\tau} \mathbf{I}_3) {}^B \mathbf{R}_{p_i} \mathbf{e}_3 f_i \end{aligned} \quad (7)$$

where  $c_{\tau}^{\tau} = c_{\tau}/c_f$ . We indicated with  $\mathbf{f}_i^B$  the thrust force of the  $i$ -th propeller expressed in  $\mathcal{F}_B$ , and with  $\boldsymbol{\tau}_{f_i}^B$  and  $\boldsymbol{\tau}_{d_i}^B$  the

torque produced by the same propeller’s thrust at the CoM and the effect of the propeller’s drag torque at the CoM, respectively.

We denote with  $\mathbf{u}$  the input of the system, consisting of the single morphing angle  $\alpha$  and the rotor speeds contained in the vector  $\mathbf{u}_w$  such that

$$\mathbf{u}_w = [u_{w_1} \cdots u_{w_n}]^T = [w_1 |w_1| \cdots w_n |w_n|]^T \in \mathbb{R}^n. \quad (8)$$

Hence, one has  $\mathbf{u} = [\mathbf{u}_w^T \alpha]^T \in \mathbb{R}^9$ . Based on Eqs. 6 and 7, we have that the actuation wrench is

$$\mathbf{w}(\mathbf{u}) = \begin{bmatrix} \mathbf{f}_B^B \\ \boldsymbol{\tau}_B^B \end{bmatrix} = \mathbf{A}(\alpha) \mathbf{u}_w. \quad (9)$$

The full expression of  $\mathbf{A}(\alpha)$  is at the end of Section 2.3.

### 2.3 Actuation Properties

First, we define the *full allocation matrix*  $\mathbf{F}(\mathbf{u}) \in \mathbb{R}^{6 \times 9}$  as in [6]:

$$\mathbf{F}(\mathbf{u}) = [\mathbf{F}_1(\alpha) \quad \mathbf{F}_2(\mathbf{u}_w, \alpha)] \quad (10)$$

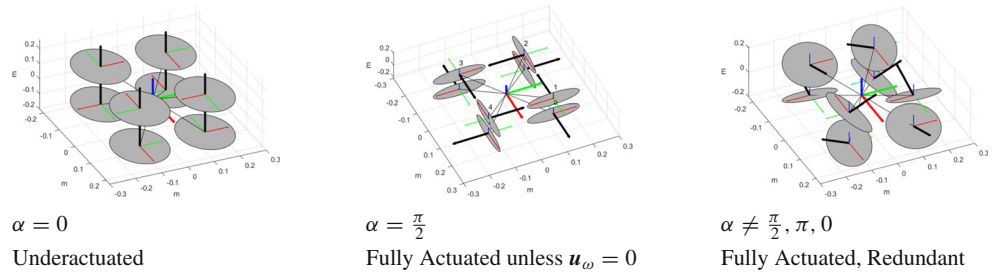
with  $\mathbf{F}_1(\alpha) = \frac{\partial \mathbf{w}}{\partial \alpha}(\alpha) \in \mathbb{R}^{6 \times 8}$ ,  $\mathbf{F}_2(\mathbf{u}) = \frac{\partial \mathbf{w}}{\partial \mathbf{u}_w}(\mathbf{u}) \in \mathbb{R}^6$ . Note that  $\mathbf{F}_1$  only depends on  $\alpha$  because  $\mathbf{w}$  in Eq. 9 depends *linearly* on  $\mathbf{u}_w$ . Particularly, following the definition given above and considering Eq. 9, we have that  $\mathbf{F}_1 = \mathbf{A}(\alpha)$  and  $\mathbf{F}_2 = \frac{\partial(\mathbf{A}(\alpha)\mathbf{u}_w)}{\partial \alpha}$ . Following the method provided in [6], we study the actuation properties of the platform by looking at the rank of  $\mathbf{F}(\mathbf{u})$  in Eq. 10. For all the points in the input space in which  $\text{rank}(\mathbf{F}(\mathbf{u})) = 6$  (full rank), the platform is fully actuated. In those points, the actuation wrench can be changed in every direction by suitably acting on the input. Instead, in all points in which  $\text{rank}(\mathbf{F}(\mathbf{u})) < 6$  (rank deficient) the platform is underactuated, this means that when the input is in those singular points there are some forbidden directions along which the actuation wrench cannot be changed.

First, we note that studying the rank of  $\mathbf{F}_1(\alpha)$  helps us study the rank of the full allocation matrix,  $\mathbf{F}(\mathbf{u})$ :

- $\text{rank}(\mathbf{F}_1(\alpha)) = 4$  is a sufficient condition for the platform to be underactuated: the rank of the full allocation matrix  $\mathbf{F}(\mathbf{u})$  can be at most  $5 < 6$ .
- $\text{rank}(\mathbf{F}_1(\alpha)) = 6$  is a sufficient condition for the system to be fully actuated;
- If  $\text{rank}(\mathbf{F}_1(\alpha)) = 5$ , the full-/under-actuation needs further analysis of  $\mathbf{F}_2(\mathbf{u}) \in \mathbb{R}^{6 \times 1}$  in relation to  $\mathbf{F}_1(\alpha)$ .

<sup>1</sup> In general, we indicate as  ${}^A \mathbf{R}_A$  the rotation from frame  $\mathcal{F}_A$  to  $\mathcal{F}_A$ .

**Table 1** Actuation properties for different values of  $\alpha$



Grey circles are the propellers' planes. Each is associated with a local frame, and a body-fixed frame is in the middle of the robot. The RGB convention is used to indicate the x-, y-, and z-axis, respectively. The direction of the thrust produced by the individual propellers is represented with a thick black arrow

Hence, we study the rank of  $\mathbf{F}_1(\alpha)$  for various values of  $\alpha$ , finding that

$$\text{rank}(\mathbf{F}_1(\alpha)) = \begin{cases} 4 & \text{if } \alpha = 0 + k\pi, k \in \mathbb{N} \\ 5 & \text{if } \alpha = \frac{\pi}{2} \\ 6 & \text{otherwise} \end{cases} \quad (11)$$

So, from what has been said so far, OmniMorph is underactuated for  $\alpha = 0$ . Moreover,  $\text{rank}(\mathbf{F}(\mathbf{u}_w, 0)) = 5$  if  $\mathbf{u}_w \neq \mathbf{0}$  and if not all  $u_{wi}$  are equal to each other, i.e., the total wrench can be instantaneously changed on a 'reduced' five-dimensional manifold.

