



**HAL**  
open science

# Omnidirectional Aerial Vehicles with Unidirectional Thrusters: Analysis, Optimal Design, and Motion Control

Marco Tognon, Antonio Franchi

► **To cite this version:**

Marco Tognon, Antonio Franchi. Omnidirectional Aerial Vehicles with Unidirectional Thrusters: Analysis, Optimal Design, and Motion Control. IEEE Robotics and Automation Letters, 2018, 3 (3), pp.2277-2282. <10.1109/LRA.2018.2802544>. <hal-01704054>

**HAL Id: hal-01704054**

**<https://laas.hal.science/hal-01704054v1>**

Submitted on 8 Feb 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



HAL Authorization

# Omnidirectional Aerial Vehicles with Unidirectional Thrusters: Analysis, Optimal Design, and Motion Control

Marco Tognon<sup>1</sup> and Antonio Franchi<sup>1</sup>

**Abstract**—This paper presents a theoretical study on omnidirectional aerial vehicles with body-frame fixed unidirectional thrusters. Omniplus multi-rotor designs are defined as the ones that allow to exert a total wrench in any direction using positive-only lift force and drag moment (i.e., positive rotational speed) for each rotor blade. Algebraic conditions for a design to be omniplus are derived, a simple necessary condition being the fact that at least seven propellers have to be used. An energy optimal design strategy is then defined as the one minimizing the maximum norm of the input set needed to span a certain wrench ellipsoid for the adopted input allocation strategy. Two corresponding major design criteria are then introduced: firstly, a minimum allocation-matrix condition number aims at an equal sharing of the effort needed to generate wrenches in any direction; secondly, imposing a balanced design guarantees an equal sharing of the extra effort needed to keep the input in the non-negative orthant. We propose a numerical algorithm to solve such optimal design problem and a control algorithm to control any omnidirectional platform. The work is concluded with informative simulation results in non-ideal conditions.

## I. INTRODUCTION

Aerial vehicles have been thoroughly studied and applied in several fields and for several tasks, from simple remote sensing to the more challenging physical interaction with the environment and humans. The latter have been firstly targeted using *unidirectional-thrust vehicles* actuated by multiple collinear rotors and endowed with cables [1], rigid tools [2], [3] or more complex robotic arms [4]–[6]. These vehicles are energy efficient but underactuated because of the unidirectionality of the total thrust in the body frame. Therefore i) the vehicle orientation is coupled with its translational motion, and ii) the system cannot *instantaneously* react to forces with any direction. Recent solutions to these issues consist in using *multidirectional-thrust vehicles* that can generate a force in multiple directions and can control both position and orientation independently. Examples are the platforms with tilted *unidirectional-thrust rotors* (i.e., propellers generating lift in only one direction), see, e.g., [7] and [8]. However, in these platforms the set of feasible forces does not span any direction in  $\mathbb{R}^3$ .

A special case is made by *omnidirectional-thrust vehicles*, that can produce a force in any direction in the body frame. This sub-class of vehicles is the most preferable, especially for physical interaction, because it can be oriented in any direction and can compensate/exert any force independently,

thus allowing applications that are impossible with other platforms, including safe human interaction, 360° aerial photography, etc.

In [9] and [10] two omnidirectional-thrust vehicles are proposed with 6 and 8 tilted *bidirectional-thrust rotors*, respectively. Such rotors are able to invert the direction of the lift force by inverting either the motor rotation or the propeller angle of attack. However such rotors have several issues: i) scarceness of reversible Electronic Speed Controllers (ESC) for brushless motors, ii) lower energetic efficiency compared to unidirectional rotors, iii) lower controllability of the exerted force at low speeds, and iv) extra mechanical complexity and increased weight and thus energy consumption (in case of variable pitch propellers). A solution to obtain an omnidirectional-thrust vehicle using instead unidirectional-thrust rotors is to actively tilt the whole propeller groups [11]–[13]. This also requires extra actuation and weight, and cannot in general guarantee instantaneous force exertion because of the non-negligible time the propellers need to re-orient themselves.

At the best of our knowledge, there are no works thoroughly investigating *if and how* it is instead possible to obtain *omnidirectional-thrust vehicles* with *fixed* (non-tiling) and *uni-directional* thrusters, a solution that would overcome all the problems of the aforementioned solutions. An attempt can be found in [14], where an ad-hoc optimization for an hexarotor is performed using an additional thruster whose position and orientation depend on the other six. The method cannot be easily extended to generic multirotor platforms, and the general theoretical problem still remains mostly open.

Instead, in this work we provide the fundamental definitions, properties, and conditions needed to rigorously address the problem in the general case of  $n$  propellers having any arrangement. For example, it turns out that an omnidirectional-thrust vehicle needs to have at least 7 fixedly attached unidirectional-thrust rotors. We propose an algorithm computing the best (fixed) directions of the  $n \geq 7$  propellers that make the vehicle omnidirectional-thrust and minimize the range of required control inputs to hover in any orientation. Finally, we propose a full-pose controller ensuring the input unidirectionality.

## II. MULTIROTOR MODEL

We start by defining an inertial world frame  $\mathcal{F}_W = \{O_W, \mathbf{x}_W, \mathbf{y}_W, \mathbf{z}_W\}$  where  $O_W$  is its origin, placed arbitrarily, and  $(\mathbf{x}_W, \mathbf{y}_W, \mathbf{z}_W)$  are the orthogonal unit vectors. We consider  $\mathbf{z}_W$  parallel and opposite to the gravity vector. Then we

<sup>1</sup>LAAS-CNRS, Université de Toulouse, CNRS, Toulouse, France, [mtognon@laas.fr](mailto:mtognon@laas.fr), [antonio.franchi@laas.fr](mailto:antonio.franchi@laas.fr)

This work has been funded by the European Union's Horizon 2020 research and innovation programme under grant agreement No 644271 AEROARMS.

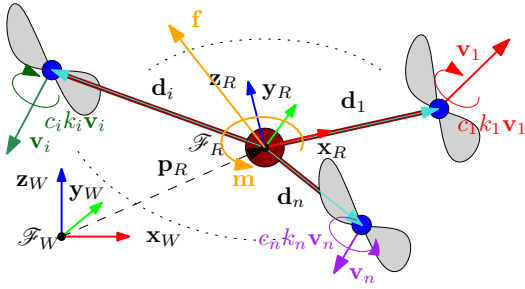


Fig. 1: Schematic representation of a multirotor and its main quantities. Only three of the  $n$  propellers are shown.

define the body frame  $\mathcal{F}_R = \{O_R, \mathbf{x}_R, \mathbf{y}_R, \mathbf{z}_R\}$  rigidly attached to the vehicle and centered in  $O_R$ , the vehicle center of mass (CoM). Position of  $O_R$  and orientation of  $\mathcal{F}_R$  w.r.t.  $\mathcal{F}_W$  are described by the vector  $\mathbf{p}_R \in \mathbb{R}^3$  and the rotation matrix  $\mathbf{R}_R \in SO(3)$ , respectively. Then we define by the vector  $\mathbf{v}_R \in \mathbb{R}^3$  the translational velocity of  $O_R$  expressed in  $\mathcal{F}_W$ , and by  $\boldsymbol{\omega}_R \in \mathbb{R}^3$  the angular velocity of  $\mathcal{F}_R$  w.r.t.  $\mathcal{F}_W$  and expressed in  $\mathcal{F}_R$ . The generic vehicle is depicted in Fig. 1.

The vehicle is modeled as a rigid body with mass  $m_R \in \mathbb{R}_{>0}$  and moment of inertia about  $O_R$ , defined w.r.t.  $\mathcal{F}_R$ , described by the positive definite matrix  $\mathbf{J}_R \in \mathbb{R}^{3 \times 3}$ . The dynamics of the system is computed applying the Newton-Euler equations, thus obtaining  $\dot{\mathbf{p}}_R = \mathbf{v}_R$ ,  $\dot{\mathbf{R}}_R = \mathbf{R}_R \boldsymbol{\Omega}_R$ , and

$$\underbrace{\begin{bmatrix} m_R \mathbf{I}_3 & \mathbf{0} \\ \mathbf{0} & \mathbf{J}_R \end{bmatrix}}_{\mathbf{M}_R} \underbrace{\begin{bmatrix} \dot{\mathbf{v}}_R \\ \dot{\boldsymbol{\omega}}_R \end{bmatrix}}_{\mathbf{a}} = \underbrace{\begin{bmatrix} -g\mathbf{e}_3 \\ -\mathbf{J}_R \boldsymbol{\omega}_R \times \boldsymbol{\omega}_R \end{bmatrix}}_{\mathbf{b}_R} + \underbrace{\begin{bmatrix} \mathbf{R}_R & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_3 \end{bmatrix}}_{\mathbf{G}_R} \underbrace{\begin{bmatrix} \mathbf{f} \\ \mathbf{m} \end{bmatrix}}_{\mathbf{f}},$$

where  $\mathbf{e}_3 = [0 \ 0 \ 1]^\top$ ,  $\boldsymbol{\Omega}_R = \mathbf{S}(\boldsymbol{\omega}_R)$  is the skew symmetric matrix relative to  $\boldsymbol{\omega}_R$ ,  $\mathbf{f} \in \mathbb{R}^3$  and  $\mathbf{m} \in \mathbb{R}^3$  are the controllable total input force and torque expressed in  $\mathcal{F}_R$ , respectively.

