Dynamic Decentralized Control for Protocentric Aerial Manipulators

Marco Tognon[1], Burak Yüksel[2], Gabriele Buondonno[3] and Antonio Franchi[1]

***Abstract*— We present a control methodology for underactuated aerial manipulators that is both easy to implement on real systems and able to achieve highly dynamic behaviors. The method is composed by two parts: *i)* a nominal input/state trajectory generator that takes into account the full-body dynamics of the system exploiting its differential flatness property; *ii)* a decentralized feedback controller acting on the actuated degrees of freedom that confers the needed robustness to the closed-loop system. We demonstrate that the proposed controller is able to precisely track dynamic trajectories when implemented on a standard hardware. Comparative experiments clearly show the benefit of using the nominal input/state generator.**

I. INTRODUCTION

Aerial robots are attracting increasing interest from scientists in the robotics society, due to their agility and great workspace. The interest in control of such robots concerns tracking of complex trajectories, human-robot interaction [1], and recently aerial manipulation [2] as well.

An aerial manipulator is a robotic system, which has the capacity to fly and at the same time to manipulate objects in its environment. Most commonly they consist of a flying robot and at least one manipulator arm. This system breaks ground to many different robotic applications, e.g., pick and place [3], aerial physical interaction [4], and aerial grasping (first introduced in [5] and then shown also in [6] with the help of an arm). In [7] a passive decomposition method is shown for dynamic modeling and control of a quadrotor equipped with a redundant rigid arm. A kinematic control of a rigid manipulator attached on a quadrotor was recently experimented in [8]. Despite the fact that rigid manipulators are the most common tools, other types of aerial manipulators are also studied as, e.g., compliant actuators [4], [9].

Although aerial manipulators open new doors for various robotic tasks, their control is not trivial, since they are an interconnection of multiple nonlinear robotic systems. For this reason, it is important to analyze their system dynamics, and develop control algorithms dealing not only with the problem of tracking the outputs we are interested in, but also with the internal dynamics of the system and how they are coupled with each other. Such sophisticated control methods using deep system knowledge often require torque control of the manipulating arms [7], [9]. However small-size light-weight arms with torque-controlled actuators are either not available at a low price or not reliable enough in the torque control modality. On the other hand, position/velocity controlled servo motors are cheap and easy to reach, making them preferable to be used in the experimental setups. Thus, it is relevant to seek for a controller that, while taking into account the system dynamics, can also be used with light-weight manipulators built using off-the-shelf servo motors.

The *differential flatness* property allows to analytically compute all nominal system states and inputs from a desired differentiable trajectory of certain *flat outputs* [10]. In [2] we showed that, in the 2D vertical plane, for Protocentric Aerial Manipulators (PAMs), with rigid or compliant joints, the end-effector position together with the arm configuration is a flat output. This was exploited to develop an *exact linearization controller*, and the same was done in [9] for a single-link PAM equipped with a Variable Stiffness Actuator (VSA).

A drawback of such controllers is that they require very good knowledge of the model parameters and torque-controlled motors. As an alternative, in this paper, we present a *decentralized flatness-based control* for the output tracking problem of PAMs. This is different from the one used in [2], since it is not exactly linearizing, but uses differential flatness to compute the feed-forward terms in the control law. In [6] a decentralized controller was presented for the simple case of a single-DoF PAM (a PAM equipped with an arm having one Degree of Freedom). The controller presented here is instead thought for a more complex system where *i)* the *Center of Mass* (CoM) of the aerial vehicle can be different from the geometric center of its actuation, *ii)* the PAM can have any number of arms, each having any number of DoFs. The controller presented in this paper best performs for robot arms equipped with torque-controlled actuators. However, it is possible to obtain very good results also with kinematically controlled motors, thanks to a simple variant.

The paper is organized as follows. In Sec. II we introduce a generic dynamic model of an aerial manipulator. In Sec. III we describe the decentralized controller which requires the nominal states and inputs. In Sec. IV we show how to compute these nominal quantities for a PAM in 2D vertical plane. Then in Sec. V we experimentally test our controller comparing it with two other standard approaches. Final discussions are conducted in Sec. VI.

II. Model of an Aerial Manipulator

Consider a generic aerial manipulator, consisting of a *Vertical Take-off and Landing* vehicle, VTOL for short, equipped with m robotic arms. We denote with $\mathcal{F}_W$: $\{\mathrm{P_W}, \mathbf{x}_W, \mathbf{y}_W, \mathbf{z}_W\}$ and $\mathcal{F}_0$: $\{\mathrm{P}_0, \mathbf{x}_0, \mathbf{y}_0, \mathbf{z}_0\}$ the world inertial frame (chosen according to the common *North-East-Down* convention) and the VTOL frame, respectively, where P_0 is the CoM of the VTOL. The orien-

[1]LAAS-CNRS, Université de Toulouse, CNRS, Toulouse, France, antonio.franchi@laas.fr, marco.tognon@laas.fr

[2]Max Planck Institute for Biological Cybernetics, Spemanstr. 38, 72076, Tübingen, Germany. burak.yueksel@tuebingen.mpg.de

[3]Dipartimento di Ingegneria Informatica, Automatica e Gestionale, Sapienza Università di Roma, Via Ariosto 25, 00185 Roma, Italy. buondonno@diag.uniroma1.it

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

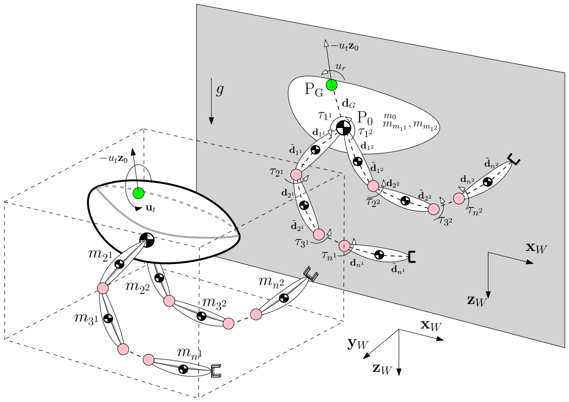

Fig. 1: Sketch of a two-arm protocentric aerial manipulator (PAM) and its projection on the $\mathbf{x}_W - \mathbf{z}_W$ plane. In the 3D model the axis of rotation for each joint is parallel to $\mathbf{y}_W$.