For  $\alpha = \pi/2$ , OmniMorph is in general under-actuated. However, it is fully actuated ( $\text{rank}(\mathbf{F}(\mathbf{u}_w, \pi/2)) = 6$ ) if  $\mathbf{u}_w \neq \mathbf{0}$ . An intuitive interpretation is provided in the following. When  $\alpha = \pi/2$ , all propellers' thrusts are on the same plane in  $\mathcal{F}_B$ . The yaw acceleration can be changed as the individual thrusts generate a torque around the vertical axis of the robot. The pitch and roll accelerations can be changed thanks to the drag moment of the propellers. The translational acceleration in the horizontal plane is clearly changeable by the thrust, but the translational acceleration along  $\mathbf{z}_B$  is not. As soon as  $\alpha$  changes, the propellers' thrusts have a component in the vertical direction as well, which gives control of the vertical acceleration, unless the propellers' thrusts are actually zero (mathematically,  $\mathbf{u}_w = \mathbf{0}$ ).

Even if the platform is fully actuated it might not be able to fly properly because the propeller thrust intensity may be not large enough to sustain the platform's weight.

Eventually, for all other values of  $\alpha$ , the platform is fully actuated. A schematic summary of the OmniMorph actuation properties is in Table 1.

For sufficiently large values of  $\alpha$ , the polygon of the achievable forces, see Fig. 2 contains a sphere whose radius corresponds to the weight of the platform: the robot can sustain its weight in any orientation; thus, it is omnidirectional.

For the sake of completeness, the full allocation matrix is reported in the following, where  $s_\alpha$  and  $c_\alpha$  indicate the sin and cosine of  $\alpha$ , respectively:

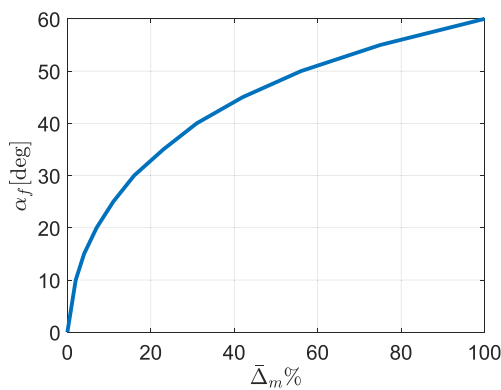
$$\mathbf{F} = c_f \begin{bmatrix} -s_\alpha & 0 & s_\alpha & 0 \\ 0 & s_\alpha & -s_\alpha & 0 \\ c_\alpha & c_\alpha & c_\alpha & c_\alpha \\ Lc_\alpha - \frac{c_\tau s_\alpha}{c_f} & Lc_\alpha + Ls_\alpha \frac{c_\tau s_\alpha}{c_f} & -Lc_\alpha & -Lc_\alpha - Ls_\alpha \\ -Lc_\alpha - Ls_\alpha & Lc_\alpha + \frac{c_\tau s_\alpha}{c_f} & Lc_\alpha + Ls_\alpha & -Lc_\alpha - \frac{c_\tau s_\alpha}{c_f} \\ Ls_\alpha + \frac{c_\tau c_\alpha}{c_f} & Ls_\alpha - \frac{c_\tau c_\alpha}{c_f} & Ls_\alpha + \frac{c_\tau c_\alpha}{c_f} & Ls_\alpha - \frac{c_\tau c_\alpha}{c_f} \\ s_\alpha & 0 & 0 & -s_\alpha \\ 0 & -s_\alpha & s_\alpha & 0 \\ c_\alpha & c_\alpha & c_\alpha & c_\alpha \\ Lc_\alpha + \frac{c_\tau s_\alpha}{c_f} & -Lc_\alpha - Ls_\alpha & -Lc_\alpha + Ls_\alpha & -Lc_\alpha - \frac{c_\tau s_\alpha}{c_f} \\ -Lc_\alpha - Ls_\alpha & \frac{c_\tau s_\alpha}{c_f} - Lc_\alpha & Lc_\alpha - \frac{c_\tau s_\alpha}{c_f} & Lc_\alpha + Ls_\alpha \\ \frac{c_\tau c_\alpha}{c_f} - Ls_\alpha & -Ls_\alpha - \frac{c_\tau c_\alpha}{c_f} & -Ls_\alpha - \frac{c_\tau c_\alpha}{c_f} & \frac{c_\tau c_\alpha}{c_f} - Ls_\alpha \end{bmatrix} \frac{\partial \mathbf{w}}{\partial \alpha}$$

### 2.4 Power Consumption Analysis

To analyze the power consumption of the OmniMorph in comparison with a non-morphing fully actuated platform, let us start by computing the input needed to enforce the static equilibrium when the platform is hovering with orientation  $\mathbf{R} = \mathbf{I}_3$ ; we indicate that input as  $\mathbf{u}_w^h(\alpha, m)$  and retrieve it by inverting Eq. 9 as

$$\mathbf{u}_w^h(\alpha, m) = (\mathbf{A}(\alpha))^\dagger \begin{bmatrix} mge_3 \\ \mathbf{0} \end{bmatrix}. \quad (12)$$

The power consumed by each brushless motor is equal to the scalar product between the motor torque and its angular velocity. Given that in hovering we can approximate the motor torque with the drag moment and that the drag moment is always opposing the angular velocity, we can write the power consumed by the  $i$ -th brushless motor as  $P_{\tau i}(u_{wi}) = \|\boldsymbol{\tau}_{d_i}\| |w_i| = c_\tau |w_i|^3 = c_\tau |u_{wi}|^3/2$ . The total



**Fig. 4** For a given fixed propeller tilting angle  $\alpha_f$ , a morphing design is convenient, provided that the relative mass of the tilting mechanism is lower than  $\bar{\Delta}_m$

power consumed by all the motors is then

$$P_\tau(\mathbf{u}_w) = \sum_{i=1}^8 P(u_{w_i}) = c_\tau \sum_{i=1}^8 |u_{w_i}|^{3/2}. \tag{13}$$

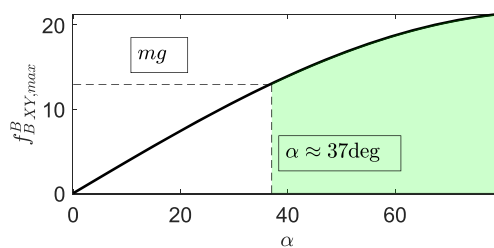
Therefore,  $P_\tau(\mathbf{u}_w^h(\alpha, m))$  represents the power consumed at hovering by a platform with a certain mass  $m$  and a given tilting angle  $\alpha$ .

Let us now write the robot’s mass as follows:  $m = m_0(1 + \Delta_m)$ , where  $\Delta_m$  is the ratio between the mass of the propeller tilting mechanism and  $m_0$ , which is the mass of all other components; indeed,  $\Delta_m$  is given by  $\frac{m-m_0}{m_0}$ .

