

# Full-Pose Trajectory Tracking of Overactuated Multi-Rotor Aerial Vehicles With Limited Actuation Abilities

Mahmoud Hamandi <sup>1</sup>, Ismail Al-Ali, Lakmal Seneviratne <sup>2</sup>, Antonio Franchi <sup>3</sup>, *Fellow, IEEE*,  
and Yahya Zweiri <sup>4</sup>, *Member, IEEE*

**Abstract**—This letter presents a novel optimization-based full-pose trajectory tracking method to control overactuated multi-rotor aerial vehicles with limited actuation abilities. The proposed method allocates feasible control inputs to track a reference trajectory, while ensuring the tracking of the reference position, and while tracking the closest feasible attitude. The optimization simultaneously searches for a feasible trajectory and corresponding feasible control inputs from the infinite possible solutions, while ensuring smooth control inputs. The proposed real-time algorithm is tested in extensive simulation on multiple platforms with fixed and actuated propellers. The simulation experiments show the ability of the proposed approach to exploit the complex set of feasible forces and moments of overactuated platforms while allocating smooth feasible control inputs.

**Index Terms**—Aerial systems: Applications, aerial systems: Mechanics and control, optimization and optimal control.

Manuscript received 28 January 2023; accepted 31 May 2023. Date of publication 28 June 2023; date of current version 7 July 2023. This letter was recommended for publication by Associate Editor Cunjia Liu and Editor Pauline Pounds upon evaluation of the reviewers' comments. The work of Mahmoud Hamandi was supported by the NYUAD Center for Artificial Intelligence and Robotics, funded by Tamkeen through the NYUAD Research Institute Award CG010. The work of Antonio Franchi was supported by the H2020 Research and Innovation Program under Grant 871479 (AERIAL-CORE). This work was supported by the Khalifa University of Science and Technology under Award RC1-2018-KUCARS and Project CIRA-2020-082. (*Corresponding author: Mahmoud Hamandi.*)

Mahmoud Hamandi is with the Department of Aerospace Engineering, Khalifa University, Abu Dhabi 52799, UAE, and with the Khalifa University Center for Autonomous Robotic Systems, Khalifa University, Abu Dhabi 52799, UAE, and also with the Center for Artificial Intelligence and Robotics, New York University Abu Dhabi, Abu Dhabi 129188, UAE (e-mail: mahmoud.hamandi@nyu.edu).

Ismail Al-Ali is with the Khalifa University Center for Autonomous Robotic Systems, Khalifa University, Abu Dhabi 52799, UAE (e-mail: ismail.alali@ku.ac.ae).

Lakmal Seneviratne is with the Department of Mechanical Engineering, Khalifa University, Abu Dhabi 52799, UAE, and also with the Khalifa University Center for Autonomous Robotic Systems, Khalifa University, Abu Dhabi 52799, UAE (e-mail: lakmal.seneviratne@ku.ac.ae).

Antonio Franchi is with the Robotics and Mechatronics Lab, Faculty of Electrical Engineering, Mathematics and Computer Science, University of Twente, Enschede 7500 AE, The Netherlands, and with the Department of Computer, Control and Management Engineering, Sapienza University of Rome, 00185 Rome, Italy, and also with the LAAS-CNRS, Université de Toulouse, 31000 Toulouse, France (e-mail: a.franchi@utwente.nl).

Yahya Zweiri is with the Department of Aerospace Engineering, Khalifa University, Abu Dhabi 52799, UAE, and also with the Advanced Research and Innovation Center, Khalifa University, Abu Dhabi 52799, UAE (e-mail: yahya.zweiri@ku.ac.ae).

Digital Object Identifier 10.1109/LRA.2023.3290422

## I. INTRODUCTION

UNMANNED Aerial Vehicles (UAV)s gained popularity in the last two decades due to their versatility, and wide range of applications spanning aerial-physical inspection, photography and others. With the expansion of applications, many new UAV designs emerged aiming at expanding the work space beyond what was possible with the classical quadrotor. Such new designs changed the number and configuration of propellers to construct *fully actuated* and *overactuated* platforms [1].

Overactuated platforms are UAVs with redundant control inputs, such that the number of control inputs is larger than the dimensionality of the 6D force-moment work space. Such platforms provide redundancy in the case of propeller failures and ensure full actuation when some of their actuation limits have been reached. For example, [2] prove that for a platform with fixed propellers and a uni-directional thrust constraint, at least seven propellers are required to achieve omnidirectional thrust (i.e., full actuation in all directions). Many other overactuated platforms are presented in the literature, such as [3], [4], [5], etc... However, most overactuated platforms operate away from actuation limits and thus do not fully exploit the full range of the platforms' control inputs. In spite of the advantages of overactuated platforms, the larger control set provides many challenges.

One of the challenges of interest in this letter is the allocation of feasible control inputs to track a reference trajectory, in the presence of an infinite number of such possible controls and while satisfying actuation limits. In many designs, the minimum energy solution (i.e., moore-penrose pseudo inverse allocation [6]) provides feasible controls to hover around some of the operating points, and could break actuation limits when diverging from those points. For example, the minimum energy solution for the dual-tilt quadrotor shown in [7] provides a feasible solution when the desired force is parallel to the platform's vertical axis, and violates the actuation constraints in other operating points. On the other hand, the authors in [8] show that for the fixed propeller overactuated heptarotor they designed, the minimum energy solution is guaranteed to break at least one of the actuation constraints. To avoid actuation constraints, the authors in [9] suggest a null space-based approach to find feasible control inputs to achieve a desired force and moment. Their approach optimizes the incremental control input added

to the currently applied controls. The optimization is relaxed by adding an error term, assumed to be in the null space of the allocation matrix. Similar to [9], the authors in [10] suggest to allocate incremental control inputs added to the currently applied controls; however, their proposed overactuated platform had only thrust constraints, and thus it was possible to operate away from actuation limits.

While overactuated and fully actuated platforms have the ability to track independent position and attitude [11] away from actuation limits, the full 6D trajectory tracking have to be relaxed near actuation limits to ensure the satisfaction of the corresponding constraints. Correspondingly, a trajectory could be modified online to ensure that the allocated forces and moments are inside the platform's set of feasible forces and moments. Most solutions in the literature follow the reference position while compromising the reference attitude. This approach is similar to the classical approach applied to classical quadrotors. In classical quadrotors, the set of feasible forces is simply a line along the body's vertical axis [1]. As such, classical controllers point the platform's vertical axis in the direction of the desired force [12]. As the set of feasible forces and moments is higher dimensional for overactuated and fully actuated platforms, multiple letters from the literature estimate the corresponding set to find a feasible solution. The authors in [13] define a maximum roll and pitch angle the platform could reach, and allocate forces and moments within the defined constraints, implicitly estimated as a cone. Note that multiple letters from the literature follow a similar approach. On the other hand, multiple letters in the literature present solutions after estimating the set of feasible forces while applying zero moments, referred to as the force set. For example, the authors in [14] generate 6D trajectory primitives, and test the feasibility of each primitive offline. While their method is promising, it requires the definition of geometrical limits of the platform's thrust and angular velocity. As such, their method can ensure the feasibility of the trajectory, but it does not fully exploit the set of feasible control. On the other hand, the authors in [15] estimate the platform's force set as a cylinder, and limit the maximum rotation of the platform such that the applied force is within the defined bounds. Similarly, the authors in [16] estimate the force set as a cone, and allocate angular velocities to track the closest attitude to the reference, while tracking the desired position and satisfying the cone constraints. These letters provide a promising solution for trajectory tracking of overactuated and fully actuated platforms. However, all these solutions estimate the set of feasible forces of the platform, parameterized by a single parameter (maximum angle of the cone, radius of the cylinder, etc.). As shown in [17], the estimation of this parameter could either be conservative, and thus do not fully exploit the platform's actuation limits, or could be liberal and risk allocating unfeasible forces and moments. In addition, the complexity of the set of feasible forces and moments grows with the number of added control inputs [1]. Note also that in the above solutions, the force set is estimated while applying zero-moment, and the variable parameter is set conservatively such that the allocated controls are always feasible.

