Hierarchical Control of the Over-Actuated ROSPO Platform via Static Input Allocation ⋆

C. Nainer * **M. Furci** ** **A. Seuret** ** **L. Zaccarian** **,* **A. Franchi** **

* *DII, University of Trento, Italy (e-mail: carlo.nainer@gmail.com).*
** *LAAS-CNRS, Université de Toulouse, CNRS, Toulouse, France (e-mail: michele.furci,aseuret,zaccarian,antonio.franchi@laas.fr).*

Abstract: This paper addresses the problem of control allocation applied to an over-actuated hovercraft-type vehicle. A hierarchical control architecture, consisting of a high level controller for trajectory tracking, and a control allocation algorithm, is developed and proved to be effective in tracking a desired trajectory while optimizing some cost related to actuator constraints. The control allocation algorithm exploits the redundancy of the system in order to keep the actuator states inside their saturation limits and tries to minimize the power consumption of the propellers. Unlike other papers on control allocation, actuator dynamics is taken into account. The control architecture is tested through simulations that well illustrate the capabilities of the proposed control design.

Keywords: Mobile Robots, Robot Control, Allocation, Over-actuated Robots, Hierarchical Control.

1. INTRODUCTION

Mini and micro Unmanned Aerial Vehicles (UAV) are being subject of many research topics in the last decade. The main motivations are the rich research problems they offer and the enormous amount of potential applications they have. Pertaining research topics include: control, motion planning, multi-agent systems, task allocation, coordination and many more. On the application side, we find: surveillance, industrial inspection, search and rescue, photography, 3D mapping, delivery and many more. A classical multirotor consists of a frame with 4 coplanar motor-propeller actuators that generate forces and torques. This is an under-actuated system, where the classical application consists of tracking a particular position trajectory, with the attitude dependent of the position trajectory. This can be a limitation for certain applications such as aerial manipulation, where the robot has to exert a specific force/torque in contact with the environment. Only in the last years researchers tried to overcome the limitation of planar multi-propeller UAV in terms of under-actuation by adding additional actuators. In (Odelga et al., 2016) the authors propose to use two extra servos to tilt the angle of the propellers and obtain a fully actuated quadrotor. In (Rajappa et al., 2015) six tilted propellers, with a fixed tilting angle can generate forces and torques in all directions. In (Ryll et al., 2016) an exa-rotor can tilt all the propellers in a coordinated fashion by means of an extra actuator, resulting in an overactuated system and finally in (Ryll et al., 2015) a quadrotor can rotate each tilt angle of the four propellers for an over-actuation. Additional recent examples of fully-actuated platform designs are provided in (Brescianini and D'Andea, 2015) and (Park et al., 2016) In all these works the inputs (propeller speed and tilt angle) are considered as ideal inputs, with no actuator dynamics and with unlimited bandwidth. These limitations can be however important when wanting to maximize the bandwidth of a certain closed-loop control action and should be taken into account in the control design. In particular, adding extra actuator now introduces new redundant input directions that can be exploited for a maximally effective use of the many actuators now available.

Allocation of redundant actuators has been a subject of intensive works in the past two decades, especially with reference to specific applications, such as the aerospace Oppenheimer et al. (2006) and underwater Fossen and Johansen (2006) areas. Historical solutions deciding the optimal distribution of input authority among the available actuators typical rely on the online solution of an optimization problem, which can be computationally intensive, especially in light of the fact that different actuators are often in place to best deal with hard or soft limitations, which then result in nonlinear cost functions to be miminized (see, e.g., the scheme surveyed in Johansen and Fossen (2013) and Oppenheimer et al. (2010)).

In this paper we consider a simplified platform (the ROSPO platform) that can emulate the over-actuated multi-rotor configuration. It consists of a frame with a variable number of modules composed by a tilting turret and a motor-propeller actuator. As compared to a UAV, this platform is simplified because its dynamics is constrained to a 2D plane, having thus only 3 degrees of freedom. Despite the simplified setting, this platform well represents the main challenges behind having a large number of actuators, thereby introducing redundancy and requiring some input allocation strategy. The goal is to test a control paradigm suitable merging a high level controller with a static input allocation technique for an over-actuated platform with actuators with a first order dynamics and limited input magnitude. In particular, we

⋆ This work has been partially funded by the European Unions 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.

