



HAL
open science

Input Allocation for the Propeller-Based Overactuated Platform ROSPO

Michele Furci, Carlo Nainer, Luca Zaccarian, Antonio Franchi

► **To cite this version:**

Michele Furci, Carlo Nainer, Luca Zaccarian, Antonio Franchi. Input Allocation for the Propeller-Based Overactuated Platform ROSPO. IEEE Transactions on Control Systems Technology, 2020, 28 (6), pp.2720 - 2727. 10.1109/TCST.2019.2944341 . hal-02537332

HAL Id: hal-02537332

<https://laas.hal.science/hal-02537332>

Submitted on 8 Apr 2020

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.

Input Allocation for the Propeller-Based Overactuated Platform ROSPO

Michele Furci, Carlo Nainer, Luca Zaccarian, and Antonio Franchi

Abstract—We apply input allocation to a redundantly actuated platform driven by tilting aerodynamic propulsion units: the Rotor graSPing Omnidirectional (ROSPO). This platform represents a novel testbed for redundancy allocation designs in propeller driven platforms. The control solution is based on a hierarchical architecture, made of a high level controller for trajectory tracking, and a nonlinear input allocation algorithm. The algorithm exploits the input redundancy to take into account soft constraints associated to physical saturation limits of the actuators, and also induce reduced energy consumption. The actuator dynamics is fully taken into account in the framework and a rigorous proof of asymptotic tracking of time-varying references is guaranteed despite the impossibility of an instantaneous force execution. The experiments on the ROSPO platform clearly show the practicability and effectiveness of the proposed approach, as well as its scalability with different degrees of over-actuation levels.

Index Terms—Mobile Robots, Robot Control, Allocation, Over-actuated Robots, Hierarchical Control.

I. INTRODUCTION

In recent years researchers have tried to overcome the underactuation problem of collinear multirotor UAVs with the use of more generic designs and the addition of more actuators. Two extra servos are used in [1] to tilt the propellers and achieve a multi-directional total thrust force. The platform presented in [2] has six propeller fixedly tilted and is able to generate a total thrust force in many directions by only changing the rotational speed of the propellers. The superiority of this platform, as compared to standard collinear multirotors, has been shown in [3] for tasks involving physical interaction and precise control of the exerted wrench. In [4] the authors propose to add a single servo motor to the design of [2] in order to obtain an over-actuated platform that can tilt all the propellers in a coordinated way. This additional degree of freedom is used to find the best trade-off between multi-directional total thrust and energy efficiency. Finally, the design proposed in [5] has four propellers plus four servo motors to tilt each propeller independently and thus attain an omnidirectional total thrust and over-actuation that is used to minimize the energy consumption. Six or more propellers that can reverse the rotation direction and produce negative lift have been used in [6] and [7] in order to achieve an omnidirectional total thrust force as well.

M. Furci, C. Nainer, L. Zaccarian and A. Franchi are with LAAS-CNRS, Université de Toulouse, CNRS, Toulouse, France. L. Zaccarian is also with DII, University of Trento, Italy. (e-mail: mfurci@laas.fr | cnainer@laas.fr | zaccarian@laas.fr | antonio.franchi@laas.fr)

This work has been partially funded by the European Union's Horizon 2020 research and innovation programme under grant agreement No 644271 AEROARMS and by the CARNOT post-doctoral project GRASP supported by the LAAS-CNRS.

Each one of the aforementioned works assumes that the system inputs (the propeller speed, the tilt angle, or both) are ideal, i.e., that there is no actuator dynamics and the bandwidth is unlimited. However such limitations are present and cannot be neglected if agile maneuvers are to be obtained. Furthermore, the presence of additional actuators allows exploiting redundant input directions in order to produce a maximally effective action on the system. (see, e.g., the surveys in [8] and [9]).

In this paper we consider a platform, called Rotor graSPing Omnidirectional (ROSPO), which mimics the main dynamic features of overactuated multirotor systems. The ROSPO is also suitable for the emulation of cooperative manipulation of a large load by means of several robots, see e.g., [10], [11], [12]. The ROSPO is composed by a single frame with a variable number of actuation modules, each comprising an orientable turret and a motor-propeller actuator. With respect to a MAV, the ROSPO represents a simplification because its dynamics evolves in the 2D plane and therefore it has only 3 degrees of freedom when compared to the 6 degrees of freedom of a rigid body in the 3D space. Nevertheless, it well represents the input redundancy challenges and to the non-ideal dynamics of the actuators employed in MAVs, such as propellers and servomotors.

We develop here and experimentally test a control paradigm fusing a high-level controller with a static input allocation technique presented in [13]. Our solution is able to mitigate part of the issues arising from the actuation limits by exploiting the redundancy. However we do not directly address the saturation problem as we treat those limits as soft constraints rather than hard ones. In particular, we focus here on steady-state saturations, so that a natural extension of our work would consist of designing anti-windup mechanisms for rigorously dealing with inevitable transient saturation problems (some preliminary results are discussed in [14]). Our method is scalable and its effectiveness is tested in the presence of different actuator configurations. A control law for a similar platform, but without the over-actuation, was proposed by [15]. The over-actuation on 2D motions is instead a well studied problem for ships and vessels as in [16] and is comprehensively discussed in [17], [18]. A preliminary conference version of this paper has been presented in [19], where only simulation results have been shown on a three-turrets configuration. The main differences with respect to that conference version are: 1) the presentation of the overall method is streamlined and improved; 2) we provide a more detailed level of formalism; 3) we present the design and realization of the experimental ROSPO platform; 4) we present the implementation and testing with real experiments of the proposed method on

the platform with both three and four turrets, showing the scalability and the experimental effectiveness of our solution.

The paper is organized as follows: Section II presents the platform and its dynamical model. Section III describes the control architecture for the system, and Section IV presents the experiments on the ROSPO platform.

II. MECHATRONICS DESIGN AND MODEL

A. Mechatronics Design

The ROSPO platform (in Figure 1) is composed by a rigid rectangular aluminum frame sustained by omnidirectional passive spherical wheels (“ball transfer unit”) allowing for 2D movements with relatively low friction. On top of the frame, a variable number of actuator modules can be mounted, to allow for a flexible configuration. An actuator module consists of a rotating turret carrying a propeller driven by a BLDC motor, as shown in Figure 2. The actuator modules can be attached virtually anywhere and in arbitrary number.

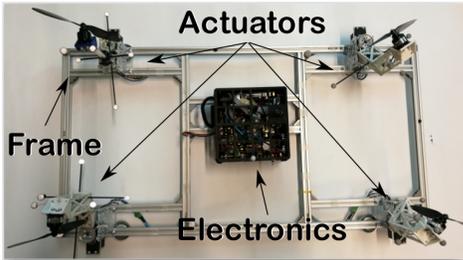


Fig. 1. The ROSPO platform in a four-turrets configuration.

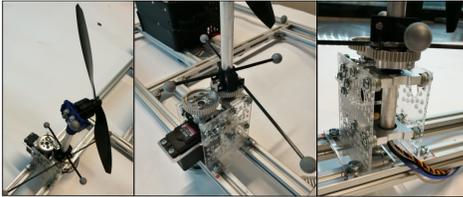


Fig. 2. Details of one actuator module of the ROSPO.