tation of $\mathcal{F}_0$ in $\mathcal{F}_W$ is described by the rotation matrix $\mathbf{R}_0 = [\mathbf{x}_0\ \mathbf{y}_0\ \mathbf{z}_0] \in \mathrm{SO}(3)$, parametrized by the roll-pitch-yaw angles $\boldsymbol{\eta} = [\phi_0\ \theta_0\ \psi_0]^T \in \mathbb{R}^3$ The μ-th arm possesses n^μ links and its joint configuration is described by the vector $\boldsymbol{\theta}_\mu = [\theta_{1^\mu}\ \cdots\ \theta_{n^\mu}]^T \in \mathbb{R}^{n^\mu}$. The total number of links of the generic aerial manipulator is $n = \sum_{\mu=1}^{m} n^\mu$. A particular example of aerial manipulator is shown on the left of Fig. 1. A set of generalized coordinates of the whole platform is $\mathbf{q} = \begin{bmatrix}\mathbf{p}_0^T & \boldsymbol{\eta}^T & \boldsymbol{\theta}^T\end{bmatrix}^T \in \mathbb{R}^{6+n}$, where $\mathbf{p}_0 = [x_0\ y_0\ z_0]^T \in \mathbb{R}^3$ is the position of P_0 expressed in $\mathcal{F}_W$ and $\boldsymbol{\theta} = [\boldsymbol{\theta}_1^T\ \cdots\ \boldsymbol{\theta}_m^T]^T \in \mathbb{R}^n$.

The ν-th joint of the μ-th manipulator (ν^μ-th joint) is actuated by the torque τ_{ν^μ}, and all actuation torques of the μ-th arm are listed as $\boldsymbol{\tau}_\mu = [\tau_{1^\mu}\ \cdots\ \tau_{n^\mu}]^T \in \mathbb{R}^{n^\mu}$. All joint torques are referred to as $\boldsymbol{\tau} = [\boldsymbol{\tau}_1^T\ \cdots\ \boldsymbol{\tau}_m^T]^T \in \mathbb{R}^n$. The VTOL is actuated by the thrust u_t, which is a scalar force value acting perpendicularly to the platform (in the direction of $-\mathbf{z}_0$), and by the torque $\mathbf{u}_r \in \mathbb{R}^3$. We denote with $\mathrm{P_G}$ the *center of actuation* of the PVTOL, where the *thrust vector* $\mathbf{f}_0 = -u_t\mathbf{z}_0 \in \mathbb{R}^3$ and the base torque $\mathbf{u}_r$ are applied; its constant position in $\mathcal{F}_0$ is denoted with $\mathbf{d}_G = [d_{G_x}\ d_{G_y}\ d_{G_z}]^T \in \mathbb{R}^3$. The overall control input of the whole aerial manipulator is $\mathbf{u} = \begin{bmatrix}u_t & \mathbf{u}_r^T & \boldsymbol{\tau}^T\end{bmatrix}^T \in \mathbb{R}^{4+n}$.

The classical Lagrangian dynamic model is given by

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{c}(\mathbf{q},\dot{\mathbf{q}}) + \mathbf{g} = \mathbf{G}(\boldsymbol{\eta})\mathbf{u} \qquad (1)$$

$$\mathbf{G}(\boldsymbol{\eta}) = \begin{bmatrix}-\mathbf{R}_0\mathbf{e}_3 & \mathbf{0}\\ \mathbf{0} & \mathbf{I}_{3+n}\end{bmatrix} \quad \mathbf{M}(\mathbf{q}) = \begin{bmatrix}m_s\mathbf{I}_3 & \mathbf{M}_{pr}\\ \mathbf{M}_{pr}^T & \mathbf{M}_r\end{bmatrix} \qquad (2)$$

where $\mathbf{M} \in \mathbb{R}^{(6+n)\times(6+n)}$ is the inertia matrix, m_s is the total mass of the system, $\mathbf{c}$ is the vector of Coriolis and centrifugal forces, $\mathbf{g}$ is the vector of gravity forces, and $\mathbf{G} \in \mathbb{R}^{(6+n)\times(4+n)}$ is the input matrix. Here, $\mathbf{I}_k$ is the $k \times k$ identity matrix, $\mathbf{e}_3$ is the third column of $\mathbf{I}_3$, and $\mathbf{0}$ is the zero matrix, of appropriate dimension.

Remark II.1. Notice that, since the control input has less elements $(4+n)$ than the configuration variables $(6+n)$, the system is underactuated. In particular we can not independently control the orientation of the aerial vehicle. Therefore, since the position of the end-effectors does not only depend on $\mathbf{p}_0$, ψ_0 and $\boldsymbol{\theta}$, but *also* on $\boldsymbol{\eta}$, it is not possible to plan exclusively for $\mathbf{p}_0$, ψ_0 and $\boldsymbol{\theta}$ if the position of the end-effectors has to be controlled.

Remark II.2. The inertia matrix $\mathbf{M}$ exhibits dynamic couplings between all elements of the state. This considerably complicates the control problem.

III. Decentralized Control

In this section we present a *decentralized* controller for a generic aerial manipulator in 3D. By decentralization, we mean that the controller does not consider the dynamic coupling of the complex system, *explicitly*. However, it does take the system dynamics *implicitly* into account, by using some *feed-forward* terms. Moreover, it uses *feed-back* terms for steering the system to a desired behavior while providing some robustness to the closed-loop system.

Now, say $\mathbf{y}^d(t)$ stands for the desired output of the system given in (1), and our objective is to track this output. If the desired output trajectory is consistent with the underactuation constraints it is in theory possible to find some corresponding *desired* states and inputs, i.e., $\mathbf{q}^d(t) = [\mathbf{p}_0^{d^T}\ \boldsymbol{\eta}^{d^T}\ \boldsymbol{\theta}^{d^T}]^T$, $\mathbf{u}^d = [u_t^d\ \mathbf{u}_r^{d^T}\ \boldsymbol{\tau}^{d^T}]^T$, $\dot{\mathbf{q}}^d(t) = [\dot{\mathbf{p}}_0^{d^T}\ \dot{\boldsymbol{\eta}}^{d^T}\ \dot{\boldsymbol{\theta}}^{d^T}]^T$, where we assume that these desired values are given; hence we will call them *feed-forward* terms. Notice that these terms can be computed as the *nominal* states and inputs using the differential flatness property of the system (e.g., as shown in Sec. IV). In fact, doing so, we will be using the knowledge of the system dynamics in a decentralized controller.