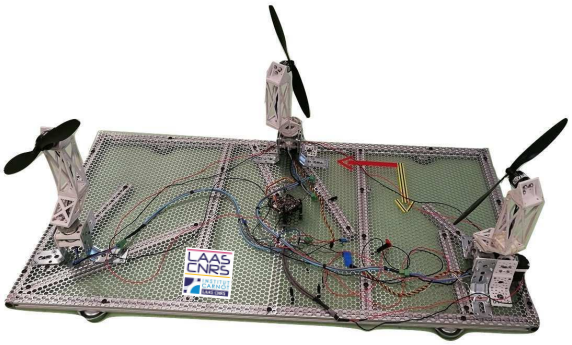


Fig. 1. The ROSPO platform.

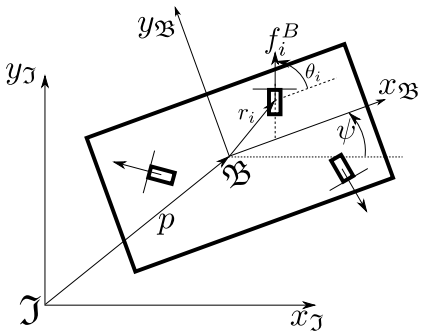


Fig. 2. Schematic representation of the ROSPO platform and main symbol definitions.

allocate the control action based on a scheme presented in Passenbrunner et al. (2016), using suitable projection matrices capable of performing some kind of decoupling between the desired wrench to be applied to the physical system, and the cost function that should be optimized during operation. The scheme has the advantage of being computationally quite cheap, even though optimality is only gauranteed asymptotically. In this paper we should the effectiveness of the proposed control paradigm on the ROSPO platform. A control law on a similar platform but without the over-actuation was proposed by Aguiar et al. (2003). The over-actuation on 2D motions is instead a well studied problem for ships and vessels as in (Berge and Fossen, 1997) and as widely overviewed in Fossen and Johansen (2006); Fossen et al. (2009).
The paper is organized as follows: in Section 2 we present the platform and its dynamical model. In Section 3 we describe the control architecture for the system, composed by an allocation strategy and a high level controller for motion tracking. Finally in Section 4 we present some simulations for the platform with the proposed controller.

2. MODEL

Figures 1 and 2 show the ROtor-graSPing Omnidirectional (ROSPO) platform and its schematic representation including the main symbol definitions, respectively. The system is composed by a rigid rectangular frame sustained by four omnidirectional passive spherical wheels and n actuating modules. Each module consists of a turret, which is orientable by means of a servo motor. Each turret carries a propeller driven by a BLDC motor. In the particular case shown in Fig 1 it is $n = 3$, however, the developed theory extends naturally to a generic number of turrets $n \geq 3$.

The actuating modules can produce a thrust by spinning the propeller and can independently rotate the whole turret to orient the exerted force. The platform can move on the ground with very low friction thanks to "ball transfer unit" wheels. 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:

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

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

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 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 $\mathcal{B}$, $\mathbf{r}_i \in \mathbb{R}^2$ are the positions of each actuator module w.r.t. the center of mass of the system, $\mathbf{\Pi} = \left[\begin{smallmatrix} 0 & 1 \\ -1 & 0 \end{smallmatrix}\right]$, 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 $i = 1, \ldots, n$ and k_w is a constant aerodynamic parameter. Equations (1a) can be referred to as the platform model, while equations (1b) as the actuators' dynamics.

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:

$$\begin{array}{cc} \underline{\theta_i} \leq \theta_i \leq \overline{\theta_i} & \underline{w_i} \leq w_i \leq \overline{w_i} \\ \underline{u_{\theta,i}} \leq u_{\theta,i} \leq \overline{u_{\theta,i}} & \underline{u_{w,i}} \leq u_{w,i} \leq \overline{u_{w,i}}, \end{array} \tag{2}$$

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:

$$\begin{cases} m\dot{\mathbf{v}} &= \mathbf{R}(\psi)\mathbf{f} \\ j\dot{\omega} &= \tau \\ \dot{\theta}_i &= u_{\theta,i} \\ \dot{w}_i &= u_{w,i} \\ \mathbf{f}_i^B &= k_w w_i^2 \begin{bmatrix}\cos(\theta_i)\\ \sin(\theta_i)\end{bmatrix}, \end{cases} \tag{3}$$

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

3. CONTROL ARCHITECTURE

In a typical task for this platform, we want to control a desired position and attitude trajectory. Hence the system has three regulated outputs (two positions and one angle), while the available inputs are $2n$. This means that for $n \geq 2$ the system is over-actuated, while for $n = 1$ the system is under-actuated. In this paper we consider the case $n \geq 2$. We can sum up the peculiarities of the system, which correspond to:

- Nonlinear plant;
- Over-actuated system;
- Actuators with nonlinear dynamics;
- Actuators with constrained state and input.