Now, consider instead, a *non-morphing* multirotor, in which a certain level of dexterity is obtained by tilting the propellers by a *fixed* angle  $\alpha_f$ . We ask ourselves which is the critical value of  $\Delta_m$ , call it  $\bar{\Delta}_m$ , below which a morphing design would be convenient for such a platform. Indeed, it has to be taken into account that the morphing design can efficiently hover with  $\alpha = 0$  (all propellers parallel to each other, so no thrust is waisted into internal forces) but it will have to sustain a larger mass due to the tilting mechanism ( $\Delta_m > 0$ ). On the contrary, the non-morphing platform is lighter ( $\Delta_m = 0$ ) but due to the propellers’ fixed tilt part of the thrust generated to hover is wasted in internal forces. To understand which is the maximum value of  $\Delta_m$  for which the morphing design is the most efficient one, namely to compute  $\bar{\Delta}_m$ , we numerically solve the following implicit relation between  $\Delta_m$  and  $\alpha$ :

$$P_\tau(\mathbf{u}_w^h(0, m_0(1 + \bar{\Delta}_m))) = P_\tau(\mathbf{u}_w^h(\alpha_f, m_0)).$$

Clearly,  $\bar{\Delta}_m$  depends on the tilting angle of the non-morphing design. The values of  $\bar{\Delta}_m$  as a function of  $\alpha_f$  for the OmniMorph are reported in Fig. 4. Intuitively, for  $\alpha_f = 0$ , namely when no dexterity is required from the fixed-propeller platform, which is then a standard uni-directional platform,



**Fig. 5** Radius of the maximum inscribed sphere is the feasible force set in Fig. 2 for different values of  $\alpha$ . When this is equal to its weight, OmniMorph can sustain its weight in all orientations

one has that  $\bar{\Delta}_m = 0$ . On the other side, for instance, for  $\alpha_f=20$  deg, the use of a morphing platform is more power-efficient for a weight of the tilting mechanism up to 7% that of all the other components. For the OmniMorph,  $\alpha_f \geq 30$  deg, which tells us that the morphing platform is more power-efficient than a non-morphing one even for a weight of the tilting mechanism that constitutes up to 16% of all the other components’ weight. Note that the minimum  $\alpha_f$  to attain omnidirectionality depends on the maximum motor speed, the propeller type, and the mass of the vehicle. We are considering here the same bi-directional propellers as in [24] with the same maximum motor speed and a mass  $m_0 = 1.128$  kg equal to the mass of the OmniMorph without any tilting mechanism. Considering also the additional mass of the tilting mechanism, the minimum value of  $\alpha$  for which OmniMorph is omnidirectional is  $\alpha \approx 37$  deg, as shown in Fig. 5. Clearly, there is also a maximum value for which OmniMorph loses omnidirectionality. OmniMorph cannot sustain its weight in hovering as soon as  $\alpha \geq 72$  deg.

The power consumption analysis here presented is at a steady state. The power consumed in a dynamic task by the OmniMorph and by an analogous platform but with fixedly tilted propellers is discussed in Section 4.1.

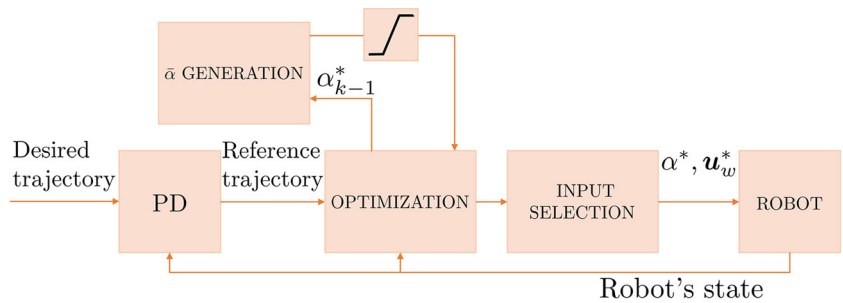
### 3 Control

This section introduces a control law that allows OmniMorph to follow a desired 6D trajectory switching between the underactuated and the fully actuated configurations to account for the minimization of trajectory tracking error and the minimization of the input  $\mathbf{u}_w$ .

Let us define  $\ddot{\mathbf{q}} = [\ddot{\mathbf{p}}^\top, \dot{\boldsymbol{\omega}}^\top]^\top$ . Given a reference trajectory for the UAV, indicated through the superscript  $(\cdot)^d$ , we compute the reference acceleration of the robot, call it  $\ddot{\mathbf{q}}^r$ , using a PD feedback control plus a feedforward term. The following desired input acceleration is obtained:

$$\ddot{\mathbf{q}}^r = \begin{bmatrix} \ddot{\mathbf{p}}^d + \mathbf{K}_{p1}\dot{\mathbf{e}}_p + \mathbf{K}_{p2}\mathbf{e}_p \\ \dot{\boldsymbol{\omega}}^d + \mathbf{K}_{\omega 1}\dot{\mathbf{e}}_\omega + \mathbf{K}_{\omega 2}\mathbf{e}_R \end{bmatrix} \tag{14}$$

**Fig. 6** Block diagram of the control method



where  $e_R = \frac{1}{2}[\mathbf{R}^\top \mathbf{R}^d - \mathbf{R}^{d\top} \mathbf{R}]_\vee$ , with  $\vee$  from  $so(3)$  to  $\mathbb{R}^3$  being the inverse of the hat map [26],  $e_p = p^d - p$ , and  $e_\omega = \omega^d - \omega$ .

We now design an inner control loop to track  $\ddot{q}^r$  thanks to suitable inputs  $\alpha^*$  and  $u_w^*$ , chosen as the solution to:

$$\begin{aligned} & \min_{\alpha, u_w, \ddot{q}} J_1 + J_2 + J_3 \\ & s.t. \quad \mathbf{M}\ddot{q} = \mu + \mathbf{J}_R \mathbf{F}(\alpha) u_w, \\ & \quad \mathbf{C} u_w < \mathbf{b}, \\ & \quad -\epsilon_\alpha \leq \alpha - \alpha_{k-1}^* \leq \epsilon_\alpha \end{aligned} \tag{15}$$

where the cost function is composed of three terms:  $J_1 = \|u_w\|_{\mathbf{W}_1}^2$  is to minimize the norm of the input,  $J_2 = \|\ddot{q}^r - \ddot{q}\|_{\mathbf{W}_2}^2$  to ensure tracking of the desired trajectory, and  $J_3 = \|u_w - u_{w,k-1}^*\|_{\mathbf{W}_3}^2$  to minimize the propeller spinning accelerations. The quantities  $\alpha_{k-1}^*$  and  $u_{w,k-1}^*$  are the solution of the optimization problem Eq. 15 at the previous time step.  $\|\cdot\|_{W_i}$  is the 2-norm weighted by the positive definite weight matrix  $\mathbf{W}_i$ . Alternatively, if the bounds on the propeller accelerations are identified, as in [23],  $J_3$  could be also expressed as a unilateral constraint.