Now, let us first address the control of the aerial platform, in this case a VTOL. We develop a hierarchical approach based on the separation of the translational and rotational dynamics, which eventually tracks the position $\mathbf{p}_0^d$. First let us compute the controlled thrust vector as:

$$\mathbf{f}_0 = \mathbf{f}_0^d + \mathbf{f}_0^\star = \mathbf{f}_0^d + \mathbf{K}_{\mathbf{p}_0}^P(\mathbf{p}_0^d - \mathbf{p}_0) + \mathbf{K}_{\mathbf{p}_0}^D(\dot{\mathbf{p}}_0^d - \dot{\mathbf{p}}_0), \qquad (3)$$

where $\mathbf{K}_{\mathbf{p}_0}^P, \mathbf{K}_{\mathbf{p}_0}^D \in \mathbb{R}_{\geq 0}^{3\times 3}$. Notice that $\mathbf{f}_0$ is computed as a combination of the feed-forward terms $(\cdot^d)$, and the feedback term $(\cdot^\star)$ proportional to the state error of the system with respect to the nominal one. From the controlled thrust vector we can retrieve the commanded thrust as

$$u_t = -(\mathbf{R}_0\mathbf{e}_3)^T\mathbf{f}_0, \qquad (4)$$

and the commanded attitude as $\mathbf{R}_0^c = [\mathbf{x}_0^c\ \mathbf{y}_0^c\ \mathbf{z}_0^c]$ where

$$\mathbf{z}_0^c = \frac{\mathbf{f}_0}{\|\mathbf{f}_0\|}, \quad \mathbf{y}_0^c = \frac{\mathbf{z}_0^c \times \mathbf{e}_1}{\|\mathbf{z}_0^c \times \mathbf{e}_1\|}, \quad \mathbf{x}_0^c = \frac{\mathbf{y}_0^c \times \mathbf{z}_0^c}{\|\mathbf{y}_0^c \times \mathbf{z}_0^c\|}. \qquad (5)$$

This closes the *outer-loop control*. The controlled attitude is then passed to the *inner-loop control* as the desired attitude, to compute the controller torque as:

$$\begin{aligned}\mathbf{e}_{\mathbf{R}_0}^{[\times]} &= \tfrac{1}{2}(\mathbf{R}_0^{c\,T}\mathbf{R}_0 - \mathbf{R}_0^T\mathbf{R}_0^c),\ \ \mathbf{e}_{\boldsymbol{\omega}_0} = \mathbf{R}_0^T\mathbf{R}_0^d\boldsymbol{\omega}_0^d - \boldsymbol{\omega}_0\\ \mathbf{u}_r &= \mathbf{u}_r^d + \mathbf{u}_r^\star = \mathbf{u}_r^d + \mathbf{K}_{\mathbf{R}_0}^P\mathbf{e}_{\mathbf{R}_0} + \mathbf{K}_{\mathbf{R}_0}^D\mathbf{e}_{\boldsymbol{\omega}_0},\end{aligned} \qquad (6)$$

where, $\cdot^{[\times]}$ represents the skew operation, $\boldsymbol{\omega}_0 \in \mathbb{R}^3$ and $\boldsymbol{\omega}_0^d \in \mathbb{R}^3$ are the current and the desired angular velocities of the VTOL body in body-fixed frame (directly computed from $\boldsymbol{\eta}$, $\boldsymbol{\eta}^d$ and their time derivatives), $\mathbf{R}_0^d$ is the desired rotation matrix computed from $\boldsymbol{\eta}^d$, and $\mathbf{K}_{\mathbf{R}_0}^P, \mathbf{K}_{\mathbf{R}_0}^D \in \mathbb{R}_{\geq 0}^{3\times 3}$.

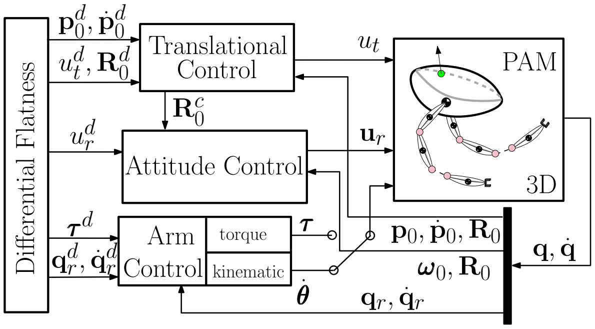
Differential Flatness
$\mathbf{p}_0^d, \dot{\mathbf{p}}_0^d$
$u_t^d, \mathbf{R}_0^d$
Translational Control
u_t
PAM
3D
$\mathbf{R}_0^c$
u_r^d
Attitude Control
$\mathbf{u}_r$
$\boldsymbol{\tau}^d$
$\mathbf{q}_r^d, \dot{\mathbf{q}}_r^d$
Arm Control
torque
kinematic
$\boldsymbol{\tau}$
$\dot{\boldsymbol{\theta}}$
$\mathbf{p}_0, \dot{\mathbf{p}}_0, \mathbf{R}_0$
$\boldsymbol{\omega}_0, \mathbf{R}_0$
$\mathbf{q}, \dot{\mathbf{q}}$
$\mathbf{q}_r, \dot{\mathbf{q}}_r$

Fig. 2: Control of the PAM depicted in Fig. 1, using the decentralized controller, exploiting its differential flatness property.

Now, let us give the control of the generic ν^μ-th joint, to track the relative desired angle. For a torque-controlled motor, we design the control law based on a PD strategy as

$$\tau_{\nu^\mu} = \tau_{\nu^\mu}^d + \tau_{\nu^\mu}^\star = \tau_{\nu^\mu}^d + k_{\nu^\mu}^P(\theta_{\nu^\mu}^d - \theta_{\nu^\mu}) + k_{\nu^\mu}^D(\dot{\theta}_{\nu^\mu}^d - \dot{\theta}_{\nu^\mu}), \quad (7)$$

where $k_{\nu^\mu}^P, k_{\nu^\mu}^D \in \mathbb{R}_{\geq 0}$. This controller ensures the best performances. Nonetheless, for kinematically controlled motors, the commanded velocity can be given as

$$\dot{\theta}_{\nu^\mu} = \dot{\theta}_{\nu^\mu}^d + k_{\nu^\mu}^P(\theta_{\nu^\mu}^d - \theta_{\nu^\mu}). \quad (8)$$

A schematic representation of the controller is shown in Fig. 2. Now let us show how to use the differential flatness of a specific type of aerial manipulator for computing the feed-forward terms of this controller.