We want to develop a hierarchical control architecture consisting of a high level control for motion control (trajectory tracking) and a control allocation algorithm to handle the over-actuation in the actuators. In this architecture, the high level controller specifies a desired wrench 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), with some optimality criterion.

The control architecture is depicted in Figure 3 and is based on the static allocation scheme of Passenbrunner et al. (2016). In that paper, both a static and a dynamic strategy are proposed, where the dynamic one is necessary in case certain signals are not available for measurement. Since the ROSPO platform is equipped with full measurement systems, we adopt here the static strategy, but the use of the dynamic strategy could be envisioned in future work to deal with uncertainties or remove measurement signals from the ROSPO platform. The adopted allocation strategy ensures regulation of a commanded virtual input ($\mathbf{u_{v,c}}$) together with optimality with respect to a given cost function. The construction is based on the assumption of an invertible input matrix (such as the one in (1b)) and first order strictly proper dynamics, and is parametrized by a cost function to be optimized. More specifically, with this allocation strategy the high level controller sees 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.

For the high level control we design a controller for trajectory tracking that considers also the effect of the allocator (namely the linear first order dynamics) and achieves global exponential stability for arbitrary class C^3 reference functions.

3.1 Static Allocation

We design the control allocation algorithm following Passenbrunner et al. (2016). We summarize briefly the fundamental of the corresponding allocation technique. Consider a virtual control $\mathbf{u_v} \in \mathbb{R}^{n_{u_v}}$, a number of redundant actuators $n_a > n_{u_v}$ and a commanded virtual control $\mathbf{u_{v,c}} \in \mathbb{R}^{n_{u_v}}$ coming from a high level controller. The actuators should obey a first-order (possibly coupled) dynamics:

$$\dot{\mathbf{x}}_\mathbf{a} = \mathbf{f}(\mathbf{x_a}) + \mathbf{g}(\mathbf{x_a})\mathbf{u}, \quad \mathbf{u_v} = \mathbf{h}(\mathbf{x_a}) \tag{4}$$

with $\mathbf{f} : \mathbb{R}^{n_a} \to \mathbb{R}^{n_a}$, $\mathbf{g} : \mathbb{R}^{n_a} \to \mathbb{R}^{n_a \times n_a}$, $\mathbf{h} : \mathbb{R}^{n_a} \to \mathbb{R}^{n_{u_v}}$. In this platform we have $n_a = 2n$ and $n_{u_v} = 3$.

The allocator generates $\mathbf{u}$ such that $\lim_{t\to\infty} \|\mathbf{u_v} - \mathbf{u_{v,c}}\| = 0$, while minimizing a cost function $J : \mathbb{R}^{n_a} \to \mathbb{R}$.

In particular, the static allocation control is given by:

$$\mathbf{u} = \mathbf{g}(\mathbf{x_a})^{-1} \left(-\mathbf{f}(\mathbf{x_a}) + \mathbf{u_y} - \mathbf{u_J}\right) \tag{5}$$

where:

$$\begin{aligned} \mathbf{u_y} &= \gamma_P \nabla \mathbf{h}(\mathbf{x_a}) \left(\nabla \mathbf{h}(\mathbf{x_a})^T \nabla \mathbf{h}(\mathbf{x_a})\right)^{-1} (\mathbf{u_{v,c}} - \mathbf{h}(\mathbf{x_a})) \\ \mathbf{u_J} &= \gamma_J \nabla_\perp \mathbf{h}(\mathbf{x_a}) \nabla \mathbf{J}(\mathbf{x_a}) \end{aligned} \tag{6}$$

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

$$\nabla_\perp \mathbf{h}(\mathbf{x_a}) = \mathbf{I} - \nabla \mathbf{h}(\mathbf{x_a}) \left(\nabla \mathbf{h}(\mathbf{x_a})^T \nabla \mathbf{h}(\mathbf{x_a})\right)^{-1} \nabla \mathbf{h}(\mathbf{x_a})^T. \tag{7}$$

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.