The equality constraint in Eq. 15 is the system’s dynamics, where we have indicated the inertia matrix as  $\mathbf{M} = \begin{bmatrix} m\mathbf{I}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{J} \end{bmatrix}$ , the vector of the gravity and Coriolis terms as  $\mu = \begin{bmatrix} -mge_3 \\ -\omega \times \mathbf{J}\omega \end{bmatrix}$ , and  $\mathbf{J}_R = \begin{bmatrix} \mathbf{R} & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{I}_3 \end{bmatrix}$ ; the second, unilateral, constraint in Eq. 15 expresses the input constraint, where  $\mathbf{C}$ ,  $\mathbf{b}$  are properly defined constant quantities; the last constraint is on the rate of change of  $\alpha$ . The maximum rate of change  $\epsilon_\alpha$  is here defined as symmetric but in the general case they may also be non-symmetric and the analysis still holds. The term  $J_1$  has an equivalent effect to minimizing the norm of the drag moments, as it differs from the input norm by a constant coefficient.

In order to solve problem Eq. 15, which is not a QP problem because the first constraint is nonlinear in the optimization variables, we proceed as follows. Consider the

following problem for a fixed value of  $\alpha$ , indicated as  $\bar{\alpha}$ .

$$\begin{aligned} & \min_{u_w, \ddot{q}} J_1 + J_2 + J_3 \\ & s.t. \quad \mathbf{M}\ddot{q} = \mu + \mathbf{J}_R \mathbf{F}(\bar{\alpha}) u_w, \\ & \quad \mathbf{C} u_w < \mathbf{b} \end{aligned} \tag{16}$$

Problem Eq. 16 is now a QP problem as  $\alpha$  is not an optimization variable anymore, and, as a consequence, the bilateral constraint is linear in the optimization variables.

At each time step Eq. 16 is solved three times for three different values of  $\bar{\alpha} = \{\alpha_{k-1}^* - \epsilon_\alpha, \alpha_{k-1}^*, \alpha_{k-1}^* + \epsilon_\alpha\}$ . Hence, at each time step, the solutions  $\alpha^*$  and  $u_w^*$  that correspond to the lowest value of the cost function are selected. A schematic representation of the proposed control scheme is in Fig. 6.

The block that feeds the optimization with the three different values of  $\alpha$  also performs a saturation to make sure that the values of  $\alpha$  provided by the controller comply with the platform kinematics constraints. Especially, the propellers are allowed to assume values between 0 and 60 deg. We saw that the theoretical maximum value of  $\alpha$  for OmniMorph to sustain its weight in any configuration was about 70 deg; however, we decided to be more conservative than the theoretical upper bound to keep always enough thrust in the vertical direction to account for model uncertainties, rejection of external disturbances, low battery levels, etc.

In principle, the selection of the parameter  $\bar{\alpha} \in [\alpha_{k-1}^* - \epsilon_\alpha, \alpha_{k-1}^* + \epsilon_\alpha]$  may be done following different methods. Here we proposed a simple method that allows keeping the value of the tilting angle constant or changing it in the two possible directions accounting for the bound on the variation of  $\alpha$ . Other approaches such as an exhaustive search or more efficient sampling algorithms are possibilities, and their assessment is left for future work.

Note that, by tuning the weights  $\mathbf{W}_i$ , the control input negotiates between low input effort and low tracking error. We recall that, thanks to its redundancy, the robot is able to follow a certain desired trajectory with different values of the control inputs.



## 4 Results

### 4.1 Simulations

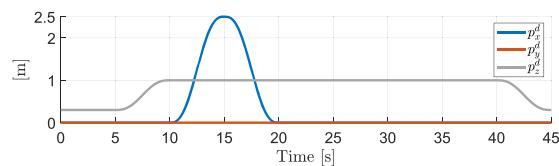
Numerical simulations have been carried out using a Unified Robot Description Format (URDF) description of the OmniMorph and ordinary differential equation (ODE) physics engine in Gazebo. The control software has been implemented in Matlab-Simulink. The interface between Matlab and Gazebo is managed by a Gazebo-genom3 plugin<sup>2</sup>. A Simulink s-function embedding qpOases QP solver<sup>3</sup> has been used for the optimization problem.

The mass and inertia of the robot are  $m = 1.3150$  kg and  $\mathbf{J} = \text{diag}(1.16 \cdot 10^{-2}, 1.13 \cdot 10^{-2}, 1.13 \cdot 10^{-2})\text{Nms}^2$ . The other parameters are

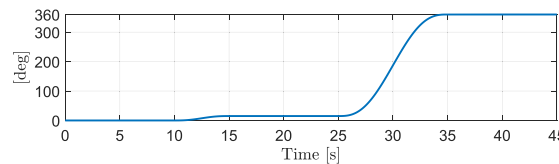
- $\mathbf{K}_{p1} = \text{diag}(30,30,30) \text{ s}^{-1}$ ,
- $\mathbf{K}_{p2} = \text{diag}(300,300,300) \text{ s}^{-2}$ ,
- $\mathbf{K}_{\omega 1} = \text{diag}(40,40,40) \text{ s}^{-1}$ ,
- $\mathbf{K}_{\omega 2} = \text{diag}(100,100,100) \text{ s}^{-2}$ ,
- $\mathbf{W}_3 = 10^{-5} \mathbf{I}_8$ .

The tilting angle  $\alpha$  is saturated at 60 deg, a value sufficient to reach the omnidirectionality, as it will be in the real robot. A PD controller has been used to generate the reference trajectory of the robot’s CoM, but any other suitable controller could be used for that purpose, e.g., based on dynamic inversion, etc. Indeed, the main contribution of this work from a control point of view is the optimization, and PD gains are not of primary importance for showing the work’s contribution. Fine gain tuning, although important for performance enhancement, is, thus, considered out of the scope of this work. The PD gains have been handcrafted and set once and for all. All the following results are purposely obtained with the same PD controller gains, to highlight the contribution of the optimization. Advanced methods for automatic gain tuning based, e.g., on neural networks[27], reinforcement learning [28], or stochastic optimization [7] have been proposed in the literature, and they can be considered for refining the performance of the controlled system in future work.