In recent years, optimization-based control allocation became very popular. Multiple letters from the literature showed that the optimization of the platform's trajectory over a time horizon could be a solution to ensure the feasibility of the trajectory [18], [19]. Such methods have also been used in the literature for the control allocation of multi-input systems, such as humanoids and aerial manipulators [20].

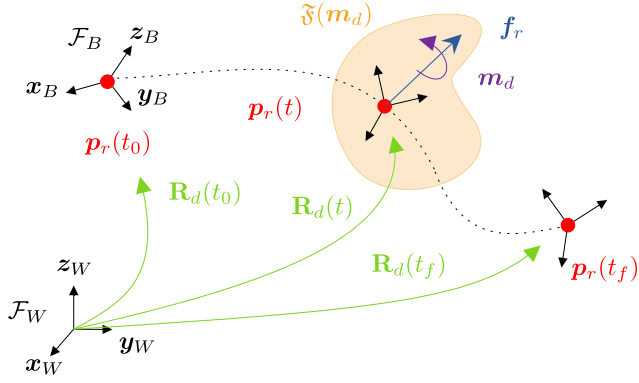
While there has been many control allocation strategies in the literature [21], in this letter we propose a novel optimization-based trajectory tracking algorithm for overactuated and fully actuated platforms with limited actuation abilities. The proposed optimization framework searches online for statics based feasible control inputs to track a reference 6D trajectory directly in the set of feasible control inputs, without the need to model the set of feasible forces and moments. In the absence of such control inputs, it searches for feasible control inputs such that the reference position is tracked, while tracking a feasible attitude that is the closest to the reference. While searching for a feasible trajectory, the optimization algorithm enforces a smooth change in control inputs to accommodate for the platform's dynamics.

The performance of the proposed real-time algorithm is tested in an exhaustive realistic simulation. We demonstrate in these simulations the ability of the proposed approach to exploit the complex topology of the set of feasible forces and moments for overactuated platforms while satisfying actuation limits. These simulations also demonstrate the generalization of the proposed algorithm to different platforms, with fixed and actuated propellers.

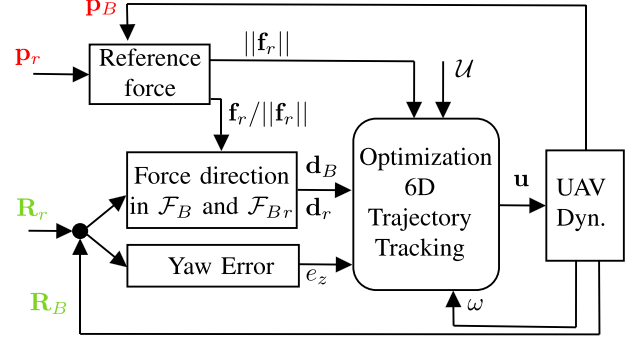
The letter is organized as follows. Section II models a generic UAV and Section III defines the generic UAV control and feasible trajectory problem. Section IV describes the optimization-based 6D trajectory tracking algorithm. Section V analyzes the proposed method experimentally, and finally, Section VI concludes the letter.

## II. MODELING

Let us first denote by  $\mathcal{F}_W$  as the world frame with axes  $\{\mathbf{x}_W, \mathbf{y}_W, \mathbf{z}_W\}$  and a body fixed frame  $\mathcal{F}_B$  with axes  $\{\mathbf{x}_B, \mathbf{y}_B, \mathbf{z}_B\}$ . The origin of the body frame  $O_B$  is assumed to coincide with the Center of Mass (CoM) of the platform. Let us denote by  $\mathbf{p}_B, \mathbf{v}_B \in \mathbb{R}^3$  as respectively the position of  $O_B$  in  $\mathcal{F}_W$ , and its velocity with respect to (w.r.t.)  $\mathcal{F}_W$ , and by  $\mathbf{R}_B \in SO(3)$  and  $\boldsymbol{\omega}_B \in \mathbb{R}^3$  as the orientation of  $\mathcal{F}_B$  in  $\mathcal{F}_W$  and its angular velocity w.r.t.  $\mathcal{F}_B$ . Furthermore, let us denote by  $N \in \mathbb{N}_{>0}$  as the number of propellers, and let us assume for the sake of this letter, the generic case where each propeller can be tilted independently in  $S^2$  (i.e., each propeller is actuated by 2 independent servo motors). Note that propellers tilting in  $S^1$  or fixed are special cases of the above [22]. Let us denote by  $\mathcal{F}_{p_i}$  as the propeller fixed frame, with its origin  $O_{p_i}$  coinciding with the fixed attachment point of the  $i^{th}$  propeller in body frame.  $\mathcal{F}_{p_i}$  is defined such that the propeller's thrust is always generated about its  $z$ -axis  $\mathbf{z}_{p_i}$ . Finally, let us denote by  $\mathbf{R}_{p_i} \in \mathcal{R}_i \subset SO(3)$  as a rotation that brings a vector in  $\mathcal{F}_{p_i}$  to  $\mathcal{F}_B$ , where  $\mathcal{R}_i$  is the set of feasible propeller orientations.



(a) Schematic illustrating the proposed full pose trajectory tracking approach. The proposed approach guarantees the platform follows the reference position, while following a desired orientation, where the desired orientation is the closest to the reference, such that  $\mathbf{f}_r$  and  $\mathbf{m}_d$  inside the set  $\mathfrak{F}(\mathbf{m}_d)$ .



(b) Illustration of the main components of the optimization algorithm.

Fig. 1. Illustrations showing the main components representing a feasible trajectory, and the proposed approach for its computation.

Depending on its orientation, each propeller can produce forces in  $\mathcal{F}_B$  denoted as  $\mathbf{u}_i = \{u_{x,i}, u_{y,i}, u_{z,i}\} \in \mathbb{R}^3$ , with norm  $u_i$ , such that  $\mathbf{u}_i = \mathbf{R}_{p_i} \mathbf{e}_3 u_i$ . It is assumed that  $u_i = c_f \omega_i^2$ , where  $\omega_i \in [\underline{\omega}, \bar{\omega}]$  is the propeller's rotational speed, and  $c_f$  its lift coefficient assumed constant. We denote by  $[\underline{u}, \bar{u}]$  as the minimum and maximum thrust that can be generated by a propeller, and corresponding to the range of  $\omega_i$ .

Let us denote by  $\mathbf{u} \in \mathcal{U} \subset \mathbb{R}^{3N}$  as the platform's control input, where  $\mathcal{U}$  is denoted as the set of feasible control inputs. It is easy to see that  $\mathcal{U}$  can be deduced from the propellers' feasible orientations  $\mathcal{R}_i$ , and from the propellers' rotational speed range  $[\underline{\omega}, \bar{\omega}]$ .

