Task-space Tracking Control for Underactuated Aerial Manipulators

Gianluca Garofalo[a], Fabian Beck[a,b] and Christian Ott[a]

Abstract— A flying vehicle equipped with a robotic manipulator poses a challenging control problem because of the dynamical coupling between both subsystems. In order to achieve highly dynamic behaviors while enabling precise end-effector tracking of the manipulator, this coupling has to be taken into account in the control design. In this paper, an inertial decomposition of the full rigid body dynamics is utilized to design a control law, which guarantees end-effector reference tracking and position trajectory control of the overall system. The stability analysis of the closed loop system is accompanied with a validation in simulation of the proposed controller.

I. INTRODUCTION

Floating base robots are nowadays present in a variety of application fields. Examples of such systems are humanoid robots, wheeled robots, as well as satellites and flying vehicles equipped with manipulators. Unlike their fixed base counterparts, these robots have the possibility to change the pose of both their base and their end-effectors within the environment in which they act, therefore having the ability to eliminate restrictions on the workspace. The increased dexterity of these systems, is nevertheless often accompanied by a considerable increase in the complexity of the planning phase and control design. The limited degree of actuation of the base can be one of the reasons for a more complex architecture. The base of the system can be fully actuated (e.g. a fully propelled satellite), partially actuated (e.g. an unmanned aerial vehicle) or passive (e.g. a humanoid robot). In this paper, the focus will be on the second class of systems. In particular, it will be considered the case of an Unmanned Aerial Vehicle (UAV) equipped with a robotic manipulator as depicted in Fig. 1.

The composition of an UAV and a robot arm is often referred as aerial manipulator [1]–[7] and there is a wide array of potential applications for these systems. Aerial manipulation tasks include for example remote sensing, disaster response, transportation and surveillance [1]. As a first step towards these tasks, the goal of this contribution is to let the end-effector of the system track a desired trajectory. This topic has been already considered for example in [3]–[8]. To the best of the knowledge of the authors, all previous contributions considering the full dynamics of the aerial manipulator system did not provide a proof of stability or made strong assumptions on the structure of the model. Only very recently first results on the stability analysis have been provided for the regulation case [8]. In [3] the static forces

Fig. 1. Left: Aerial manipulator consisting of a rotor helicopter equipped with a KUKA Lightweight Robot [12] developed at DLR.
Right: Sketch of an aerial manipulator with actuated inputs. The components of $\boldsymbol{\mu}$ have been indicated as μ_x, μ_y, μ_z and together with f constitute $\boldsymbol{\psi}$.

and moments originating from the manipulator have been feedforwarded into helicopter control, neglecting all dynamic coupling effects between helicopter and manipulator. The contributions [4], [5] incorporate these dynamic couplings, but in [4] the control problem is solved for a simplified planar model, while [5] does not provide an explicit control law and the task considers only translational coordinates. In [7] it is assumed that the robotic manipulator is attached to the center of mass of the UAV and that the robotic arm is restricted to a plane in UAV body frame. With regard to the aerial manipulator system depicted in Fig. 1 both assumptions are violated. In order to obtain a control strategy for task-space tracking also for this class of systems, an inertially decoupling transformation is applied to the dynamic model of the aerial manipulator. Inertially decoupling transformations have been extensively used to control redundant robots [9] and they have been coupled to first principles of mechanics to show structural properties of the model of a floating base robot [10]. The results from the latter allowed, for example, the derivation of control strategies for free-floating space robots [11]. Compared to [11], here the more complex problem of trajectory tracking with an underactuated base and in presence of non-constant momenta is considered.

The main contribution of the paper is the derivation of a control law that, in combination with a physically motivated coordinate transformation, allows for analysis and design of the whole system by considering separate simpler subsystems. It is worth to emphasize the importance of having a good understanding of the dynamic model as this has often resulted in improved controller design in the robotic community [13]. In particular, it is possible to split the design of the complete controller in the design for a system with similar characteristics as a quadrotor (for which the control strategies from the literature could be used [14]–[16]) and for a "modified" manipulator.