For this specific application, according to (1), we can define:

$$\mathbf{x_a} = [w_1 \ \theta_1 \ \cdots \ w_n \ \theta_n]^T \tag{8}$$

$$\mathbf{u} = [u_{w_1} \ u_{\theta,1} \ \cdots \ u_{w,n} \ u_{\theta,n}]^T \tag{9}$$

$$\mathbf{f}(\mathbf{x_a}) = 0, \qquad \mathbf{g}(\mathbf{x_a}) = \mathbf{I} \tag{10}$$

$$\mathbf{h}(\mathbf{x_a}) = \begin{bmatrix} \sum_{i=1}^{n} k_w w_i^2 \begin{bmatrix}\cos(\theta_i)\\ \sin(\theta_i)\end{bmatrix} \\ \sum_{i=1}^{n} (\mathbf{\Pi r}_i)^T k_w w_i^2 \begin{bmatrix}\cos(\theta_i)\\ \sin(\theta_i)\end{bmatrix} \end{bmatrix}. \tag{11}$$

A crucial aspect that influences the behavior and the optimality of the allocator relies on the choice of the cost function J.

For our application the cost function was chosen as follows:

$$J = \sum_{i=0}^{n} \left(\mu_1 \left(\frac{\tilde{w}_i}{\Delta_{w,i}}\right)^6 + \mu_2 \left(\frac{\tilde{\theta}_i}{\Delta_{\theta,i}}\right)^6 + \mu_w w_i^2 \right), \tag{12}$$

with:

$$\begin{aligned} \tilde{w}_i &= w_i - w_{m,i} & \tilde{\theta}_i &= \theta_i - \theta_{m,i} \\ \Delta_{w,i} &= \overline{w_i} - \underline{w_i} & \Delta_{\theta,i} &= \overline{\theta_i} - \underline{\theta_i} \\ w_{m,i} &= \frac{\overline{w_i} + \underline{w_i}}{2} & \theta_{m,i} &= \frac{\overline{\theta_i} + \underline{\theta_i}}{2} \end{aligned} \tag{13}$$

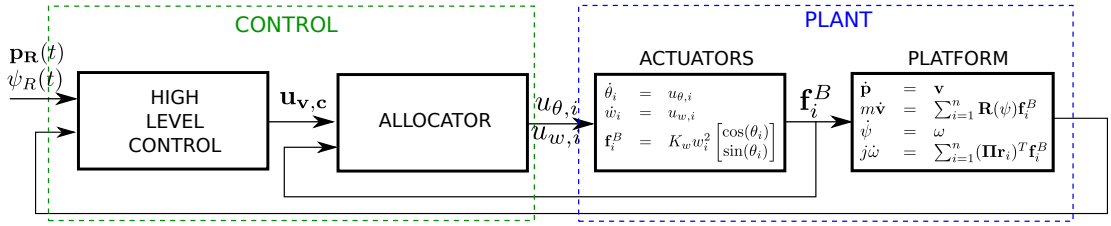


Fig. 3. The hierarchical control architecture comprising the high level controller and the low level allocator.

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

The first two terms in (12) impose a soft state constraint on the states of the actuators according to (2). The functions are indeed almost zero in the intervals $[\underline{w_i}\ \overline{w_i}]$, $[\underline{\theta_i}\ \overline{\theta_i}]$ and very steep when approaching the state limits $\overline{w_i}, \underline{w_i}, \underline{\theta_i}, \overline{\theta_i}$. The third term in (12) penalizes high rotational speeds of the propellers, to reduce the energy consumption of the platform.

Lemma 1. If

$$rank(\nabla \mathbf{h}(\mathbf{x_a})) = 3, \tag{14}$$

system (1b) augmented with allocator (5)-(11) is such that:

(1) for each initial condition $\theta_i(0), w_i(0)$ and each virtual commanded input $t \mapsto \mathbf{u_{v,c}}(t)$, the virtual control satisfies:
$$\dot{\mathbf{u}}_\mathbf{v} = \gamma_P(-\mathbf{u_v} + \mathbf{u_{v,c}})$$
(2) for any constant selection of $\mathbf{u_{v,c}}$, cost J defined in (12) converges to the constrained minimum J^*:
$$J^* = \min_{\mathbf{x_a} \in \mathbb{R}^{n_a}} J(\mathbf{x_a}), \quad \text{subject to} \quad \mathbf{h}(\mathbf{x_a}) = \mathbf{u_{v,c}}.$$