IV. Flatness and Control of PAMs

In Sec. III we presented a *decentralized* controller for aerial manipulators. As mentioned, this controller requires an algorithm computing all *nominal* states and inputs.

In our previous work [2] we have introduced the notion of *protocentricity*. A protocentric aerial manipulator (PAM) is characterized by all manipulator arms being attached at P_0. We studied the properties of such systems in the 2D vertical plane, and we found that they are differentially flat w.r.t. a set of flat outputs given by the angles of the manipulator links with respect to $\mathbf{z}_W$ and the position of any of the end-effectors (or, alternatively, the position of P_0), thus making PAMs of particular interest for aerial manipulation.

The above discussion suggests the possibility to control the motion of a PAM in the $\mathbf{x}_W - \mathbf{z}_W$ plane by combining the controller presented in Sec. III and the flatness property proven in [2]. In order to do this, consider a PAM in 3D, where the motion of all manipulators is constrained to a plane, i.e., $y_0 = 0$, $\mathbf{y}_0^T\mathbf{z}_W = \mathbf{y}_0^T\mathbf{x}_W = 0$, and each joint axis is parallel to $\mathbf{y}_0$. Hence the projection of the PAM on the $\mathbf{x}_W - \mathbf{z}_W$ plane can be modeled as a planar mechanical system. A sketch of such design is depicted in Fig. 1 (right).

In 2D, we give the generalized coordinates of a PAM as $\mathbf{q}_2 = \begin{bmatrix}\mathbf{p}_{0_{xz}}^T & \theta_0 & \mathbf{q}_r^T\end{bmatrix}^T \in \mathbb{R}^{3+n}$, where $\mathbf{p}_{0_{xz}} = [x_0\ z_0]^T \in \mathbb{R}^2$ is the position of P_0 in the $\mathbf{x}_W - \mathbf{z}_W$ plane, θ_0 is the pitch, and $\mathbf{q}_r = [\mathbf{q}_{r^1}^T \ \cdots \ \mathbf{q}_{r^m}^T]^T \in \mathbb{R}^n$ is the vector combining the *absolute* orientations of each joint of every arm, with $\mathbf{q}_{r^\mu}^T = [\theta_{01^\mu} \ \cdots \ \theta_{0n^\mu}]^T \in \mathbb{R}^{n^\mu}$ written for the μ-th manipulating arm, where $\theta_{0k^\mu} = \theta_0 + \sum_{\nu=1}^{k}\theta_{\nu^\mu}$ is the absolute orientation of the ν^μ-th joint. The set of inputs is $\mathbf{u}_2 = [u_t \ u_r \ \boldsymbol{\tau}^T]^T \in \mathbb{R}^{2+n}$, where the *scalar* $u_r \in \mathbb{R}$ is the magnitude of the base torque applied about the axis passing through P_G and parallel to $\mathbf{y}_0$. Notice that the PAM in 2D is also underactuated, as the one in 3D. Finally, we define $\mathbf{p}_{e^\mu_{xz}}$ as the position of the μ-th end-effector in the 2D plane. The flatness property can be then phrased as:

Fact 1. *Thanks to protocentricity,* $\mathbf{y} = [\mathbf{p}_{0_{xz}}^T \ \ \mathbf{q}_r^T]^T \in \mathbb{R}^{(n+2)}$ *is a flat output of a PAM modeled in 2D. Hence,* $\mathbf{y}_e = [\mathbf{p}_{e^\mu_{xz}}^T \ \mathbf{q}_r^T]^T \in \mathbb{R}^{(n+2)}$ *is also a flat output, for any* μ*.*

In the following, we will show the computation of the nominal states and control inputs as sole functions of the flat outputs trajectories, assuming each joint is actuated via a motor rotating around the corresponding joint axis.

A. Computation of Nominal Inputs and States

Define $\mathbf{p}_c \in \mathbb{R}^2$ as the position, in the $\mathbf{x}_W - \mathbf{z}_W$ plane, of the CoM of the whole PAM (i.e., flying base plus all the manipulators) . The (translational) dynamics of $\mathbf{p}_c \in \mathbb{R}^2$, see [2], implies that the vector

$$\mathbf{w} = \mathbf{w}(\mathbf{y}, \dot{\mathbf{y}}, \ddot{\mathbf{y}}) = \ddot{\mathbf{p}}_c - [0 \ g]^T = [w_x \ w_z]^T \in \mathbb{R}^2, \quad (9)$$

is equal to $\mathbf{w} = -\frac{u_t}{m_s}[\sin(\theta_0) \ \cos(\theta_0)]^T$, g being the gravitational constant. Hence

$$\begin{aligned} \theta_0 &= \theta_0(\ddot{\mathbf{p}}_c) = \mathrm{atan2}(-w_x, -w_z) \\ \dot{\theta}_0 &= \dot{\theta}_0(\ddot{\mathbf{p}}_c, \dddot{\mathbf{p}}_c) = \tfrac{w_z\dot{w}_x - w_x\dot{w}_z}{w_x^2 + w_z^2} \\ \ddot{\theta}_0 &= \tfrac{\ddot{w}_x w_z - w_x\ddot{w}_z}{w_x^2 + w_z^2} - \tfrac{2[(w_z^2 - w_x^2)\dot{w}_x\dot{w}_z + (\dot{w}_x^2 - \dot{w}_z^2)w_x w_z]}{(w_x^2 + w_z^2)^2} \\ u_t &= u_t(\ddot{\mathbf{p}}_c) = m_s\|\mathbf{w}\|. \end{aligned} \quad (10)$$