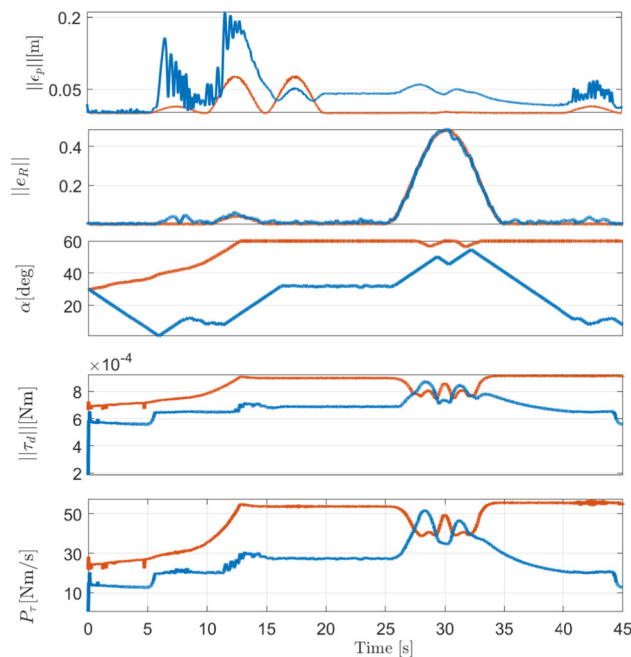
We show two different simulation scenarios, let us refer to them as Case A and Case B, in which the robot’s desired trajectory is the same, but the optimization gains are different. Especially, the trajectory includes moving upwards/downwards, translating with a tilted attitude, and rotating by 360 degrees on the spot— see Figs. 7a and 7b. It has been chosen to cover the whole spectrum of the actuation modes: the vertical motion with constant upward orientation is feasible in the underactuated mode, the lateral translations with constant upward orientation require full actuation, and



(a) Desired position



(b) Desired attitude expressed as a rotation around  $\mathbf{x}_W$



(c) Case A: in orange line; Case B: blue line

**Fig. 7** Desired trajectory (7a and 7b) and results (7c) of the Gazebo simulations

the full 360 deg rotation on the spot about a horizontal axis requires omnidirectionality.

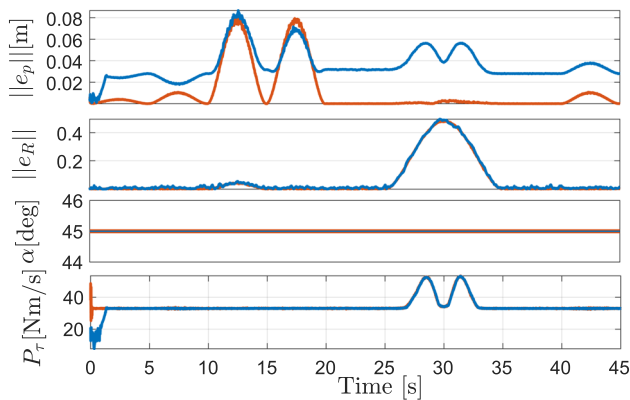
What distinguishes the two scenarios is that Case A is characterized by a lower weight on the input,  $\mathbf{W}_1$ , and a higher weight on the tracking error,  $\mathbf{W}_2$ , than Case B.

The optimization weights for Case A are  $\mathbf{W}_1 = 10^{-8} \mathbf{I}_8$  and  $\mathbf{W}_2 = \text{diag}(3 \cdot 10^6, 3 \cdot 10^6, 3 \cdot 10^6, 10^3, 10^3, 10^3)$ . The optimization weights for Case B are  $\mathbf{W}_1 = 10^{-5} \mathbf{I}_8$  and  $\mathbf{W}_2 = \text{diag}(3 \cdot 10^4, 3 \cdot 10^4, 3 \cdot 10^4, 10, 10, 10)$ .

In Fig. 7c, we report a comparison between the results of the two simulation scenarios. One can appreciate as in Case A the robot performs better in terms of tracking (the average position and attitude errors are, respectively, 0.011 m and 0.062), at the expenses of a higher motor torque; the tilting

<sup>2</sup> <https://git.openrobots.org/projects/mrsim-gazebo>

<sup>3</sup> <https://github.com/coin-or/qpOASES>



**Fig. 8** Results of the Gazebo simulations in Case A (red) and Case B (blue) for the propeller tilting angle fixed at  $\alpha = 45$  deg. The power consumption remains almost identical in the two cases

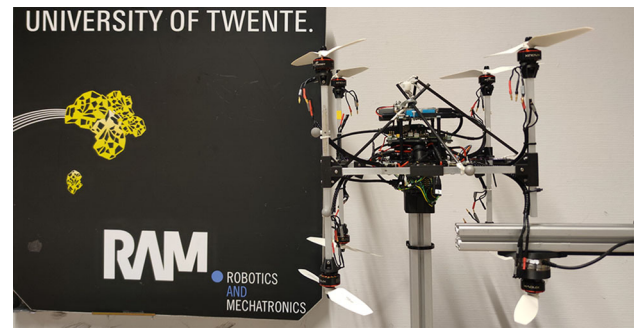
angle is increased and kept at the highest value along the task execution. On the contrary, in Case B, the errors are higher (mean square position and attitude errors are, respectively, 0.044 m and 0.068), but the motor torque is, as expected, lower, and the propeller tilting angle is decreased as soon as the robot only moves vertically (as soon as the underactuated mode is enough to perform the desired motion). The efficiency is higher in Case B with the value of energy related to the drag moment equal to 1162.3 J against 2080.7 J in Case A.

Note that, in a trajectory tracking task like the one shown here, there is another source of power consumption besides the power consumed by the drag: it is the power needed to accelerate the propellers. If  $dE$  is the instantaneous variation of the kinetic energy of all the eight propellers, the instantaneous power is computed as  $\frac{dE}{dt}$ . Considering the mass of each propeller equal to 0.01 kg, the total energy consumed to accelerate all the propellers is equal to 65.5 J in Case A and 225.3 J in Case B.

This suggests that only a minor part of the energy is spent to accelerate the propellers, and Case B remains the most efficient scenario. One can act on  $\mathbf{W}_3$  to reduce the propeller acceleration. We carried out the same simulation as in Case B but increasing the weight on the propeller acceleration to  $\mathbf{W}_3 = 10^{-4}$ ; we obtained a total consumed energy to accelerate the propellers equal to 87.7 J. The total corresponding tracking errors were comparable to Case B: 0.0412 cm in position and 0.063 in attitude.

#### 4.1.1 Simulations at a Fixed $\alpha$

We performed simulations in Case A and Case B with a fixed value of  $\alpha = 45$  deg— see Fig. 8. The total consumed energy is around 1400 J in both cases, namely in between the energy consumed by OmniMorph to perform an analogous task in Case A and Case B. The ratio between the energy



**Fig. 9** Experimental setup for  $\alpha = 0$

consumed by the fixed-propeller robot and OmniMorph ultimately depends on the specific task, e.g., on the portion of vertical motion and hovering, not requiring large values of  $\alpha$ .

However, it is worth noting that fixedly tilted propellers seem to offer less room to tune the efficiency, as the power consumption corresponding to the same reference trajectory remains basically the same for Cases A and B. Instead, OmniMorph offers the possibility to exploit the additional degree of freedom to reach the desired balance between efficiency and performance.