To understand the effect of the propellers' thrust on the platform's motion, let us write the platform's Newton-Euler equations as follows:

$$\begin{bmatrix} m_B \dot{\mathbf{v}}_B \\ \mathbf{J}_B \dot{\boldsymbol{\omega}}_B \end{bmatrix} = - \begin{bmatrix} m_B g \mathbf{z}_W \\ \boldsymbol{\omega}_B \times \mathbf{J}_B \boldsymbol{\omega}_B \end{bmatrix} + \mathbf{G} \begin{bmatrix} \mathbf{f} \\ \mathbf{m} \end{bmatrix} \quad (1)$$

where  $m_B \in \mathbb{R}_{>0}$  and  $\mathbf{J}_B \in \mathbb{R}^{3 \times 3}_{>0}$  are the platform's mass and inertia matrix,  $g$  is the gravitational constant,  $\mathbf{f}, \mathbf{m} \in \mathbb{R}^3$  are propeller induced forces and moments applied on the platform's CoM in  $\mathcal{F}_B$ , and  $\mathbf{G}$  is a matrix that rotates  $\mathbf{f}$  from  $\mathcal{F}_B$  to  $\mathcal{F}_W$ . The platform's forces and moments for a platform with propellers tilting in  $\mathcal{S}^2$  can be written in the following format:

$$\begin{bmatrix} \mathbf{f} \\ \mathbf{m} \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{I}_3 & \dots & \mathbf{I}_3 \\ \mathbf{S}(\mathbf{p}_1^p) + r\mathbf{I} & \dots & \mathbf{S}(\mathbf{p}_N^p) + r\mathbf{I} \end{bmatrix}}_{\mathbf{F}} \mathbf{u} \quad (2)$$

where  $\mathbf{S}(\cdot)$  is the skew-symmetric operator,  $r = c_\tau/c_f$  is the ratio between the drag coefficient and the lift coefficient, and  $\mathbf{I}$  is the identity matrix. The matrix  $\mathbf{F}$ , referred to as the full allocation matrix, is constant throughout the operation of the platform [22]. The full allocation matrix can be written as  $\mathbf{F} = [\mathbf{F}_1^T \ \mathbf{F}_2^T]^T$ , where  $\mathbf{f} = \mathbf{F}_1 \mathbf{u}$  and  $\mathbf{m} = \mathbf{F}_2 \mathbf{u}$ .

Finally, let us define the platform's force set at a specific moment as follows:

$$\mathfrak{F}(\mathbf{m}) = \left\{ \forall \mathbf{f} \in \mathbb{R}^3 \text{ s.t. } \begin{bmatrix} \mathbf{f} \\ \mathbf{m} \end{bmatrix} = \mathbf{F} \mathbf{u} \mid \mathbf{u} \in \mathcal{U} \right\} \quad (3)$$

### III. PLATFORM CONTROL

In this section we discuss the control structure behind the Full-pose trajectory tracking algorithm. The complete architecture is illustrated in Fig. 1.

A full-pose reference trajectory is denoted by  $\mathbf{q}_r(t) = (\mathbf{p}_r(t), \mathbf{R}_r(t)) : [t_0, t_f] \rightarrow SE(3)$ , where  $\mathbf{p}_r(t) \in \mathbb{R}^3$  is the reference position trajectory and  $\mathbf{R}_r(t) \in SO(3)$  is the reference attitude trajectory. It is assumed that  $\mathbf{q}_r(t)$  is a smooth polynomial trajectory up to the 4<sup>th</sup> order (i.e., continuous jerk), similar to the one presented in [23]. Let us denote by  $\mathbf{f}_r(t)$  as the force required to follow a desired position  $\mathbf{p}_r(t)$ , and by  $\mathbf{m}_r(t)$  as the moment required to follow a desired orientation  $\mathbf{R}_r(t)$ .

For simplicity, let us assume that any direction of  $\mathbf{m} \in \mathbb{R}^3$  is admissible as long as the corresponding control input does not reach saturation. Following this assumption, a feasible trajectory is illustrated in Fig. 1(a), and can be defined as follows:

*Definition 1:* A trajectory  $\mathbf{q}_r(t)$  is feasible if  $\mathbf{R}_r^T(t) \mathbf{f}_r^W(t) \in \mathfrak{F}(\mathbf{m}_r) \forall t \in [t_0, t_f]$ .

where  $\mathbf{f}_r^W$  is the reference force in  $\mathcal{F}_W$ , and  $\mathbf{R}_r^T(t) \mathbf{f}_r^W(t)$  is the equivalent force in  $\mathcal{F}_B$ , and  $\mathbf{m}_r$  is the moment required to follow  $\mathbf{R}_r^T(t)$ .

We denote by  $\mathbf{q}_d(t) = (\mathbf{p}_d(t), \mathbf{R}_d(t)) : [t_0, t_f] \rightarrow SE(3)$  as a desired feasible trajectory as follows:

$$\mathbf{p}_d(t) = \mathbf{p}_r(t), \quad \mathbf{R}_d(t) = \arg \min_{\mathbf{R}} \text{dist}(\mathbf{R}, \mathbf{R}_r(t))$$

$$\text{s.t. } \mathbf{R}_d^T(t) \mathbf{f}_d^W(t) \in \mathfrak{F}(\mathbf{m}_d) \forall t \in [t_0, t_f] \quad (4)$$

where  $\mathbf{m}_d$  is the feasible desired moment required to follow the feasible rotation  $\mathbf{R}_d^T(t)$  and  $\text{dist}$  is any function that represents the distance between the two rotations, such that the function is minimal when the two rotations are equivalent. Given the above definition of a feasible trajectory, the objective of this letter is to

find a feasible trajectory  $\mathbf{q}_d(t)$  that is the closest to the reference trajectory as stated in (4). This problem can be stated as follows:

*Problem 1:* Given a reference trajectory  $\mathbf{q}_r(t) : [t_0, t_f]$ , find a feasible trajectory:

$$\begin{aligned} \mathbf{q}_d(t) &: [t_0, t_f] \text{ s.t.} \\ \mathbf{R}_d(t) &= \arg \min_{\mathbf{R}} \text{dist}(\mathbf{R}, \mathbf{R}_r(t)) \\ \begin{bmatrix} \mathbf{R}_d^T(t) \mathbf{f}_d^W(t) \\ \mathbf{m}_d(t) \end{bmatrix} &= \mathbf{F} \mathbf{u}(t), \text{ s.t } \mathbf{u}(t) \in \mathcal{U} \forall t \in [t_0, t_f] \end{aligned} \quad (5)$$

where it was assumed that  $\mathbf{f}_d^W(t) = \mathbf{f}_r^W(t)$ .

#### A. Feedback Controller

First, let us state the position and attitude controllers:

$$\mathbf{f}_d^W = m_B(\dot{\mathbf{v}}_d + g\mathbf{e}_3) - \mathbf{K}_p \mathbf{e}_p - \mathbf{K}_v \mathbf{e}_v \quad (6)$$

$$\mathbf{m}_d = \boldsymbol{\omega} \times \mathbf{J}_B \boldsymbol{\omega} - \mathbf{K}_R \mathbf{e}_R - \mathbf{K}_w \mathbf{e}_w - \text{FF} \quad (7)$$

$$\text{FF} = \mathbf{J}_B (\boldsymbol{\omega} \times \mathbf{R}^T \mathbf{R}_d \boldsymbol{\omega}_d - \mathbf{R}^T \mathbf{R}_d \dot{\boldsymbol{\omega}}_d) \quad (8)$$

where  $\mathbf{K}_p$ ,  $\mathbf{K}_v$ ,  $\mathbf{K}_R$ ,  $\mathbf{K}_w$  are control gains,  $\mathbf{e}_p$ ,  $\mathbf{e}_v$ ,  $\mathbf{e}_R$  and  $\mathbf{e}_w$  are respectively the error terms in position, velocity, orientation, and angular velocity, and FF is the feedforward attitude tracking term. While the existence of the feedforward term enhances the trajectory tracking of the platform, the platform can still converge to the desired trajectory in the absence of such term [15]. The implementation of the feedforward term requires the computation of  $\boldsymbol{\omega}_d$  and  $\dot{\boldsymbol{\omega}}_d$ , in addition to the computed  $\mathbf{R}_d$ . While it is straightforward to compute the feedforward term for the fixed trajectory  $\mathbf{q}_r(t)$ , it might be impractical to compute it for the varying feasible trajectory  $\mathbf{q}_d(t)$ . As such, the feedforward term will be ignored in the optimization formulation of this letter.