3.2 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 4 and can be written as:

$$\begin{aligned} m\dot{\mathbf{v}} &= \mathbf{R}(\psi)\mathbf{S_1}\mathbf{u_v} \\ j\dot{\omega} &= \mathbf{S_2}\mathbf{u_v} \\ \dot{\mathbf{u}}_\mathbf{v} &= \gamma_P(-\mathbf{u_v} + \mathbf{u_{v,c}}), \end{aligned} \tag{15}$$

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, $\mathbf{S}_1 = \left[\begin{smallmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{smallmatrix}\right]$ and $\mathbf{S}_2 = [\,0\ 0\ 1\,]$.

The goal is to design a high level controller that stabilizes (15) for a trajectory tracking purpose. The desired trajectory can be expressed as $\mathbf{p_R}(t), \dot{\mathbf{p}}_\mathbf{R}(t), \cdots, \mathbf{p_R}^{(n)}(t)$, $\psi_R(t), \dot{\psi}_R(t), \cdots, \psi_R^{(n)}(t)$, 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:

$$\begin{aligned} \tilde{\mathbf{p}}(\mathbf{t}) &= \mathbf{p}(t) - \mathbf{p_R}(t) \\ \tilde{\psi}(t) &= \psi(t) - \psi_R(t). \end{aligned} \tag{16}$$

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

To this end, we propose the following selection:

$$\mathbf{u_{v,c}} = \mathbf{B}^{-1}(\dot{\mathbf{u}}_\mathbf{v}^* - \mathbf{A}\mathbf{u}_\mathbf{v}^*) - \mathbf{K}\tilde{\mathbf{u}}_\mathbf{v} \tag{17}$$

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 (17) uses the feedforward signal

$$\mathbf{u}_\mathbf{v}^* = \begin{bmatrix} m\mathbf{R}^T(\psi)\left(-k_P\tilde{\mathbf{p}} - k_D\dot{\tilde{\mathbf{p}}} + \ddot{\mathbf{p}}_\mathbf{R}\right) \\ j\left(-k_{P,\psi}\tilde{\psi} - k_{D,\psi}\dot{\tilde{\psi}} + \ddot{\psi}_R\right) \end{bmatrix} \tag{18}$$

with $\tilde{\mathbf{u}}_\mathbf{v} = \mathbf{u_v} - \mathbf{u}_\mathbf{v}^*$ 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 (17), (18) ensures a desirable closed-loop behavior with the equivalent dynamics (15). This is established in the next lemma.

Lemma 2. Consider the feedback interconnection between plant (15) and controller (17), (18). For these dynamics, the point $(\tilde{\mathbf{p}}, \tilde{\psi}, \dot{\tilde{\mathbf{p}}}, \dot{\tilde{\psi}}, \tilde{\mathbf{u}}_\mathbf{v}) = 0$ is globally exponentially stable.

3.3 Overall Control Structure

The overall control input is defined as:

$$[u_{w,1}\quad u_{\theta,1}\quad \cdots\quad u_{w,n}\quad u_{\theta,n}]^T = \mathbf{u_y} - \mathbf{u_J} \tag{19}$$

with $\mathbf{u_y}$ and $\mathbf{u_J}$ defined in (6) and with a commanded virtual control $\mathbf{u_{v,c}}$ defined as in (17).

With this overall scheme in place, the following main result can be proven by combining the results proven below in Lemmas 1 and 2.

Theorem 3. 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),(19),(6)-(11),(17),(18). Consider a cost function $J : \mathbb{R}^{n_a} \to \mathbb{R}$. The overall closed loop system is such that:

- The trajectories $\mathbf{p}(t), \psi(t)$ of system (1) converge globally and exponentially to $\mathbf{p_R}(t), \psi_R(t)$.
- The commanded virtual control $\mathbf{u_{v,c}}(t)$ converges to: $\overline{\mathbf{u_{v,c}}}(\mathbf{p_R}(t), \psi_R(t))$ that depends only by the reference trajectories and its derivatives.
- For any constant reference trajectory leading to a constant value of $\mathbf{u_{v,c}}$, $\mathbf{x_a}$ converges to a stationary point of
$$J(\mathbf{x_a}), \quad \text{subject to} \quad \mathbf{h}(\mathbf{x_a}) = \mathbf{u_{v,c}}. \tag{20}$$
In particular, if J is strongly convex, then $\mathbf{x_a}$ converges to the global minimum of (20).