Each module generates a force, whose intensity and direction in the plane can be changed by varying the speed of the propeller and the orientation of the turret, respectively. The turret is driven by a multi-turn servo motor, allowing for multiple complete rotations of the turret. The rotations are however also limited by the winding of the power cables of the motors. At the center of the platform the on-board electronics is placed, as well as the battery to power the system. The electronics include: an autopilot board, a board with 6 Electronic Speed Controllers (ESC), an Arduino board, and an ODROID single board computer. The autopilot board includes an IMU that allows retrieving the angular velocities from the gyroscopes and the accelerations from the accelerometers. The ESC board allows to control in a closed-loop fashion (with a refresh rate of 1000 kHz) the speed of the BLDC motors, with a spinning frequency error of ± 1.5 Hz. The Arduino board runs the servo control algorithm and generates suitable reference signals for the rotation of the turrets. Finally, the ODROID computer is in

charge of running the control algorithm present in the paper, the high-level software architecture, the wi-fi communication, the motion capture interface and the estimation filters.

For the experiments, two sample configurations were used, one involving three turrets and a second one involving four turrets. Infrared reflective markers were placed on the structure and the turrets, for the motion capture system to retrieve the position and orientation of the platform and turrets at 100 Hz, to be used in the state estimation filters.

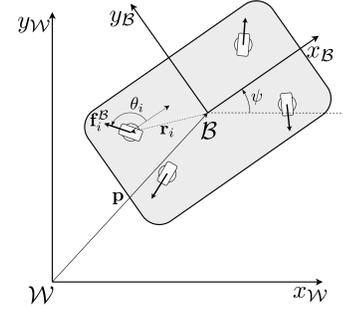


Fig. 3. Schematic representation of the ROSPO platform in the four-turrets configuration, and main symbol definitions.

B. Equations of Motion

Figure 3 shows the ROSPO platform schematic representation including the main symbol definitions. To describe the dynamical model of the platform we have to define two reference systems \mathcal{W} and \mathcal{B} . \mathcal{W} is the fixed ‘world’ inertial frame while \mathcal{B} is the *body frame* whose origin is the Center of Mass (CoM) of the platform and whose axes are aligned with the platform itself. The system dynamics can be defined by the following differential equations:

$$\begin{cases} \dot{\mathbf{p}} &= \mathbf{v} \\ m\dot{\mathbf{v}} &= \sum_{i=1}^n \mathbf{R}(\psi)\mathbf{f}_i^B + \mathbf{f}_{r,p}(\mathbf{v}) \\ \dot{\psi} &= \omega \\ j\dot{\omega} &= \sum_{i=1}^n (\mathbf{\Pi}\mathbf{r}_i)^T \mathbf{f}_i^B + f_{r,\psi}(\omega) \end{cases} \quad (1a)$$

$$\begin{cases} \dot{\theta}_i &= u_{\theta,i} \\ \dot{w}_i &= u_{w,i} \\ \mathbf{f}_i^B &= k_w w_i^2 \begin{bmatrix} \cos(\theta_i) \\ \sin(\theta_i) \end{bmatrix} \end{cases} \quad \text{for } i = 1, \dots, n \quad (1b)$$

where m and j are constant scalars representing the total mass and moment of inertia of the system, $\mathbf{p} = [p_x \ p_y]^T \in \mathbb{R}^2$ is the position of the center of mass of the system in \mathcal{W} , $\mathbf{v} \in \mathbb{R}^2$ the velocity of the center of mass of the system in \mathcal{W} , $\psi \in \mathbb{R}$ is the yaw angle (attitude) of the system defined as the angle between the x axes of \mathcal{B} and \mathcal{W} , ω is the angular velocity of the platform, $\mathbf{R}(\psi) = \begin{bmatrix} \cos(\psi) & -\sin(\psi) \\ \sin(\psi) & \cos(\psi) \end{bmatrix} \in SO(2)$ is the rotation matrix transforming the orientation from \mathcal{B} to \mathcal{W} , $\mathbf{f}_i^B \in \mathbb{R}^2$ are the forces produced by each actuator module, expressed in \mathcal{B} , $\mathbf{f}_{r,p}$ and $f_{r,\psi}$ the friction forces and torque respectively, $\mathbf{r}_i \in \mathbb{R}^2$ are the positions of each actuator module expressed in \mathcal{B} , $\mathbf{\Pi} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$, θ_i is the angle of the the i -th actuator module expressed in \mathcal{B} , defined as the angle between the produced force and the x axis of \mathcal{B} , $u_{\theta,i}$ is the spinning rate of the i -th actuator angle, w_i is the spinning rate of the propeller of

each actuator module, $u_{w,i}$ is the spinning acceleration, with $i = 1, \dots, n$ and $k_w > 0$ is a constant aerodynamic parameter. Equations (1a) can be referred to as the platform model, while equations (1b) as the actuators' dynamics. The reaction torque induced on the the platform dynamics by the turret actuation is neglected because of its negligible effect (the two inertias are roughly 2 orders of magnitude apart).

Note that we wrote the actuator dynamics (1b) by recognizing that an instantaneous variation of θ_i or w_i is not possible. Therefore, it makes sense to consider $\dot{\theta}_i = u_{\theta,i}$ and $\dot{w}_i = u_{w,i}$ as the actual inputs. With this strategy we may well represent the rate limitations of the actuators as input constraints on $u_{\theta,i}$, $u_{w,i}$ and their maximum excursions as state constraints on θ_i and w_i . In particular we enforce:

$$\begin{aligned} \underline{\theta}_i \leq \theta_i \leq \overline{\theta}_i & \quad \underline{w}_i \leq w_i \leq \overline{w}_i \\ \underline{u_{\theta,i}} \leq u_{\theta,i} \leq \overline{u_{\theta,i}} & \quad \underline{u_{w,i}} \leq u_{w,i} \leq \overline{u_{w,i}}, \end{aligned} \quad (2)$$

where all the underlined and overlined terms are constant scalar values. Constraints (2) imply that the spinning rates of the propellers and the angle of the actuator turrets are both limited, as well as the inputs to control them (due to electro-mechanical and power limitations of the actuators). The angle of each turret is limited due to servo-motors limitations, while the maximum propeller speed is limited for power and dissipation reasons. Finally the minimum propeller speed is due to the fact that Electronic Speed Controllers (ESC) without hall sensors (namely most of the commercial ESC) do not perform well at low speed.

Equation (1) can be written in a more compact way as:

$$\begin{cases} m\dot{\mathbf{v}} &= \mathbf{R}(\psi)\mathbf{f} + \mathbf{f}_{r,p}(\mathbf{v}) \\ j\dot{\omega} &= \tau + f_{r,\psi}(\omega) \\ \dot{\theta}_i &= u_{\theta,i} \\ \dot{w}_i &= u_{w,i} \\ \mathbf{f}_i^B &= k_w w_i^2 \begin{bmatrix} \cos(\theta_i) \\ \sin(\theta_i) \end{bmatrix}, \end{cases} \quad \text{for } i = 1, \dots, n \quad (3)$$

where $\mathbf{f} = \sum_{i=1}^n \mathbf{f}_i^B \in \mathbb{R}^2$ and $\tau = \sum_{i=1}^n (\mathbf{I}\mathbf{r}_i)^T \mathbf{f}_i^B \in \mathbb{R}$ can be well understood as the total force and torque applied to the system by the actuators.