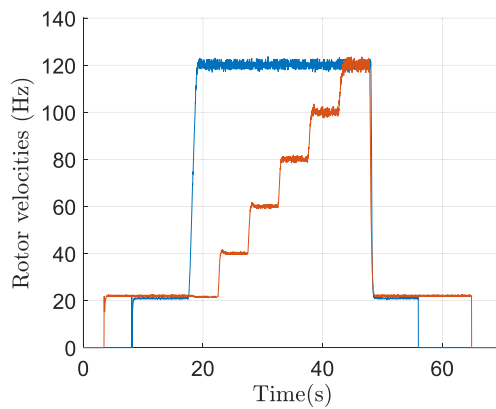
## 4.2 Propellers' Mutual Aerodynamic Interference

### 4.2.1 Experiments

In this section, we present preliminary results to experimentally assess the propellers' mutual aerodynamic interference in the OmniMorph prototype. A picture of the experimental setup is in Fig. 9. A prototype of OmniMorph without the tilting mechanism is attached to a fixed structure. A propeller was fixed to an ATI Mini40<sup>4</sup> force-torque sensor at the same location it would have on the OmniMorph body. The propeller measured by the force torque sensor was commanded to spin at a constant rate of 120Hz. The remaining 7 propellers were accelerated in a stepwise fashion with 20Hz increments to a maximum propeller speed of 120Hz, as reported in Fig. 10. Electronic speed controllers Kiss Racing 32A<sup>5</sup> were used. The effect of aerodynamic interference was estimated by measuring the change in force on the single propeller due to the increasing speed of the remaining propellers. A propeller in the lower part has been purposely chosen to consider the worst-case scenario as it gets invested by the downward air-flow of the top propeller. Two values of  $\alpha = [0, 50]$  deg have been tested by fixing the propellers with custom 3D-printed parts.

<sup>4</sup> [https://www.ati-ia.com/products/ft/ft\\_models.aspx?id=Mini40](https://www.ati-ia.com/products/ft/ft_models.aspx?id=Mini40)

<sup>5</sup> <https://www.getfpv.com/kiss-32a-32bit-esc-2-6s.html>



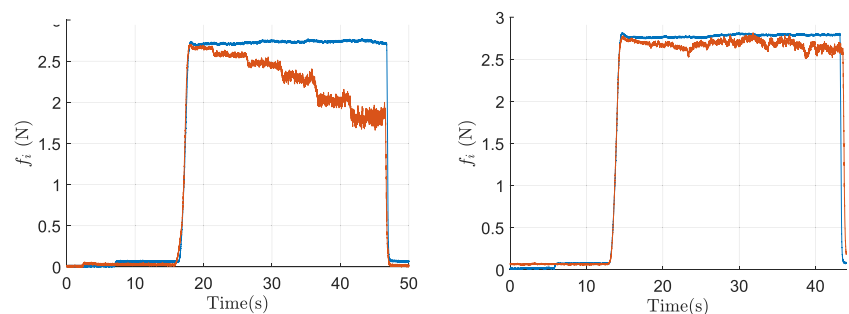
**Fig. 10** Profiles of the propellers' velocities during the tests. In blue, the constant velocity of the propeller attached to the force sensor; in red the velocity of the others

It emerges that, as intuitively expected, the aerodynamic interference is higher when  $\alpha = 0$ , as the entire airflow of the top propellers is directed downwards toward the bottom ones. In that case, the measured force generated by the individual propeller goes from 2.7 N to about 1.8 N, i.e., it decreases by a maximum of about 33% of its nominal value. On the other hand, the maximum decrease when  $\alpha = 50\%$  is around 7%. These results are reported in Fig. 11. The non-trivial computation of the perturbed allocation matrix as a function of all propellers' individual speed and of the angle  $\alpha$  is left as future work. However, in the next section, we show simulation results to assess the robustness of the proposed controller to aerodynamic interferences.

#### 4.2.2 Simulations with Diminished Propeller Performance

Based on the previous results, in this section, we show two simulations to assess the robustness of the proposed controller against decreased propeller performance. Especially, the actual  $c_f$  of OmniMorph is lower than the nominal one known by the controller. In reality, the amount varies for each propeller depending on the others' spinning velocity and  $\alpha$ . The assessment of an update law for  $c_f$  based on experimental data is left for future work. As a worst-case scenario, we considered in the simulations that the actual thrust coefficient

**Fig. 11** The blue values are the force sensor measures when the propeller attached to the sensor is the only one spinning. The red values are the measures when the other propellers spin as in Fig. 10.  $\alpha = 0$  on the left;  $\alpha = 50$  deg on the right

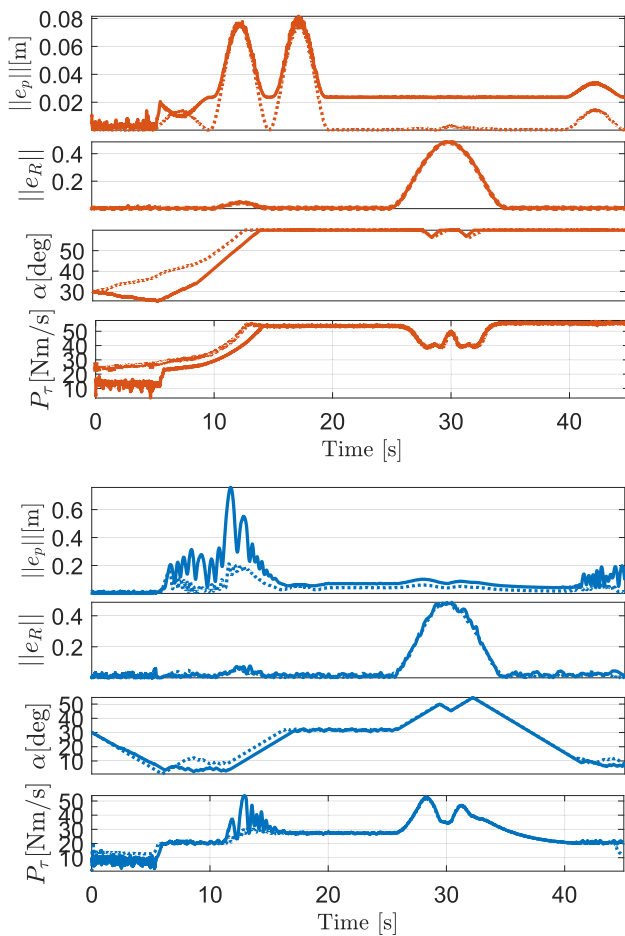


is constant and 35% lower than the nominal one. While the platform successfully carries out the task in Case A, although with a slightly higher tracking error than in the ideal case, in Case B the robot fails to recover from high tracking errors seemingly due to the scarce dexterity in the horizontal directions caused by low values of  $\alpha$ ; however, Case B is still successful when  $c_f$  is 30% lower than the nominal one. That suggests that, in the presence of such a mismatch between the model and the system, high priority should be given to the tracking performance to improve robustness. However, we simulated the worst-case scenario in which the mismatch is constantly high and is the *same for all propellers*, while in the real platform, the top propellers would likely suffer from considerably lower interference. Figure 12 collects the results in the two aforementioned cases. The position error is the most affected in both cases. Recordings of the simulations are provided in the attached multimedia material.

