Dynamic Decentralized Control for Protocentric Aerial Manipulators

Marco Tognon, Burak Yüksel, Gabriele Buondonno, Antonio Franchi

Dynamic Decentralized Control for Protocentric Aerial Manipulators

Marco Tognon[1] Burak Yüksel[2] Gabriele Buondonno[3] 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, a nominal input/state generator that takes into account the full-body nonlinear and coupled dynamics of the system, and a decentralized feedback controller acting on the actuated degrees of freedom that confers the needed robustness to the closed-loop system. We show how to apply the method to Protocentric Aerial Manipulators (PAM) by first using their differential flatness property on the vertical 2D plane in order to generate dynamical input/state trajectories, then statically extending the 2D structure to the 3D, and finally closing the loop with a decentralized controller having the dual task of both ensuring the preservation of the proper static 3D immersion and tracking the dynamic trajectory on the vertical plane. We demonstrate that the proposed controller is able to precisely track dynamic trajectories when implemented on a standard hardware composed by a quadrotor and a robotic arm with servo-controlled joints (even if no torque control is available). Comparative experiments clearly show the benefit of using the nominal input/state generator, and also the fact that the use of just static gravity compensation might surprisingly perform worse, in dynamic maneuvers, than the case of no compensation at all. We complement the experiments with additional realistic simulations testing the applicability of the proposed method to slightly non-protocentric aerial manipulators.**

I. INTRODUCTION

Aerial robots are attracting increasing interest from scientists in the robotics society, due to their agility and great workspace. Especially multi-rotors (quadrotors, hexarotors, etc.) are nowadays available to a broad public, thanks to their symmetric design allowing simplified mathematical models to be used, their light-weight and highly efficient electronics, and also their affordable prices. The control of such robots has been a challenge for the researchers since over two decades, concerning tracking of complex trajectories [1], human-robot interaction [2], and so on, which paved the way for surveillance and monitoring-like tasks.

Manipulator robots are well known mechanisms that have been studied intensively for a very long time [3], especially on the modeling and control of fixed-base manipulators.

An aerial manipulator is a robotic system, which has the capacity to fly and at the same time manipulate objects in its environment by applying reasonable forces and torques. Most commonly they consist of a flying robot and at least one manipulator arm [4]. In such robots, the great workspace and the agility of aerial robots meet with the dexterity of conventional manipulators. This breaks ground to many different robotic applications, e.g., pick and place of objects [5], aerial physical interaction [6], and aerial grasping [7]. In [8] 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 [9]. Despite the fact that rigid manipulators are the most common tools used so far, other types of aerial manipulators are also studied as, e.g., compliant actuators [6], [10].

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 tracking problem of 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 the torque control of the manipulating arms [8], [10]. 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. For this reason, it is relevant to seek for a controller that, while taking into account the system dynamics, can also be used together with light-weight manipulators built using off-the-shelf servo motors.

The *differential flatness* property could turn handy in such cases, since it allows to analytically compute in advance all (nominal) states and control inputs of the system from the so called *flat outputs* and their derivatives up to a finite order [11]. However the end-effector configuration of an aerial manipulator or the center of the flying base on which the aerial manipulator is mounted (where many of the sensors are typically hosted) are not in general part of a flat output of the system. In [12] we have shown that, in the 2D vertical plane, Protocentric Aerial Manipulators (PAMs) with any number of manipulating arms, each having any number of joints with either rigid or compliant actuators are *differentially flat systems*. In [10], the authors showed that this property can be employed together with an *exact linearizing controller* for tracking control of a quadrotor equipped with an elastic-joint arm - using a Variable Stiffness

[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.

Actuator (VSA). Such controller requires however a good knowledge of the model parameters, and motors with torque-control modality.

Alternatively, in this paper, we present and experimentally validate a *decentralized flatness-based control* for the output tracking problem of PAMs. Using their flatness property, as studied in [12], we develop a controller which can steer the outputs of the system to their desired values. The method is compared with other two standard techniques as well.

Notice that the controller presented here is different from the one used in [12], since there the authors presented an *exact linearizing controller*, while the controller presented here is decentralized and uses the differential flatness property (see Sections III and IV). In [7] 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 manipulator arms, each having any number of DoFs.

The controller presented in this paper best performs for the robot arms equipped with torque-controlled actuators. However, it is possible to obtain very good results also with kinematically controlled motors, by employing a simple variant. This is a great advantage over the torque-based controllers like the ones in [10], [12], as it makes it implementable to the readily available hardware with less effort, while still fully considering the dynamics of the nonlinear system, unlike the controllers only based on the system kinematics as, e.g., [9].

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 for such system, which requires the computation of its nominal states and inputs. In Sec. IV we show how to compute these nominal states and inputs for a PAM in 2D vertical plane, using the differential flatness of the system. Then in Sec. V we experimentally test our controller using an aerial manipulator which consists of a quadrotor VTOL and a 2-DoF rigid manipulating arm for three different trajectories. The proposed method is also compared with other two standard approaches. Finally, we conclude our work in Sec. VI with remarks on the future directions. Supplementary simulation results are given in a technical report[1] for the limited size of the paper.