Considering a multirotor with  $n$  rotors, each of them produces a lift force and a moment due to the drag force [15]. All together they generate the total force (or thrust) and moment,  $\mathbf{f}$  and  $\mathbf{m}$ , respectively, expressed as:

$$\mathbf{w} = [\mathbf{f}^\top \ \mathbf{m}^\top]^\top = [\mathbf{F}_1^\top \ \mathbf{F}_2^\top]^\top [u_1 \ \dots \ u_n]^\top = \mathbf{F}\mathbf{u}. \quad (1)$$

The matrixes  $\mathbf{F} \in \mathbb{R}^{6 \times n}$ ,  $\mathbf{F}_1 \in \mathbb{R}^{3 \times n}$ , and  $\mathbf{F}_2 \in \mathbb{R}^{3 \times n}$  are called the *full allocation matrix*, the *force allocation matrix* and the *moment allocation matrix*, respectively. The control  $u_i \in \mathbb{R}$  is typically equal to  $w_i/|w_i|$ , where  $w_i \in \mathbb{R}$  is the  $i$ -th propeller rotational speed.  $\mathbf{F}_1$  and  $\mathbf{F}_2$  have the following structure

$$\mathbf{F}_1 = [\mathbf{v}_1 \ \dots \ \mathbf{v}_n], \quad (2)$$

$$\mathbf{F}_2 = [\mathbf{d}_1 \times \mathbf{v}_1 \ \dots \ \mathbf{d}_n \times \mathbf{v}_n] + [c_1 k_1 \mathbf{v}_1 \ \dots \ c_n k_n \mathbf{v}_n], \quad (3)$$

where i)  $\mathbf{v}_i \in \mathbb{R}^3$  are the coordinates, in  $\mathcal{F}_R$ , of the lift force generated by the  $i$ -th propeller when  $u_i = 1$ . In this formulation the aerodynamic coefficient that maps propeller speed into thrust intensity, typically called lift factor  $c_f$ , is  $c_{fi} = \|\mathbf{v}_i\| = v_i$ ; ii)  $\mathbf{d}_i$  is the position of the center of the  $i$ -th propeller in body frame; iii)  $c_i = -1$  ( $c_i = 1$ ) if the  $i$ -th propeller angular velocity vector has the same direction of  $\mathbf{v}_i$  ( $-\mathbf{v}_i$ ) when  $u_i > 0$ , i.e., the propeller spins CCW (CW)

when watched from its top; iv)  $k_i \in \mathbb{R}$  is the constant ratio between the  $i$ -th propeller lift force and the drag moment, typically denoted with  $c_\tau/c_f$  in the literature. In conclusion, we recall a well known fact.

**Fact 1** (translation invariance).  $\mathbf{F}$  does not change if  $\mathbf{d}_i$  is replaced with  $\mathbf{d}_i + \lambda_i \mathbf{v}_i$  for any  $i = 1, \dots, n$  and  $\lambda_1, \dots, \lambda_n \in \mathbb{R}$ .

### III. OMNIPLUS DESIGNS

We introduce now the basic concept of multirotor design. Let us first define  $\mathbf{c} = [c_1 \ \dots \ c_n]^\top$  and  $\mathbf{k} = [k_1 \ \dots \ k_n]^\top$ .

**Definition 1.** A multirotor *design* is a tuple  $\mathcal{D} = (n, \mathbf{c}, \mathbf{k}, \mathbf{d}_1, \dots, \mathbf{d}_n, \mathbf{v}_1, \dots, \mathbf{v}_n)$ , which describes the number of propellers  $n$ , their aerodynamic characteristics, locations and orientations w.r.t.  $\mathcal{F}_R$ . We call the tuples  $(\mathbf{v}_1, \dots, \mathbf{v}_n)$  and  $(n, \mathbf{c}, \mathbf{k}, \mathbf{d}_1, \dots, \mathbf{d}_n)$  the *vectoring part* and the *etero-vectoring part* of  $\mathcal{D}$ , respectively.

We denote with  $\mathbf{1}$  the column vector with all ones. Its size is understood from the context. Given two vectors  $\mathbf{x}$  and  $\mathbf{y}$ , the notations  $\mathbf{x} \geq \mathbf{y}$ ,  $\mathbf{x} > \mathbf{y}$  have to be intended component-wise.

**Definition 2.** Given  $\underline{u} \geq 0$ , a multirotor design  $\mathcal{D}$  is  *$\underline{u}$ -omniplus* ( $\underline{u}O_+$ ) if the corresponding full allocation matrix  $\mathbf{F}$ , satisfies

$$\forall \mathbf{w} \in \mathbb{R}^6 \quad \exists \mathbf{u} \geq \underline{u}\mathbf{1} \quad \text{s.t.} \quad \mathbf{F}\mathbf{u} = \mathbf{w}. \quad (4)$$

A design that is  $\underline{u}O_+$  for any  $\underline{u} \in \mathbb{R}_{\geq 0}$  is said *omniplus* ( $O_+$ ). Considering  $\underline{u} \geq 0$  accounts for the possible presence of a minimum rotational speed constraint for the propellers.

**Proposition 1** (Theorem 1 in [16] extended to the  $\mathbb{R}^6$  case). *The following two conditions are equivalent*

$$\forall \mathbf{w} \in \mathbb{R}^6 \quad \exists \mathbf{u} \geq \mathbf{0} \quad \text{s.t.} \quad \mathbf{F}\mathbf{u} = \mathbf{w}, \quad (5)$$

$$\text{rank}(\mathbf{F}) = 6 \quad \text{and} \quad \exists \mathbf{b} > \mathbf{0} \quad \text{s.t.} \quad \mathbf{F}\mathbf{b} = \mathbf{0}. \quad (6)$$

*Proof.* The same as in [16] but replacing 3-dimensional vectors with 6-dimensional vectors.  $\square$

**Corollary 1.** *Condition (4) is equivalent to (5) and (6), and as a consequence, any  $\underline{u}O_+$  design is also  $O_+$ .*

*Proof.* Sufficiency is trivial. For necessity, consider a  $\mathbf{u}$  satisfying (5). Thanks to (6) consider  $\mathbf{u}' = \mathbf{u} + \underline{u}\mathbf{b}/\|\mathbf{b}\|$  which satisfies both  $\mathbf{u}' \geq \underline{u}\mathbf{1}$  and  $\mathbf{F}\mathbf{u}' = \mathbf{w}$  and therefore fulfills (4).  $\square$

**Corollary 2.** *For any design that is  $O_+$  it must be  $n \geq 7$ .*

*Proof.* We have that it must be  $\text{rank}(\mathbf{F}) = 6$  and at the same time  $\mathbf{0} \neq \mathbf{b} \in \text{null}(\mathbf{F})$ , therefore it must be at least  $n = 7$ .  $\square$

*Remark.* It is interesting to note that Prop. 1 and Corol. 2 find their counterpart in the literature of frictionless contact grasping (see [17] and the references therein).

**Proposition 2.** *Let be given an  $O_+$  design  $\mathcal{D}$ . Any design  $\mathcal{D}'$  with the same eterovectoring part of  $\mathcal{D}$  and a new vectoring part  $(\alpha \mathbf{v}_1, \alpha \mathbf{v}_2, \dots, \alpha \mathbf{v}_n)$ , where  $\alpha \neq 0$ , is also  $O_+$ .*

*Proof.* Denote with  $\mathbf{F}$  and  $\mathbf{F}'$  the full allocation matrixes of  $\mathcal{D}$  and  $\mathcal{D}'$ , respectively. We have that  $\mathbf{F}' = \alpha\mathbf{F}$ , therefore the properties (6) are valid also for  $\mathbf{F}'$  as long as  $\alpha \neq 0$ .  $\square$

In the following we denote with  $\mathbf{I}_j$  the  $j$ -by- $j$  identity matrix. We also use the following notation  $\mathbf{v} = [\mathbf{v}_1^\top \cdots \mathbf{v}_n^\top]^\top$  and  $\mathbf{d} = [\mathbf{d}_1^\top \cdots \mathbf{d}_n^\top]^\top$ . We can then rewrite (2) and (3) as

$$\mathbf{F}_1 = [\mathbf{I}_3 \mathbf{v}_1 \cdots \mathbf{I}_3 \mathbf{v}_n] \quad (7)$$

$$\mathbf{F}_2 = [(\mathbf{S}(\mathbf{d}_1) + c_{1k_1} \mathbf{I}_3) \mathbf{v}_1 \cdots (\mathbf{S}(\mathbf{d}_n) + c_{nk_n} \mathbf{I}_3) \mathbf{v}_n]. \quad (8)$$

**Proposition 3.** A multirotor design is  $O_+$  if and only if

$$\text{rank}(\mathbf{F}) = 6, \quad (9)$$

and  $\exists \mathbf{b} = [b_1 \cdots b_n]^\top > \mathbf{0}$  s.t.

$$\underbrace{\begin{bmatrix} b_1 \mathbf{I}_3 & \cdots & b_n \mathbf{I}_3 \\ b_1 (\mathbf{S}(\mathbf{d}_1) + c_{1k_1} \mathbf{I}_3) & \cdots & b_n (\mathbf{S}(\mathbf{d}_n) + c_{nk_n} \mathbf{I}_3) \end{bmatrix}}_{\mathbf{A}(\mathbf{c}, \mathbf{k}, \mathbf{d}_1, \dots, \mathbf{d}_n, \mathbf{b})} \mathbf{v} = \mathbf{0} \quad (10)$$

*Proof.* The condition (9) is the first part of (6). The condition (10) is obtained from the second part of (6) by using (7) and (8) and imposing  $\mathbf{F}\mathbf{b} = \mathbf{0}$ .  $\square$

#### IV. ALLOCATION STRATEGIES FOR $O_+$ DESIGNS

In this section we introduce two different input allocation strategies for  $O_+$  designs. The first one, defined as the solution of Prob. 1, is the optimal one but is hard to be exploited for an analytically sound optimization of the design (unless a completely numerical algorithm is used). The second one, defined as the solution of Prob. 2, is suboptimal, but is amenable of a clear geometrical interpretation which can be used for an analytically sound design optimization.