Therefore, we need to compute the second to fourth time derivative of $\mathbf{p}_c$ from the flat outputs. To this end, define $\mathrm{P}_{\mathrm{M}_{\nu^\mu}}$ and P_{ν^μ} as the CoM of the ν^μ-th motor and link, with coordinates $\mathbf{p}_{m_{\nu^\mu}} \in \mathbb{R}^2$ and $\mathbf{p}_{\nu^\mu} \in \mathbb{R}^2$ in $\mathcal{F}_W$, we have

$$\mathbf{p}_c = \frac{1}{m_s}\left(m_0\mathbf{p}_{0_{xz}} + \sum_{j=1}^{m}\left(\sum_{i=1}^{n^j}(m_{ij}\mathbf{p}_{ij} + m_{m_{ij}}\mathbf{p}_{m_{ij}})\right)\right), \quad (11)$$

where $m_s = m_0 + \sum_{j=1}^{m}\left(\sum_{i=1}^{n^j} m_{ij} + m_{m_{ij}}\right)$, with m_0, m_{ν^μ} and $m_{m_{\nu^\mu}}$ the masses of the aerial platform and the ν^μ-th link and motor, respectively. This expression is easily differentiated from the second up to the fourth time derivative, as a function of the corresponding derivatives of $\mathbf{p}_{ij}$ and $\mathbf{p}_{m_{ij}}$. These are computed as follows. For each link, define the local frame $\mathcal{F}_{\nu^\mu}$, attached to the link CoM and with orientation $\mathbf{R}_{0\nu^\mu}$ w.r.t. $\mathcal{F}_W$, where $\mathbf{R}_* \in \mathrm{SO}(2)$ denotes a rotation of angle $\theta_* \in \mathbb{R}$ in the plane. The constant position of $\mathrm{P}_{\mathrm{M}_{\nu^\mu}}$ and $\mathrm{P}_{\mathrm{M}_{(\nu+1)^\mu}}$ in $\mathcal{F}_{\nu^\mu}$ is denoted with $-\mathbf{d}_{\nu^\mu} \in \mathbb{R}^2$ and $\tilde{\mathbf{d}}_{\nu^\mu} \in \mathbb{R}^2$, respectively. Thus we obtain

$$\mathbf{p}_{m_{\nu^\mu}}(\mathbf{y}) = \mathbf{p}_{0_{xz}} + \sum_{i^\mu=1}^{\nu^\mu-1}\mathbf{R}_{0i^\mu}\bar{\mathbf{d}}_{i^\mu} \quad (12)$$

$$\mathbf{p}_{\nu^\mu}(\mathbf{y}) = \mathbf{p}_{m_{\nu^\mu}} + \mathbf{R}_{0\nu^\mu}\mathbf{d}_{\nu^\mu}, \quad (13)$$

which are easily differentiated to obtain the second to fourth time derivative, with $\bar{\mathbf{d}}_* = \mathbf{d}_* + \tilde{\mathbf{d}}_*$. Here, (12) is used in (13), both are used in (11), and finally (9) and (10) are applied.

The ν^{μ}-th nominal torque is given recursively as

$$\tau_{\nu^{\mu}} = \tau_{\nu^{\mu}+1} + \mathbf{m}_{0\nu^{\mu}}^{T}(\theta_{0\nu^{\mu}})\ddot{\mathbf{p}}_{0_{xz}} + c_{r_{\nu^{\mu}}}(\mathbf{q}_{r^{\mu}}, \dot{\mathbf{q}}_{r^{\mu}}) + \mathbb{J}_{\nu^{\mu}}\ddot{\theta}_{0\nu^{\mu}} + g_{r_{\nu^{\mu}}}(\theta_{0\nu^{\mu}}) + \sum_{l=1, l\neq\nu^{\mu}}^{n^{\mu}} m_{l\nu^{\mu}}(\theta_{0l^{\mu}}, \theta_{0\nu^{\mu}})\ddot{\theta}_{0l^{\mu}}, \quad (14)$$

where $\tau_{n^{\mu}+1} = 0$, while $c_{r_{\nu^{\mu}}}$ and $g_{r_{\nu^{\mu}}}$ are the ν^{μ}-th elements of the Coriolis and gravitational force vectors acting on the center of the ν^{μ}-th link, respectively. The coupling term between the aerial platform and the ν^{μ}-th joint is given with $\mathbf{m}_{0\nu^{\mu}}$, and it is an element of $\mathbf{M}_{pr}$ in (2). Both terms $\mathbb{J}_{\nu^{\mu}}$ and $m_{l\nu^{\mu}}$ are inertial terms of $\mathbf{M}_r$ in (2), where the former one is the diagonal element and the latter is the coupling term between the l-th and the ν-th joint of the μ-th manipulator. Using these quantities, it is possible to compute $\tau_{\nu^{\mu}} = \tau_{\nu^{\mu}}(\mathbf{y}, \dot{\mathbf{y}}, \ddot{\mathbf{y}})$. Then the flying base torque is computed:

$$u_r = u_r(\mathbf{y}, \dot{\mathbf{y}}, \ddot{\mathbf{y}}, \dddot{\mathbf{y}}, \ddddot{\mathbf{y}}) = J_0\ddot{\theta}_0 + \sum_{j=1}^{m} \tau_{1^j} - d_{G_x}u_t, \quad (15)$$

where J_0 is the inertia of the aerial platform and d_{G_x} is the constant position of $\mathrm{P_G}$ with respect to P_0 in $\mathcal{F}_0$ along $\mathbf{x}_0$.

Finally, the relative joint angles can be easily obtained from the absolute ones and the value of θ_0. Hence, all state and input variables can be computed from (or coincide with) the flat outputs $\mathbf{y}$ and their time derivatives up to the fourth order. We refer the meticulous reader to [2], [11] for the details of the notation and the computations.

The 2D flatness can be exploited in 3D by setting

$$\mathbf{R}_0^d = \begin{bmatrix} c_{\theta_0^d} & 0 & s_{\theta_0^d} \\ 0 & 1 & 0 \\ -s_{\theta_0^d} & 0 & c_{\theta_0^d} \end{bmatrix}, \qquad \boldsymbol{\omega}_0^d = \dot{\theta}_0^d\mathbf{e}_2 \quad (16)$$

$$\mathbf{f}_0^d = -u_t^d\mathbf{R}_0^d\mathbf{e}_3, \qquad \mathbf{u}_r^d = u_r^d\mathbf{e}_2, \qquad y_0^d \equiv 0. \quad (17)$$

Notice that $\theta_0^d, \dot{\theta}_0^d, u_t^d, \tau_{\nu^{\mu}}^d, u_r^d$ are computed as in (10), (14) and (15) for $\mathbf{y}^d$. Then we can use these values as the feed-forward terms of the controller presented in Sec. III. See Fig. 2 for a representation of the overall control method.

V. Experimental Validation

In this section we show the results of some experiments aimed at validating the controller proposed in this paper. Furthermore, we analyze its performances by comparing it with other standard control techniques.