## 5 Conclusions

This work presented the design concept of OmniMorph, a novel morphing multirotor that can range between omnidirectionality and underactuation thanks to actively tilting propellers. The design leverages one single servomotor to synchronously tilt all the propellers, thus reducing the mechanical complexity and the additional payload. The dynamics model was presented, and the actuation properties depending on the propeller tilting angles were studied. Hence, a controller was proposed to negotiate between input effort and tracking performance. Simulations in a realistic Gazebo environment were presented. A preliminary prototype was built and tests were carried out to assess the propeller aerodynamic interferences on the real prototype, and additional simulation results to assess the robustness against them were shown.

In the future, experiments on the real prototype will be carried out, and predictive controllers will be tested. Computing the online changes of the allocation matrix due to aerodynamic interferences depending on all individual propeller speeds and on  $\alpha$ , and incorporating such information on the optimal control is left as interesting future work. Increas-



**Fig. 12** Top: results for Case A with actual  $c_f$  lower than the nominal one by 35%. Bottom: results for Case B with actual  $c_f$  lower than the nominal one by 30%. The results of the ideal cases are superimposed in dotted lines

ing the dimensions of the prototype to allow better spacing of the propellers will also be considered. Equipping the OmniMorph with physical interaction capabilities in an interesting future direction.

**Acknowledgements** The authors thank Gianluca Corsini for contributing to the simulator and Aaron Saini for the code to plot the feasible force sets.

**Author Contributions** Y. Aboudorra contributed to deriving the modeling, the control law, designing the simulator, and writing the manuscript, as well as to the background literature research. C. Gabellieri contributed to the theoretical analysis, controller implementation, and manuscript writing, as well as to the background literature research; she worked on the simulation results and the multimedia files. R. Brančes contributed to testing the prototype and to the manuscript writing. Q. Sabl contributed to the OmniMorph design and to building and testing the prototype. A. Franchi is the initiator of the OmniMorph research idea, he did the fundraising and is the PI of the research project. He contributed to the modeling and the control law, and to the manuscript writing, as well as to the background literature research.

**Funding** This work was partially funded by European Commission projects Horizon2020 AUTOASSESS (101120732) and MSCA project Flyflic (101059875).

**Code Availability** Data sets generated during the current study are available from the corresponding author on reasonable request.

## Declarations

**Conflicts of interest** The authors have no relevant financial or non-financial interests to disclose.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

- Ollero, A., Tognon, M., Suarez, A., Lee, D., Franchi, A.: Past, Present, and Future of Aerial Robotic Manipulators. *IEEE Trans. Robot.* **38**(1), 626–645 (2022)
- Tognon, M., Alami, R., Siciliano, B.: Physical Human-Robot Interaction With a Tethered Aerial Vehicle: Application to a Force-Based Human Guiding Problem. *IEEE Trans. Robot.* **37**(3), 723–734 (2021)
- Tognon, M., Tello Chávez, H.A., Gasparin, E., Sablé, Q., Bicego, D., Mallet, A., Lany, M., Santi, G., Revaz, B., Cortés, J. et al.: A truly-redundant aerial manipulator system with application to push-and-slide inspection in industrial plants. *IEEE Robot. Autom. Lett.* **4**(2), 1846–1851 (2019)
- Gabellieri, C., Tognon, M., Sanalidro, D., Franchi, A.: Equilibria, stability, and sensitivity for the aerial suspended beam robotic system subject to model uncertainty, (2023). [arXiv preprint arXiv:2302.07031](https://arxiv.org/abs/2302.07031)
- Corsini, G., Jacquet, M., Das, H., Afifi, A., Sidobre, D., Franchi, A.: Nonlinear model predictive control for human-robot handover with application to the aerial case. In: 2022 IEEE/RSJ International conference on intelligent robots and systems (IROS), IEEE, pp. 7597–7604 (2022)
- Hamandi, M., Usai, F., Sablé, Q., Staub, N., Tognon, M., Franchi, A.: Design of multirotor aerial vehicles: A taxonomy based on input allocation. *Int. J. Robot. Res.* **40**(8-9), 1015–1044 (2021) Publisher: SAGE Publications Ltd STM
- Kose, Oguz, Oktay, Tugrul: Simultaneous design of morphing hexarotor and autopilot system by using deep neural network and spsa. *Aircr. Eng. Aerosp. Technol.* **95**(6), 939–949 (2023)
- Şahin, Hüseyin., Kose, Oguz, Oktay, Tugrul: Simultaneous autonomous system and powerplant design for morphing quadrotors. *Aircr. Eng. Aerosp. Technol.* **94**(8), 1228–1241 (2022)