**Problem 1.** Consider a given  $O_+$  design with full allocation matrix  $\mathbf{F}$ . Given a desired  $\mathbf{w} \in \mathbb{R}^6$ , with  $\mathbf{w} \neq \mathbf{0}$ , find the input  $\mathbf{u} \in \mathbb{R}^n$  s.t.  $\mathbf{F}\mathbf{u} = \mathbf{w}$ ,  $\mathbf{u} \geq \underline{u}\mathbf{1}$ , and  $\|\mathbf{u}\|$  is minimized.

The solution to Prob. 1 without the constraint  $\mathbf{u} \geq \underline{u}\mathbf{1}$  is  $\mathbf{u}^* = \mathbf{F}^\dagger \mathbf{w}$ , where  $\mathbf{F}^\dagger$  is the Moore-Penrose pseudo-inverse of  $\mathbf{F}$ . However, for a  $O_+$  design it is never  $\mathbf{u}^* \geq \mathbf{0}$ , a part from the trivial case  $\mathbf{w} = \mathbf{0}$ , as stated next.

**Proposition 4.** Let  $\mathbf{F}$  be the full allocation matrix of an  $O_+$  design and  $\mathbf{F}^\dagger$  its Moore-Penrose pseudo-inverse. For any desired wrench  $\mathbf{w} \neq \mathbf{0}$ , the minimum norm solution of  $\mathbf{F}\mathbf{u} = \mathbf{w}$ , i.e.,  $\mathbf{u}^* = \mathbf{F}^\dagger \mathbf{w}$ , has always at least a negative entry, hence it is never a solution to Problem 1.

*Proof.* We have that  $\text{im}(\mathbf{F}^\dagger) = \text{im}(\mathbf{F}^\top)$  [18] and that  $\exists \mathbf{b} \in \text{null}(\mathbf{F})$  s.t.,  $\mathbf{b} > \mathbf{0}$ . Since  $\text{im}(\mathbf{F}^\top)$  is orthogonal to  $\text{null}(\mathbf{F})$ , we have that  $\mathbf{b}^\top \mathbf{u}^* = \mathbf{0}$ . If  $\mathbf{w} \neq \mathbf{0}$  then  $\mathbf{u}^* \neq \mathbf{0}$ , and since  $\mathbf{b} > \mathbf{0}$ ,  $\mathbf{u}^*$  must have at least a negative entry for  $\mathbf{b}^\top \mathbf{u}^* = \mathbf{0}$  to hold.  $\square$

Prop. 4 implies that the solution of Prob. 1 is always of the form  $\mathbf{u} = \mathbf{u}^* + \mathbf{y}$  with  $\mathbf{y} \in \text{null}(\mathbf{F})$ . In particular, exploiting the fact that  $\mathbf{u}^* \perp \mathbf{y}$ , the solution structure is  $\mathbf{u}^{**} = \mathbf{u}^* + \mathbf{y}^*$ , where

$$\mathbf{y}^* = \arg \min_{\substack{\mathbf{y} \geq \underline{u}\mathbf{1} - \mathbf{u}^* \\ \mathbf{F}\mathbf{y} = \mathbf{0}}} \|\mathbf{y}\|, \quad (11)$$

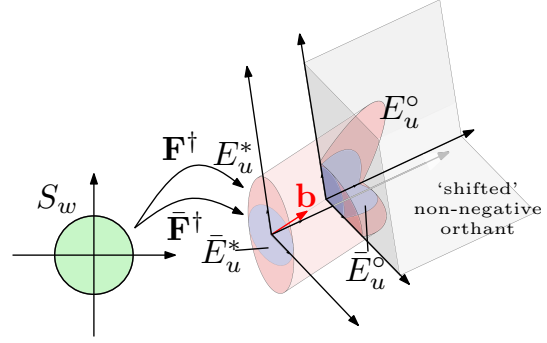


Fig. 2: Simplified representation, in a reduced space, of  $S_w$  being mapped by  $\mathbf{F}^\dagger$  into  $E_u^*$ , in turn projected, according to (13), on the facets of the shifted non-negative orthant along  $\mathbf{b}$  producing  $E_u^{\circ}$ .  $\mathbf{F}$  and  $\bar{\mathbf{F}}$  correspond to a non-optimized and a optimized design, respectively (see Sec. V for the details). Reducing  $E_u^*$  (represented in pink) to a sphere  $\bar{E}_u^*$  (represented in blue), by the minimization of the condition number of  $\mathbf{F}$ , the maximum norm of the projection on the shifted positive orthant is also minimized.

which can be efficiently solved with any constrained QP solver.

To provide a geometrical understanding of the structure of the solutions of the input allocation problems, let us consider an ellipsoid that may, e.g., represent the set of desired attainable wrenches  $S_w = \{\mathbf{w} \in \mathbb{R}^6 \mid \mathbf{w}^\top \Sigma \mathbf{w} \leq 1\} \subset \mathbb{R}^6$ , where  $\Sigma \in \mathbb{R}^{6 \times 6}$  is a positive definite matrix. The ellipsoid  $S_w$  is mapped by  $\mathbf{F}^\dagger$  to the set  $E_u^* = \{\mathbf{u} \in \mathbb{R}^n \mid \mathbf{u} = \mathbf{F}^\dagger \mathbf{w}, \forall \mathbf{w} \in S_w\} \subset \mathbb{R}^n$  – an idealized representation from  $\mathbb{R}^2$  to  $\mathbb{R}^3$ , and with  $\Sigma = \mathbf{I}$  is shown in Fig. 2. The set  $E_u^*$  is a 6-dimensional ellipsoid of  $\mathbb{R}^n$ , contained in the subspace  $\text{im}(\mathbf{F}^\top)$ , whose shape is defined by the singular value decomposition of  $\mathbf{F}$  and  $\Sigma$ . There is a one to one correspondence between each  $\mathbf{w} \in S_w$  and each  $\mathbf{u} \in E_u^*$ . However, according to Prop. 4, any vector  $\mathbf{u} \in E_u^*$  has always at least a negative entry (a part from  $\mathbf{u} = \mathbf{0}$ ). In order to satisfy the constraint  $\mathbf{u} \geq \underline{u}\mathbf{1}$  one has to project each point  $\mathbf{u}^*$  of  $E_u^*$  onto one of the external facets of the shifted non-negative orthant denoted from now on with  $\mathbb{R}_{\geq \underline{u}\mathbf{1}}^n$ . The projection must be done by adding to  $\mathbf{u}^*$  a perpendicular vector that belongs to  $\text{null}(\mathbf{F})$  and has minimum norm, i.e., obtaining  $\mathbf{y}^*$  by solving (11). By doing so for all the points in  $E_u^*$  we obtain the set of solutions of Prob. 1 defined as  $E_u^{**} = \{\mathbf{u}^* + \mathbf{y}^* \in \mathbb{R}^n \mid \mathbf{u}^* \in E_u^*, \text{ and } \mathbf{y}^* \text{ solves (11)}\}$ .

Denote with  $\mathbb{R}_{++}^n$  the positive orthant of  $\mathbb{R}^n$  and let us consider the following alternative Problem.

**Problem 2.** Consider a given  $O_+$  design with full allocation matrix  $\mathbf{F}$  and let be given a constant vector  $\mathbf{b} \in \text{null}(\mathbf{F}) \cap \mathbb{R}_{++}^n$ . For any desired  $\mathbf{w} \in \mathbb{R}^6$ , with  $\mathbf{w} \neq \mathbf{0}$ , find the input  $\mathbf{u} = \mathbf{u}^* + \lambda \mathbf{b} \in \mathbb{R}^n$ , where  $\lambda > 0$ , s.t.  $\mathbf{u} \geq \underline{u}\mathbf{1}$ , and  $\|\mathbf{u}\|$  is minimized.