4. SIMULATIONS

In this section we first describe the parameters of our system and then we present the main results of the simulations. All the parameters are taken from the real ROSPO platform and estimated by experiments. The experimental platform has $n = 3$ actuator modules, but is modular and other modules can be easily attached. For our system ((1a) and (1b)) we identified $m = 5.1\ kg$, $j = 0.42\ kgm^2$ and $k_w = 6.5 \cdot 10^{-4}\ Ns^2$. The inertia was

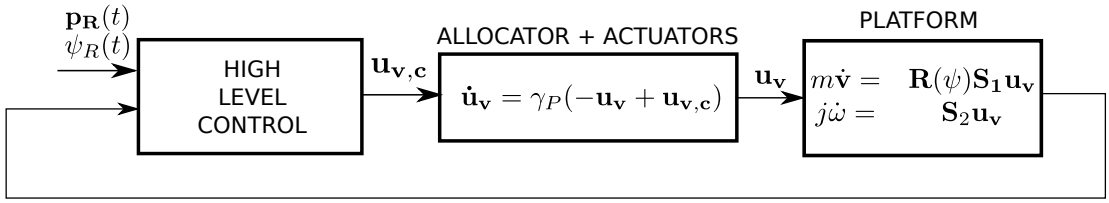


Fig. 4. The equivalent first-order linear dynamics seen by the high level controller due to the hierarchical control architecture.

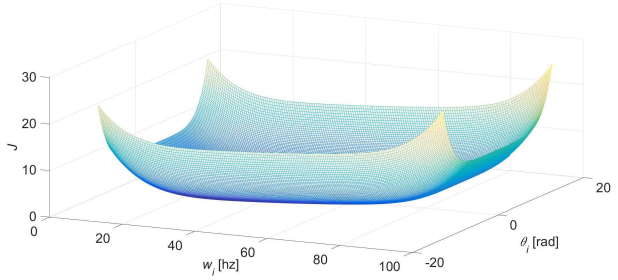


Fig. 5. Cost function J in (12) for a single actuator module (turret and propeller). The overall cost involves the sum of these terms.

estimated considering the mass of the platform structure (3.6 kg) equally distributed over its area (80cm x 40cm) and considering the mass of each single turret (0.5 kg) as a point mass. 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. The three actuator modules have the following positions $\mathbf{r_1} = [0.2\,, 0.35]$, $\mathbf{r_2} = [0.2\,, -0.35]$ and $\mathbf{r_3} = [-0.2\,, 0]$.

For the high-level controller we considered the following proportional and derivative gains ensuring a desirable linear response of the linear error dynamics $(\dot{\tilde{\mathbf{p}}}, \dot{\tilde{\mathbf{v}}}, \dot{\tilde{\psi}}, \dot{\tilde{\omega}})$: $k_P = 0.15$, $k_{P,\psi} = 0.1$, $k_D = 0.6$ and $k_{D,\psi} = 0.4$. Matrix $\mathbf{K}$ was selected with a pole placement algorithm, placing the eigenvalues of the dynamics at $[-2, -4, -3]$ 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 (12) in the allocator block we had to consider the following physical constraints of the actuators: $\overline{w_i} = 95\,\text{Hz}$, $\underline{w_i} = 9\,\text{Hz}$, $\overline{\theta_i} = 4\pi$ and $\underline{\theta_i} = -4\pi$. The weight parameters in (12) 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 (12) and corresponding to the different soft saturations imposed by the allocation strategy. The cost function for a single actuator module can be seen in Figure 5, where one can appreciate its increasing behavior close to the soft actuator limits.

For the simulation we set the following allocation gains: $\gamma_P = 1$ and $\gamma_J = 4$. The reference trajectory for the high-level controller is an ∞-shaped track. The corresponding function is the following:

$$\begin{cases} p_{R,x}(t) &= \rho\cos(c_1 t) \\ p_{R,y}(t) &= \rho\sin(c_2 t) \\ \psi_R(t) &= c_3 t + \pi/2 \end{cases}, \qquad (21)$$

with $\rho = 4$, $c_1 = 0.2$, $c_2 = 0.4$ and $c_3 = -0.18$. The last equation of the reference ψ_R for ψ imposes a constant yaw rate along the ∞-shaped path.