[a]The authors are with the Institute of Robotics and Mechatronics, German Aerospace Center (DLR), Wessling, Germany.
[b]The author is with the Control Engineering Group, Technische Universität Ilmenau, Germany.
`gianluca.garofalo(at)dlr.de`

The paper is organized as follows. In Section II the model and the coordinate transformation are presented. Based on the latter, the control law is derived and analyzed in Section III. Section IV presents the simulation results and Section V summarizes the paper and presents possible extensions for future works.

II. DYNAMICS

The dynamic equations of motion of the system are

$$M(q)\dot{v} + C(q,v)v + g(x) = Q^T\tau + J_b^T(x)B\psi \ , \quad (1)$$

where $x \in \mathbb{X} = \mathbb{R}^3 \times SO(3) \times \mathbb{R}^n$ and $v \in \mathbb{R}^{(6+n)}$ constitute together the state of the system[1]. The configuration x includes the joint angles of the robot $q \in \mathbb{R}^n$ and the floating base coordinates $x_b \in \mathbb{R}^3 \times SO(3)$. The matrix $Q \in \mathbb{R}^{n\times(6+n)}$ selects the joint velocities $\dot{q}$ out of all the velocity coordinates v, i.e. $\dot{q} = Qv$. Moreover, $M(q) \in \mathbb{R}^{(6+n)\times(6+n)}$ denotes the symmetric and positive definite inertia matrix, $C(q,v) \in \mathbb{R}^{(6+n)\times(6+n)}$ a Coriolis matrix[2] satisfying the so-called passivity property $\dot{M}(q,v) = C(q,v) + C^T(q,v)$ and $g(x) \in \mathbb{R}^{(6+n)}$ the gravity torque vector. The torques $\tau \in \mathbb{R}^n$ produced by the motors of the robot are an input to the system, together with the wrench at the base $w_b = B\psi \in \mathbb{R}^6$, where B is a constant matrix used to model the presence of underactuation and ψ the actuated input. Unlike $\begin{bmatrix} J_b^T(x) & Q^T \end{bmatrix}$ where $J_b(x) \in \mathbb{R}^{6\times(6+n)}$ is the Jacobian matrix of the base, the matrix B is not invertible. The combination of the latter two matrices determines how the inputs enter in the system.

In the case under consideration, which is depicted in Fig. 1, $B \in \mathbb{R}^{6\times4}$ is the selection matrix mapping the torques $\mu \in \mathbb{R}^3$ around the body-frame axis and the thrust $f \in \mathbb{R}$ along the vertical axis (z-axis) of the UAV (i.e. the entries in ψ) to the body wrench w_b

$$w_b = B\psi = \begin{bmatrix} e_3 & O_3 \\ 0 & E_3 \end{bmatrix}\begin{bmatrix} f \\ \mu \end{bmatrix} \ , \quad (2)$$

where $e_3 = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T$, $E_3 \in \mathbb{R}^{3\times3}$ is the identity matrix and $O_3 \in \mathbb{R}^{3\times3}$ is a matrix of zeros.

For ease of presentation, the dependencies will be dropped in the remainder of the paper.

A. Coordinates transformation

The goal of this section is to provide a change of coordinates for the system that allows to rewrite it in a form more suitable for the design of the controller. To this end, a transformation that leads to a block-diagonal inertia matrix is sought after (called inertial decoupling), such that a block corresponds to the linear dynamics of the base, one to its angular dynamics and one for the dynamics of the manipulator. In this way, the control design can be split

into three independent steps (see Section III). The required inertial decoupling is achieved via the transformation

$$\xi := \begin{bmatrix} \dot{x}_g \\ \omega_b \\ \rho \end{bmatrix} = \begin{bmatrix} J_g \\ J_{b,\omega} \\ N \end{bmatrix} v = Tv \ , \quad (3)$$

where the Center of Mass (CoM) velocity $\dot{x}_g \in \mathbb{R}^3$ of the overall system, the angular velocity of the base $\omega_b \in \mathbb{R}^3$ and the nullspace velocity[3] $\rho \in \mathbb{R}^n$ are obtained through the correspondent matrices in $T \in \mathbb{R}^{(6+n)\times(6+n)}$. While the meaning of the Jacobian matrices J_g and $J_{b,\omega}$ in T is clear, N is a dynamically consistent nullspace projector [9] and its expression will be provided later on.