Problem 2 represents a restriction of Prob. 1 in the sense that a solution of Problem 2 satisfies the constraints of Prob. 1 but is in general sub-optimal, since the solutions are searched only of the form  $\mathbf{u} = \mathbf{u}^* + \lambda \mathbf{b}$  where  $\mathbf{b}$  is a fixed

vector in  $\text{null}(\mathbf{F}) \cap \mathbb{R}_{++}^n$  (which always exists, thanks to (6)), and  $\lambda > 0$  is a large enough positive scalar that ensures that each entry of  $\mathbf{u}$  is not smaller than  $\underline{u}$ . Since it is structurally  $\mathbf{u}^* \perp \lambda \mathbf{b}$ , in order to minimize the norm of  $\mathbf{u}^* + \lambda \mathbf{b}$ , one has to choose

$$\lambda = \lambda^\circ(\mathbf{u}^*, \mathbf{b}, \underline{u}) = \min_{\mu | \mathbf{u}^* + \mu \mathbf{b} \geq \underline{u}\mathbf{1}} \mu, \quad (12)$$

thus obtaining

$$\mathbf{u}^\circ := \mathbf{u}^* + \lambda^\circ \mathbf{b}. \quad (13)$$

By doing so we are projecting the set  $E_u^*$  on the facets of  $\mathbb{R}_{\geq \underline{u}\mathbf{1}}^n$  following the constant direction defined by  $\mathbf{b}$ . We denote this projection with  $E_u^\circ = \{\mathbf{u}^* + \lambda^\circ \mathbf{b} | \mathbf{u}^* \in E_u^* \text{ and } \lambda^\circ \text{ solves (12)}\}$ . The geometric relations between  $S_w$ ,  $E_u^*$  and  $E_u^\circ$  are shown in Fig. 2 for an idealized, smaller, dimensional space.

## V. OPTIMAL OMNIPLUS DESIGN

Our concept of optimal design follows the chosen allocation strategy. W.l.o.g., in the following we assume that all the propellers have a common lift factor, i.e.,  $\|\mathbf{v}_i\| = v$ ,  $\forall i = 1 \dots n$ .

**Definition 3.** A design is optimal if  $\max_{\mathbf{u} \in E} \|\mathbf{u}\|$  is minimized, where  $E$  is the set of inputs that the given allocation strategy maps to  $S_w$  (e.g.,  $E = E_u^{**}$  for  $\mathbf{u}^{**}$  or  $E = E_u^\circ$  for  $\mathbf{u}^\circ$ ).

Even if minimization (11) can be efficiently solved with any constrained QP solver, there is, at the best of our knowledge, no analytical form to express  $\mathbf{y}^*$ . Furthermore,  $\mathbf{y}^*$  may in general change (in both norm and direction) depending on the particular  $\mathbf{u}^* \in E_u^*$ . This makes hard to understand how is the shape of  $E_u^{**}$  and, especially, how the value of  $\max_{\mathbf{u} \in E_u^{**}} \|\mathbf{u}\|$  are influenced by the changes of the design parameters. Hence, developing an analytically sound design optimization around the first allocation policy is left as future investigation.

The second allocation strategy is instead amenable of a more clear geometrical interpretation that leads naturally to the definition of a design optimization problem. First of all let us assume that the etero-vectoring part of the design is given and that one has to optimize only the vectoring part. This is what happens in practice most of the time. Since  $\mathbf{u}^* \perp \mathbf{b}$  in (13), in order to make a design optimal according to Definition 3, it makes sense, first of all, to minimize the eccentricity of the set  $E_u^*$ , i.e., condition number of  $\Sigma^{-1}\mathbf{F}$ . Furthermore, in order to optimally project the set  $E_u^*$  onto the facets of  $\mathbb{R}_{\geq \underline{u}\mathbf{1}}^n$ , it is easy to be convinced that the best choice would be  $\mathbf{b} = \mathbf{1}$  in (13) if the design would allow it, i.e., if  $\mathbf{F}\mathbf{1} = \mathbf{0}$ . This leads to:

**Definition 4.** An  $O_+$  design is *balanced* if  $\mathbf{F}\mathbf{1} = \mathbf{0}$ .

For balanced  $O_+$  designs the  $n$  propellers *equally share* the *extra effort* needed to actively satisfy the constraint  $\mathbf{u} \geq \underline{u}\mathbf{1}$  in (13). In this way the risk of obtaining too large inputs, due to unbalanced sharing of the extra effort, is reduced. All these considerations lead to the following optimization problem.

**Problem 3.** Let be given an etero-vectoring part ( $n \geq 7, \mathbf{c}, \mathbf{k}, \mathbf{d}_1, \dots, \mathbf{d}_n$ ). Find a vectoring part  $(\mathbf{v}_1, \dots, \mathbf{v}_n)$  that solves

$$\min \text{cond}(\Sigma^{-1}\mathbf{F}) \quad (14)$$

subject to

$$\mathbf{v}^\top \mathbf{D}_1 \mathbf{v} = v, \dots, \mathbf{v}^\top \mathbf{D}_n \mathbf{v} = v \quad (15)$$

$$\text{rank}(\mathbf{F}(\mathbf{c}, \mathbf{k}, \mathbf{d}_1, \dots, \mathbf{d}_n, \mathbf{v})) = 6 \quad (16)$$

$$\mathbf{A}(\mathbf{c}, \mathbf{k}, \mathbf{d}_1, \dots, \mathbf{d}_n, \mathbf{1})\mathbf{v} = \mathbf{0}, \quad (17)$$

where  $\mathbf{D}_i = \text{diag}(\mathbf{D}_{i1}, \dots, \mathbf{D}_{in})$  is a  $3n$ -by- $3n$  diagonal matrix in which  $\mathbf{D}_{ij}$ , for  $j = 1 \dots n$ , are a 3-by-3 matrices such that  $\mathbf{D}_{ij} = \mathbf{0}$  for  $j \neq i$  and  $\mathbf{D}_{ii} = \mathbf{I}_3$ .

### A. On the Existence of Solutions

Determining which are the conditions on the etero-vectoring part that ensure the existence of a solution for Problem 3, and how to analytically compute a solution  $\mathbf{v}$ , are both still open questions which are left as future work. In the following we shall assume that a solution is computed in a numerical fashion. We empirically noticed that it is not practically hard to find numerical solutions for an etero-vectoring part whose parameters are chosen following the next common sense rules.

Firstly, the vectors  $\mathbf{d}_1, \dots, \mathbf{d}_n$  are chosen coplanar and in a star-shaped configuration, i.e., selecting any  $\mathbf{d}_1$  such that  $\mathbf{d}_1 \times \mathbf{e}_3 \neq \mathbf{0}$ , and then choosing  $\mathbf{d}_i = \mathbf{R}_z(2\pi(i-1)/n)\mathbf{d}_1$  for  $i = 2 \dots n$ , where  $\mathbf{R}_z(\theta)$  is the canonical rotation matrix about the  $z$ -axis of an angle  $\theta$ . The coplanar constraint does not restrict the generality of the results. One could use any other etero-vectoring part. However, any 3D configuration of  $\mathbf{d}$  can be reduced to a planar one. Indeed, once obtained the vectoring part, one can move the  $i$ -th thruster along the  $\mathbf{v}_i$  direction exploiting Fact.1. This feature might be also exploited to avoid collisions between propellers and the main frame. The constraint  $\|\mathbf{d}_i\| = \|\mathbf{d}_1\| \forall i = 2 \dots n$  is instead added for mechanical simplicity. Secondly, the vector  $\mathbf{c}$  showing a balanced set of  $-1$  and  $1$  entries, e.g.,  $c_i = (-1)^i$  for  $i = 1, \dots, n$ . Thirdly, it is imposed  $\|\mathbf{v}_i\| = v$  and  $k_i = k \forall i = 1 \dots n$ , since it is common to use the same propellers in the same multirotor. Based on our experience, the algorithm described next has always been able to find a solution to Problem 3 with any etero-vectoring part of the class defined by the three rules above.

### B. Algorithm

A simple but effective method to solve Prob. 3 is provided in Algorithm 1 and explained in the following.