Following the computation of the desired force and moment, a tentative control input can be computed as follows:

$$\mathbf{u}_c = \underbrace{\mathbf{F}^\dagger \begin{bmatrix} \mathbf{f}_d \\ \mathbf{m}_d \end{bmatrix}}_{\tilde{\mathbf{u}}_c} + \underbrace{\mathbf{B} \mathbf{x}}_y \quad (9)$$

where  $\mathbf{F}^\dagger \in \mathbb{R}^{3 \times 6}$  is the moore-penrose pseudo-inverse of the allocation matrix  $\mathbf{F}$ ,  $\tilde{\mathbf{u}}_c$  is the minimum energy solution,  $\mathbf{B} \in \mathbb{R}^{3 \times \max(0, 3N-6)}$  its null space basis, and  $\mathbf{x} \in \mathbb{R}^{\max(0, 3N-6)}$  is an arbitrary variable, such that  $\forall \mathbf{x} \in \mathbb{R}^{\max(0, 3N-6)} \mathbf{F} \mathbf{y} = \mathbf{0}$ . In the case where  $3N - 6 > 0$ , even if  $\mathbf{f}_d \in \mathfrak{F}(\mathbf{m}_d)$ , it is possible that  $\exists \mathbf{u}_c \in \mathcal{U}$  while  $\tilde{\mathbf{u}}_c \notin \mathcal{U}$ . In this case, the allocated control input in the null space of the allocation matrix allows changing the control input such that  $\mathbf{u}_c \in \mathcal{U}$  without changing the applied forces and moments. Then for a feasible force and moment, the control input in the null space can be found as follows:

$$\mathbf{y}^* = \mathbf{B} \arg \min_{\mathbf{x} \in \mathcal{U}} \|\mathbf{x}\|_Q^2 \quad (10)$$

where  $\mathbf{Q} = \mathbf{B}^T \mathbf{B}$ . Following (10) we can find a feasible control input for any feasible force and moment.

#### IV. FEASIBLE TRAJECTORY OPTIMIZATION

To find a feasible trajectory  $\mathbf{q}_d(t)$  for an overactuated platform, one has to solve the minimization from (4), while allocating optimal control inputs for the desired forces and moments following (10).

If there exists a control input that allows the platform to follow  $\mathbf{q}_r(t)$ , then  $\mathbf{q}_d(t) = \mathbf{q}_r(t)$ . In the case where  $\mathbf{q}_r(t)$  is not feasible, our optimization will advantage following the reference position while following a feasible desired orientation closest to the reference (similar to [15]), i.e.,  $\mathbf{f}_d^W(t) = \mathbf{f}_r^W(t)$ . Then the problem would resolve in finding a platform orientation such that  $\mathbf{R}_d^T(t) \mathbf{f}_d^W(t) \in \mathfrak{F}(\mathbf{m}_d(t))$ .

Obviously, the solution depends on the shape of the platform's force set. For example, in the case of a classical quadrotor, the force set is a line [1], and the trivial solution will always be to orient such line in the direction of the desired force, such that  $\mathbf{R}_d^T(t) \mathbf{f}_d^W(t) \propto \mathbf{z}_B$ . Multiple approaches in the literature fit a predefined geometrical shape into the force set (such as a cone [16] or a cylinder [15]). As shown in [17], such assumptions can either be conservative, and thus do not fully exploit the feasible control set, or elaborate and risk allocating infeasible control inputs. In this letter, we propose to find a feasible solution by searching directly in  $\mathcal{U}$ . As such, the allocated controls can take advantage of the entire  $\mathcal{U}$  while ensuring the feasibility of any allocated control input.

Let us write the optimization function  $J$  at each time step as follows (in what follows we will drop the time dependence for clarity):

$$\begin{aligned} J &= \min_{\mathbf{R}, \mathbf{x}} \text{dist}(\mathbf{R}_r, \mathbf{R}) + \frac{1}{2} \|\mathbf{x}\|_Q^2 \\ &\text{s.t.} \\ \mathbf{F}^\dagger \begin{bmatrix} \mathbf{R}^T \mathbf{f}_d^W \\ \mathbf{m}_d \end{bmatrix} + \mathbf{B} \mathbf{x} &\in \mathcal{U} \end{aligned} \quad (11)$$

Let us assume for simplicity that the constraint  $\mathbf{u}_c \in \mathcal{U}$  can be expressed as a group of linear and quadratic constraints. Despite this assumption, the optimization from (11) is still non convex due to the optimization of the rotation matrix. Instead, let us define the force direction in body frame as  $\mathbf{d} = \mathbf{R}^T \mathbf{f}^W / \|\mathbf{f}^W\|$ , and let us note by  $\mathbf{d}_B$ ,  $\mathbf{d}_r$  and  $\mathbf{d}_d$  as the corresponding force direction in current body frame  $\mathbf{R}_B$ , in reference body frame  $\mathbf{R}_r$  and in desired body frame  $\mathbf{R}_d$ . As such, the minimization between  $\mathbf{R}_r$  and  $\mathbf{R}_d$  can be expressed as a minimization between  $\mathbf{d}_r$  and  $\mathbf{d}_d$ . To be able to minimize the distance between the two vectors, let us define a tangent space shown in Fig. 2, tangent to  $\mathbf{d}_B$ . The basis of the tangent space, referred to as  $\mathbf{C} \in \mathbb{R}^{3 \times 2}$  can be computed as follows:

$$\mathbf{C} = \begin{bmatrix} \mathbf{c}_1 & \mathbf{c}_2 \end{bmatrix} \quad (12)$$

where  $\mathbf{c}_2 = \mathbf{d}_B \times \mathbf{e}_3$  and  $\mathbf{c}_1 = \mathbf{d}_B \times \mathbf{c}_2$ . In the degenerate case where  $\mathbf{d}_B = \mathbf{e}_3$ ,  $\mathbf{C}$  is chosen such that  $\mathbf{c}_1 = \mathbf{e}_1$  and  $\mathbf{c}_2 = \mathbf{e}_2$ . Let us denote by  $\mathbf{v} = [\delta x \ \delta y]$  as a vector in the tangent space defined by  $\mathbf{C}$ .  $\mathbf{v}$  represents any deviation along  $\mathbf{x}_B$  and  $\mathbf{y}_B$ . Let

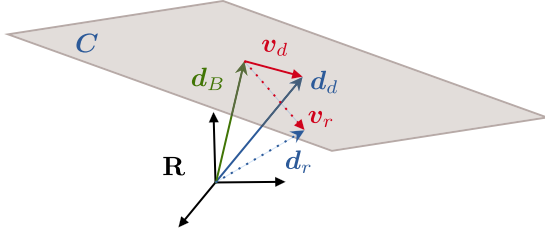


Fig. 2. Illustration of the tangent space defined by the basis  $\mathbf{C}$ , tangent to  $\mathbf{d}_B$ , and the corresponding transformations  $\mathbf{v}_r$  and  $\mathbf{v}_d$ .

us also denote by  $\mathbf{v}_r$  and  $\mathbf{v}_d$  as the deviations corresponding to  $\mathbf{d}_r$  and  $\mathbf{d}_d$ .