To understand how (3) is derived, firstly recall that the dynamics of the CoM is affected only by the external forces and, therefore, it is independent from the rest of the system. Clearly, the use of $\dot{x}_g$ will provide a partial inertial decoupling. This has been formally proven[4] in [10] and it is briefly reported in the Appendix for the specific case under consideration. To further diagonalize the inertia matrix and achieve the desired structure, it is first computed a nullspace base matrix [9] $Z \in \mathbb{R}^{n\times(6+n)} : Z\begin{bmatrix} J_g^T & J_{b,\omega}^T \end{bmatrix} = 0$, then Z is used to compute $N \in \mathbb{R}^{n\times(6+n)}$ as

$$N = \left(ZMZ^T \right)^{-1}ZM \ , \quad (4)$$

which satisfies by construction the condition $NM^{-1}\begin{bmatrix} J_g^T & J_{b,\omega}^T \end{bmatrix} = 0$ and guarantees that T is always invertible. Based on these properties, it is possible to show that pre-multiplying (1) by T^{-T} and using (3), the model in the new coordinates is given by

$$m\ddot{x}_g + mge_3 = u_1' \ , \quad (5a)$$
$$\Lambda_{2,2}\dot{\omega}_b + \Gamma_{2,2}\omega_b + \Gamma_{2,3}\rho = u_2' \ , \quad (5b)$$
$$\Lambda_{3,3}\dot{\rho} - \Gamma_{2,3}^T\omega_b + \Gamma_{3,3}\rho = u_3' \ , \quad (5c)$$

where $\Lambda_{i,i}$, $\Gamma_{i,i}$, $i = \{2,3\}$ and $\Gamma_{2,3}$ are the blocks of the transformed inertia matrix Λ and Coriolis matrix Γ, while m is the total mass and g the gravitational acceleration constant (see the Appendix). It is important to highlight that the transformation preserves the passivity property [9] and, since Λ is block-diagonal, it can be expressed as

$$\dot{\Lambda}_{i,i} = \Gamma_{i,i} + \Gamma_{i,i}^T \ . \quad (6)$$

Finally, u_j' with $j = \{1,2,3\}$ are the transformed inputs, i.e.

$$T^{-T}\begin{bmatrix} J_b^T & Q^T \end{bmatrix}\begin{bmatrix} fe_3 \\ \mu \\ \tau \end{bmatrix} = \begin{bmatrix} u_1' \\ u_2' \\ u_3' \end{bmatrix} := u' \ , \quad (7)$$

which represents a linear mapping between the original input $\begin{bmatrix} f & \mu^T & \tau^T \end{bmatrix}^T$ and u', with a noninvertible matrix (due to the underactuation). In a fully actuated scenario (e.g. a fully propelled satellite equipped with a manipulator), the

---

[1]SO(3) is the special orthogonal group of the orthogonal matrices with determinant 1.

[2]The centrifugal and Coriolis terms can be always factorized as $C(q,v)v$, but the choice of $C(q,v)$ is not unique [17].

[3]The nullspace velocity describes the motion which does not interfere with the task that a redundat robot is executing [9].