First of all, `randomVectoring( $n, v$ )` generates the  $\mathbf{v}_i$ 's, for  $i = 1, \dots, n$  as  $\mathbf{v}_i = v\mathbf{n}_i$ , where  $\mathbf{n}_i \in S_2 = \{\mathbf{n} \in \mathbb{R}^3 \mid \|\mathbf{n}\| = 1\}$  are sampled randomly with a uniform probability. The resulting vectoring part fulfills only (15) among the constraints, and cannot be used as initial guess for a nonconvex numerical solver since constraint (17) is not yet satisfied. In order to find an initial guess that satisfies (17) we use an iterative algorithm which tries to find the solution to  $\min_{\mathbf{v}} \|\mathbf{A}\mathbf{v}\|^2$  subject

---

**Algorithm 1: Optimal Omniplus Design**


---

**Input:** etero-vectoring part,  $(n, \mathbf{c}, \mathbf{k}, \mathbf{d}_1, \dots, \mathbf{d}_n)$   
**Output:** design  $\mathcal{D}^* = (n, \mathbf{c}, \mathbf{k}, \mathbf{d}_1, \dots, \mathbf{d}_n, \mathbf{v}_1, \dots, \mathbf{v}_n)$

- 1:  $\mathcal{D}^* \leftarrow \emptyset$
- 2: **for**  $k = 1$  to  $N$  **do**
- 3:  $\mathbf{v}^0 \leftarrow \text{randomVectoring}(n, \mathbf{v})$
- 4: **for**  $j = 0$  to  $N_N - 1$  **do**
- 5:  $\mathbf{v}^{j+1} \leftarrow \mathbf{v}^j - \mathbf{A}^\dagger \mathbf{A} \mathbf{v}^j$  % Netwon-Raphson
- 6:  $\mathbf{v}^{j+1} \leftarrow \text{renormalization}(\mathbf{v}^{j+1}, \mathbf{v})$
- 7:  $\mathcal{D}^k \leftarrow \text{locOptimumDesign}(n, \mathbf{c}, \mathbf{k}, \mathbf{d}_1, \dots, \mathbf{d}_n, \mathbf{v}_1^{N_N}, \dots, \mathbf{v}_n^{N_N})$
- 8: **if**  $\mathcal{D}^* = \emptyset$  **or**  $0 < \text{condNumF}(\mathcal{D}^k) < \text{condNumF}(\mathcal{D}^*)$  **then**
- 9:  $\mathcal{D}^* \leftarrow \mathcal{D}^k$
- 10: **return**  $\mathcal{D}^*$

---

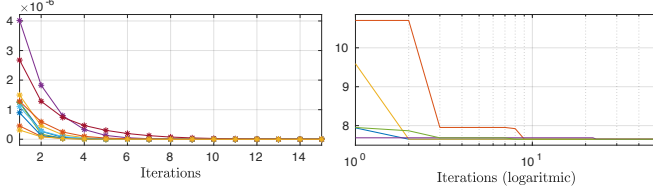


Fig. 3: Performances of the algorithm. Left: inner loop error for 10 different initial guesses. Right:  $\text{cond}(\mathbf{F})$  for 5 trials of the algorithm.

to (15) (lines 4-6). At each step of the iteration, the vectoring part is updated using Newton-Raphson update rule (line 5) and renormalized (line 6). A number  $N_N$  of iterations is executed, which guarantees to obtain an initial guess with  $\|\mathbf{A}\mathbf{v}\|^2$  that is practically zero. The obtained initial guess is provided to  $\text{optimumDesign}(n, \mathbf{c}, \mathbf{k}, \mathbf{d}_1, \dots, \mathbf{d}_n, \mathbf{v}_1^{N_N}, \dots, \mathbf{v}_n^{N_N})$  which applies an interior point (IP) algorithm to solve locally the full Prob. 3. This approach finds a local minima, and therefore the whole process is repeated  $N$  times from the random generation step in order to find the best of the several local minima found by each call of the IP method with different initial guesses.

To give an idea of the computation factors, in Fig. 3-left we show the convergence of  $\|\mathbf{A}\mathbf{v}\|$  to zero in the Newton-Raphson iterations for a meaningful case. On the right we instead show the convergence of the condition number of  $\mathbf{F}$  at each iteration of the outer loop. For the case in exam, the Newton-Raphson loop takes less than 15 iterations to converge to a valid initial guess, while the whole optimization algorithm needs around 40 iterations. The overall algorithm implemented in Matlab, for  $N_N = 40$  and  $N = 40$ , takes a mean time of 15 [s] to run on a standard laptop. The computation time was not a real problem here since the algorithm has to be run offline before starting the mechanical design of the vehicle.

As an example, in Fig. 4 we report one of the designs found by setting  $n = 7$  and running the previous optimization algorithm (see Sec. VII for more details). One can notice that there are some symmetries between rotors orientation. Indeed, considering motor 2 (the one with no symmetries) as the “front of the vehicle”, then thrusters (1,3) point ahead/interior/up, (4,7) point back/interior/up and (5,6) point back/interior/down. This fact could be exploited in future works to further reduce the condition number.

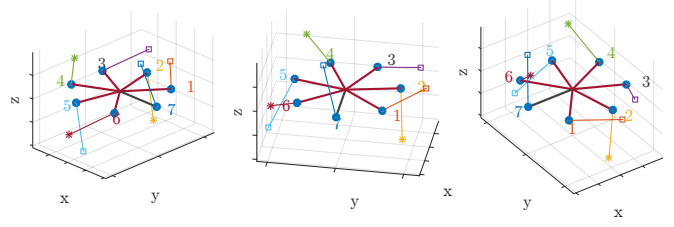


Fig. 4: Optimized omniplus design ( $\text{cond}(\mathbf{F}) = 3.59$ ) with 7 propellers. The red tick lines correspond to the vectors  $\mathbf{d}_1, \dots, \mathbf{d}_6$  arms all departing from  $O_R$ . The black one defines  $\mathbf{d}_7$ . The blue spheres correspond to the positions of the motors. The colored slim lines indicate the lift force direction of each propeller. The star and the square symbols indicate CCW and CW propellers, respectively.

## VI. CONTROL STRATEGY

Given desired position and orientation trajectories, i.e.,  $\mathbf{p}_R^d(t)$  and  $\mathbf{R}_R^d(t)$ , respectively, the control strategy of a platform with an omniplus design is rather straightforward. In fact, one has to first decide the force and moment vector  $\mathbf{w}^d$  to be applied to the body to steer the output along the desired trajectory. One can use a nonlinear model inversion combined with a Feed Forward plus a PID inner loop:

$$\mathbf{w}^d = \mathbf{G}_R^{-1} \left( \mathbf{M}_R \mathbf{a}^d + \mathbf{K}_D \dot{\mathbf{e}} + \mathbf{K}_P \mathbf{e} + \mathbf{K}_I \int_0^t \mathbf{e} d\tau \right), \quad (18)$$

where  $\mathbf{K}_D, \mathbf{K}_P, \mathbf{K}_I \in \mathbb{R}^{6 \times 6}$  are positive diagonal definite matrices,  $\dot{\mathbf{e}} = [(\mathbf{v}_R^d - \mathbf{v}_R)^\top (\mathbf{R}_R^d \boldsymbol{\omega}_R^d - \boldsymbol{\omega})^\top]^\top$ ,  $\mathbf{e} = [(\mathbf{p}_R^d - \mathbf{p}_R)^\top \mathbf{e}_{\mathbf{R}_R}^\top]^\top$ ,  $\mathbf{e}_{\mathbf{R}_R} = [1/2(\mathbf{R}_R^{d\top} \mathbf{R}_R - \mathbf{R}_R^\top \mathbf{R}_R^d)]^\wedge$ . The  $[\cdot]^\wedge$  is the un-skew operator. The  $\mathbf{w}^d$  is then implemented choosing  $\mathbf{u} = \mathbf{u}^\circ$ .

## VII. NUMERICAL SIMULATIONS

In this section we shall present the simulation results validating the algorithm to find an optimal omniplus design and the proposed controller. We chose to use the minimum number of propellers, i.e.,  $n = 7$ . Then we used an etero-vectoring part where  $\mathbf{d}_i = 0.4\mathbf{R}_z(2\pi(i-1)/n)[1 \ 0 \ 0]^\top$  and  $k_i = 0.0192$  [m] for  $i = 1, \dots, n$ . Furthermore, considering a standard motor-propeller with diameter 0.30 [m] available in the market, we have that  $\nu = 9.9 \cdot 10^{-4}$  [N/Hz<sup>2</sup>] and  $\underline{u} = 16^2$  [Hz<sup>2</sup>]. On the other hand, the maximum control input is equal to  $\bar{u} = 130^2$  [Hz<sup>2</sup>]. Mass and inertia of the vehicle are  $m_R = 1.3$  [Kg] and  $\mathbf{J}_R = \text{diag}(0.030, 0.030, 0.030)$  [Kg · m<sup>2</sup>], respectively.

We finally completed the omniplus design running the proposed Algorithm 1. Figure 4 shows the design used in the following simulations for which it has been achieved  $\text{cond}(\mathbf{F}) = 3.59$ . The optimized vectoring part is equal to:

$$[\mathbf{v}_1 \ \dots \ \mathbf{v}_n] = \begin{bmatrix} -0.71 & 0.11 & 0.41 & 0.44 & 0.57 & -0.64 & -0.17 \\ 0.67 & 0.04 & 0.85 & -0.35 & -0.38 & -0.58 & -0.26 \\ 0.11 & -0.98 & 0.31 & 0.81 & -0.72 & -0.48 & 0.94 \end{bmatrix}.$$

In order to fully show the capability of the proposed design, we ask the vehicle to translate and rotate at the same time. The translational trajectory is a spline from the initial position to a desired final one. For the orientation, we planned a trajectory such that the  $z$ -axis of  $\mathcal{F}_R$  circles many times around the one-radius sphere. In this way we

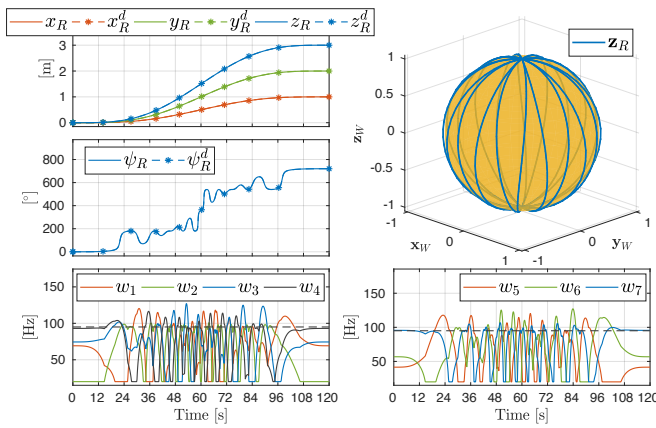


Fig. 5: Simulations results with the optimized platform for which  $\text{cond}(\mathbf{F}) = 3.59$ . Evolution of the translational and rotational outputs and rotor speeds  $w_1, \dots, w_7$ . The black dashed line represent the maximum angular velocity of the motors. The sphere is centered in  $O_R$  and follows the vehicle motion. The blue line shows the tip of  $\mathbf{z}_R$  w.r.t. a frame centered in  $O_R$  and parallel to  $\mathcal{F}_W$ .