C. Friction forces

Particular attention should be given to the modeling of non-negligible friction forces and torques $\mathbf{f}_{r,p}$ and $f_{r,\psi}$ that must be compensated for in the control law. The friction is mainly generated by the four ball-transfer-unit attached to the base, to allow for omni-directional movements. Based on [20], we use here the differentiable friction force model:

$$\mu(s) = \gamma_1(\tanh(\gamma_2 s) - \tanh(\gamma_3 s)) + \gamma_4 \tanh(\gamma_5 s) + \gamma_6 s \quad (4)$$

where $\gamma_1, \gamma_2, \gamma_3, \gamma_4, \gamma_5, \gamma_6$ are constant coefficients to be identified and s is the scalar velocity. As discussed in [20], despite the absence of an actual discontinuity, static friction can be approximated by $\gamma_1 + \gamma_4$, the Stribeck effect is captured by the term $\tanh(\gamma_2 s) - \tanh(\gamma_3 s)$, Coulomb friction is modeled by coefficients γ_4 and γ_5 and viscous dissipation is associated to coefficient γ_6 . Directly applying (4) to the two components of \mathbf{v} results in an imprecise model, because the friction

force should always be opposite to the velocity vector \mathbf{v} . We then propose the following modification, preserving the differentiability property when $\mathbf{v} \neq \mathbf{0}$:

$$\mathbf{f}_{r,p}(\mathbf{v}) = \begin{cases} \mu(|\mathbf{v}|) \frac{\mathbf{v}}{|\mathbf{v}|}, & \text{if } \mathbf{v} \neq \mathbf{0} \\ \mathbf{0}, & \text{if } \mathbf{v} = \mathbf{0} \end{cases} \quad (5)$$

On the other hand, for the rotational friction we may use:

$$f_{r,\psi}(\omega) = d\mu(\omega) \quad (6)$$

where d is the distance of the wheels from the center of gravity (CoG), supposing that the four wheels are placed along a circle of radius d around the CoG.

The time derivative of (5) for $\mathbf{v} \neq \mathbf{0}$ is computed as follows by using $\frac{d}{dt} \frac{\mathbf{v}}{|\mathbf{v}|} = \frac{\dot{\mathbf{v}}}{|\mathbf{v}|} + \frac{d}{dt} \mathbf{v}(\mathbf{v}^T \mathbf{v})^{-1/2} = \left(I - \frac{\mathbf{v}\mathbf{v}^T}{\mathbf{v}^T \mathbf{v}} \right) \frac{\dot{\mathbf{v}}}{|\mathbf{v}|}$ and $\frac{d}{dt} \mu((\mathbf{v}^T \mathbf{v})^{1/2}) = \mu'(|v|) \frac{\dot{\mathbf{v}}}{|\mathbf{v}|} \mathbf{v}^T$:

$$\dot{\mathbf{f}}_{r,p}(\mathbf{v}) = \left(\left(I - \frac{\mathbf{v}\mathbf{v}^T}{\mathbf{v}^T \mathbf{v}} \right) \mu(|\mathbf{v}|) + \frac{\mathbf{v}\mathbf{v}^T}{|\mathbf{v}|} \mu'(|\mathbf{v}|) \right) \frac{\dot{\mathbf{v}}}{|\mathbf{v}|}, \quad (7)$$

where μ' is the derivative of μ . In our practical implementation, for small values of $|\mathbf{v}|$ we have implemented a smooth dead-zone, as customarily done in these cases to avoid problems induced by the velocity measurement noise.

Disregarding the Stribeck effect (coefficients $\gamma_1, \gamma_2, \gamma_3$ are set to zero) and viscous dissipation, which is negligible at the considered speed (coefficient γ_6 is set to zero), we may account for Coulomb friction by properly tuning coefficients γ_4, γ_5 within the simplified model

$$\mu(s) = \gamma_4 \tanh(\gamma_5 s) \quad (8)$$

$$\mu'(s) = \gamma_4 (1 - \tanh^2(\gamma_5 s)) \gamma_5. \quad (9)$$

Finally, the time derivative of expression (6) is given by:

$$\dot{f}_{r,\psi}(\omega) = d\gamma_4 (1 - \tanh^2(\gamma_5 \omega)) \gamma_5 \dot{\omega}. \quad (10)$$

III. CONTROL ARCHITECTURE

We design a hierarchical control architecture with a high level unit providing trajectory tracking of the three degrees of freedom of the ROSPO, and a low level control allocation unit that handles the over-actuation arising from having $n \geq 2$ turrets. As shown in Figure 4, the high level controller specifies a desired wrench $\mathbf{u}_{v,c}$ to be applied to the system's center of mass, while the low level allocation action implements such a request on the actual wrench (\mathbf{f}, τ) while taking into account a soft version of constraints (2). With this allocation strategy, the high level controller perceives the actuators and the allocator as a first-order linear filter (whose eigenvalue is a design parameter), making it easy to consider this virtual dynamics in the design of the high-level controller.

A. Static Allocation

Based on [13], for the control allocation algorithm, consider a virtual control $\mathbf{u}_v \in \mathbb{R}^{n_{uv}}$, a number of redundant actuators $n_a > n_{uv}$ and a commanded virtual control $\mathbf{u}_{v,c} \in \mathbb{R}^{n_{uv}}$ coming from a high level controller. The actuators should obey a first-order (possibly coupled) dynamics:

$$\dot{\mathbf{x}}_a = \mathbf{f}(\mathbf{x}_a) + \mathbf{g}(\mathbf{x}_a)\mathbf{u}, \quad \mathbf{u}_v = \mathbf{h}(\mathbf{x}_a) \quad (11)$$

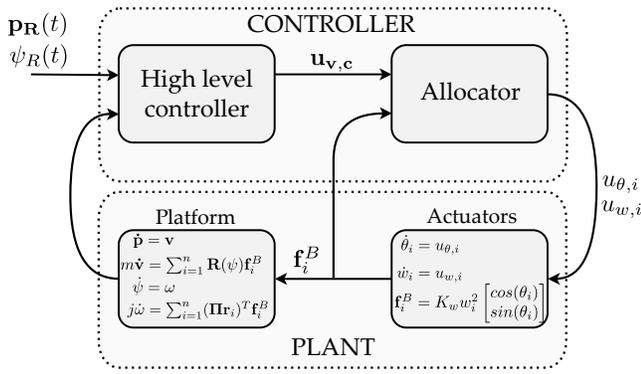


Fig. 4. The hierarchical control architecture comprising the high level controller and the low level allocator.

where \mathbf{x}_a are the actuators' states, $\mathbf{f} : \mathbb{R}^{n_a} \rightarrow \mathbb{R}^{n_a}$ represents the actuator dynamics, $\mathbf{g} : \mathbb{R}^{n_a} \rightarrow \mathbb{R}^{n_a \times n_a}$ maps the commanded virtual control to the actuator states and $\mathbf{h} : \mathbb{R}^{n_a} \rightarrow \mathbb{R}^{n_{uv}}$ maps the actuator states to the virtual control. For the ROSPO and from (1), we have $n_a = 2n$ and $n_{uv} = 3$ with selections

$$\mathbf{x}_a = [w_1 \ \theta_1 \ \cdots \ w_n \ \theta_n]^T \quad (12a)$$

$$\mathbf{u} = [u_{w,1} \ u_{\theta,1} \ \cdots \ u_{w,n} \ u_{\theta,n}]^T \quad (12b)$$

$$\mathbf{f}(\mathbf{x}_a) = \mathbf{0}, \quad \mathbf{g}(\mathbf{x}_a) = \mathbf{I} \quad (12c)$$

$$\mathbf{h}(\mathbf{x}_a) = k_w \begin{bmatrix} \sum_{i=1}^n w_i^2 \begin{bmatrix} \cos(\theta_i) \\ \sin(\theta_i) \end{bmatrix} \\ \sum_{i=1}^n (\mathbf{\Pi} \mathbf{r}_i)^T w_i^2 \begin{bmatrix} \cos(\theta_i) \\ \sin(\theta_i) \end{bmatrix} \end{bmatrix}. \quad (12d)$$

showing that maps \mathbf{f} and \mathbf{g} are linear, while the challenging nonlinearity resides in the output function \mathbf{h} .