The aerial manipulator used for the experiments (see Fig. 3) consists of a Quadrotor VTOL and a 2-DoF manipulator arm. The light-weight arm design was inspired by the work in [12]. A big difference of our design is that all the actuators are placed at the base of the arm, rigidly attached to the VTOL. The first joint is directly connected to its actuator (a dynamixel MX-64 motor), while the second one is connected to its motor (a dynamixel MX-28) via a metal-reinforced plastic belt. Such design allows us to reduce the mass of each joint and in particular their inertia. This in turn lets us use a relatively small and weak quadrotor (diameter $0.4\,[\mathrm{m}]$, maximum thrust per propeller of $5.26\,[\mathrm{N}]$) with respect to the platforms normally used for arms of similar length (e.g., in [8] a larger octorotor is used). For the physical parameters of the system, please refer to Tab. I. Since the motors cannot be controlled in torque but at best in velocity, we used (8) slightly modified to cope with the fact that the second link is not directly attached to its motor.

Phys. param.	VTOL	1th-Link	2th-Link
Mass $[\mathrm{Kg}]$	1.3	0.145	0.123
Rot. inertia $[\mathrm{Kgm}^2]$	0.03	$1.2 \cdot 10^{-3}$	$0.9 \cdot 10^{-3}$
Length $[\mathrm{m}]$	0.4 (diam.)	0.29	0.25

Controller	$\mathbf{K}_{\mathbf{p}_0}^P$	$\mathbf{K}_{\mathbf{p}_0}^D$	$\mathbf{K}_{\mathbf{R}_0}^P$	$\mathbf{K}_{\mathbf{R}_0}^D$	k_1^P	k_2^P
Gain	$12\mathbf{I}_3$	$7\mathbf{I}_3$	$3\mathbf{I}_3$	$0.3\mathbf{I}_3$	1.8	1.6

Traj. Param.	$a_{\mathbf{p}_0}^x$ [m]	$a_{\mathbf{q}_r}^1$ [°]	$a_{\mathbf{q}_r}^2$ [°]	ω [rad/s]
(a)	0	30	60	$2\pi/3$
(b)	0.5	-40	-70	$2\pi/3$
(c)	0.5	40	70	$2\pi/3$

TABLE I: Starting from the top: physical parameters of the real system; controller gains; and the parameters of the three trajectories. Length and the inertia are the one on the 2D vertical plane.

The aerial vehicle hardware is the one of a Mikrokopter quadrotor endowed with an IMU, and four brushless motor controllers (BLDC ESC) regulating the propeller speed using an in-house developed closed-loop speed controller [13].

The control law presented in Sec. III, implemented in Matlab–Simulink, runs on a desktop PC sending the commanded propeller velocities at $500\,[\mathrm{Hz}]$ and the commanded arm motor velocities at $250\,[\mathrm{Hz}]$ through a serial communication. The gains used for the controller are given in Tab. I. The control loop is then closed based on the measurements of: i) the position and attitude of the vehicle provided at $1\,[\mathrm{kHz}]$ by a UKF that fuses the Motion Capture (Mo-Cap) System measurements at $120\,[\mathrm{Hz}]$ with the IMU measurements at $1\,[\mathrm{kHz}]$; ii) the linear and angular velocities of the vehicle provided by the same UKF filter at $1\,[\mathrm{KHz}]$; and finally iii) the position and velocity of the arm motors provided by their internal absolute encoders at $250\,[\mathrm{Hz}]$. In order to read the motor values corresponding to zero joint angles, a calibration procedure is implemented *once*, using the Mo-Cap markers on the manipulator arm (see Fig. 3).

We tested the proposed controller with a parametric and multi-DoF sinusoidal-like trajectory, i.e., $\mathbf{y}^d = \left[\mathbf{p}_{0_{xy}}^{d^T}\ \mathbf{q}_r^{d^T}\right]^T = \left[a_{\mathbf{p}_0}^x\ 0\ a_{\mathbf{q}_r}^1\ a_{\mathbf{q}_r}^2\right]^T \sin(\omega t)$, for three different sets of parameters corresponding to three qualitatively different task trajectories:

(a) the arm oscillates and the quadrotor remains still,
(b) the arm and quadrotor oscillate with opposite phases,
(c) the arm and quadrotor oscillate with the same phase.

These task trajectories are understandable from Fig. 4, and the parameters of the trajectories are given in Tab. I.

For each of the three task trajectory, we compared the performance of the proposed controller using three different types of *feedforward methods*:

1) *minimal compensation*: on the quadrotor side only the total mass is compensated, i.e., $u_t^d = -m_s\mathbf{e}_3^T\mathbf{R}_0\mathbf{e}_3$. In this way the VTOL and the arm virtually are treated as two independent systems (even if in practice they are not).
2) *static compensation*: only the static effects due to gravity are compensated, i.e., the nominal state and inputs are computed considering all the derivatives of the desired trajectory equal to zero, i.e., $\mathbf{y}^{d^{(l)}} = \mathbf{0}$ for $l = 1, \ldots, 4$,

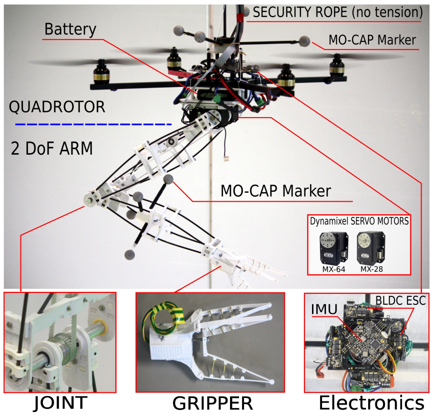

Fig. 3: Experimental setup of the aerial manipulator. A quadrotor VTOL is equipped with a 2 DoF manipulating arm.