9. Kose, Oguz, Oktay, Tugrul: Simultaneous quadrotor autopilot system and collective morphing system design. *Aircr. Eng. Aerosp. Technol.* **92**(7), 1093–1100 (2020)
10. Kose, O., Oktay, T., Özen, E.: Simultaneous arm morphing quadcopter and autonomous flight system design. *Air. Eng. Aerosp. Technol.* (2023)
11. Hamandi, M., Sable, Q., Tognon, M., Franchi, A.: Understanding the omnidirectional capability of a generic multi-rotor aerial vehicle. In: 2021 Aerial robotic systems physically interacting with the Environment (AIRPHARO)
12. Hamandi, M., Sawant, K., Tognon, M., Franchi, A.: Omni-Plus-Seven (O7+): An Omnidirectional Aerial Prototype with a Minimal Number of Unidirectional Thrusters. In: 2020 International conference on unmanned aircraft systems (ICUAS), pp. 754–761 (2020). ISSN: 2575–7296
13. Brescianini, D., D'Andrea, R.: Design, modeling and control of an omni-directional aerial vehicle. In: 2016 IEEE International conference on robotics and automation (ICRA), pp. 3261–3266(2016)
14. Park, S., Her, J., Kim, J., Lee, D.: Design, modeling and control of omni-directional aerial robot. In: 2016 IEEE/RSJ International conference on intelligent robots and systems (IROS), pp. 1570–1575 (2016) ISSN: 2153–0866
15. Park, Sangyul, Lee, Jeongseob, Ahn, Joonmo, Kim, Myungsin, Her, Jongbeom, Yang, Gi-Hun., Lee, Dongjun: ODAR: Aerial Manipulation Platform Enabling Omnidirectional Wrench Generation. *IEEE/ASME Trans. Mechatron.* **23**(4), 1907–1918 (August 2018)
16. Tognon, M., Franchi, A.: Omnidirectional Aerial Vehicles With Unidirectional Thrusters: Theory, Optimal Design, and Control. *IEEE Robot. Autom. Lett.* **3**(3):2277–2282
17. Ryll, M., Bühlhoff, H.H., Giordano, P.R.: A Novel Overactuated Quadrotor Unmanned Aerial Vehicle: Modeling, Control, and Experimental Validation. *IEEE Transactions on Control Systems Technology*, **23**(2), 540–556, (2015)
18. Kamel, M., Verling, S., Elkhatib, O., Sprecher, C., Wulkop, P., Taylor, Z., Siegwart, R., Giltschenski, I.: The Voliro Omnidirectional Hexacopter: An Agile and Maneuverable Tilttable-Rotor Aerial Vehicle. *IEEE Robotics & Automation Magazine* **25**(4), 34–44 (2018)
19. Allenspach, M., Bodie, K., Brunner, M., Rinsoz, L., Taylor, Z., Kamel, M., Siegwart, R., Nieto, J.: Design and optimal control of a tiltrotor micro-aerial vehicle for efficient omnidirectional flight. *Int. J. Robot. Res.* **39**(10–11), 1305–1325 (2020). Publisher: SAGE Publications Ltd STM
20. Sal, F.: Simultaneous swept anhedral helicopter blade tip shape and control-system design. *Aircr. Eng. Aerosp. Technol.* **95**(1), 101–112 (2022)
21. Ryll, M., Bicego, D., Franchi, A.: Modeling and control of FAST-Hex: A fully-actuated by synchronized-tilting hexarotor. In: 2016 IEEE/RSJ International conference on intelligent robots and systems (IROS), pp. 1689–1694 (2016). ISSN: 2153–0866
22. Ryll, M., Bicego, D., Giurato, M., Lovera, M., Franchi, A.: FAST-Hex-A Morphing Hexarotor: Design, Mechanical Implementation, Control and Experimental Validation. *IEEE/ASME Trans. Mechatronics* **27**(3):1244–1255 (2022). Conference Name: IEEE/ASME Transactions on Mechatronics
23. Bicego, D., Mazzetto, J., Carli, R., Farina, M., Franchi, A.: Non-linear model predictive control with enhanced actuator model for multi-rotor aerial vehicles with generic designs. *J. Intell. & Robot. Syst.* **100**, 1213–1247 (2020)
24. Brescianini, D., D'Andrea, R.: An omni-directional multirotor vehicle. *Mechatronics*, **55**, 76–93 (2018)
25. Michieletto, G., Ryll, M., Franchi, A.: Fundamental actuation properties of multirotors: Force-moment decoupling and fail-safe robustness. *IEEE Trans. Rob.* **34**(3), 702–715 (2018)
26. Lee, T., Leok, M., McClamroch, N.H.: Geometric tracking control of a quadrotor UAV on SE(3). In: 49th IEEE Conference on decision and control, IEEE, pp. 5420–5425 (2010)
27. Lin, F.-J., Lin, C.-H.: On-line gain-tuning ip controller using rfnn. *IEEE Trans. Aerosp. Electron. Syst.* **37**(2), 655–670 (2001)
28. El Hakim, A., Hindersah, H., Rijanto, E.: Application of reinforcement learning on self-tuning pid controller for soccer robot multi-agent system. In: 2013 Joint international conference on rural information & communication technology and electric-vehicle technology, IEEE, pp. 1–6 (2013)

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

**Youssef Aboudorra** received his B.Sc. degree in Mechatronics Engineering in 2014 from the German University in Cairo (GUC), Egypt. In 2019, he received his M.Sc. degree (Laurea Magistrale) in Control Engineering from Sapienza University of Rome, Italy. He did his master thesis as a visiting student at the RIS team of LAAS-CNRS, Toulouse, France. Since 2020 he has been a Ph.D. Candidate at the Robotics and Mechatronics group (RaM) of the University of Twente, Enschede, The Netherlands. His research interests include: aerial robotics, multi-rotor aerial vehicles (MRAV, UAVs), modelling and control of robotic systems.

**Chiara Gabellieri** is a tenured Assistant Professor at the Robotics and Mechatronics group of the University of Twente, Enschede, The Netherlands. She was a postdoc in the same group from 2021 to 2023. She is a grant recipient of the Marie Skłodowska-Curie postdoctoral fellowship for the Flyflic project and a work package leader in the Horizon Europe coordination and support action AeroSTREAM. She is an Associate Editor for IEEE RA-L. Her research interests include environmental robotics, aerial robotics, and robotic manipulation. Chiara received her Ph.D. in Information Engineering in 2021 from the University of Pisa and, from the same institution, her MSc. in Robotics and Automation Engineering with honors in 2017 and her BSc. in Bioengineering in 2014. She was an intern at LAAS-CNRS in Toulouse, France, from November 2017 to May 2018 and a visiting Ph.D. student at the German Aerospace Center (DLR) in Oberpfaffenhofen from November 2019 to May 2020.

**Ralph Brantjes** is a student in the MSc program of Electrical Engineering at the University of Twente, Enschede, the Netherlands. He is doing his MSc thesis on omnidirectional aerial multirotor vehicles at the Robotics and Mechatronics (RaM) group of the same institution. He obtained his BSc degree in Advanced Technology from the University of Twente.

**Quentin Sablé** obtained his graduate degree in mechanical engineering in 2017 from the National Institute of Applied Science (INSA) in Strasbourg, France. From 2017 to 2020, he worked as a research engineer at the RIS team of LAAS-CNRS, Toulouse, France. Since 2020, he has been a research engineer in the Robotics and Mechatronics group (RaM) of the University of Twente, Enschede, The Netherlands. His research interests include mechanical design, mechatronics, aerial manipulation, and multi-rotor aerial vehicles (MRAV, UAVs).

**Antonio Franchi** (F'23–SM'16–M'07) received the Ph.D. degree in system engineering from the Sapienza University of Rome, Rome, Italy in 2010, and the HDR degree in Science, from the National Polytechnic Institute of Toulouse in 2016. From 2014 to 2019 he was a tenured CNRS researcher at LAAS-CNRS, Toulouse, France. From 2019 to 2021 he was an Associate Professor and then since 2022 he has been a Full Professor in aerial robotics control, both in the Robotics and Mechatronics group at the University of Twente, Enschede, The Netherlands. Since 2023 he is also a Full Professor in the Department of Computer, Control and Management Engineering at Sapienza University of Rome. He co-authored more than 160 publications in peer-reviewed international journals, books, and conferences on design, control, and estimation for robotic systems applied to multi-robot systems and aerial robots.