The allocator generates \mathbf{u} such that $\lim_{t \rightarrow \infty} \|\mathbf{u}_v - \mathbf{u}_{v,c}\| = 0$, while using the gradient of a cost function $J : \mathbb{R}^{n_a} \rightarrow \mathbb{R}$ arising from a soft version of constraints (2), taking into account via suitable penalty terms. In particular, the static allocation control is given by:

$$\mathbf{u} = \mathbf{g}(\mathbf{x}_a)^{-1} (-\mathbf{f}(\mathbf{x}_a) + \mathbf{u}_v - \mathbf{u}_J) = \mathbf{u}_y - \mathbf{u}_J \quad (13a)$$

$$\mathbf{u}_y = \gamma_P \nabla \mathbf{h}(\mathbf{x}_a) \left(\nabla \mathbf{h}(\mathbf{x}_a)^T \nabla \mathbf{h}(\mathbf{x}_a) \right)^{-1} (\mathbf{u}_{v,c} - \mathbf{h}(\mathbf{x}_a))$$

$$\mathbf{u}_J = \gamma_J \nabla_{\perp} \mathbf{h}(\mathbf{x}_a) \nabla J(\mathbf{x}_a)$$

with $\nabla(\cdot)$ indicating the gradient of a function and $\nabla_{\perp} \mathbf{h}(\mathbf{x}_a)$ being the following projection matrix:

$$\nabla_{\perp} \mathbf{h}(\mathbf{x}_a) = \mathbf{I} - \nabla \mathbf{h}(\mathbf{x}_a) \left(\nabla \mathbf{h}(\mathbf{x}_a)^T \nabla \mathbf{h}(\mathbf{x}_a) \right)^{-1} \nabla \mathbf{h}(\mathbf{x}_a)^T. \quad (13b)$$

The two terms \mathbf{u}_y and \mathbf{u}_J guarantee respectively the regulation condition and the optimality condition. The positive gain $\gamma_P \in \mathbb{R}^+$ is used to adjust the speed of the linear first order filter governing the dynamical relationship between $\mathbf{u}_{v,c}$ and \mathbf{u}_c . Instead, gain $\gamma_J \in \mathbb{R}^+$ is used to adjust the speed of the allocation term that minimizes cost J . Based on (2), we intuitively select the cost function J as

$$J = \sum_{i=0}^n \left(\mu_1 \left(\frac{\tilde{w}_i}{\Delta_{w,i}} \right)^6 + \mu_2 \left(\frac{\tilde{\theta}_i}{\Delta_{\theta,i}} \right)^6 + \mu_w w_i^2 \right), \quad (14)$$

with:

$$\begin{aligned} \tilde{w}_i &= w_i - w_{m,i} & \tilde{\theta}_i &= \theta_i - \theta_{m,i} \\ \Delta_{w,i} &= \bar{w}_i - \underline{w}_i & \Delta_{\theta,i} &= \bar{\theta}_i - \underline{\theta}_i \\ w_{m,i} &= \frac{\bar{w}_i + \underline{w}_i}{2} & \theta_{m,i} &= \frac{\bar{\theta}_i + \underline{\theta}_i}{2} \end{aligned} \quad (15)$$

and where μ_1, μ_2, μ_w are positive scalar weights.

The first two terms in (14) impose a soft version of the actuator constraints (2) focusing on steady-state saturation effects, but not fully addressing and solving a hard version of these constraints. This more challenging goal would require additional anti-windup mechanisms to be added on top of our solution (see [14] for some preliminary results). In this paper we provide theoretical results about the *unconstrained* dynamics (1) controlled by (13) essentially disregarding the transient saturation effects. The third term in (14) penalizes high rotational speeds of the propellers, to reduce the energy consumption of the platform.

The following lemma establishes two desirable properties of the allocation scheme, comprising first-order linear dynamics between the commanded virtual control and the actual virtual control, and asymptotic convergence to the optimal allocation, as established by cost J . Its proof is omitted because it is an immediate consequence of the results in [13].

Lemma 1: If $\text{rank}(\nabla \mathbf{h}(\mathbf{x}_a)) = 3$, then system (1b) augmented with allocator (13a)-(12d) is such that:

- 1) for each initial condition $\theta_i(0), w_i(0)$ and virtual commanded input $t \mapsto \mathbf{u}_{v,c}(t)$, the virtual control satisfies:

$$\dot{\mathbf{u}}_v = -\gamma_P (-\mathbf{u}_v + \mathbf{u}_{v,c})$$

- 2) for any constant selection of $\mathbf{u}_{v,c}$, cost J defined in (14) converges to the constrained minimum J^* :

$$J^* = \min_{\mathbf{x}_a \in \mathbb{R}^{n_a}} J(\mathbf{x}_a), \quad \text{subject to } \mathbf{h}(\mathbf{x}_a) = \mathbf{u}_{v,c}.$$

Since Lemma 1 requires $\text{rank}(\nabla \mathbf{h}(\mathbf{x}_a)) = 3$, we prove below that, as long as $n \geq 2$ (i.e., there is some level of redundancy), this condition is verified as long as the turrets are placed in different positions.

Lemma 2: If $\mathbf{r}_i \neq \mathbf{r}_j$ for all index pairs $i \neq j$ in $\{1, \dots, n\}$ and there exist indexes h, k in $\{1, \dots, n\}$ such that $w_h \neq 0, w_k \neq 0$, then function \mathbf{h} in (12d) satisfies $\text{rank}(\nabla \mathbf{h}(\mathbf{x}_a)) = 3$.

Proof: Note that $\nabla \mathbf{h}$ is the stacking of rows $k_w \mathbf{W}_i^T \mathbf{T}_i := k_w \begin{bmatrix} 2w_i & 0 \\ 0 & w_i^2 \end{bmatrix} \begin{bmatrix} \cos(\theta_i) & \sin(\theta_i) & r_{y_i} \cos(\theta_i) - r_{x_i} \sin(\theta_i) \\ -\sin(\theta_i) & \cos(\theta_i) & -r_{y_i} \sin(\theta_i) - r_{x_i} \cos(\theta_i) \end{bmatrix}$, for $i = 1, \dots, n$. Let us only consider a selection \mathbf{M} consisting in the four rows of $\nabla \mathbf{h}$ associated to the nonzero speeds w_h, w_k . Clearly $\text{rank}(\mathbf{M}) \leq \text{rank}(\nabla \mathbf{h})$. In addition, $k_w > 0$ and $w_k \neq 0, w_h \neq 0$ imply that $\text{rank}(\mathbf{M}) = \text{rank}(\overline{\mathbf{M}})$, where $\overline{\mathbf{M}} := \begin{bmatrix} \mathbf{T}_h \\ \mathbf{T}_k \end{bmatrix}$. After some calculations, we get

$$\overline{\mathbf{M}}^T \overline{\mathbf{M}} = \begin{bmatrix} 2 & 0 & \mathbf{r}_{y_h} + \mathbf{r}_{y_k} \\ 0 & 2 & -\mathbf{r}_{x_h} - \mathbf{r}_{x_k} \\ \mathbf{r}_{y_h} + \mathbf{r}_{y_k} & -\mathbf{r}_{x_h} - \mathbf{r}_{x_k} & |\mathbf{r}_h|^2 + |\mathbf{r}_k|^2 \end{bmatrix},$$

whose determinant corresponds to $2(\mathbf{r}_{x_h} - \mathbf{r}_{x_k})^2 + 2(\mathbf{r}_{y_h} - \mathbf{r}_{y_k})^2$, which is clearly zero if and only if $\mathbf{r}_h = \mathbf{r}_k$. ■

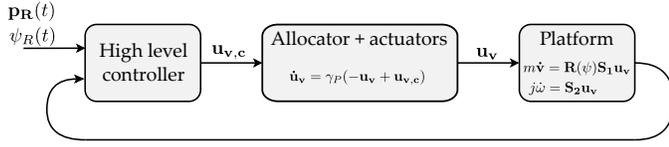


Fig. 5. Equivalent first-order linear dynamics seen by the high level controller due to the hierarchical architecture.