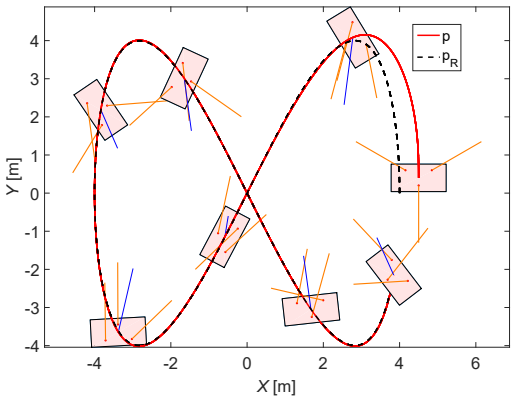


Fig. 6. Path tracking of the ∞-shaped path.

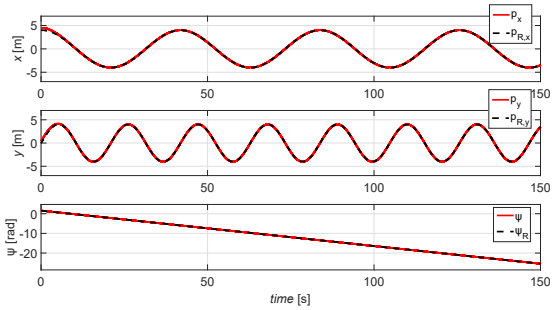


Fig. 7. Reference and actual x-position, y-position and yaw angle, respectively.

The simulation step was set equal to 10 ms. We imposed an error at the initial position of the platform in order to show the transient behavior of our controller. Figure 6 shows in dashed line the desired path and in solid red line the actual path followed by the platform. The rectangles on the path show the position and orientation of the platform at some time instants. The three thin orange lines appearing on each rectangle show the orientation of the turrets. The length of those lines is proportional to the force generated by each module. Only the first 30 seconds out of the 150 of the simulation were plotted, to avoid making the plot too crowded and to better highlight the initial transient. These 30 seconds correspond to an almost complete "∞" track.

A clearer representation of the desired positions and actual positions for the whole simulation is presented in Figure 7, which shows how the closed-loop system quickly reaches the desired position and the error converges to zero asymptotically.

The effect of the allocator and of its term u_J are evident in Figure 8, showing the evolution of the cost function in two simulation runs with $\gamma_J = 0$ and $\gamma_J = 4$, respectively. Clearly, the case $\gamma_J = 0$ corresponds to ignoring the cost

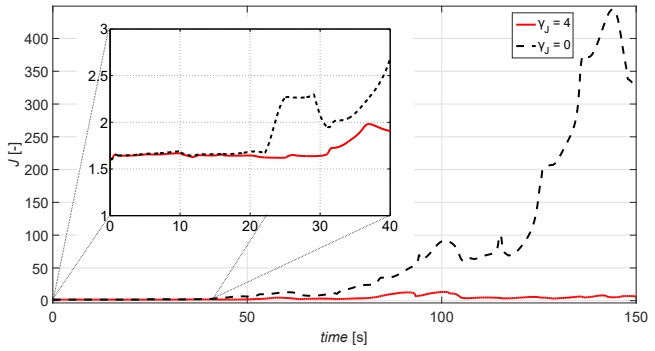
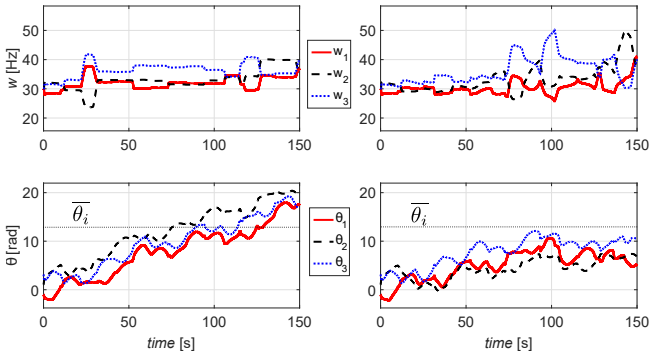


Fig. 8. Allocator with $\mathbf{u_J}$ term in solid red line; allocator without $\mathbf{u_J}$ term in dashed black line

J, because $\mathbf{u_J}$ becomes identically zero. From the figure we may appreciate how in the case $\gamma_J = 4$ the allocator is able to keep the cost function quite low, and in our particular case this keeps the actuator states, w_i and θ_i, inside their limits (see the graphs on the right side of Figure 9). Without $\mathbf{u_J}$ the actuators states (in particular θ_i) exceed their saturation limits just after 100 s of simulation.