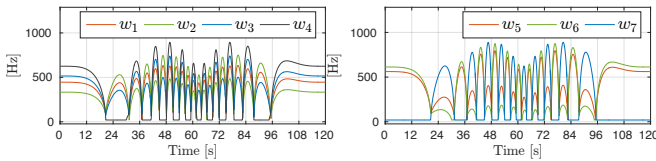


Fig. 6: Control inputs required to tracking the desired trajectory in Fig. 5 by a non-optimized omnipus design with  $\text{cond}(\mathbf{F}) = 186.84$ .

can span a vast variety of orientations. A representation of this trajectory is shown in Fig. 5 and in the attached video, where a possible realistic design is also shown. Looking at the plots one can see that the vehicle is able to track the desired trajectory requiring propeller rotational speeds  $w_1, \dots, w_7$  that are always in the limits. On the other hand, a non-optimized platform requires input peaks that go beyond the propeller limits (see Fig. 6 and the attached technical report for more details).

We also conducted a thorough simulation campaign to check the robustness of the proposed method against: i) noisy measurements, ii) parameters uncertainties, iii) non ideal motors, iv) control input delay, and v) external disturbances. This analysis showed good tracking performance for standard non-ideal scenarios, and allowed us to understand its limits. Furthermore, we investigated some interesting characteristic of the simulated platform as the maximum feasible forces and torques in every direction, the maximum and minimum thrust to weight ratio and the energy consumption. Finally, to show that our algorithm can be used for any number of propellers, we computed and simulated an optimized design for  $n = 8$ . Due to the limited space, we added all those results in the attached technical report.

## VIII. CONCLUSIONS

In this work we formalize the problem of designing an omnidirectional-thrust vehicle using only body-frame fixed unidirectional thrusters. We provide the main definitions, concepts and properties of such sought design. We

show the conditions that have to be satisfied to obtain the omnidirectional-thrust property and propose an algorithm to generate such design in an optimal way. We also propose a nonlinear controller to track position and orientation trajectories demonstrating the lowest possible inputs for the optimized platform.

Based on those fundamental results many other works could sprout up from the community. An example could be the mentioned improvement of the optimization algorithm, perhaps exploiting the noticed symmetries on an optimized platform, and considering the thrust position as well. The formal proof that balanced design with  $\min \text{cond}(\mathbf{F})$  minimizes the norm of the input is also left as future work, as well as the real implementation of such optimized platform.

## REFERENCES

- [1] M. Tognon and A. Franchi, "Dynamics, control, and estimation for aerial robots tethered by cables or bars," *IEEE Trans. on Robotics*, vol. 33, no. 4, pp. 834–845, 2017.
- [2] H. Nguyen and D. Lee, "Hybrid force/motion control and internal dynamics of quadrotors for tool operation," in *2013 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Tokyo, Japan, Nov. 2013, pp. 3458–3464.
- [3] B. Yüksel, C. Secchi, H. H. Bühlhoff, and A. Franchi, "Aerial physical interaction via reshaping of the physical properties: Passivity-based control methods for nonlinear force observers," in *ICRA 2014 Workshop: Aerial robots physically interacting with the environment*, Hong Kong, China, May. 2014.
- [4] M. Fumagalli, R. Naldi, A. Macchelli, R. Carloni, S. Stramigioli, and L. Marconi, "Modeling and control of a flying robot for contact inspection," in *2012 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Vilamoura, Portugal, Oct 2012, pp. 3532–3537.
- [5] G. Muscio, F. Pierri, M. A. Trujillo, E. Cataldi, G. Giglio, G. Antonelli, F. Caccavale, A. Viguria, S. Chiaverini, and A. Ollero, "Experiments on coordinated motion of aerial robotic manipulators," in *2016 IEEE Int. Conf. on Robotics and Automation*, Stockholm, Sweden, May 2016, pp. 1224–1229.
- [6] M. Tognon, B. Yüksel, G. Buondonno, and A. Franchi, "Dynamic decentralized control for protocentric aerial manipulators," in *2017 IEEE Int. Conf. on Robotics and Automation*, Singapore, May 2017, pp. 6375–6380.
- [7] S. Rajappa, M. Ryll, H. H. Bühlhoff, and A. Franchi, "Modeling, control and design optimization for a fully-actuated hexarotor aerial vehicle with tilted propellers," in *2015 IEEE Int. Conf. on Robotics and Automation*, Seattle, WA, May 2015, pp. 4006–4013.
- [8] H. Romero, S. Salazar, A. Sanchez, and R. Lozano, "A new UAV configuration having eight rotors: dynamical model and real-time control," in *46th IEEE Conf. on Decision and Control*, New Orleans, LA, Dec. 2007, pp. 6418–6423.
- [9] S. Park, J. J. Her, J. Kim, and D. Lee, "Design, modeling and control of omni-directional aerial robot," in *2016 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Daejeon, South Korea, 2016, pp. 1570–1575.
- [10] D. Brescianini and R. D'Andrea, "Design, modeling and control of an omni-directional aerial vehicle," in *2016 IEEE Int. Conf. on Robotics and Automation*, Stockholm, Sweden, May 2016, pp. 3261–3266.
- [11] M. Ryll, D. Bicego, and A. Franchi, "Modeling and control of FAST-Hex: a fully-actuated by synchronized-tilting hexarotor," in *2016 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Daejeon, South Korea, Oct. 2016, pp. 1689–1694.
- [12] M. Ryll, H. H. Bühlhoff, and P. Robuffo Giordano, "A novel over-actuated quadrotor unmanned aerial vehicle: modeling, control, and experimental validation," *IEEE Trans. on Control Systems Technology*, vol. 23, no. 2, pp. 540–556, 2015.
- [13] Y. Long and D. J. Cappelleri, *Omnicopter: A Novel Overactuated Micro Aerial Vehicle*. Heidelberg: Springer International Publishing, 2013, pp. 215–226.
- [14] A. Nikou, G. C. Gavridis, and K. J. Kyriakopoulos, "Mechanical design, modelling and control of a novel aerial manipulator," in *2015 IEEE Int. Conf. on Robotics and Automation*, May 2015, pp. 4698–4703.

- [15] K. P. Valavanis, *Advances in Unmanned Aerial Vehicles: State of the Art and the Road to Autonomy*, ser. Intelligent Systems, Control and Automation: Science and Engineering. Springer, 2007, vol. 33.
- [16] R. S. Sanchez Pena, R. Alonso, and P. A. Anigstein, "Robust optimal solution to the attitude/force control problem," *IEEE Trans. on Aerospace and Electronic System*, vol. 36, no. 3, pp. 784–792, 2000.
- [17] A. Bicchi and V. Kumar, "Robotic grasping and contact: a review," in *2000 IEEE Int. Conf. on Robotics and Automation*, vol. 1, April 2000, pp. 348–353.
- [18] S. Campbell and C. Meyer, *Generalized Inverses of Linear Transformations*. Society for Industrial and Applied Mathematics, 2009.
- [19] R. A. Waltz, J. L. Morales, J. Nocedal, and D. Orban, "An interior algorithm for nonlinear optimization that combines line search and trust region steps," *Mathematical Programming*, vol. 107, no. 3, pp. 391–408, Jul 2006.

# Additional Analysis and Simulations for an Omnidirectional-thrust vehicle with Only Fixed Unidirectional Thrusters

Technical report of:

“Omnidirectional Aerial Vehicles with Unidirectional Thrusters: Theory, Optimal Design, and Control”  
*IEEE Robotics and Automation Letters*

Marco Tognon<sup>1</sup> and Antonio Franchi<sup>1</sup>

**Abstract**—This document is a technical attachment to [1] as an extension of the numerical validation part. Here we present additional simulations in presence of non-ideal conditions as noise, parameter variations, non-ideal motors, control input delays and external disturbances. A through validation of the robustness of the proposed method against the previously mentioned non-idealities is conducted.

## I. HOW TO CITE THIS WORK

This technical report is accompanying our IEEE Robotics and Automation Letters paper [1]. If you wish to reference this work, please cite this paper as follows:

```
@Article{Tognon18ral,
  author = {M. Tognon and A. Franchi},
  title = {Omnidirectional Aerial Vehicles
    with Unidirectional Thrusters:
    Theory, Optimal Design,
    and Control},
  journal = {{IEEE} Robot. Autom. Lett.},
  year = {2018},
  doi = {10.1109/LRA.2018.2802544},
}
```

## II. ADDITIONAL SIMULATION

In this section we present some additional simulations performed to validate the proposed method in ideal and non-ideal conditions.