[4]To be precise, in [10] it was considered the linear momentum p, but it is well known that $p = m\dot{x}_g$ with m the total mass of the system.

system (5) could be controlled by choosing $\boldsymbol{u}'$ and then inverting (7) in order to obtain the value of the input needed in (1). Due to the underactuation, $\boldsymbol{u}'$ cannot be freely chosen, but it has to lie in the range of the matrix of the linear mapping (7). This is easily achieved (as shown in Section III) thanks to the special structure of the transformation (3). Indeed, the first three rows of the matrix $\boldsymbol{T}^{-T} \begin{bmatrix} \boldsymbol{J}_b^T & \boldsymbol{Q}^T \end{bmatrix}$ have the notable expression:

$$\begin{bmatrix} \boldsymbol{R}_b & \boldsymbol{O} & \boldsymbol{O} \end{bmatrix} \ ,$$

where $\boldsymbol{R}_b$ is the rotation matrix mapping a vector from the UAV frame to the fixed world frame. The expression is a direct consequence of Newton's second law, since the torques $\boldsymbol{\mu}$ and $\boldsymbol{\tau}$ in (7) cannot have any effect on the acceleration of the CoM, which is on the other hand affected by the thrust expressed in world frame.

III. CONTROL

Thanks to the special structure of $\boldsymbol{T}^{-T} \begin{bmatrix} \boldsymbol{J}_b^T & \boldsymbol{Q}^T \end{bmatrix}$, choosing $\boldsymbol{u}_1' \in \mathbb{R}^3$ proportional to $\boldsymbol{R}_b \boldsymbol{e}_3$ allows to solve (7) for every choice of $\boldsymbol{u}_2' \in \mathbb{R}^3$ and $\boldsymbol{u}_3' \in \mathbb{R}^n$. The latter are used to compensate the velocity couplings, i.e.

$$\begin{bmatrix} \boldsymbol{u}_1' \\ \boldsymbol{u}_2' \\ \boldsymbol{u}_3' \end{bmatrix} = \begin{bmatrix} u_1 \boldsymbol{R}_b \boldsymbol{e}_3 \\ \boldsymbol{\Gamma}_{2,3} \boldsymbol{\rho} + \boldsymbol{u}_2 \\ -\boldsymbol{\Gamma}_{2,3}^T \boldsymbol{\omega}_b + \boldsymbol{u}_3 \end{bmatrix} \ , \tag{8}$$

leading to the system

$$m\ddot{\boldsymbol{x}}_g + mg\boldsymbol{e}_3 = u_1 \boldsymbol{R}_b \boldsymbol{e}_3 \ , \tag{9a}$$

$$\boldsymbol{\Lambda}_{2,2}\dot{\boldsymbol{\omega}}_b + \boldsymbol{\Gamma}_{2,2}\boldsymbol{\omega}_b = \boldsymbol{u}_2 \ , \tag{9b}$$

$$\boldsymbol{\Lambda}_{3,3}\dot{\boldsymbol{\rho}} + \boldsymbol{\Gamma}_{3,3}\boldsymbol{\rho} = \boldsymbol{u}_3 \ , \tag{9c}$$

in which $\boldsymbol{u} = \begin{bmatrix} u_1 & \boldsymbol{u}_2^T & \boldsymbol{u}_3^T \end{bmatrix}^T$ has to be chosen in such a way to guarantee the specifications given in Section III-A. Notice that in this way the underactuated structure of the system is "preserved", in the sense that the underactuation appears only in the first three lines both for (1) and (9).

A. Control strategy and specifications

The main goal of the control strategy is to track a desired time varying task-space reference denoted by $\boldsymbol{y}_d \in \mathcal{C}^2$, where $\mathcal{C}^p$ denotes the space of continuous functions that have continuous first p derivatives. This task is achievable with the n degrees of freedom of the manipulator and there remains no redundancy with respect to the task. With regard to the aerial manipulation tasks mentioned in the introduction, $\boldsymbol{y}_d$ may contain a desired pose in inertial frame. Due to the limitation of the manipulators workspace, the tracking of the end-effector task is in general not possible relying only on the manipulator itself. The aerial manipulator is repositioned by stabilizing a desired CoM position trajectory $\boldsymbol{x}_d \in \mathcal{C}^4$. It is important to note that $\boldsymbol{x}_g$ is not equivalent to the UAV position, but they both provide information about the global position of the aerial manipulator. In both cases, the design of the global positioning requires in general a planning phase, which is beyond the scope of this paper. Therefore, it will be assumed that both $\boldsymbol{y}_d$ and $\boldsymbol{x}_d$ are provided.