Since  $\mathbf{d}$  is defined as a unit vector, then following the angle-axis representation,  $\mathbf{d}$  can be written as follows:

$$\mathbf{d} = \frac{\sin(\theta)}{\theta} \mathbf{C}\mathbf{v} + \cos(\theta) \mathbf{d}_B \quad (13)$$

where  $\theta = \|\mathbf{C}\mathbf{v}\|$ . Let us assume  $\theta$  to represent a small change in the platform's orientation, and rewrite  $\mathbf{d}$  as follows:

$$\mathbf{d} = \mathbf{C}\mathbf{v} + \mathbf{d}_B \sqrt{1 - \|\mathbf{C}\mathbf{v}\|^2} \quad (14)$$

Note that  $\mathbf{v}$  represents a vector change in the platform's orientation. To find the corresponding angular change, let us denote by  $\mathbf{R}_e$  as the error rotation matrix such that  $\mathbf{R}_e = \mathbf{R}_B^T \mathbf{R}_d$ . The angular error about  $\mathcal{F}_B$  can be written  $\mathbf{e}_R = \mathbf{R}_e^\wedge$ , where  $[\ ]^\wedge$  is the inverse skew symmetric operator. The error rotation matrix can be written in the Rodrigues' form as follows:

$$\begin{aligned} \mathbf{k} &= \mathbf{d}_B \times \mathbf{d}_d \\ \theta &= \|\mathbf{k}\| \\ \mathbf{R}_e &= \mathbf{I}_3 + \frac{\sin(\theta)}{\theta} \mathbf{k}_\times + \frac{1 - \cos(\theta)}{\theta^2} (\mathbf{k}\mathbf{k}^T - \mathbf{I}_3) \end{aligned} \quad (15)$$

where  $\mathbf{I}_n$  is an  $n \times n$  identity matrix, and  $[\ ]_\times$  is the skew-symmetric operator.  $\mathbf{k} = \mathbf{d}_B \times (\mathbf{C}\mathbf{v}_d + \mathbf{d}_B \sqrt{1 - \|\mathbf{C}\mathbf{v}_d\|^2}) = \mathbf{d}_B \times (\mathbf{C}\mathbf{v}_d)$ . Since  $\mathbf{C}\mathbf{v}_d$  is orthogonal to  $\mathbf{d}_B$ , then  $\mathbf{k} = [-\delta y_d \ \delta x_d \ 0]^T$ . Following the small angle approximation made earlier, we can simplify (15) as follows:

$$\mathbf{e}_R = \mathbf{R}_e^\wedge = (\mathbf{I}_3 + \mathbf{k}_\times)^\wedge = \mathbf{k} = [-\delta y_d \ \delta x_d \ 0]^T \quad (16)$$

Let us update (17) for a computed  $\mathbf{v}_d$  as follows:

$$\begin{aligned} \mathbf{m}_d &= \boldsymbol{\omega} \times \mathbf{J}_B \boldsymbol{\omega} - \mathbf{K}_R \mathbf{e}_R - \mathbf{K}_w \mathbf{e}_w \\ \mathbf{e}_R &= [-\delta y_r \ \delta x_r \ e_z], \quad \mathbf{e}_w = \boldsymbol{\omega} \end{aligned} \quad (17)$$

where  $e_z$  is the error between  $\mathcal{F}_B$  and  $\mathcal{F}_{B_r}$  w.r.t.  $\mathbf{z}_B$  computed similar to [24]. Note that since  $e_z$  is computed directly from the error between  $\mathcal{F}_B$  and  $\mathcal{F}_{B_r}$ , the corresponding expected rotation is accommodated for in the computation of  $\mathbf{d}_B$  and  $\mathbf{d}_r$ , where  $\mathcal{F}_{B_r}$  is the reference body frame. Also, note that the desired angular velocity  $\boldsymbol{\omega}_d$  was not included in the error computation due to the difficulty of calculating its desired value for an online changing trajectory.

Following the above derivations, the optimization problem from (11) can be written as follows:

$$J = \min_{\mathbf{v}, \mathbf{x}} \frac{\alpha}{2} \|\mathbf{v}_r - \mathbf{v}\|^2 + \frac{\beta}{2} \|\mathbf{x}\|_Q^2 + \frac{\gamma}{2} \|\mathbf{u}_c - \mathbf{u}_c(t-1)\|^2 \quad (18)$$

$$s.t. \quad \|\mathbf{v}\|^2 \leq \min(\|\mathbf{v}_r\|^2, \bar{v}^2) \quad (19)$$

$$\begin{aligned} \mathbf{u}_c &= \mathbf{F}_1^\dagger \|\mathbf{f}_d^W\| (\mathbf{C}\mathbf{v} + \mathbf{d}_B \sqrt{1 - \|\mathbf{C}\mathbf{v}\|^2}) + \mathbf{F}_2^\dagger \mathbf{K}_R \begin{bmatrix} 0 & 1 \\ -1 & 0 \\ 0 & 0 \end{bmatrix} \mathbf{v} \\ &+ \mathbf{F}_2^\dagger (\boldsymbol{\omega} \times \mathbf{J}_B \boldsymbol{\omega} - \mathbf{K}_w \boldsymbol{\omega} - \mathbf{K}_R e_3 e_z) + \mathbf{B}\mathbf{x} \in \mathcal{U} \end{aligned} \quad (20)$$

where  $\alpha$ ,  $\beta$  and  $\gamma$  are the weighting terms of the multi-objective optimization problem and  $\bar{v}$  is the maximum allowed norm for  $\mathbf{v}$ . The goal of the regularization term  $\|\mathbf{u}_c - \mathbf{u}_c(t-1)\|^2$  in the objective function is to avoid large changes in control inputs between one control step and another. In addition, the constraint imposed on the norm of  $\mathbf{v}$  serves two objectives. First, it should be noted that the tangent space defined by (14) is only valid locally. As  $\|\mathbf{v}\|$  increases, norm of  $\mathbf{d}$  diverges from unity. As such, while optimizing the value of  $\mathbf{v}$ , (19) ensures the validity of the assumptions made to write (14). In addition, if  $\|\mathbf{v}_r\| < \bar{v}$ , the constraint from (19) ensures that the total rotation angle  $\theta = \|\mathbf{C}\mathbf{v}\|$  is chosen s.t.  $\theta \leq \theta_r$ . Gradients of the objective function and possible constraint functions are derived in the appendix of the technical report found in the following link [Technical Report].