### A. Ideal Conditions

Figure 1 shows more detailed plots of the simulation done in ideal condition presented in Sec. VII of the paper.

### B. Standard Non-ideal Conditions

We simulated the system in non-ideal condition considering:

- Gaussian noise added to the state measurement with standard deviation equal to  $\sigma_{\mathbf{p}_R} = 0.01$  [m],  $\sigma_{\mathbf{v}_R} = 0.02$  [m/s],  $\sigma_{\mathbf{R}_R} = 3$  [°] and  $\sigma_{\omega_R} = 0.1$  [rad/s] for the position, linear velocity, attitude and angular velocity, respectively. This corresponds to the standard deviation

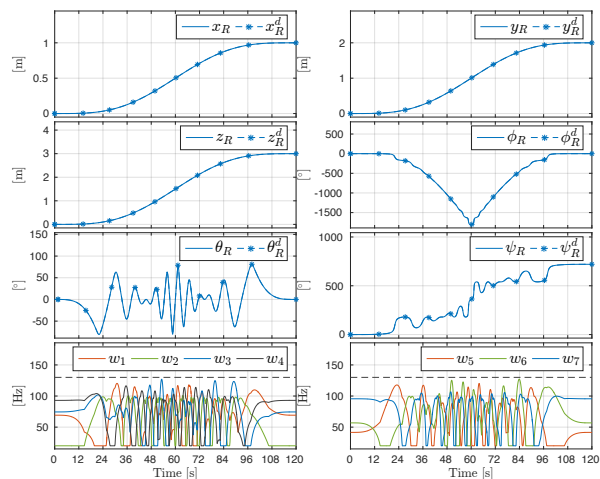


Fig. 1: Simulations results with the optimized platform with 7 propellers for which  $\text{cond}(\mathbf{F}) = 3.59$ . Evolution of the translational and rotational outputs and rotor speeds  $w_1, \dots, w_7$ . The black dashed line represent the maximum angular velocity of the motors.

of standard pose estimators for aerial vehicles, in order to simulate real sensors;

- non-ideal motors modeled as a first order system with time constant equal to 0.08 [s];
- parameters uncertainty for mass and inertia matrix equal to 5% of the nominal value.

Fig. 2 shows the tracking performance and the inputs of the closed loop system between 40 [s] and 45 [s]. Outside of this interval the behavior is the same. One can notice that the tracking error is relatively small and, most importantly, the control inputs do not increase with respect to the ideal case.

### C. Noise Robustness

Here we investigate the performances of the proposed method under increasing noise intensity. In particular, we performed several simulations in which the standard deviation of the noise varies from 0 to a maximum value of  $\bar{\sigma}_{\mathbf{p}_R} = 0.01$  [m],  $\bar{\sigma}_{\mathbf{v}_R} = 0.02$  [m/s],  $\bar{\sigma}_{\mathbf{R}_R} = 3$  [°] and  $\bar{\sigma}_{\omega_R} = 0.1$  [rad/s], that corresponds to a very bad sensorial setup. This analysis shows how the tracking performance would get worse with the degradation of the sensorial set-up, e.g., moving from

<sup>1</sup>LAAS-CNRS, Université de Toulouse, CNRS, Toulouse, France, [mtognon@laas.fr](mailto:mtognon@laas.fr), [antonio.franchi@laas.fr](mailto:antonio.franchi@laas.fr)

This work has been funded by the European Union’s Horizon 2020 research and innovation programme under grant agreement No 644271 AEROARMS.

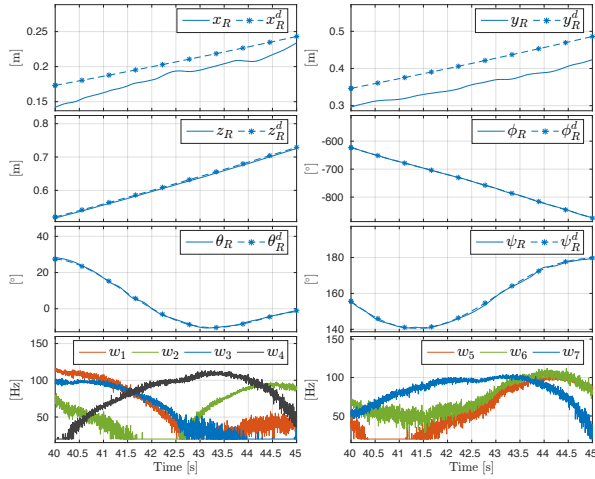


Fig. 2: Simulations with non-idealities. The plot zooms only on the time from 40 to 45 [s], the rest looks very similar.

motion capture system-like setup to a very poor gps. Notice that the chosen maximum standard deviations are higher than typical values obtained from standard state estimators, even using on-board sensors as vision or gps.

In order to show the results with an increasing noise, for each performed simulation we set  $\sigma_* = \Delta\sigma\bar{\sigma}_*$  with  $\Delta\sigma \in [0, 1]$  for each noise.

For every value of  $\Delta\sigma$ , in Fig. 3, we show the mean value and the standard deviation of the norm of the tracking error in position and attitude, i.e.,  $\bar{e}_{\mathbf{p}_R}$ ,  $\sigma_{e_{\mathbf{p}_R}}$ ,  $\bar{e}_{\mathbf{R}_R}$  and  $\sigma_{e_{\mathbf{R}_R}}$ , respectively. Those quantities are computed as in the following:

$$\begin{aligned}
 e_{\mathbf{p}_R} &= \|\mathbf{p}_R^d - \mathbf{p}_R\|^2 \\
 e_{\mathbf{R}_R} &= \|\mathbf{e}_{\mathbf{R}_R}\|^2 \\
 \bar{e}_{\mathbf{p}_R} &= \frac{1}{T} \int_0^T e_{\mathbf{p}_R}(t) dt \\
 \sigma_{e_{\mathbf{p}_R}} &= \sqrt{\frac{1}{T} \int_0^T (e_{\mathbf{p}_R}(t) - \bar{e}_{\mathbf{p}_R})^2 dt} \\
 \bar{e}_{\mathbf{R}_R} &= \frac{1}{T} \int_0^T e_{\mathbf{R}_R}(t) dt \\
 \sigma_{e_{\mathbf{R}_R}} &= \sqrt{\frac{1}{T} \int_0^T (e_{\mathbf{R}_R}(t) - \bar{e}_{\mathbf{R}_R})^2 dt}.
 \end{aligned} \tag{1}$$

In Fig. 3 one can see how the tracking error obviously gets worse with the increasing of the noise intensity. However, for reasonable level of noise, the mean tracking error is always limited and sufficiently small.

Furthermore, and more importantly, the platform never gets unstable even if the measurements are extremely degraded.

#### D. Motor Time Constant Robustness

To test the robustness considering non-ideal motors, we modeled each of them as a first order system characterized by a time constant  $\tau_M \in \mathbb{R}_{>0}$ . In Fig. 4 we assess the robustness with respect to it. In particular we plot  $\bar{e}_{\mathbf{p}_R}$ ,  $\sigma_{e_{\mathbf{p}_R}}$ ,  $\bar{e}_{\mathbf{R}_R}$  and  $\sigma_{e_{\mathbf{R}_R}}$ , varying the time constant from the value

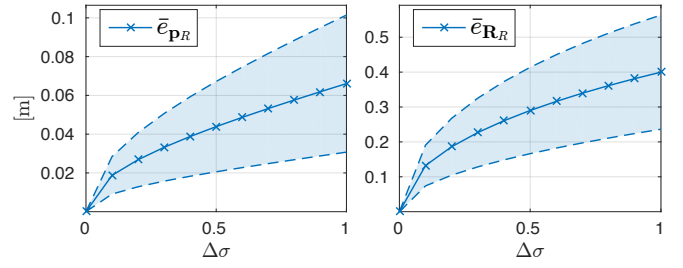


Fig. 3: Tracking performances with respect to noise intensity. The mean value of the norm of the tracking error with respect to the noise intensity, is plotted as a solid line. The opaque region shows the area between plus and minus the standard deviation.

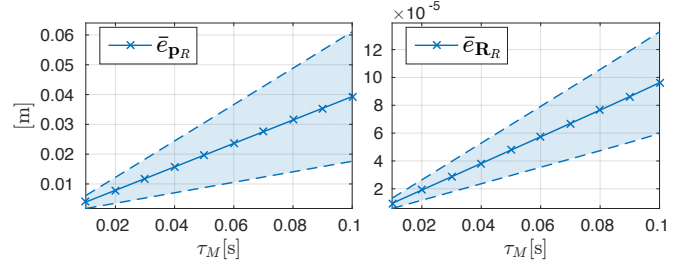


Fig. 4: Tracking performances considering non-ideal motors modeled as first order systems with time constant  $\tau_M$ . The mean value of the norm of the tracking error with respect to the time constant of the motors is plotted as a solid line; the opaque region shows the area between plus and minus the standard deviation.

of 0.01 [s] to the one of 0.1 [s]. As expected, the tracking performance gets worse until the system becomes unstable for time constants larger than 0.1 [s]. For larger values one could easily incorporate the motor dynamics in the model, including the motor speed in the system state and considering its derivative as the new input. Being the new model fully controllable, a design similar to the one presented in [1] would make the job of stabilizing the platform. However, for a standard brushless motors with the closed-loop speed controller presented in [2], the time constant is about 0.03 [s]. For this value the corresponding tracking error is sufficiently good without the need of an extended model.