Figure 9 shows the state values of the actuators (w_i and θ_i) during the whole simulation, showing that the saturation limits are never reached when using the full scheme with allocation (right plots).



(a) Actuator states without $\mathbf{u_J}$. (b) Actuator states with $\mathbf{u_J}$.

Fig. 9. Actuators states in two cases: in (a) the optimization term u_J is disabled (left plots); in (b) the optimization term u_J is enabled (right plots).

5. CONCLUSIONS

We considered the control allocation problem in an over-actuated hovercraft vehicle. A hierarchical control architecture has been proposed, consisting of a high level controller and a control allocation algorithm. Global exponential convergence to zero of the trajectory tracking error has been proved for the proposed scheme. Simulation results have been given, based on a realistic model of an experimental platform, which confirm the desirable features of the proposed control scheme. In particular, our simulations show desirable convergence properties and also show how the control allocation algorithm is capable of keeping the actuator states inside their saturation bounds. Future work will consider friction of the hovercraft, will analyze the robustness of the control scheme and will implement an anti-windup controller to handle actuator saturation phenomena that may occur during the transient responses. Finally, the new control design will be tested on the real ROSPO experimental platform.

ACKNOWLEDGEMENTS

We thank Dorian Petard for his help in the design and development of the ROSPO platform.

REFERENCES

Aguiar, A.P., Cremean, L., and Hespanha, J.P. (2003). Position tracking for a nonlinear underactuated hovercraft: Controller design and experimental results. In *42nd IEEE Conf. on Decision and Control*.

Berge, S.P. and Fossen, T.I. (1997). Robust control allocation of overactuated ships; experiments with a model ship. In *in Preprints IFAC Conference on Maneuvering and Control of Marine Craft*.

Brescianini, D. and D'Andea, R. (2015). Design, modeling and control of an omni-directional aerial vehicle. In *2016 IEEE Int. Conf. on Robotics and Automation*. Stockholm, Sweden.

Fossen, T. and Johansen, T. (2006). A survey of control allocation methods for ships and underwater vehicles. In *14th Mediterranean Conference on Control and Automation*. Ancona (Italy).

Fossen, T., Johansen, T., and Perez, T. (2009). A survey of control allocation methods for underwater vehicles. In A. Inzartsev (ed.), *Underwater Vehicles*, 109–128. INTECH Open Access Publisher.

Johansen, T. and Fossen, T. (2013). Control allocation – a survey. *Automatica*, 49(5), 1087–1103.

Odelga, M., Stegagno, P., and Bülthoff, H.H. (2016). 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.

Oppenheimer, M., Doman, D., and Bolender, M. (2010). Control allocation. In W. Levine (ed.), *The Control Handbook, Control System Applications*, 8–1–8–24. CRC Press.

Oppenheimer, M., Doman, D., and Bolender, M. (2006). Control allocation for over-actuated systems. In *14th Mediterranean Conference on Control and Automation*. Ancona (Italy).

Park, S., J. Her, J., Kim, J., and Lee, D. (2016). Design, modeling and control of omni-directional aerial robot. In *2016 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 1570–1575. Daejeon, South Korea.

Passenbrunner, T.E., Sassano, M., and Zaccarian, L. (2016). Optimality-based dynamic allocation with nonlinear first-order redundant actuators. *European Journal of Control*, 3–40.

Rajappa, S., Ryll, M., Bülthoff, H.H., and Franchi, A. (2015). Modeling, control and design optimization for a fully-actuated hexarotor aerial vehicle with tilted propellers. In *2015 IEEE Int. Conf. on Robotics and Automation*, 4006–4013. Seattle, WA.

Ryll, M., Bicego, D., and Franchi, A. (2016). Modeling and control of FAST-Hex: a fully-actuated by synchronized-tilting hexarotor. In *2016 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 1689–1694. Daejeon, South Korea.

Ryll, M., Bülthoff, H.H., and Robuffo Giordano, P. (2015). A novel overactuated quadrotor unmanned aerial vehicle: modeling, control, and experimental validation. *IEEE Trans. on Control Systems Technology*, 23(2), 540–556.