B. High Level Controller

Based on item (1) of Lemma 1, the allocator and the actuators can be considered by the high level controller as a first order linear filter having a pole in $-\gamma_P$. The equivalent system is represented in Figure 5 and can be written as:

$$\begin{aligned} m\dot{\mathbf{v}} &= \mathbf{R}(\psi)\mathbf{S}_1\mathbf{u}_v + \mathbf{f}_{r,p}(\mathbf{v}) \\ j\dot{\boldsymbol{\omega}} &= \mathbf{S}_2\mathbf{u}_v + f_{r,\psi}(\boldsymbol{\omega}) \\ \dot{\mathbf{u}}_v &= \gamma_P(-\mathbf{u}_v + \mathbf{u}_{v,c}), \end{aligned} \quad (16)$$

where $\mathbf{u}_{v,c} \in \mathbb{R}^3$, $\mathbf{u}_v \in \mathbb{R}^3$ are the states of the first order linear filter, $\mathbf{S}_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$, $\mathbf{S}_2 = [0 \ 0 \ 1]$ and functions $\mathbf{f}_{r,p}(\mathbf{v})$, $f_{r,\psi}(\boldsymbol{\omega})$ represent friction effect as defined in Section II-C.

The goal is to design a high level controller that stabilizes (16) for a trajectory tracking purpose. The desired trajectory can be expressed as $\mathbf{p}_R(t), \dot{\mathbf{p}}_R(t), \dots, \mathbf{p}_R^{(n)}(t)$, $\psi_R(t), \dot{\psi}_R(t), \dots, \psi_R^{(n)}(t)$, where $\mathbf{p}_R(t) \in \mathbb{R}^2$ is a desired position trajectory in time, $\psi_R(t) \in \mathbb{R}$ a desired attitude trajectory and the subsequent functions are their time derivatives. We define the tracking error variables as:

$$\begin{aligned} \tilde{\mathbf{p}}(t) &= \mathbf{p}(t) - \mathbf{p}_R(t) \\ \tilde{\psi}(t) &= \psi(t) - \psi_R(t). \end{aligned} \quad (17)$$

The goal of the high-level controller is to ensure $\lim_{t \rightarrow \infty} \tilde{\mathbf{p}} = 0$ and $\lim_{t \rightarrow \infty} \tilde{\psi} = 0$, namely the trajectory converges to the desired path.

To this end, we propose the following selection:

$$\mathbf{u}_{v,c} = \mathbf{B}^{-1}(\dot{\mathbf{u}}_v^* - \mathbf{A}\mathbf{u}_v^*) - \mathbf{K}\tilde{\mathbf{u}}_v \quad (18)$$

where $\mathbf{A} = -\gamma_P\mathbf{I}_3$, $\mathbf{B} = \gamma_P\mathbf{I}_3$ and $\mathbf{K} \in \mathbb{R}^{3 \times 3}$ is a gain matrix such that $\mathbf{A} - \mathbf{BK}$ is Hurwitz. Selection (18) uses the feedforward signal

$$\mathbf{u}_v^* = \begin{bmatrix} m\mathbf{R}^T(\psi) \left(-k_P\tilde{\mathbf{p}} - k_D\dot{\tilde{\mathbf{p}}} + \ddot{\mathbf{p}}_R - \frac{\mathbf{f}_{r,p}(\mathbf{v})}{m} \right) \\ j \left(-k_{P,\psi}\tilde{\psi} - k_{D,\psi}\dot{\tilde{\psi}} + \ddot{\psi}_R - \frac{f_{r,\psi}(\boldsymbol{\omega})}{j} \right) \end{bmatrix} \quad (19)$$

with $\tilde{\mathbf{u}}_v = \mathbf{u}_v - \mathbf{u}_v^*$ and where the four positive gains, $k_P \in \mathbb{R}^{>0}$, $k_D \in \mathbb{R}^{>0}$, $k_{P,\psi} \in \mathbb{R}^{>0}$, $k_{D,\psi} \in \mathbb{R}^{>0}$, assign a PD like behavior of the translational and rotational dynamics.

Controller (18), (19) ensures a desirable closed-loop behavior with the equivalent dynamics (16). This is established in the next lemma.

Lemma 3: Consider the feedback interconnection between plant (16) and controller (18), (19). For these dynamics, the point $(\tilde{\mathbf{p}}, \tilde{\psi}, \dot{\tilde{\mathbf{p}}}, \dot{\tilde{\psi}}, \tilde{\mathbf{u}}_v) = 0$ is globally exponentially stable.

Proof: Let us write the error system (avoiding the (t) for the sake of compactness) by initially replacing \mathbf{u}_v in (16) with the feedforward input \mathbf{u}_v^* :

$$\begin{cases} \dot{\tilde{\mathbf{p}}} = \tilde{\mathbf{v}} \\ \dot{\tilde{\mathbf{v}}} = \tilde{\mathbf{v}} - \ddot{\mathbf{p}}_R \\ \dot{\tilde{\psi}} = \tilde{\boldsymbol{\omega}} \\ \dot{\tilde{\boldsymbol{\omega}}} = \boldsymbol{\omega} - \ddot{\psi}_R \end{cases} \Rightarrow \begin{cases} \dot{\tilde{\mathbf{p}}} = \tilde{\mathbf{v}} \\ \dot{\tilde{\mathbf{v}}} = \tilde{\mathbf{v}} - k_P\tilde{\mathbf{p}} - k_D\dot{\tilde{\mathbf{p}}} \\ \dot{\tilde{\psi}} = \tilde{\boldsymbol{\omega}} \\ \dot{\tilde{\boldsymbol{\omega}}} = \tilde{\boldsymbol{\omega}} - k_{P,\psi}\tilde{\psi} - k_{D,\psi}\dot{\tilde{\psi}} \end{cases} \quad (20)$$

Dynamics (20) is linear and globally exponentially stable (GES) for any positive selection of the PD gains k_P , k_D , $k_{P,\psi}$, $k_{D,\psi}$, and from linearity it is also input-to-state stable from any additive input perturbing the right hand side.