II. Model of an Aerial Manipulator

Consider a generic aerial manipulator, consisting of a multi-rotor aerial platform (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, and the frame attached to the VTOL, respectively, where P_0 is the CoM of the VTOL. The world frame is chosen according to the common *North-East-Down* (NED) convention, and the orientation 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 the generic aerial manipulator model described so far is shown on the left of Fig. 1. A possible set of generalized coordinates of the whole platform is

$$\mathbf{q} = \begin{bmatrix}\mathbf{p}_0^T & \boldsymbol{\eta}^T & \boldsymbol{\theta}^T\end{bmatrix}^T \in \mathbb{R}^{6+n}, \tag{1}$$

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 of the overall system) is controlled by the actuation torque τ_{ν^μ}, and the vector of all actuation torques of the μ-th arm is denoted as $\boldsymbol{\tau}_\mu = [\tau_{1^\mu}\ \cdots\ \tau_{n^\mu}]^T \in \mathbb{R}^{n^\mu}$. All joint torques are collectively referred to as $\boldsymbol{\tau} = [\boldsymbol{\tau}_1^T\ \cdots\ \boldsymbol{\tau}_m^T]^T \in \mathbb{R}^n$. The flying base (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, whose 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 *thrust vector* $\mathbf{f}_0 = -u_t\mathbf{z}_0 \in \mathbb{R}^3$ and the base torque $\mathbf{u}_r$ are applied at and around $\mathrm{P_G}$, respectively. The overall control input of the whole aerial manipulator is

$$\mathbf{u} = \begin{bmatrix}u_t & \mathbf{u}_r^T & \boldsymbol{\tau}^T\end{bmatrix}^T \in \mathbb{R}^{4+n}. \tag{2}$$

The system is modeled dynamically using the classical Lagrangian notation

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{c}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{g} = \mathbf{G}(\boldsymbol{\eta})\mathbf{u}, \tag{3}$$

where $\mathbf{M}$ is the inertia matrix, $\mathbf{c}$ is the vector of Coriolis and centrifugal forces, $\mathbf{g}$ is the vector of gravity forces, and

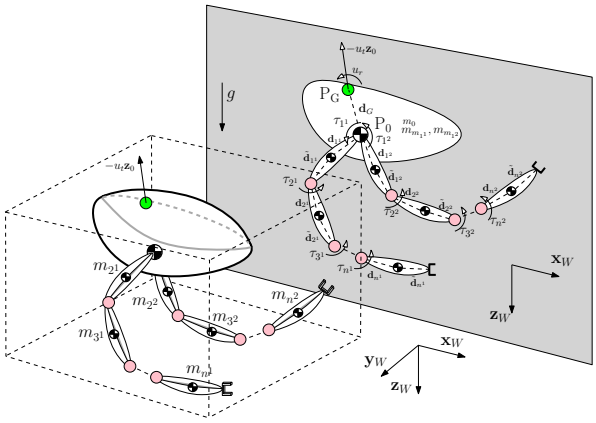

Fig. 1: Sketch of a protocentric aerial manipulator (PAM) with two arms, described with the same notation of [12]. Notice that in the 3D model of the PAM (the one on the left down), the axis of rotation for each individual joint of each arm is parallel to $\mathbf{y}_W$. Hence its projection on $\mathbf{x}_W - \mathbf{z}_W$ plane looks like the one in the grayed area (right up). Notice the possibility that the geometric center of base actuation, $\mathrm{P_G}$, and the CoM of the flying base, P_0 are not coinciding in the same point.

[1]Link to report: https://homepages.laas.fr/afranchi/files/2016/MBGA16.zip

$\mathbf{G}$ is the input matrix

$$\mathbf{G}(\boldsymbol{\eta}) = \begin{bmatrix} -\mathbf{R}_0\mathbf{e}_3 & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{3+n} \end{bmatrix} \in \mathbf{R}^{(6+n)\times(4+n)}, \quad (4)$$

where $\mathbf{I}_k$ 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.

Since the control input has less elements $(4+n)$ than the configuration variables $(6+n)$, the system is underactuated. Further, the inertia matrix has the following structure

$$\mathbf{M}(\mathbf{q}) = \begin{bmatrix} m_s\mathbf{I}_3 & \mathbf{M}_{pr} \\ \mathbf{M}_{pr}^T & \mathbf{M}_r \end{bmatrix} \in \mathbf{R}^{(6+n)\times(6+n)}, \quad (5)$$

where m_s is the total mass of the system.

Remark II.1. Because of the underactuation, commonly in multi-rotor platforms the position $\mathbf{p}_0$ and the yaw ψ_0 are controlled, while the roll ϕ_0 and the pitch θ_0 are used as virtual inputs or they are left uncontrolled. In the case of aerial manipulation, the position of the end-effectors does not only depend on $\mathbf{p}_0$, ψ_0 and $\boldsymbol{\theta}$, but *also* on ϕ_0 and θ_0. Consequently, it is not possible to plan exclusively for $\mathbf{p}_0$, ψ_0 and $\boldsymbol{\theta}$ if the position of the end-effectors is 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 for Aerial Manipulators

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 (3), and our objective is to track this output. If the desired output trajectory is consistent with the undeactuation it is in theory possible to find some corresponding *desired* states and inputs

$$\begin{aligned} \mathbf{q}^d(t) &= [\mathbf{p}_0^{dT}\ \boldsymbol{\eta}^{dT}\ \boldsymbol{\theta}^{dT}]^T, \quad \mathbf{u}^d = [u_t^d\ \mathbf{u}_r^{dT}\ \boldsymbol{\tau}^{dT}]^T, \\ \dot{\mathbf{q}}^d(t) &= [\dot{\mathbf{p}}_0^{dT}\ \dot{\boldsymbol{\eta}}^{dT}\ \dot{\boldsymbol{\theta}}^{dT}]^T, \end{aligned} \quad (6)$$