Since the CoM dynamics (9a) is underactuated and the applied thrust u_1 is in $\boldsymbol{R}_b \boldsymbol{e}_3$ direction, i.e. along UAV body z-axis, the UAV is reoriented to guarantee that the trust is along the direction needed to stabilize the CoM dynamics. Such a control strategy has been already employed for quadrotors control. In particular, the derivation of the CoM control and orientation control is along the same lines as the geometric control for quadrotors proposed in [14].

B. CoM control

In order to stabilize the desired CoM position trajectory $\boldsymbol{x}_d$, a possible choice would be to have the right-hand side in (9a) equal to the desired force

$$\boldsymbol{f}_d = m\ddot{\boldsymbol{x}}_d + mg\boldsymbol{e}_3 - \boldsymbol{K}_x \tilde{\boldsymbol{x}} - \boldsymbol{D}_x \dot{\tilde{\boldsymbol{x}}} \ , \tag{10}$$

where the CoM position error is defined by $\tilde{\boldsymbol{x}} := \boldsymbol{x}_g - \boldsymbol{x}_d$ and $\boldsymbol{K}_x, \boldsymbol{D}_x \in \mathbb{R}^{3 \times 3}$ are positive definite matrices. In general, this force cannot be applied to the CoM of the system since there is actuation only along the body z-axis, i.e. $\boldsymbol{R}_b \boldsymbol{e}_3$. Therefore, the thrust is chosen to be the projection of the desired force at the CoM $\boldsymbol{f}_d$ onto the body z-axis, resulting in

$$u_1 := \boldsymbol{f}_d^T \boldsymbol{R}_b \boldsymbol{e}_3 \ . \tag{11}$$

In order to obtain the actual $\boldsymbol{f}_d$, then $\boldsymbol{R}_b \boldsymbol{e}_3$ has to converge to the desired body z-axis $\boldsymbol{b}_{3_d}$ defined as

$$\boldsymbol{b}_{3_d} := \frac{\boldsymbol{f}_d}{\|\boldsymbol{f}_d\|} \ . \tag{12}$$

Since a rotation around $\boldsymbol{b}_{3_d}$ does not change the applied thrust, there remains one free parameter in the orientation characterization. As proposed in [14] one can choose $\boldsymbol{b}_{1_d}$ as the desired heading direction of the body x-axis. The resulting desired UAV attitude is

$$\boldsymbol{R}_d := \begin{bmatrix} \boldsymbol{b}_{2_d} \times \boldsymbol{b}_{3_d} & \boldsymbol{b}_{2_d} & \boldsymbol{b}_{3_d} \end{bmatrix} \ , \tag{13}$$

where $\boldsymbol{b}_{2_d} = (\boldsymbol{b}_{3_d} \times \boldsymbol{b}_{1_d}) / \|\boldsymbol{b}_{3_d} \times \boldsymbol{b}_{1_d}\|$ and the heading direction is chosen such that $\boldsymbol{b}_{3_d} \times \boldsymbol{b}_{1_d} \neq \boldsymbol{0}$.

In the following subsection the orientation control which stabilizes the desired UAV attitude $\boldsymbol{R}_d$ is derived. Here, the assumed sufficient smoothness of the desired CoM trajectory $\boldsymbol{x}_d \in \mathcal{C}^4$ is necessary to obtain the desired orientation reference trajectory.