In the next step of the proof, we study the evolution of $\tilde{\mathbf{u}}_v = \mathbf{u}_v - \mathbf{u}_v^*$ to take into account that dynamics (16) is actually driven by \mathbf{u}_v (and not \mathbf{u}_v^* as in (20)). With simple manipulations and using selection (18), we obtain

$$\dot{\tilde{\mathbf{u}}}_v = \dot{\mathbf{u}}_v - \dot{\mathbf{u}}_v^* = \mathbf{A}\mathbf{u}_v + \mathbf{B}\mathbf{u}_{v,c} - \dot{\mathbf{u}}_v^* \quad (21)$$

$$= (\mathbf{A} - \mathbf{BK})\tilde{\mathbf{u}}_v, \quad (22)$$

which is clearly GES because $\mathbf{A} - \mathbf{BK}$ is Hurwitz by construction. The overall error system is the cascaded interconnection of two exponentially stable linear systems, where $\tilde{\mathbf{u}}_v$ can be seen as a disturbance driving the lower subsystem (20). Global exponential stability of the origin then follows from standard results on cascaded linear systems. ■

C. Overall Control Structure

According to the scheme of Figure 4, the overall control architecture corresponds to selecting \mathbf{u} as in (13) with the commanded virtual control $\mathbf{u}_{v,c}$ defined in (18). The term $\dot{\mathbf{u}}_v^*$ appearing in (18) can be rewritten explicitly as:

$$\dot{\mathbf{u}}_v^* = \begin{bmatrix} m\mathbf{R}^T(\psi) (\mathbf{S}(\psi)\mathbf{v} + \mathbf{v}_d) \\ j \left(-k_{P,\psi}\dot{\tilde{\psi}} - k_{D,\psi}\ddot{\tilde{\psi}} + \ddot{\psi}_R - \frac{f_{r,\psi}(\boldsymbol{\omega})}{j} \right) \end{bmatrix}$$

with:

$$\begin{aligned} \mathbf{v} &= -k_P\tilde{\mathbf{p}} - k_D\dot{\tilde{\mathbf{p}}} + \ddot{\mathbf{p}}_R - m^{-1}\mathbf{f}_{r,p}(\mathbf{v}), \\ \mathbf{v}_d &= -k_{P,\psi}\dot{\tilde{\psi}} - k_{D,\psi}\ddot{\tilde{\psi}} + \ddot{\psi}_R - m^{-1}\dot{f}_{r,\psi}(\boldsymbol{\omega}), \end{aligned}$$

and $\mathbf{S}(\psi) = \begin{bmatrix} 0 & \psi \\ -\psi & 0 \end{bmatrix}$. It can be noticed that to implement this controller, one needs: (a) to measure \mathbf{u}_v , which actually depends on the forces \mathbf{f}_i^B , depending in turn by the states w_i and θ_i , which are provided by the ESC controller and the turret servo-motor encoder, respectively; (b) to measure the acceleration $\ddot{\mathbf{p}}$ and the angular acceleration $\ddot{\psi}$, which can be both measured using a gyroscope and a pair of accelerometers placed on two different locations on the platform, or by mean of a simple observer for a double integrator system; (c) availability of the third derivatives of the reference $\ddot{\mathbf{p}}_R, \ddot{\psi}_R$ which can be provided by a suitable trajectory generation; (d) knowledge of the derivatives of the friction forces, as defined in Section II-C. Because of these requirements the reference trajectories have to be class C^3 functions of time.

Regarding the availability of the measurements for feedback, we emphasize that the linear error dynamics (20), (22) helps in dealing with possible estimation transients. The use of asymptotic estimators instead of direct measurements not only helps in the case when certain sensors are not available,

but may also be a useful means to suitably filter some possibly noisy measurement outputs.

With this overall scheme in place, the following main result is a direct consequence of Lemmas 1, 2 and 3.

Theorem 1: Consider some class C^3 functions $\mathbf{p}_R(t), \psi_R(t)$ as desired position and attitude trajectories for the platform. Consider the overall closed-loop system (1), (12), (13), (18), (19). Consider a cost function $J : \mathbb{R}^{n_a} \rightarrow \mathbb{R}$. The overall closed-loop system is such that:

- (i) The trajectories $\mathbf{p}(t), \psi(t)$ of system (1) converge globally and exponentially to $\mathbf{p}_R(t), \psi_R(t)$.
- (ii) The commanded virtual control $\mathbf{u}_{v,c}(t)$ converges to: $\bar{\mathbf{u}}_{v,c}(\mathbf{p}_R(t), \psi_R(t))$ that only depends on the reference trajectories and its derivatives.
- (iii) For any constant reference trajectory leading to a constant value of $\mathbf{u}_{v,c}, \mathbf{x}_a$ converges to a stationary point of

$$J(\mathbf{x}_a), \quad \text{subject to} \quad \mathbf{h}(\mathbf{x}_a) = \mathbf{u}_{v,c}. \quad (23)$$

In particular, if J restricted to $\mathbf{h}(\mathbf{x}_a) = \mathbf{u}_{v,c}$ is strongly convex, then \mathbf{x}_a converges to the global minimum of (23).

Sketch of the proof. Items (i) and (ii) follow from combining Lemma 3 and Lemma 1, where the assumptions of Lemma 1 hold based on Lemma 2. About item (iii), the proof follows from [13, Thm 1] after noticing that any constant (therefore bounded) reference, combined with the result of items (i), (ii) produces a bounded commanded virtual control $\mathbf{u}_{v,c}(t)$. ■

We emphasize that Theorem 1 establishes results for the *unconstrained* dynamics (1), where (2) are treated as “soft” constraints embedded in the allocation design. Extending this result to the case of hard constraints requires characterizing suitable classes of reference signals and possibly bounded regions of attraction, which go beyond the scope of this work.

IV. EXPERIMENTS

Two sets of experiments were conducted on the ROSPO to validate our design. In the first set (Section IV-A) we illustrate the ability of the allocator to induce a first order response between the virtual commanded input and the commanded input. The second set of experiments (Section IV-B) apply the overall hierarchical control architecture to two configurations of the ROSPO with three and four turrets, when tracking an eight-shaped reference trajectory. The results well illustrate the scalability of the proposed design.

A. Experiment 1: Allocator

In this experiment, a sequence of steps of desired wrenches are requested to the allocator while varying γ_P , to validate the *linear dynamic behavior* established in Lemma 1. In particular the following wrenches are requested in sequence: a force of $8N$ on the x -axis, a force of $-8N$ on x -axis, a torque of $2Nm$, and finally a torque of $-2Nm$.

The results can be seen in Figure 6. From the figure we well appreciate the linear first-order response associated to gain γ_P . The larger the γ_P , the smaller the settling time. However for γ_P larger than 4 we start to notice some slight overshoots effects, especially in the torque dynamics. Such

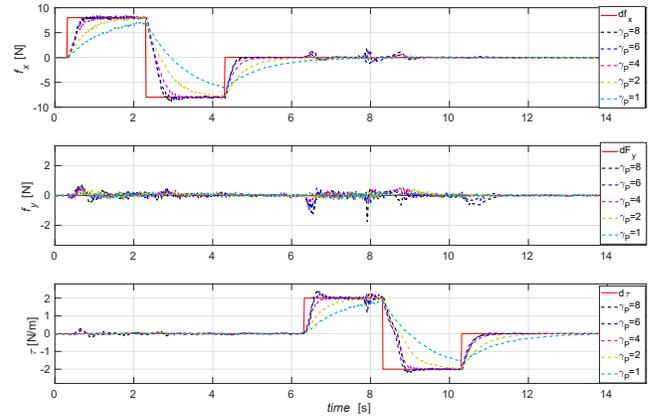


Fig. 6. Allocator responses to step requests: requested wrench (solid red) and obtained wrench (different dashed colors). The linear behavior as function of γ_P is apparent.

an overshoot can be well explained by effects of unmodeled dynamics, which start becoming relevant at those speeds. This is why $\gamma_P = 4$ was chosen for the next experiment.

B. Experiment 2: Trajectory Tracking

In this experiment the platform tracks an eight-shaped trajectory. The experiment allows to test the whole control architecture presented in this paper. We comparatively test two configurations with three and four actuator turrets.