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 the 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 $\mathbf{p}_0^d$. Firstly let us define the controlled thrust vector as:

$$\mathbf{f}_0 = \mathbf{f}_0^d + \mathbf{f}_0^\star = \mathbf{f}_0^d + \mathbf{K}_{\mathbf{p}_0}^P(\mathbf{p}_0^d - \mathbf{p}_0) + \mathbf{K}_{\mathbf{p}_0}^D(\dot{\mathbf{p}}_0^d - \dot{\mathbf{p}}_0), \quad (7)$$

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

$$u_t = -(\mathbf{R}_0\mathbf{e}_3)^T\mathbf{f}_0, \quad (8)$$

and the commanded attitude as

$$\begin{aligned} \mathbf{z}_0^c &= \mathbf{f}_0 / \left\|\mathbf{f}_0\right\|, \qquad \mathbf{y}_0^c = \mathbf{z}_0^c \times \mathbf{e}_1, \\ \mathbf{x}_0^c &= \mathbf{y}_0^c \times \mathbf{z}_0^c, \qquad \mathbf{R}_0^c = [\mathbf{x}_0^c\ \mathbf{y}_0^c\ \mathbf{z}_0^c]. \end{aligned} \quad (9)$$

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:

$$\begin{aligned} \mathbf{e}_{\mathbf{R}_0}^{[\times]} &= 1/2(\mathbf{R}_0^{cT}\mathbf{R}_0 - \mathbf{R}_0^T\mathbf{R}_0^c) \\ \mathbf{e}_{\boldsymbol{\omega}_0} &= \boldsymbol{\omega}_0^d - \boldsymbol{\omega}_0 \\ \mathbf{u}_r &= \mathbf{u}_r^d + \mathbf{u}_r^\star = \mathbf{u}_r^d + \mathbf{K}_{\mathbf{R}_0}^P\mathbf{e}_{\mathbf{R}_0} + \mathbf{K}_{\mathbf{R}_0}^D\mathbf{e}_{\boldsymbol{\omega}_0}, \end{aligned} \quad (10)$$

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[2], and $\mathbf{K}_{\mathbf{R}_0}^P, \mathbf{K}_{\mathbf{R}_0}^D \in \mathbb{R}_{\geq 0}^{3\times 3}$.

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

$$\tau_{\nu^\mu} = \tau_{\nu^\mu}^d + \tau_{\nu^\mu}^\star = \tau_{\nu^\mu}^d + k_{\nu^\mu}^P(\theta_{\nu^\mu}^d - \theta_{\nu^\mu}) + k_{\nu^\mu}^D(\dot{\theta}_{\nu^\mu}^d - \dot{\theta}_{\nu^\mu}), \quad (11)$$

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, it is possible to adapt the controller for achieving good results. For instance, for a velocity-controlled motor, the commanded velocity can be given as

$$\dot{\theta}_{\nu^\mu} = \dot{\theta}_{\nu^\mu}^d + k_{\nu^\mu}^P(\theta_{\nu^\mu}^d - \theta_{\nu^\mu}). \quad (12)$$

See also Sec. V for its implementation.

In summary, the VTOL thrust, u_t, is computed in (8); its torque is given in (10); and the control input of the individual motors of the manipulators, τ_{ν^μ} in (11) which collectively builds the torque input $\boldsymbol{\tau}$. Hence, we have all the control inputs $\mathbf{u}$ of the system in (3). A simple variant, as in (12), allows the use of this controller for kinematically-controlled motors. 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 Decentralized Control for PAMs

In Sec. III we presented a *decentralized* controller for aerial manipulators, whose model in 3D was shown in Sec. II. As also mentioned in Sec. III, for this controller to track a desired output properly, an algorithm computing all the nominal states and inputs (feed-forward terms) is required.

[2]Notice that $\boldsymbol{\omega}_0$ can be easily computed using $\boldsymbol{\eta}$ and $\dot{\boldsymbol{\eta}}$. This also applies to $\boldsymbol{\omega}_0^d$ using $\boldsymbol{\eta}^d$ and $\dot{\boldsymbol{\eta}}^d$ available from (6).

In our previous work [12] we have introduced the notion of *protocentricity*. A protocentric aerial manipulator (PAM) (with any number of manipulator arms, each having any number of DoFs, with rigid or compliant transmission) is characterized by all manipulator arms being attached to the CoM of the flying base. In [12], 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 CoM position of the flying base and the absolute rotations of the manipulator links. The choice of the absolute joint angles as system coordinates, together with the protocentric design, overcomes both difficulties highlighted in Remarks II.1 and II.2. In particular, also the position of the end-effector and the absolute rotations of the manipulator links are flat outputs, which makes such platforms of particular interest.

In this section, we show how to use the flatness property of PAMs in 2D together with the decentralized controller (see Sec. II) in order to track a desired output trajectory of a PAM in 3D. Now, consider a PAM in 3D, where the motion of all manipulators are constrained to a plane, i.e. $y_0 = 0$ and $\mathbf{y}_0^T \mathbf{z}_W = \mathbf{y}_0^T \mathbf{x}_W = 0$. A sketch of such design is depicted in Fig. 1, where each joint of all manipulators rotate around an axis parallel to $\mathbf{x}_0 \times \mathbf{z}_0$. Now, notice the similarity between the projection of the considered PAM on the $\mathbf{x}_W - \mathbf{z}_W$ plane, and the system discussed in [12]; they are the same for the case when all joints are rigid.