C. Orientation control

To avoid singularities and to maintain the uniqueness of the coordinates, the orientation error is based on SO(3) and defined as

$$\boldsymbol{e}_R := \frac{1}{2}(\boldsymbol{R}_d^T \boldsymbol{R}_b - \boldsymbol{R}_b^T \boldsymbol{R}_d)^\vee \ , \tag{14}$$

where $.^\vee$ is the vee operator[5]. The tracking error for angular velocity is represented in UAV body frame and is given by

$$\tilde{\boldsymbol{\omega}} := \boldsymbol{\omega}_b - \boldsymbol{\omega}_d \ , \tag{15}$$

---

[5]The vee operator $.^\vee : \text{so}(3) \to \mathbb{R}^3$ is the inverse of the hat operator, which is defined by $\boldsymbol{a} \times \boldsymbol{b} = \hat{\boldsymbol{a}}\boldsymbol{b}$. Here, so(3) denotes the Lie algebra of SO(3) consisting of all skew-symmetric matrices in $\mathbb{R}^{3 \times 3}$.

where $\boldsymbol{\omega}_d := \boldsymbol{R}_b^T \boldsymbol{R}_d (\boldsymbol{R}_d^T \dot{\boldsymbol{R}}_d)^\vee$ is the desired angular velocity in UAV body frame[6]. For details on the derivation of these error quantities the interested reader is referred to [14]. The tracking control law is then chosen as

$$\boldsymbol{u}_2 := \boldsymbol{\Gamma}_{2,2} \boldsymbol{\omega}_b + \boldsymbol{\Lambda}_{2,2} \Big[\dot{\boldsymbol{\omega}}_d - \bar{\boldsymbol{\Lambda}}_2^{-1} \Big(k_R \boldsymbol{e}_R + k_\omega \tilde{\boldsymbol{\omega}} \Big) \Big] \ , \quad (16)$$

where $\bar{\boldsymbol{\Lambda}}_2 \in \mathbb{R}^{3\times3}$ is a constant positive definite matrix and $k_R, k_\omega > 0$. The computation of the feedforwarded angular acceleration needs the CoM jerk, which can be obtained by the model (9a). In practical implementations filtering techniques can be employed as well.

D. Task control

The goal of this section is to design $\boldsymbol{u}_3$ to guarantee that the task coordinates $\boldsymbol{y} \in \mathbb{R}^n$ track the desired trajectory $\boldsymbol{y}_d(t)$, i.e. $\tilde{\boldsymbol{y}} := \boldsymbol{y} - \boldsymbol{y}_d(t) \to \boldsymbol{0}$ as $t \to \infty$. Differentiating the task error with respect to time leads to

$$\dot{\tilde{\boldsymbol{y}}} = \boldsymbol{J}\boldsymbol{\xi} - \dot{\boldsymbol{y}}_d = \boldsymbol{J}_1 \dot{\boldsymbol{x}}_g + \boldsymbol{J}_2 \boldsymbol{\omega}_b + \boldsymbol{J}_3 \boldsymbol{\rho} - \dot{\boldsymbol{y}}_d \ , \quad (17)$$

where $\boldsymbol{J} = \begin{bmatrix} \boldsymbol{J}_1 & \boldsymbol{J}_2 & \boldsymbol{J}_3 \end{bmatrix} \in \mathbb{R}^{n\times6+n}$ is the Jacobian matrix providing $\dot{\boldsymbol{y}}$. Defining $\boldsymbol{z} := \boldsymbol{J}_3 \boldsymbol{\rho}$ with $\boldsymbol{J}_3 \in \mathbb{R}^{n\times n}$ and $\boldsymbol{z}_d := \dot{\boldsymbol{y}}_d - \boldsymbol{J}_1 \dot{\boldsymbol{x}}_g - \boldsymbol{J}_2 \boldsymbol{\omega}_b - \boldsymbol{K}_y \tilde{\boldsymbol{y}}$, the control objective can be recast as $\tilde{\boldsymbol{z}} := \boldsymbol{z} - \boldsymbol{z}_d \to \boldsymbol{0}$ as $t \to \infty$. In fact, it follows