It should be noted that the presented optimization algorithm is solved online at each time instance, and the ensuing optimal control input is commanded to the corresponding motors to actuate the platform.

a) *Note on convexity:* The problem in (18)–(20) is the dual of the following optimization problem:

$$J_{dual} = \min_{\mathbf{d}, \mathbf{x}} \frac{\alpha}{2} \|\mathbf{d}_r - \mathbf{d}\|^2 + \frac{\beta}{2} \|\mathbf{x}\|_Q^2 + \frac{\gamma}{2} \|\mathbf{u}_c - \mathbf{u}_c(t-1)\|^2 \quad (21)$$

$$s.t. \quad \|\mathbf{d}\|^2 = 1 \quad (22)$$

$$\begin{aligned} \mathbf{u}_c &= \mathbf{F}_1^\dagger \|\mathbf{f}_d^W\| \mathbf{d} + \mathbf{F}_2^\dagger \mathbf{S}(\mathbf{d}_B) \mathbf{d} + \mathbf{F}_2^\dagger (\boldsymbol{\omega} \times \mathbf{J}_B \boldsymbol{\omega} \\ &- \mathbf{K}_w \boldsymbol{\omega} - \mathbf{K}_R e_3 e_z) + \mathbf{B}\mathbf{x} \in \mathcal{U} \end{aligned} \quad (23)$$

First, it is assumed that a solution exists for the above problem, i.e., there exists a direction  $\mathbf{d}$  at which we can apply the desired force, and there exists a corresponding moment that rotates the platform from the current direction  $\mathbf{d}_R$  to the desired direction  $\mathbf{d}$ . The solution for the corresponding problem when a solution to  $J_{dual}$  does not exist is left as future work.

Let us write (23) as  $\mathbf{u}_c = \mathbf{A}\mathbf{d} + \mathbf{B}\mathbf{x} + \hat{\mathbf{u}}$ .

It is easy to prove that  $\mathbf{A}^T \mathbf{A}$  and  $\mathbf{B}^T \mathbf{B}$  are positive semidefinite. We omit the corresponding proof in this letter due to space limitations. In the case where  $\mathcal{U}$  represents linear (quadratic) inequality constraints,  $J_{dual}$  is a quadratic programming with linear (quadratic) constraints solved on the tangent space of the Riemannian manifold  $\|\mathbf{d}\| = 1$ . Since both types of quadratic

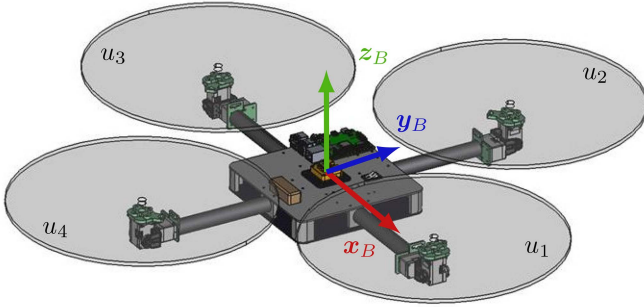


Fig. 3. CAD drawing showing a quadrotor where each propeller can be tilted independently in  $S^2$  constrained s.t. each propeller can generate thrust in a semi-sphere.

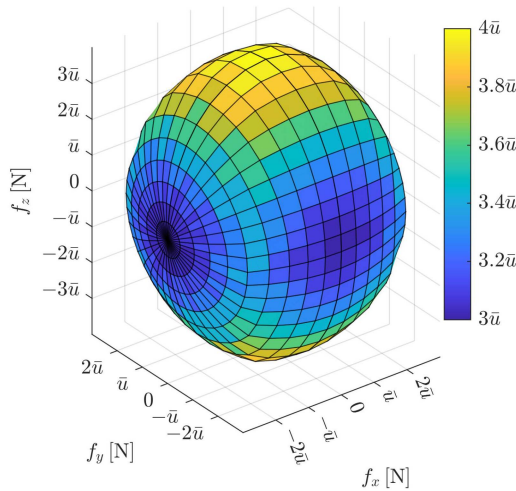


Fig. 4. Force set of the dual-tilt quadrotor shown in Fig. 3 while applying zero moment  $\mathfrak{F}(\mathbf{0})$ . This figure should be viewed in color.

programming optimizations are convex [25], and the optimization over the tangent space of a Riemannian manifold conserves convexity [26],  $J_{dual}$  is convex and correspondingly  $J$  is convex.

## V. EXPERIMENTAL RESULTS

### A. Experimental Setup

We validate the proposed approach in a realistic simulation environment implemented in MATLAB/Simulink. The environment simulates the platform shown in Fig. 3 with the corresponding actuator dynamics. The platform is a modification of the dual-tilt quadrotor platform from [7] with a higher thrust to weight ratio such that  $\bar{u}/m_{BG} = 1.05$ . With the current configuration, each propeller can generate thrust in a semi-sphere to avoid propellers' collision with the platform's arms. As such, the platform's constraints can be written as follows:

$$u_{x,1} \geq 0, u_{y,1} \geq 0 \quad (24)$$

$$u_{x,3} \leq 0, u_{y,4} \leq 0 \quad (25)$$

$$0 \leq \|\mathbf{u}_i\|^2 \leq \bar{u}_i^2 \quad \forall i = \{1, 2, 3, 4\} \quad (26)$$

where we assumed  $\bar{u} = 0$ . The force set of the platform while applying zero moment is shown in Fig. 4. It is easy to see

that, it is difficult to fit a cylinder or a cone into the force set (similar to [15], [16]) without either reducing the reachable force set, or breaking the actuation constraints. Similarly, while  $\mathfrak{F}(\mathbf{0})$  is similar in shape to an ellipsoid, it can be observed that the lateral diagonal force (directions such that  $\mathbf{f} = [\pm\|\mathbf{f}\|\sqrt{2}/2, \pm\|\mathbf{f}\|\sqrt{2}/2, 0]^T$ ) is larger than those along the  $x_B$  and  $y_B$  axes. As such, an algorithm that takes full advantage of the feasible control set could exploit such non-symmetries in the force set, which could be otherwise assumed unreachable when modeling the force set as a standard shape. Note that the corresponding force set changes depending on the applied moment.

Finally, we will demonstrate the generalizability of the proposed method to Fully Actuated (FA) platforms, where we will exhibit the trajectory tracking performance of the proposed algorithm when used on the Fully Actuated platform from [15]. The FA platform can apply forces and moments in  $\mathbb{R}^6$  with bounded lateral forces. As such, the FA platform has limited hovering orientations near the horizontal hovering direction.

### B. Algorithm Performance

We first study the performance of the proposed algorithm while applying the following trajectory:

$$\mathbf{q}(t_0) : \{\mathbf{p}_r(t_0) = [0 \ 0 \ 2]^T, \quad \mathbf{\Omega}(t_0) = [0 \ 0 \ 0]^T\}$$

$$\mathbf{q}(t = 20 \text{ s}) : \{\mathbf{p}_r(t_f) = [0 \ 0 \ 3]^T, \quad \mathbf{\Omega}(t_f) = [0 \ 90^\circ \ 0]^T\}$$

$$\mathbf{q}(t > 20 \text{ s}) = \mathbf{q}(t = 20 \text{ s}) \quad t \in [t_0 = 0 \text{ s}, t_f = 30 \text{ s}] \quad (27)$$

where  $\mathbf{\Omega}$  represents the euler angles of the platform, and where  $\mathbf{p}_r$  is represented in  $[m]$  and  $\mathbf{\Omega}$  is represented in degrees. This trajectory is feasible for the dual-tilt quadrotor. To demonstrate the performance of the proposed approach with unfeasible trajectories, we will analyze the performance of the platform after a fault, such that  $\bar{u}_{fault} = \bar{u}/3$ . This induced fault introduces unfeasible patches in the positive and negative  $x_B$  and  $y_B$  directions.