Given the above, the generalized coordinates of a PAM in 2D are chosen as

$$\mathbf{q}_2 = \begin{bmatrix} \mathbf{p}_{0_{xz}}^T & \theta_0 & \mathbf{q}_r^T \end{bmatrix}^T \in \mathbb{R}^{3+n}, \tag{13}$$

where $\mathbf{p}_{0_{xz}} = [x_0 \ z_0]^T \in \mathbb{R}^2$ is the position of the CoM of the flying base 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. Notice that $\theta_{0\nu^\mu}$ is the absolute orientation of the ν-th joint of the μ-th arm, and that $\theta_{0k^\mu} = \theta_0 + \sum_{\nu=1}^{k} \theta_{\nu^\mu}$. For more details on the naming convention we refer the reader to [12] and Section II of this paper. The set of inputs is

$$\mathbf{u}_2 = \begin{bmatrix} u_t & u_r & \boldsymbol{\tau}^T \end{bmatrix}^T \in \mathbb{R}^{2+n}. \tag{14}$$

where the *scalar* $u_r \in \mathbb{R}$ is the magnitude of the base torque, which is applied around the axis passing through $\mathrm{P_G}$ and parallel to $\mathbf{y}_0 = \mathbf{z}_0 \times \mathbf{x}_0$. Notice that the PAM in 2D is also underactuated, as the one in 3D. Finally, we define $\mathbf{p}_{e_{xz}^\mu}$ as the position of the μ-th end-effector in the considered plane.

Recalling [12] and its technical report in [13], we have the following fact:

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

This means we can control the motion of a PAM as shown in Fig. 1 in the $\mathbf{x}_W - \mathbf{z}_W$ plane, by combining the controller presented in Sec. III and the flatness property proven in [12].

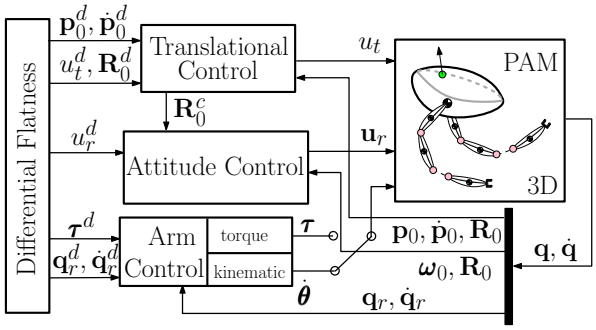

Fig. 2: Control of the PAM depicted in Fig. 1, using the decentralized controller explained in Sec. III and its differential flatness property detailed in Sec. IV.

By setting the desired motions of all the other DoFs to *zero*, we can control the overall system in 3D.

Let us then compute the nominal states and the control inputs of the PAM in the $\mathbf{x}_W - \mathbf{z}_W$ plane, as functions of $\mathbf{y}, \dot{\mathbf{y}}, \ddot{\mathbf{y}}, \dddot{\mathbf{y}}, \ddddot{\mathbf{y}}$. Considering Fact 1, the nominal states and the control inputs to be computed as sole functions of the flat ouputs are $\theta_0, \dot{\theta}_0$ (flying base pitch and its time derivatives), and u_t, u_r, τ_{ν^μ}. In the following, we assume that all the joints are actuated via a motor and the rotational center of this motor is the same with the center of the revolute joint that is attached to it.

A. Computation of the Nominal θ_0, $\dot{\theta}_0$, and u_t

As given in the technical report of [12], the translational dynamics of the CoM position of the PAM in the $\mathbf{x}_W - \mathbf{z}_W$ plane, $\mathbf{p}_c \in \mathbb{R}^2$, can be written as

$$m_s \ddot{\mathbf{p}}_c = \begin{bmatrix} -\sin(\theta_0) \\ -\cos(\theta_0) \end{bmatrix} u_t + \begin{bmatrix} 0 \\ m_s g \end{bmatrix}, \tag{15}$$

where $m_s = m_0 + \sum_{j=1}^{m} \left(\sum_{i=1}^{n^j} m_{i^j} + m_{m_{i^j}} \right)$ is the total mass of the PAM, with $m_0 \in \mathbb{R}_{>0}$ the mass of the aerial platform (base), $m_{\nu^\mu} \in \mathbb{R}_{>0}$ and $m_{m_{\nu^\mu}} \in \mathbb{R}_{>0}$ the masses of the ν^μ-th link and the motor, respectively. The gravitational constant is shown with g. We then define the vector

$$\mathbf{w} = \mathbf{w}(\mathbf{y}, \dot{\mathbf{y}}, \ddot{\mathbf{y}}) = \ddot{\mathbf{p}}_c - [0 \ g]^T = [w_x \ w_z]^T \in \mathbb{R}^2, \tag{16}$$

which is direct function of the flat outputs. Notice that $\mathbf{w} = -\frac{u_t}{m_s}[\sin(\theta_0) \ \cos(\theta_0)]^T$. Hence,

$$\begin{aligned}
\theta_0 &= \theta_0(\ddot{\mathbf{p}}_c) = \operatorname{atan2}(-w_x, -w_z) \\
\dot{\theta}_0 &= \dot{\theta}_0(\ddot{\mathbf{p}}_c, \dddot{\mathbf{p}}_c) = \frac{w_z \dot{w}_x - w_x \dot{w}_z}{w_x^2 + w_z^2} \\
\ddot{\theta}_0 &= \ddot{\theta}_0(\ddot{\mathbf{p}}_c, \dddot{\mathbf{p}}_c, \ddddot{\mathbf{p}}_c) = \frac{\ddot{w}_x w_z - w_x \ddot{w}_z}{w_x^2 + w_z^2} - \\
&\quad - \frac{2[(w_z^2 - w_x^2)\dot{w}_x \dot{w}_z + (\dot{w}_x^2 - \dot{w}_z^2) w_x w_z]}{(w_x^2 + w_z^2)^2} \\
u_t &= u_t(\ddot{\mathbf{p}}_c) = m_s ||\mathbf{w}|| \,.
\end{aligned} \tag{17}$$