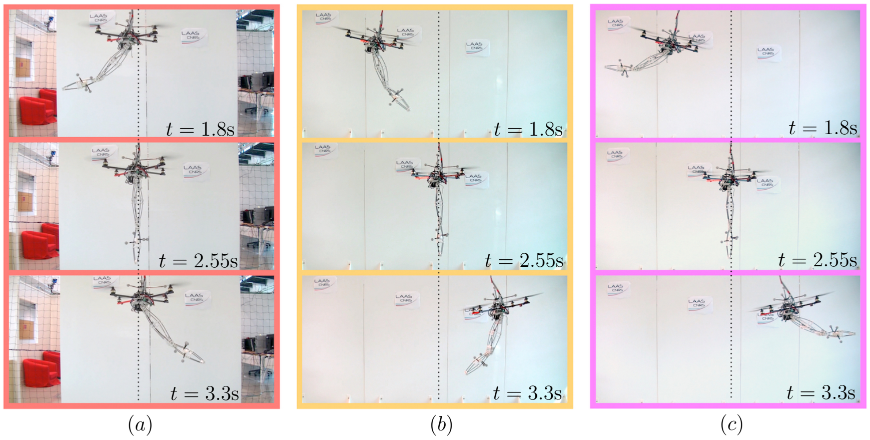

Fig. 4: Nine moments from the experiments using method 3 (dynamic compensation). From left to right trajectories (a), (b) and (c) are shown, respectively. From top to bottom the start, intermediate and end moments of half period of each the trajectory are shown.

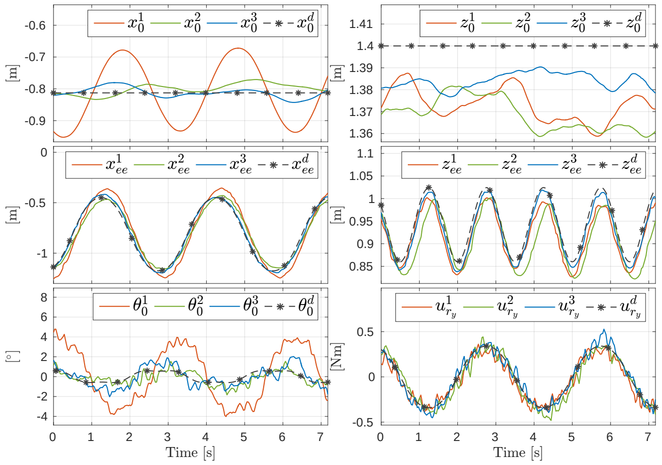

Fig. 5: Experimental results for trajectory (a) (see Fig. 4.a). In all plots, the flat outputs and the nominal states/inputs are depicted with starred black dashed lines, while red, green and blue curves show the results of the controller with *minimal compensaton*, *static compensation*, and *dynamic compensation*, respectively. A more detailed version of the plots can be found in [11].

($\mathbf{y}^d \neq \mathbf{0}$). This method is often used for the control of aerial manipulators for so called *quasi-static* operations, in order to partially compensate the effects of the manipulator on the aerial vehicle.

3) *dynamic compensation*: this corresponds to our proposed method where we exploit the flatness of the system. We compute the nominal states and inputs as functions of the desired trajectory to be tracked, and provide them to the controller as explained in Sec. III and IV.

The performances of these three methods are shown in Figs. 5, 6 and 7, and Tab. II. We encourage the reader to watch the attached video in order to appreciate even better the nature and results of the performed tests.

Looking at the tracking of the desired VTOL CoM and end-effector position one can see that the minimal compensation (method 1) shows good tracking performances (similar to the one with our method) only for trajectory (b). On the other hand, for trajectories (a) and (c) the tracking error is considerably larger than the one with dynamic compensation.

For the static compensation (method 2), the tracking performances result to be good (similar to the one with our proposed method 3) only for trajectory (a). Indeed, since trajectory (a) is the less dynamic one (quadrotor not moving), the static compensation is enough to obtain good performances. However, for more dynamical trajectories as (b) and (c) the performances rapidly get worse.

		method 1	**method 2**	**method 3**
Traj. (a)	$\bar{e}_{track}$	0.058	0.033	0.014
	$\sigma_{\bar{e}_{track}}$	0.021	0.012	0.005
Traj. (b)	$\bar{e}_{track}$	0.056	0.174	0.054
	$\sigma_{\bar{e}_{track}}$	0.023	0.085	0.023
Traj. (c)	$\bar{e}_{track}$	0.171	0.209	0.066
	$\sigma_{\bar{e}_{track}}$	0.078	0.106	0.034

TABLE II: Mean tracking error $\bar{e}_{track}$ and relative standard deviation $\sigma_{\bar{e}_{track}}$ for each trajectory and control method.

On the contrary, our proposed method 3 shows good tracking performances for all the types of trajectories validating the fact that flatness-based dynamic compensation is a good control strategy for both static and dynamic trajectories. Moreover, thanks to the feedback, the controller is robust enough to the non-perfect protocentricity of the real system. Indeed in the testbed used during the experiments, along the z-axis of $\mathcal{F}_0$ there is a non-zero offset of about $6\,[\mathrm{cm}]$ between the position of the CoM of the VTOL and the first joint. Nevertheless the controller is able to keep the tracking error small even for dynamic trajectories. For the interested reader, the effects of the non-protocentricity are investigated in the technical report [11] by numerical simulations.

In addition to the good results obtained with our method, it is also very interesting to notice that for trajectory (b), method 1, based on minimal compensation, is better than method 2, based on static compensation, in terms of tracking error. This brings us two interesting results.

The first one is highlighting how for some dynamic trajectories it is more suitable to just compensate the effect of the total mass rather than try to compensate the static configuration only. Indeed the last compensation term could result considerably wrong since it is computed for a different condition. This error in the compensation leads to undesired effects and in turn to a large tracking error, as seen in Fig. 6.

The second fascinating aspect is that for some particular

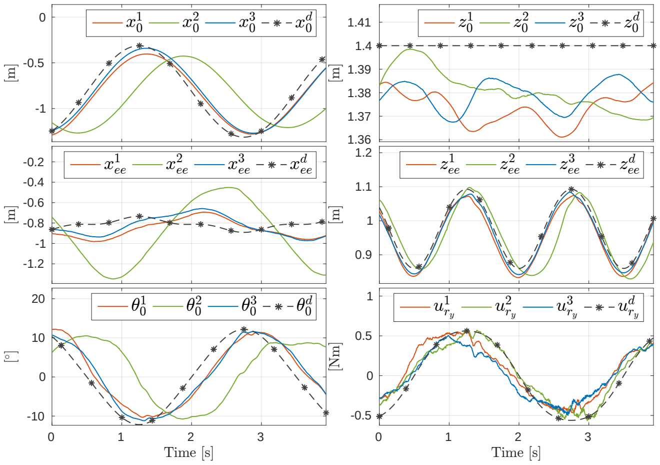

Fig. 6: Experimental results for trajectory (b) (see Fig. 4.b). The proposed controller (blue) always achieves a better performance.