Fig. 5 shows the performance of the dual-tilt quadrotor in both cases while following the desired trajectory, where  $\theta$ ,  $\phi$ ,  $\psi$  correspond to the platform's orientation in Euler angle representation. While the desired position is tracked with a transient error, in both cases the platform converges to the reference position correctly. However, while the fully functional platform is able to track correctly the desired orientation, the faulty platform compromises the reference orientation to be able to track the reference position. As we can see from Fig. 4, the dual-tilt quadrotor can achieve hover at any orientation with all its propellers functional. On the other hand, while the faulty platform could generate forces to hover at the desired vertical orientation, it can only do so while applying zero-moment. As such, the optimization algorithm was not able to find a feasible control input that rotates the platform to hover at the desired orientation. In addition, as the force set is non-symmetrical, with larger force authority near the diagonals of the force set, while the desired trajectory required only a change in pitch, the optimizer increased the roll angle to be able to correspondingly increase the pitch angle without jeopardizing the desired force.

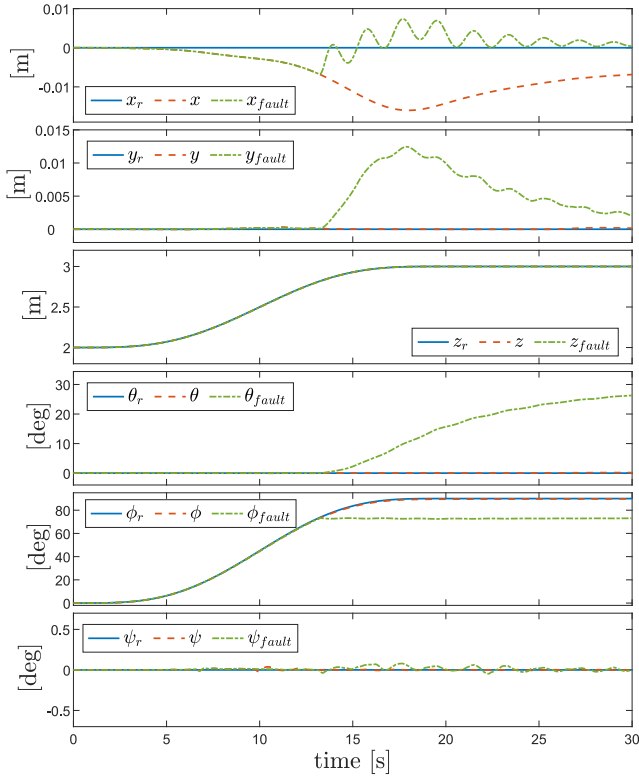


Fig. 5. Trajectory tracking performance of the dual-tilt quadrotor in normal mode and after propeller fault. The desired trajectory is shown in (27).

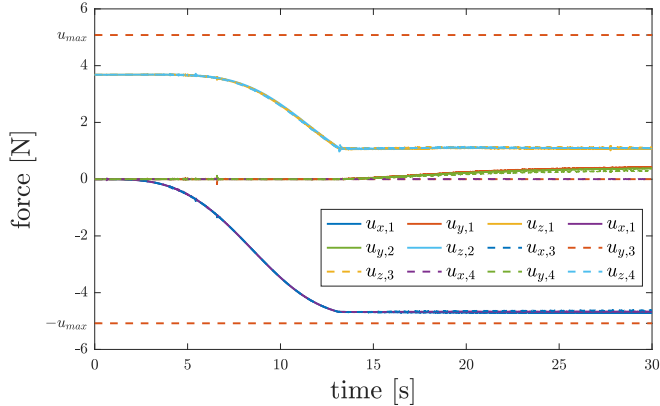


Fig. 6. Propeller commands of the faulty dual-tilt quadrotor while following the desired trajectory shown in (27).

Fig. 6 shows the propeller commands of the faulty platform while following the trajectory from (27). From this figure, we can see that in spite of the trajectory being very close to the actuation limits, the commanded control inputs are smooth, due to the regularization term in the optimization function.

Finally, note that the proposed algorithm runs in MATLAB/simulink at 750 [Hz].

### C. Circular Trajectory

One of the advantages of the Overactuated (OA) dual-tilt quadrotor is its ability to track a reference position while tracking an independent reference orientation. This advantage allows the

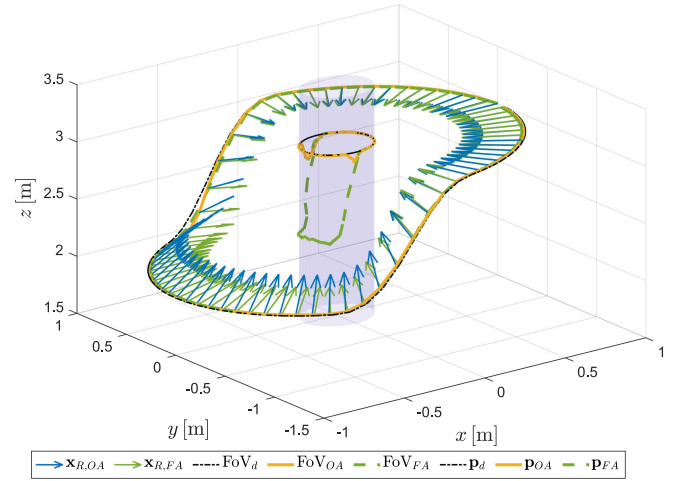


Fig. 7. Circular 6D trajectory tracking of the dual-tilt OA quadrotor and a FA hexarotor. The trajectory consists of circular trajectory around a cylinder, while changing platform height, and while pointing the platform's  $x_B$  towards a desired height along the cylinder.

platform, for example, to record a video of a desired object without jeopardizing its position. To highlight this ability, in this section we demonstrate the trajectory tracking performance of the platform while following a circular trajectory around a cylinder, while changing its altitude (for example to avoid a possible obstacle) and while continuously observing a desired line along the cylinder. To demonstrate the generalization of the proposed algorithm to different platforms, the same trajectory is also tracked by the FA platform discussed in Section V-A.

Fig. 7 shows the corresponding circular trajectory tracking performance of the OA and FA platforms. In this figure, we refer to the Field of View (FoV) as the center of the platform's field of view projected on the observed cylinder, and calculated as the projection of the platform's  $x_B$  onto the observed cylinder. From this figure we can first see that the proposed approach allows the FA platform to track the desired trajectory, while advantaging tracking of the desired position over the desired FoV. As such, once the desired pitch angle is large enough, the platform tracks the closest feasible pitch angle, while allocating feasible control inputs to track the desired position. On the other hand, we can see that the OA platform is able to track the desired trajectory, while simultaneously tracking the desired FoV. While following the desired trajectory, the OA platform was able to track the desired position with a maximum error of 0.0426 [m], while the FA platform tracked the desired trajectory with a maximum error of 0.0352 [m]. On the other hand, the OA platform was able to track the desired FoV with a maximum error of 0.0632 [m], while the FA platform had a significantly larger FoV tracking error of 0.6294 [m].

Note that in the case of the OA platform, and since the trajectory was not modified by the optimizer, it is possible to compute the FF term and to accommodate for the reference angular speed, calculated from the reference trajectory. Such modification can improve the tracking performance such that the maximum position tracking error is reduced to 0.0167 [m], and the FoV tracking error is reduced to 0.0201 [m]. However,

in Fig. 7 we show the tracking results of both platforms without the computation of the desired angular velocity and acceleration for consistency.