The reference trajectory for the high-level controller is the following:

$$\begin{cases} p_{R,x}(t) &= \rho_x \cos(c_1 t) \\ p_{R,y}(t) &= \rho_y \sin(c_2 t) \\ \psi_R(t) &= 0 \end{cases}, \quad (24)$$

with $\rho_x = 1$, $\rho_y = 0.6$, $c_1 = 0.4$, $c_2 = 0.8$. The reference yaw is zero all along the trajectory. To have a C^3 reference function as requested by Theorem 1, a degree 8 polynomial trajectory is used to join the initial condition of the platform to the eight-shaped trajectory. Degree 8 is needed to impose the initial and final conditions and to allow continuity up to the third derivative.

We repeated the experiment with $n = 4$ and $n = 3$, i.e. with 4 and 3 turret modules. Between the two experiments, no control parameters were modified, to test the validity of the algorithm across multiple configuration scenarios. From a practical viewpoint, the 3 turrets configuration was simply obtained by disabling one of the 4 installed turrets. The physical parameters appearing in dynamics ((1a) and (1b)) were identified as $m = 6 \text{ kg}$, $j = 0.61 \text{ kgm}^2$ and $k_w = 6.5 \cdot 10^{-4} \text{ N s}^2$. The inertia was estimated considering the mass of the main components of the system (bars of the frame, turrets, electronics, wheels, etc) and applying Huygens-Steiner theorem. The scalar gain k_w was estimated on a test bench with a 6-axis force sensor, using a commercial BLDC motor, and a custom ESC with closed-loop speed control [21]. The four actuator modules have the following positions $\mathbf{r}_1 = [0.37, -0.195]$, $\mathbf{r}_2 = [0.302, 0.196]$, $\mathbf{r}_3 = [-0.358, 0.189]$ and $\mathbf{r}_4 = [-0.391, 0.19]$. In the experiment with $n = 3$, the

4-th module was installed but disabled, so that the mass and inertia of the platform were maintained.

For the high-level controller we considered the following proportional and derivative gains ensuring a desirable linear response of the linear error dynamics $(\dot{\mathbf{p}}, \dot{\mathbf{v}}, \dot{\psi}, \dot{\omega})$: $k_P = 20.7$, $k_{P,\psi} = 10.4$, $k_D = 9.1$ and $k_{D,\psi} = 6.45$. Matrix \mathbf{K} was selected with a pole placement algorithm, placing the eigenvalues of the dynamics at $[-3, -3, -3]$ which is a reasonable choice in terms of the experienced input size. These values were chosen in order to keep the inputs $u_{\theta,i}$ and $u_{w,i}$ reasonably below their maximal values.

Finally, for the cost function (14) in the allocator block we had to consider the following physical constraints of the actuators: $\bar{w}_i = 100$ Hz, $\underline{w}_i = 18$ Hz, $\bar{\theta}_i = 4\pi$ and $\underline{\theta}_i = -4\pi$. The weight parameters in (14) have been selected as $\mu_1 = 300$, $\mu_2 = 750$ and $\mu_w = 1/2000$, in order to have comparable values of the penalty terms appearing in (14) and corresponding to the different soft saturations imposed by the allocation strategy.

The following allocation gains were used: $\gamma_P = 4$ and $\gamma_J = 30$ and the sampled-data control loop was executed at 100 Hz.

Figures 7 and 8 show in dashed line the desired path and in solid red line the actual path followed by the platform. The maximum error is 0.035 m on the x and y axes and 3 degrees for the yaw angle for the experiment with $n = 4$ turrets. The maximum error is 0.07 m on the x and y axes and 10 degrees for the yaw angle for the experiment with $n = 3$ turrets. The position along the y axis has also a tracking error bias of about 0.02 m, probably due to the floor not being perfectly flat, which results in a constant disturbance arising from the gravity force.

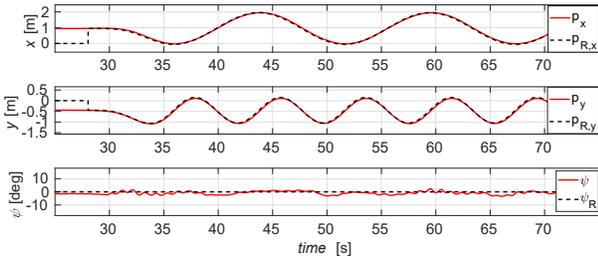


Fig. 7. Path tracking of the eight-shaped path with $n = 4$ turrets.

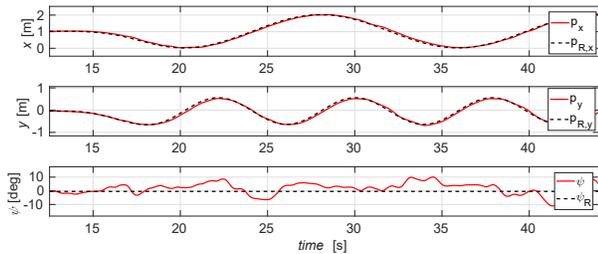


Fig. 8. Path tracking of the eight-shaped path with $n = 3$ turrets.

A top view of the reference trajectory and the position tracking for the two experiments is clear in Figure 9. The reasons for the reduced tracking precision with $n = 3$ turrets may include the following aspects:

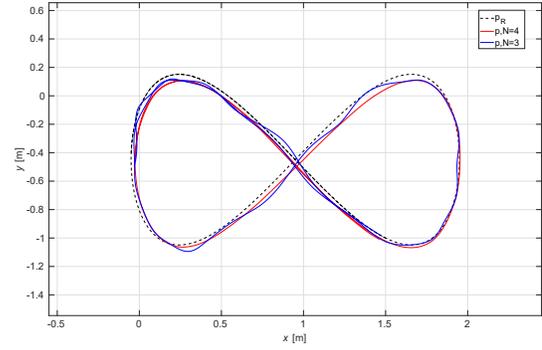


Fig. 9. Top view of the desired path and the tracking with both $n = 4$ (red) and $n = 3$ (blue) turrets.

- *Positiveness*: since the propellers can only revolve at positive rotational speed, they can produce only positive thrust. Hence 3 positive thrusts (considering θ_i fixed) cannot span the whole \mathbb{R}^3 space of the virtual input. To obtain the desired input, the allocator has to make large use of the turret's orientation, reaching very often input saturation.
- *Aggressive trajectory*: the requested trajectory is quite demanding in terms of virtual input, so with 3 modules only, both state and input saturations are often reached.

Both the previously described phenomena can be appreciated in Figure 10, where we can notice that the state saturation for w_i is often reached, and both w_i, θ_i change very rapidly (θ_i reaching often the input saturation limits).

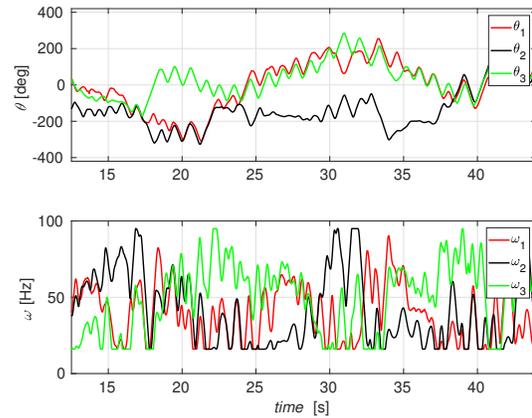


Fig. 10. Actuators states for $n = 3$. The state saturation is often reached by the speed w and the input saturation is often reached by the position θ .