Therefore, we need to compute the time derivatives of $\mathbf{p}_c$ from second up to the fourth order, as sole functions of the flat outputs. To this end, define P_{ν^μ} as the CoM of the ν^μ-th link. The ν^μ-th *link frame* is denoted with $\mathcal{F}_{\nu^\mu} : \{\mathrm{P}_{\nu^\mu}, \mathbf{x}_{\nu^\mu}, \mathbf{z}_{\nu^\mu}\}$. Also, define $\mathrm{P}_{\mathrm{M}_{\nu^\mu}}$ as the center

of the ν^μ-th motor. The ν^μ-th joint and the motor rotate around an axis parallel to $\mathbf{z}_0 \times \mathbf{x}_0$ while passing through $\mathrm{P}_{\mathrm{M}_{\nu^\mu}}$. The constant position of $\mathrm{P}_{\mathrm{M}_{\nu^\mu}}$ and of $\mathrm{P}_{\mathrm{M}_{(\nu+1)^\mu}}$ in $\mathcal{F}_{\nu^\mu}$ are denoted with $-\mathbf{d}_{\nu^\mu} = [-d_{\nu^\mu x} \; -d_{\nu^\mu z}]^T \in \mathbb{R}^2$ and $\tilde{\mathbf{d}}_{\nu^\mu} = [\tilde{d}_{\nu^\mu x} \; \tilde{d}_{\nu^\mu z}]^T \in \mathbb{R}^2$, respectively. The time-varying positions of P_{ν^μ} and $\mathrm{P}_{\mathrm{M}_{\nu^\mu}}$ in $\mathcal{F}_W$ are denoted with $\mathbf{p}_{\nu^\mu} \in \mathbb{R}^2$, $\mathbf{p}_{m_{\nu^\mu}} \in \mathbb{R}^2$ respectively. Given an angle $\theta_* \in \mathbb{R}$ between the z-axes of two frames, we use the usual rotation matrix definition $\mathbf{R}_* \in \mathrm{SO}(2)$. Therefore, the orientations of $\mathcal{F}_0$ and $\mathcal{F}_{\nu^\mu}$ in $\mathcal{F}_W$ are expressed by the rotation matrices $\mathbf{R}_0(\theta_0)$, and $\mathbf{R}_{0\nu^\mu}(\theta_{0\nu^\mu})$, respectively. Recall from [12] that the CoM of the overall system can be written as

$$\mathbf{p}_c = \frac{1}{m_s}\left(m_0\mathbf{p}_{0_{xz}} + \sum_{j=1}^{m}\left(\sum_{i=1}^{n^j}(m_{ij}\,\mathbf{p}_{ij} + m_{m_{ij}}\,\mathbf{p}_{m_{ij}})\right)\right). \quad (18)$$

This expression is easily differentiated from the second up the fourth time derivative (given in the technical report), as a function of the corresponding link and motor CoM position derivatives. Thus, it is clear that we need to compute these quantities. For the ν-th motor of the μ-th manipulator (ν^μ-th motor of the system) it is

$$\mathbf{p}_{m_{\nu^\mu}}(\mathbf{y}) = \mathbf{p}_{0_{xz}} + \underbrace{\sum_{i^\mu=1}^{\nu^\mu-1}\mathbf{R}_{0i^\mu}\bar{\mathbf{d}}_{i^\mu}}_{:=0, \quad if \;\; \nu^\mu=1}, \quad (19)$$

$$\mathbf{p}_{\nu^\mu}(\mathbf{y}) = \mathbf{p}_{m_{\nu^\mu}} + \mathbf{R}_{0\nu^\mu}\mathbf{d}_{\nu^\mu}, \quad (20)$$

where (19) is used in (20), and $\bar{\mathbf{d}}_* = \mathbf{d}_* + \tilde{\mathbf{d}}_*$. Equations (19) and (20) can be easily differentiated to obtain the time derivatives from the second up to the fourth order, as sole function of the flat outputs (see the technical report for the computations). Substituting both (19) and (20) in (18) we do the same for the time derivatives of the overall system CoM. Finally applying this to (17), taking (16) into account, we compute the nominal values of $\theta_0(\mathbf{y}, \dot{\mathbf{y}}, \ddot{\mathbf{y}})$, $\dot{\theta}_0(\mathbf{y}, \dot{\mathbf{y}}, \ddot{\mathbf{y}}, \dddot{\mathbf{y}})$, and $u_t(\mathbf{y}, \dot{\mathbf{y}}, \ddot{\mathbf{y}})$ as sole functions of the flat outputs and a finite number of the corresponding derivatives.