## VI. CONCLUSION

In this letter we proposed a novel optimization-based approach for the full-pose trajectory tracking of overactuated platforms with limited actuation abilities. The proposed approach runs in real-time, and allocates online feasible control inputs while modifying the reference attitude to satisfy actuation constraints. The proposed method was tested in a realistic simulation.

The simulative experiments, conducted on an overactuated and a fully actuated platform, show the ability of the proposed algorithm to allocate smooth control inputs trackable by the platform's dynamics. When the reference trajectory is feasible, the optimization method allocated control inputs to follow accurately the reference trajectory. On the other hand, when the reference trajectory is not feasible, the proposed method ensured accurate tracking of the reference position, while fully exploring the set of feasible forces to find the closest attitude to the reference one.

In the future, our team is working on manufacturing the platform shown in Fig. 3 to test the proposed algorithm to fly the manufactured drone.

In addition, we hope to expand the proposed approach in multiple directions: 1) to include the actuator dynamics into the optimization algorithm, and thus accommodate for the relatively slow dynamics of the servomotors; 2) to calculate online the desired angular velocities, and thus reduce the transient attitude tracking error; 3) to include vision requirements in the optimization loop, and thus track a desired field of view while modifying both the position and attitude parts of the reference trajectory.

## REFERENCES

- [1] M. Hamandi, F. Usai, Q. Sablé, N. Staub, M. Tognon, and A. Franchi, "Design of multirotor aerial vehicles: A taxonomy based on input allocation," *Int. J. Robot. Res.*, vol. 40, no. 8/9, pp. 1015–1044, 2021.
- [2] M. Tognon and A. Franchi, "Omnidirectional aerial vehicles with unidirectional thrusters: Theory, optimal design, and control," *IEEE Robot. Automat. Lett.*, vol. 3, no. 3, pp. 2277–2282, Jul. 2018.
- [3] S. Park et al., "ODAR: Aerial manipulation platform enabling omnidirectional wrench generation," *IEEE/ASME Trans. Mechatron.*, vol. 23, no. 4, pp. 1907–1918, Aug. 2018.
- [4] M. Zhao, T. Anzai, F. Shi, X. Chen, K. Okada, and M. Inaba, "Design, modeling, and control of an aerial robot DRAGON: A dual-rotor-embedded multilink robot with the ability of multi-degree-of-freedom aerial transformation," *IEEE Robot. Automat. Lett.*, vol. 3, no. 2, pp. 1176–1183, Apr. 2018.
- [5] D. Brescianini and R. D'Andrea, "An omni-directional multirotor vehicle," *Mechatronics*, vol. 55, pp. 76–93, 2018.
- [6] C. D. Pose, J. I. Giribet, and A. S. Ghersin, "Hexacopter fault tolerant actuator allocation analysis for optimal thrust," in *Proc. Int. Conf. Unmanned Aircr. Syst.*, 2017, pp. 663–671.
- [7] A. B. Junaid, A. D. De Cerio Sanchez, J. B. Bosch, N. Vitzilaios, and Y. Zweiri, "Design and implementation of a dual-axis tilting quadcopter," *Robotics*, vol. 7, no. 4, 2018, Art. no. 65.
- [8] M. Hamandi, K. Sawant, M. Tognon, and A. Franchi, "Omni-plus-seven (O7+): An omnidirectional aerial prototype with a minimal number of unidirectional thrusters," in *Proc. IEEE Int. Conf. Unmanned Aircr. Syst.*, 2020, pp. 754–761.
- [9] Y. Su, P. Yu, M. J. Gerber, L. Ruan, and T.-C. Tsao, "Nullspace-based control allocation of overactuated UAV platforms," *IEEE Robot. Automat. Lett.*, vol. 6, no. 4, pp. 8094–8101, Oct. 2021.
- [10] M. Ryll, H. Bühlhoff, and P. Giordano, "A novel overactuated quadrotor unmanned aerial vehicle: Modeling, control, and experimental validation," *IEEE Trans. Control Syst. Technol.*, vol. 23, no. 2, pp. 540–556, Mar. 2015.
- [11] G. Michieletto, M. Ryll, and A. Franchi, "Fundamental actuation properties of multi-rotors: Force-moment decoupling and fail-safe robustness," *IEEE Trans. Robot.*, vol. 34, no. 3, pp. 702–715, Jun. 2018.
- [12] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2011, pp. 2520–2525.
- [13] P. Zheng, X. Tan, B. B. Kocer, E. Yang, and M. Kovac, "TiltDrone: A fully-actuated tilting quadrotor platform," *IEEE Robot. Automat. Lett.*, vol. 5, no. 4, pp. 6845–6852, Oct. 2020.
- [14] D. Brescianini and R. D'Andrea, "Computationally efficient trajectory generation for fully actuated multirotor vehicles," *IEEE Trans. Robot.*, vol. 34, no. 3, pp. 555–571, Jun. 2018.
- [15] A. Franchi, R. Carli, D. Bicego, and M. Ryll, "Full-pose tracking control for aerial robotic systems with laterally-bounded input force," *IEEE Trans. Robot.*, vol. 34, no. 2, pp. 534–541, Apr. 2018.
- [16] D. Invernizzi and M. Lovera, "Trajectory tracking control of thrust-vectoring UAVs," *Automatica*, vol. 95, pp. 180–186, 2018.
- [17] D. Horla, M. Hamandi, W. Giernacki, and A. Franchi, "Optimal tuning of the lateral-dynamics parameters for aerial vehicles with bounded lateral force," *IEEE Robot. Automat. Lett.*, vol. 6, no. 2, pp. 3949–3955, Apr. 2021.
- [18] D. Bicego, J. Mazzetto, R. Carli, M. Farina, and A. Franchi, "Nonlinear model predictive control with enhanced actuator model for multi-rotor aerial vehicles with generic designs," *J. Intell. Robot. Syst.*, vol. 100, no. 3, pp. 1213–1247, 2020.
- [19] A. Romero, S. Sun, P. Foehn, and D. Scaramuzza, "Model predictive contouring control for time-optimal quadrotor flight," *IEEE Trans. Robot.*, vol. 38, no. 6, pp. 3340–3356, Dec. 2022.
- [20] G. Nava, Q. Sablé, M. Tognon, D. Pucci, and A. Franchi, "Direct force feedback control and online multi-task optimization for aerial manipulators," *IEEE Robot. Automat. Lett.*, vol. 5, no. 2, pp. 331–338, Apr. 2020.
- [21] T. A. Johansen and T. I. Fossen, "Control allocation—A survey," *Automatica*, vol. 49, no. 5, pp. 1087–1103, 2013.
- [22] M. Hamandi, L. Seneviratne, and Y. Zweiri, "Static hovering realization for multirotor aerial vehicles with tilttable propellers," *J. Mechanisms Robot.*, vol. 16, 2023, Art. no. 031004.
- [23] A. Boeuf, J. Cortés, R. Alami, and T. Siméon, "Enhancing sampling-based kinodynamic motion planning for quadrotors," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2015, pp. 2447–2452.
- [24] T. Lee, M. Leok, and N. H. McClamroch, "Geometric tracking control of a quadrotor UAV on SE(3)," in *Proc. IEEE 49th Conf. Decis. Control*, 2010, pp. 5420–5425.
- [25] C. J. Albers, F. Critchley, and J. C. Gower, "Quadratic minimisation problems in statistics," *J. Multivariate Anal.*, vol. 102, no. 3, pp. 698–713, 2011.
- [26] C. Udriste, *Convex Functions and Optimization Methods on Riemannian Manifolds*, vol. 297. Berlin, Germany: Springer, 1994.