The effect of the allocator and of its term u_J is evident in Figure 11, showing the evolution of the cost function in two experiments with $\gamma_J = 0$ and $\gamma_J = 30$ (we consider the case $n = 4$ in this figure), respectively. Clearly, the case $\gamma_J = 0$ corresponds to ignoring the cost J , because \mathbf{u}_J becomes identically zero. From the figure we may appreciate that in the case $\gamma_J = 30$ the allocator is able to keep the cost function lower, and in our particular case this keeps the actuator states, w_i and θ_i , more balanced and less prone to hit the state saturation (see on the right side of Figure 12). In the case without allocation ($\mathbf{u}_J = 0$), the actuators states (in

particular w_i) reach their saturation limits.

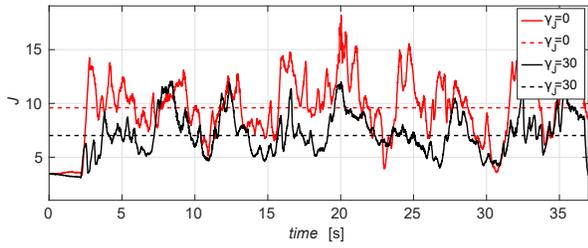
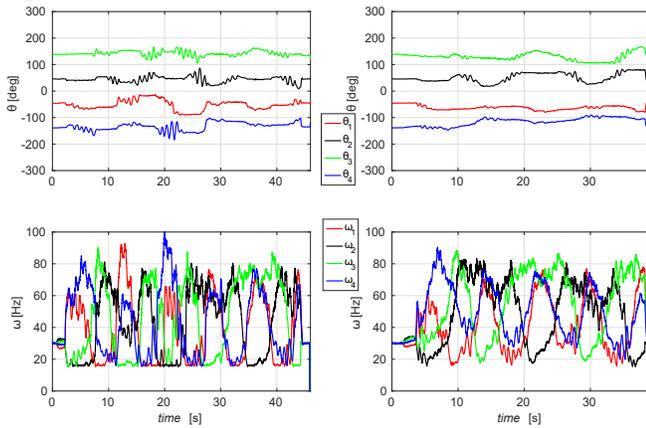


Fig. 11. ROSPO with $n = 4$ turrets. Allocator with \mathbf{u}_J term in black line; allocator without \mathbf{u}_J term in red line. In dashed lines the average of the cost.



(a) Actuator states without \mathbf{u}_J . (b) Actuator states with \mathbf{u}_J .

Fig. 12. Actuators states in two cases ($n = 4$): in (a) the optimization term \mathbf{u}_J is disabled (left plots), while in (b) it is enabled (right plots).

V. ACKNOWLEDGMENTS

We thank Anthony Mallet and Mohamed Hosameldin for their help in the hardware/software development of the platform. We also thank Alexandre Seuret for useful discussions.

VI. CONCLUSIONS

We considered the control allocation problem in an over-actuated hovercraft vehicle called ROSPO, resembling some important challenges of propeller-actuated mechatronic systems. A hierarchical control architecture has been proposed, consisting of a high level controller and a control allocation algorithm. The proposed allocation algorithm is easily adapted to any number of actuators and experiments have been performed to verify its effectiveness on two relevant scenarios with three and four propellers mounted on the ROSPO. The experimental results show desirable convergence properties and also show that the control allocation algorithm is capable of keeping the actuator states inside their “soft” saturation bounds. Future work involves addressing hard saturation bounds via suitable anti-windup augmentations addressing transient saturation phenomena that cannot be handled by the allocator scheme.

REFERENCES

- [1] M. Odelga, P. Stegagno, and H. H. Bühlhoff, “A fully actuated quadrotor UAV with a propeller tilting mechanism: Modeling and control,” in *2016 IEEE/ASME Int. Conf. on Advanced Intelligent Mechatronics*, Alberta, Canada, Dec. 2016.
- [2] 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.
- [3] M. Ryll, G. Muscio, F. Pierri, E. Cataldi, G. Antonelli, F. Caccavale, and A. Franchi, “6D physical interaction with a fully actuated aerial robot,” in *2017 IEEE Int. Conf. on Robotics and Automation*, Singapore, May 2017, pp. 5190–5195.
- [4] 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.
- [5] M. Ryll, H. H. Bühlhoff, and P. Robuffo Giordano, “A novel overactuated quadrotor unmanned aerial vehicle: modeling, control, and experimental validation,” *IEEE Trans. on Control Systems Technology*, vol. 23, no. 2, pp. 540–556, 2015.
- [6] 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.
- [7] 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.
- [8] T. Johansen and T. Fossen, “Control allocation – a survey,” *Automatica*, vol. 49, no. 5, pp. 1087–1103, 2013.
- [9] M. Oppenheimer, D. Doman, and M. Bolender, “Control allocation,” in *The Control Handbook, Control System Applications*, W. Levine, Ed. CRC Press, 2010, pp. 8–1–8–24.
- [10] A. Franchi, A. Petitti, and A. Rizzo, “Decentralized parameter estimation and observation for cooperative mobile manipulation of an unknown load using noisy measurements,” in *2015 IEEE Int. Conf. on Robotics and Automation*, Seattle, WA, May 2015, pp. 5517–5522.
- [11] A. Petitti, A. Franchi, D. Di Paola, and A. Rizzo, “Decentralized motion control for cooperative manipulation with a team of networked mobile manipulators,” in *2016 IEEE Int. Conf. on Robotics and Automation*, Stockholm, Sweden, May 2016, pp. 441–446.
- [12] G. Gioioso, A. Franchi, G. Salvietti, S. Scheggi, and D. Prattichizzo, “The Flying Hand: a formation of UAVs for cooperative aerial tele-manipulation,” in *2014 IEEE Int. Conf. on Robotics and Automation*, Hong Kong, China, May 2014, pp. 4335–4341.
- [13] T. E. Passenbrunner, M. Sassano, and L. Zaccarian, “Optimality-based dynamic allocation with nonlinear first-order redundant actuators,” *European Journal of Control*, pp. 3–40, 2016.
- [14] C. Nainer, “Static input allocation applied to a robotic platform with saturation constraints.” MS Thesis: 2017, University of Trento, Italy.
- [15] A. P. Aguiar, L. Cremean, and J. P. Hespanha, “Position tracking for a nonlinear underactuated hovercraft: Controller design and experimental results,” in *42nd IEEE Conf. on Decision and Control*, 2003.
- [16] S. P. Berge and T. I. Fossen, “Robust control allocation of overactuated ships; experiments with a model ship,” in *Preprints IFAC Conference on Maneuvering and Control of Marine Craft*, 1997.
- [17] T. Fossen and T. Johansen, “A survey of control allocation methods for ships and underwater vehicles,” in *14th Mediterranean Conference on Control and Automation*, Ancona (Italy), Jun. 2006.
- [18] T. Fossen, T. Johansen, and T. Perez, “A survey of control allocation methods for underwater vehicles,” in *Underwater Vehicles*, A. Inzartsev, Ed. INTECH Open Access Publisher, 2009, pp. 109–128.
- [19] C. Nainer, M. Furci, A. Seuret, L. Zaccarian, and A. Franchi, “Hierarchical control of the over-actuated ROSPO platform via static input allocation,” in *20th IFAC World Congress*, Toulouse, France, Jul. 2017, pp. 12 698–12 703.
- [20] W. G. S. C. Makkar, W. E. Dixon and G. Hu, “A new continuously differentiable friction model for control systems design,” in *IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, Monterey (USA), Jul. 2005.
- [21] 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, pp. 5203–5208.