#### E. Motor Communication Delay Robustness

In a real platform there will always be a certain delay in the communication with the motor controller. We have tested which is the maximum delay value for which we can obtain a stable behavior. In Fig. 5 we show the tracking performance with respect to an increasing value of the delay between the commanded angular velocity for the motor and the one received as set-point by the motor controller. The tracking error, although sufficiently small, increases until the maximum delay of 0.07 [s]; after this value some oscillatory modes appear. However notice that a delay of 0.07 [s] is incredibly large with respect to standard control input delays on aerial platform where the controller is implemented on an on-board PC. Usually, for those configurations, the delay value is below 0.002 [s].

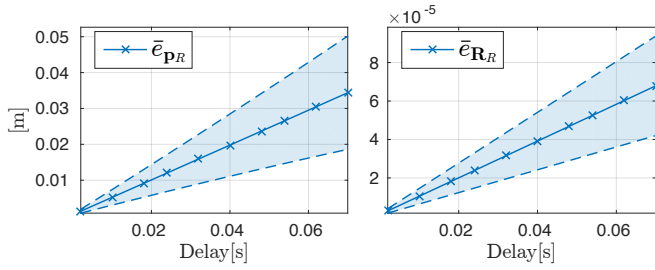


Fig. 5: Tracking performances with respect to control input delay. The mean value of the norm of the tracking error with respect to the noise intensity is plotted as a solid line; the opaque region shows the area between plus and minus the standard deviation.

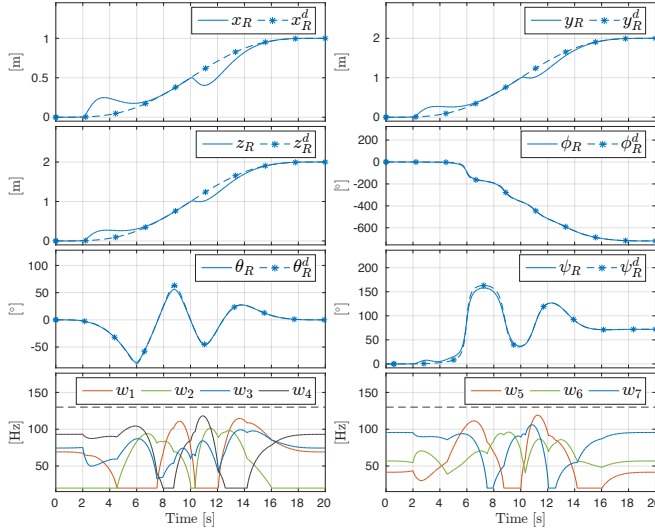


Fig. 6: Tracking performance under an external force acting on the system from time 2 [s] to time 10 [s]

### F. Disturbance Rejection

We investigate here the behavior of the system under external disturbances. In particular, in Fig. 6 we show the tracking performance under an external constant force acting at  $\mathbf{d}_7$  (the position the 7th propeller) from time 2 [s] to time 10 [s], generating both translation and orientation disturbances. This external force defined in world frame is equal to  $[2 \ 2 \ 2]^T$  [N].

As we can see from Fig. 6 at time 2 [s] the external force is “activated” and the tracking error increases. However, thanks to the integral action in the controller, after a transient the system is able to counterbalance the effects of the disturbance bringing to zero the tracking error. A similar behavior is shown when the external force is “de-activated” at time 10 [s].

Notice that the disturbance rejection performance could be further improved using a disturbance observer. However this goes beyond the scope of this manuscript.

### G. Non-optimized Omniplus Design

To better show the importance of optimizing a design to reduce the risk of saturation, we computed an omniplus design without minimizing the condition number. The resulting platform is shown in Fig. 7 on the top. We then required the vehicle to follow the same desired trajectory considered

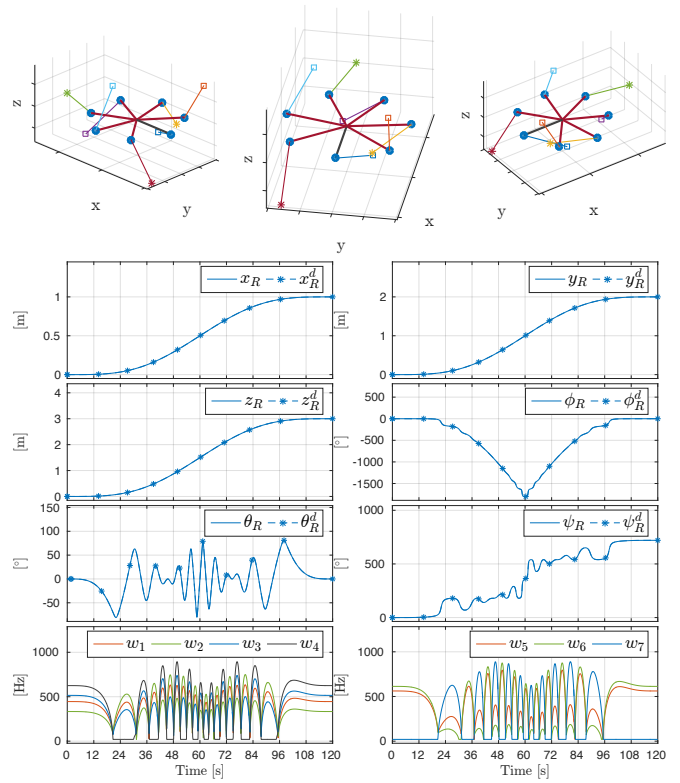


Fig. 7: Non-optimized omniplus design with  $\text{cond}(\mathbf{F}) = 186.84$  and relative control inputs required to tracking the desired trajectory in Fig. 5 of [1].

for the optimized design. In Fig. 7 one can clearly see that the trajectory is tracked as for the optimized platform but the required control inputs are much larger and they go beyond the maximum value of 130 [Hz]. The simulation does not consider the saturation in order not to obtain unstable behaviors and shows the required control inputs for the full trajectory.

### H. Optimal Omniplus Design with 8 Propellers

Finally to show that the proposed algorithm works with any number of propellers, we computed an optimal omniplus-vehicle with  $n = 8$ . Figure 8 shows the obtained design from different perspectives and the tracking performance following the usual desired trajectory. One can see that the tracking and the desired control inputs are comparable to the one of the optimal omniplus design with 7 thrusters.

### REFERENCES

- [1] M. Tognon and A. Franchi, “Omnidirectional aerial vehicles with unidirectional thrusters: Theory, optimal design, and control,” *IEEE Robotics and Automation Letters*, 2018.
- [2] A. Franchi and A. Mallet, “Adaptive closed-loop speed control of BLDC motors with applications to multi-rotor aerial vehicles,” in *2017 IEEE Int. Conf. on Robotics and Automation*, Singapore, May 2017.

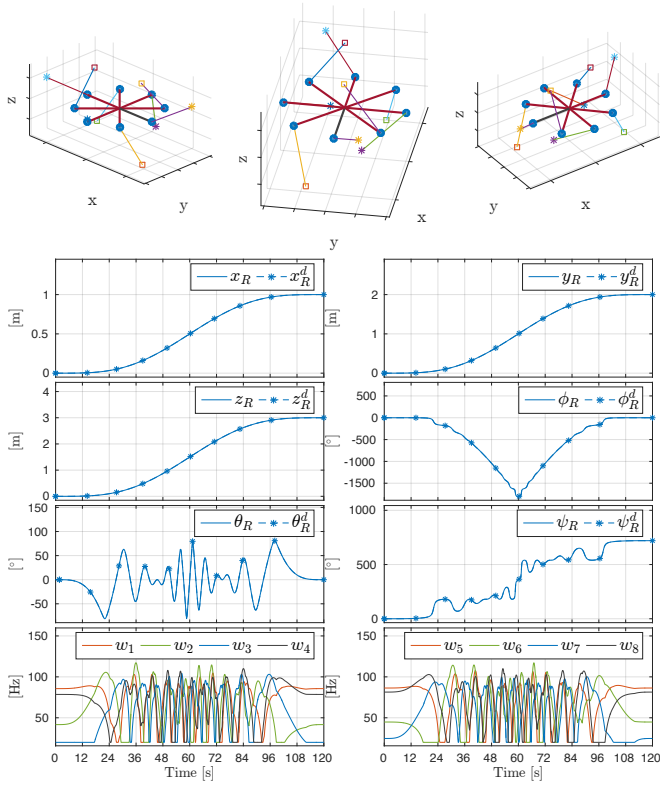


Fig. 8: Optimized omnibus design with  $n = 8$  and  $\text{cond}(\mathbf{F}) = 3.59$ , and relative control inputs required to tracking the desired trajectory.