B. Computation of Nominal τ_{ν^μ} and u_r

In [12], the nominal torque of the ν^μ-th motor is given as

$$\tau_{\nu^\mu} = \tau_{\nu^\mu+1} + \mathbf{m}_{0\nu^\mu}^T(\theta_{0\nu^\mu})\ddot{\mathbf{p}}_{0_{xz}} + c_{r_{\nu^\mu}}(\mathbf{q}_{r^\mu}, \dot{\mathbf{q}}_{r^\mu}) + \mathbb{J}_{\nu^\mu}\ddot{\theta}_{0\nu^\mu} + g_{r_{\nu^\mu}}(\theta_{0\nu^\mu}) + \sum_{l=1, l\neq\nu^\mu}^{n^\mu} m_{l\nu^\mu}(\theta_{0l^\mu}, \theta_{0\nu^\mu})\ddot{\theta}_{0l^\mu}, \quad (21)$$

where $c_{r_{\nu^\mu}}$ 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 (for $\nu^\mu = n^\mu$ it is $\tau_{\nu^\mu+1} = 0$). 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 (5). Both terms $\mathbb{J}_{\nu^\mu}$ and $m_{l\nu^\mu}$ are inertial terms of $\mathbf{M}_r$ in (5), 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. The explicit computation of all these elements are given in [12] and [13], which we omit them here for the brevity. Using them, 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:

$$u_r = u_r(\mathbf{y}, \dot{\mathbf{y}}, \ddot{\mathbf{y}}, \dddot{\mathbf{y}}, \ddddot{\mathbf{y}}) = J_0\ddot{\theta}_0 + \sum_{j=1}^{m}\tau_{1^j} - d_{G_x}u_t, \quad (22)$$

where J_0 is the inertia of the aerial platform, d_{G_x} is the constant position of P_G with respect to P_0 in $\mathcal{F}_0$, along $\mathbf{x}_0$, τ_{1^μ} comes from (21) for $\nu = 1$, and $\ddot{\theta}_0$ with u_t are available from (17). We refer the meticulous reader to [12] for the details of the notation and the computations.

Notice that the flatness of $\mathbf{y}_e$ in Fact. 1 is quite obvious, thanks to the protocentric design and the absolute joint coordinates. Let us give the following remark:

Remark IV.1. The flat outputs of a PAM are:

- $\mathbf{y} = [\mathbf{p}_{0_{xz}}^T \; \mathbf{q}_r^T]^T \in \mathbb{R}^{(n+2)}$ from [12],
- $\mathbf{y}_e = [\mathbf{p}_{e^\mu_{xz}}^T \; \mathbf{q}_r^T]^T \in \mathbb{R}^{2+n}$ for any μ, since $\exists f_e : \mathbf{p}_{0_{xz}} = f_e(\mathbf{p}_{e^\mu_{xz}} \; \mathbf{q}_r)$.

Hence, we have showed how to use the flat outputs $\mathbf{y}$ given in Fact 1 and their derivatives up to the fourth order, for computing the nominal values of $\theta_0, \dot{\theta}_0, u_t, u_r, \boldsymbol{\tau}$. It is clear that all the other states are actually the flat outputs themselves. This implies that we can say that, $\exists h : \; \mathbf{b} = h(\mathbf{y}, \dot{\mathbf{y}}, \ddot{\mathbf{y}}, \dddot{\mathbf{y}}, \ddddot{\mathbf{y}})$, where $\mathbf{b} = [\mathbf{q}_2^T \; \dot{\mathbf{q}}_2^T \; \mathbf{u}_2^T]^T$ is a vector combining all the states and inputs of the PAM in 2D. Finally, the relative joint angles can be easily obtained as $\theta_{1^\mu} = \theta_{01^\mu} - \theta_0$ and $\theta_{\nu^\mu} = \theta_{0\nu^\mu} - \theta_{0(\nu-1)^\mu}$ for $\nu > 1$.

Now, for tracking a desired $\mathbf{y}^d(t)$, where $\mathbf{y}$ is as in Fact 1, using the differential flatness property in 2D, but together with the controller developed in 3D (Sec. III), we can say

$$\mathbf{R}_0^d = \begin{bmatrix} c_{\theta_0^d} & 0 & s_{\theta_0^d} \\ 0 & 1 & 0 \\ -s_{\theta_0^d} & 0 & c_{\theta_0^d} \end{bmatrix} \quad (23)$$

$$\mathbf{f}_0^d = -u_t^d\mathbf{R}_0^d\mathbf{e}_3, \qquad \boldsymbol{\omega}_0^d = \dot{\theta}_0^d\mathbf{e}_2 \quad (24)$$

$$\mathbf{u}_r^d = u_r^d\mathbf{e}_2, \qquad y_0^d \equiv 0 \quad (25)$$

will impose the necessary constraints. Notice that $\theta_0^d, \dot{\theta}_0^d, u_t^d, u_r^d$ are computed as in (17) and (22) for $\mathbf{y}^d$. Clearly, $\tau_{\nu^\mu}^d$ will be computed in the same way using (21). 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 preliminary experiments aimed at validating the controller proposed in this paper. Furthermore, we analyze its performances by comparing it with other standard control techniques.