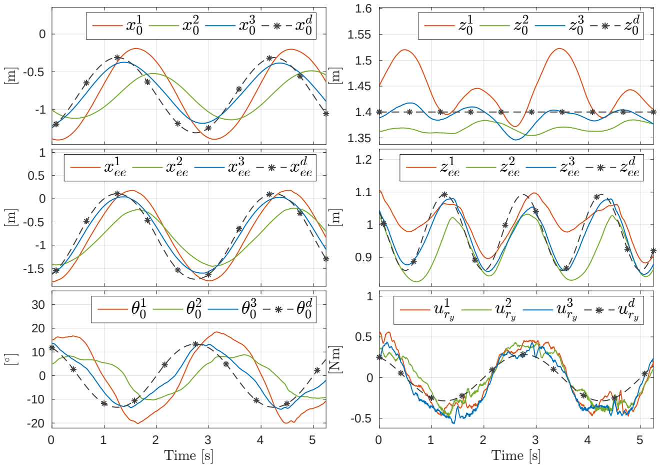

Fig. 7: Experimental results for trajectory (c) (see Fig. 4.c). The proposed controller (blue) again outperforms the other methods.

dynamic trajectories, as for trajectory (b), the arm could help the aerial vehicle to move toward the desired direction, implying the need of smaller compensations and in turn of smaller control efforts. Indeed, looking at Fig. 4.b one could notice the similarity between: i) the motion of the robotic arm and the one of the legs of a person sitting on a swing when trying to enhance the angular motion of the swing; ii) the thrust force and the tension along the cables attached to the swing to win the gravity and the centrifugal terms. This is why for trajectory (b) the minimal compensation shows similar results to the ones obtained with our method. Based on this consideration we believe that the studies on optimal trajectory generation become even more fundamental to achieve aerial manipulation tasks exploiting the dynamic properties (such as the flatness) of the systems. However this promising topic is left as future work.

VI. Conclusions

In this paper we have presented a dynamic decentralized controller for a specific type of aerial manipulators: VTOLs equipped with any number of manipulator arms, each having any number of rigid links. The aerial manipulator is assumed to be protocentric, namely all the manipulator arms are attached to the CoM of the flying base. Using the differential flatness property of PAMs (in 2D), we showed how to compute the nominal states and inputs of the system analytically in advance, and use it to track dynamic maneuvers for an aerial manipulator in the 3D space. The experimental results are in line with the proposed theory showing the advantage of using the differential flatness of aerial manipulators.

This work can be extended in many directions. One is to study the differential flatness property directly in 3D. Another is finding ways to relax the protocentric assumption. The third one is performing experiments using more than one manipulator arm.

Acknowledgements

We thank Anthony Mallet (LAAS-CNRS) for his contribution in the software architecture of the experiments, and Markus Ryll (LAAS-CNRS) and Quentin Sable (LAAS-CNRS/INSA-Strasbourg) for the design and development of the lightweight arm used in the experiments.

References

[1] A. Franchi, C. Masone, H. H. Bülthoff, and P. Robuffo Giordano, "Bilateral teleoperation of multiple UAVs with decentralized bearing-only formation control," in *2011 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, San Francisco, CA, Sep. 2011, pp. 2215–2222.

[2] B. Yüksel, G. Buondonno, and A. Franchi, "Differential flatness and control of protocentric aerial manipulators with any number of arms and mixed rigid-/elastic-joints," in *2016 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Daejeon, South Korea, Oct. 2016, pp. 561–566.

[3] S. Kim, S. Choi, and H. J. Kim, "Aerial manipulation using a quadrotor with a two dof robotic arm," in *2013 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Tokyo, Japan, November 2013, pp. 4990–4995.

[4] B. Yüksel, S. Mahboubi, C. Secchi, H. H. Bülthoff, and A. Franchi, "Design, identification and experimental testing of a light-weight flexible-joint arm for aerial physical interaction," in *2015 IEEE Int. Conf. on Robotics and Automation*, Seattle, WA, May 2015, pp. 870–876.

[5] R. Spica, A. Franchi, G. Oriolo, H. H. Bülthoff, and P. Robuffo Giordano, "Aerial grasping of a moving target with a quadrotor UAV," in *2012 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Vilamoura, Portugal, Oct. 2012, pp. 4985–4992.

[6] J. Thomas, J. Polin, K. Sreenath, and V. Kumar, "Avian-Inspired Grasping for Quadrotor Micro UAVs," in *2013 ASME Int. Design Engineering Technical Conf. and Computers and Information in Engineering Conf.*, Portland, OR, Aug. 2013.

[7] H. Yang and D. J. Lee, "Dynamics and control of quadrotor with robotic manipulator," in *2014 IEEE Int. Conf. on Robotics and Automation*, Hong Kong, China, May. 2014, pp. 5544–5549.

[8] G. Muscio, F. Pierri, M. A. Trujillo, E. Cataldi, G. Giglio, G. Antonelli, F. Caccavale, A. Viguria, S. Chiaverini, and A. Ollero, "Experiments on coordinated motion of aerial robotic manipulators," in *2016 IEEE Int. Conf. on Robotics and Automation*, Stockholm, Sweden, May. 2016, pp. 1224–1229.

[9] B. Yüksel, N. Staub, and A. Franchi, "Aerial robots with rigid/elastic-joint arms: Single-joint controllability study and preliminary experiments," in *2016 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Daejeon, South Korea, Oct. 2016, pp. 1667–1672.

[10] R. M. Murray, M. Rathinam, and W. Sluis, "Differential flatness of mechanical control systems: A catalog of prototype systems," in *ASME Int. Mechanical Eng. Congress and Exposition*, San Francisco, CA, Nov. 1995.

[11] M. Tognon, B. Yüksel, G. Buondonno, and A. Franchi, "Protocentric aerial manipulators: Explicit computations and simulations," LAAS-CNRS, Tech. Rep. hal-01474695, Feb 2017. [Online]. Available: https://hal.archives-ouvertes.fr/hal-01474695

[12] R. Cano, C. Perez, F. Pruano, A. Ollero, and G. Heredia, "Mechanical Design of a 6-DOF Aerial Manipulator for assembling bar structures using UAVs," in *2nd IFAC Work. on Research, Education and Development of Unmanned Aerial Systems*, 2013.

[13] 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.