Let us first describe briefly the testbed used for the experiments (see Fig. 3). The aerial manipulator consists of a Quadrotor VTOL and a 2 DoF manipulator arm. The arm structure is based on carbon fiber bars and printed plastic parts, whose design was inspired by the work in [14]. A big difference of our design is that the actuators of both joints are placed at the base of the arm, rigidly attached to the VTOL. The first joint is connected to its actuator (a

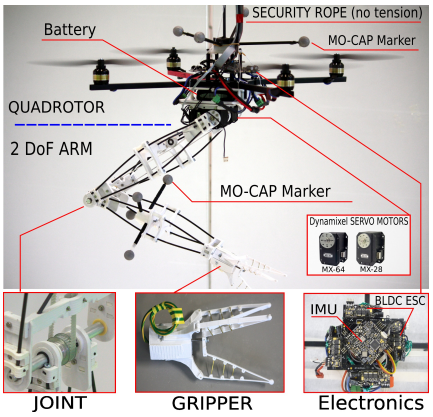

Fig. 3: Experimental setup of the aerial manipulator. A quadrotor VTOL is equipped with a 2 DoF manipulating arm.

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

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

Controller	$\mathbf{K}^P_{\mathbf{p}_0}$	$\mathbf{K}^D_{\mathbf{p}_0}$	$\mathbf{K}^P_{\mathbf{R}_0}$	$\mathbf{K}^D_{\mathbf{R}_0}$	k^P_1	k^P_2
Gain	$12\mathbf{I}_3$	$7\mathbf{I}_3$	$3\mathbf{I}_3$	$0.3\mathbf{I}_3$	1.8	1.6

Traj. Param.	$a^x_{\mathbf{p}_0}$ [m]	$a^1_{\mathbf{q}_r}$ [°]	$a^2_{\mathbf{q}_r}$ [°]	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 need to compute nominal state and inputs by the flatness.

dynamixel MX-64 motor) directly, while the second one is connected to its motor (a dynamixel MX-28) via a metal-reinforced plastic belt (with very low elasticity). A detailed description of the setup is given in Fig. 3. Such design allows us to have a light-weight arm reducing the mass of each joint and in particular their inertia. This in turn allows to use a relative small quadrotor (diameter $0.4\,[\mathrm{m}]$, maximum thrust per propeller of about $5.26\,[\mathrm{N}]$) with respect to the ones normally used in the literature for arms of similar length, as, e.g., in [15]. Since the motors are rigidly attached to the aerial vehicle, their mass can be seen as part of the total VTOL mass. 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 (as almost all the affordable motors suitable for aerial manipulation) to control the arm we use (12), except for a slight modification needed to cope with the fact that the second link is not directly attached to its motor.

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. Like for the arm, the physical parameters are given in Tab. I.

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 acceleration and the angular velocity 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.:

$$\mathbf{y}^d = \begin{bmatrix} = \mathbf{p}_{0^d_{xz}} \\ \mathbf{q}^d_r \end{bmatrix} = \begin{bmatrix} a^x_{\mathbf{p}_0} & 0 & a^1_{\mathbf{q}_r} & a^2_{\mathbf{q}_r} \end{bmatrix}^T \sin(\omega t) \quad (26)$$

for three different sets of parameters corresponding to three qualitatively different task trajectories:

(a) the arm is oscillating and quadrotor is fixed,
(b) the arm and quadrotor are oscillating with opposite phases,
(c) the arm and quadrotor are oscillating 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., $u^d_t = -m_s \mathbf{e}_3^T \mathbf{R}_0 \mathbf{e}_3$. In this way the VTOL and the arm virtually are considered as two independent systems (even if in practice they are not).

2) *static compensation*: only the static effects due to the gravity are compensated, i.e., the nominal state and the inputs are computed considering all the derivatives of the desired trajectory are equal to zero, i.e., $\mathbf{y}^{d^{(l)}} = \mathbf{0}$ for $l = 1, \dots, 4$, ($\mathbf{y}^d \neq \mathbf{0}$). This method is often used for the control of the 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 explain in Sec. III and Sec. IV.

The performances of these three methods are shown in Figs. 5, 6 and 7, using red, green and blue curves, respectively. In particular the plots show the evolution of the position of the VTOL CoM and the end-effector[3] in the first two rows, the remaining configuration variables in 3D (third and fourth row), and the inputs on the vertical plane, as well as the nominal relative quantities (with a dashed black line). 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.

On the contrary, our proposed method 3) shows good tracking performances for all the types of trajectories validating the fact the dynamic compensation based on the flatness 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 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 investigate in the technical report by numerical simulations.

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

[3]Notice that for the results of Sec. IV, showing the joint angles or the position of the end-effector is equivalent since they are both flat outputs.

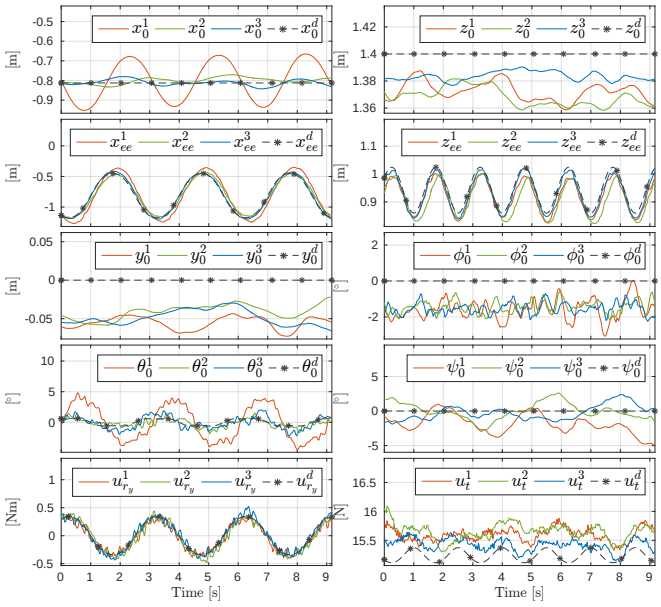

Fig. 5: Experimental results for trajectory (a) (see Fig. 4.a). In all plots, the flat outputs and the nominal states/inputs are depicted with black dashed lines (and stars). Again in all plots, red, green and blue curves stands for the results of the controller with *minimal compensaton*, *static compensation*, and *dynamic compensation*, respectively. While the first controller perform worse for tracking of the all outputs, third one (proposed controller) is always performing good especially when tracking the end-effector positions.

The first is due to the the fact that 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 one 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 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 one 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

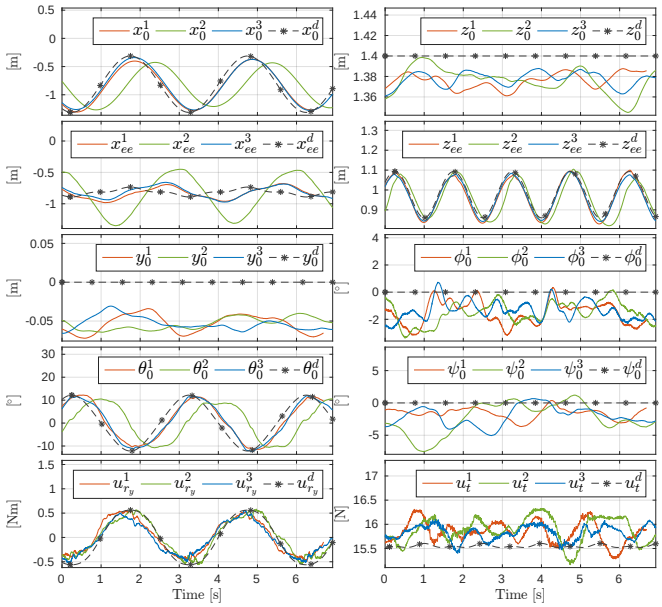

Fig. 6: Experimental results for trajectory (b) (see Fig. 4.b). Same color and line code is used as in Fig. 5. The proposed controller (blue) achieves always a better tracking performance. An interesting result is that for such a dynamic tracking task, the controller with the static compensation (green) performs worse than the one with minimum compensation (red).

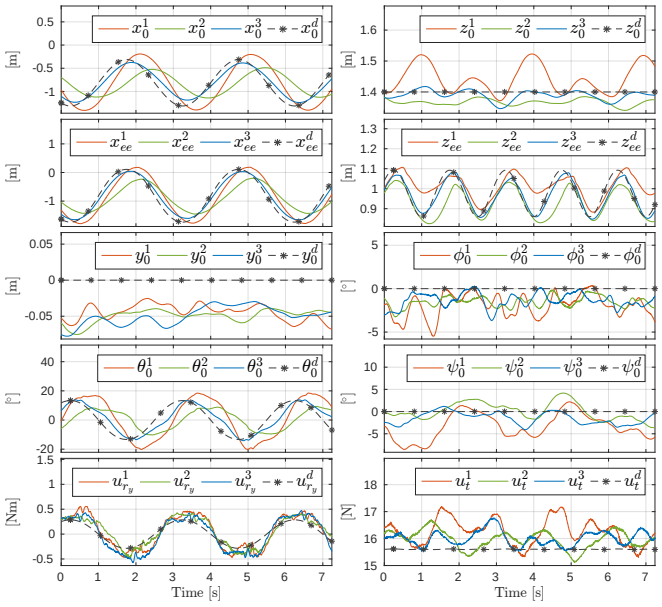

Fig. 7: Experimental results for trajectory (c) (see Fig. 4.c). Same color and line code is used as in Fig. 5. The proposed controller (blue) again outperforms the other methods.

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 the PAMs (in 2D), we showed how to compute the nominal states and the inputs of the system analytically in advance, and use it for tracking dynamic maneuvers. The experimental results are in line with the proposed theory showing the advantage of using differential flatness of the aerial manipulators.

This work can be extended in many directions. One is to study the differential property directly in 3D. Another is finding ways to relax the protocentric assumption. The third are experiments using more than one manipulator arm. The fourth is to study control-aware planning algorithms that take into account the capabilities of the controller.

VII. 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 arm used in the experiments.

References

[1] T. Lee, M. Leoky, and N. H. McClamroch, "Geometric tracking control of a quadrotor UAV on SE(3)," in *49th IEEE Conf. on Decision and Control*, Atlanta, GA, Dec. 2010, pp. 5420–5425.

[2] 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.

[3] B. Siciliano and O. Khatib, *Handbook of Robotics*. Springer, 2008.

[4] AeRoArms, "EU Collab. Project ICT-644271," www.aeroarms-project.eu, 2015-2019.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] B. Yüksel, G. Buondonno, and A. Franchi, "Protocentric aerial manipulators: Flatness proofs and simulations," LAAS-CNRS, Tech. Rep. hal-01351153, July 2016. [Online]. Available: https://hal.archives-ouvertes.fr/hal-01351153

[14] 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.

[15] A. E. Jimenez-Cano, J. Martin, G. Heredia, A. Ollero, and R. Cano, "Control of an aerial robot with multi-link arm for assembly tasks," in *2013 IEEE Int. Conf. on Robotics and Automation*, May 2013, pp. 4916–4921.