$$\tilde{\boldsymbol{z}} = \boldsymbol{0} \iff \dot{\tilde{\boldsymbol{y}}} + \boldsymbol{K}_y \tilde{\boldsymbol{y}} = \boldsymbol{0} \ , \quad (18)$$

which, with $\boldsymbol{K}_y \in \mathbb{R}^{n\times n}$ positive definite, guarantees $\tilde{\boldsymbol{y}}, \dot{\tilde{\boldsymbol{y}}} \to \boldsymbol{0}$. In the hypothesis of an invertible $\boldsymbol{J}_3$, a new change of coordinates can be performed on (9c), leading to

$$\boldsymbol{\Lambda}_z \dot{\boldsymbol{z}} + \boldsymbol{\Gamma}_z \boldsymbol{z} = \boldsymbol{J}_3^{-T} \boldsymbol{u}_3 \ , \quad (19)$$

for which the choice $\boldsymbol{u}_3 = \boldsymbol{J}_3^T \Big(\boldsymbol{\Lambda}_z \dot{\boldsymbol{z}}_d + \boldsymbol{\Gamma}_z \boldsymbol{z}_d - \boldsymbol{K}_z \tilde{\boldsymbol{z}} \Big)$, with $\boldsymbol{K}_z \in \mathbb{R}^{n\times n}$ positive definite, satisfies the required control objective.

As in the previous subsection, the computation of the feedforward term can be obtained using the model (9a) - (9b), or by filtering techniques in practical implementations. In particular, having already designed both u_1 and $\boldsymbol{u}_2$, in

$$\dot{\boldsymbol{z}}_d = \ddot{\boldsymbol{y}}_d - \boldsymbol{J}_1 \ddot{\boldsymbol{x}}_g - \dot{\boldsymbol{J}}_1 \dot{\boldsymbol{x}}_g - \boldsymbol{J}_2 \dot{\boldsymbol{\omega}}_b - \dot{\boldsymbol{J}}_2 \boldsymbol{\omega}_b - \boldsymbol{K}_y \dot{\tilde{\boldsymbol{y}}} \quad (20)$$

all the quantities can be computed using the model and the measurements of the state.

E. Closed-loop system and stability analysis

Before providing the closed-loop system and its stability analysis, the assumptions made throughout the derivation of the control law are summarized.

Assumption 1: $\boldsymbol{J}_3$ is never singular during control action[7].

Assumption 2: The desired thrust of the controller does not vanish, i.e.

$$\| m \ddot{\boldsymbol{x}}_d + mg \boldsymbol{e}_3 - \boldsymbol{K}_x \tilde{\boldsymbol{x}} - \boldsymbol{D}_x \dot{\tilde{\boldsymbol{x}}} \| \neq 0 \ . \quad (21)$$

---

[6]The expression of the angular velocity error is the same as in [14], but in this paper $\boldsymbol{\omega}_d$ is directly expressed in UAV body frame.

[7]In the planning phase $\boldsymbol{y}_d$ and $\boldsymbol{x}_d$ have to be chosen in such a way that this assumption can be fulfilled.
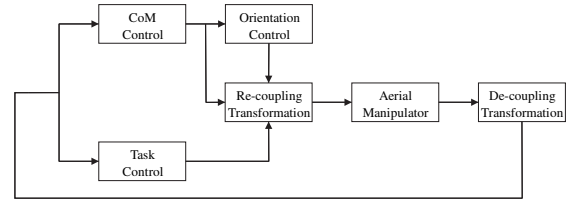

Fig. 2. Overview of the closed-loop system.

Assumption 3: $\| m \ddot{\boldsymbol{x}}_d + mg \boldsymbol{e}_3 \| < \gamma$ for a given $\gamma > 0$. The first assumption is similar to the requirement used in space robotics to have a nonsingular generalized Jacobian matrix [18], [19] and therefore a similar treatment could be considered in future works. If in (3) $\boldsymbol{\omega}_b$ is replaced by the total angular momentum, then it was shown in [10] that $\boldsymbol{J}_3$ is exactly the generalized Jacobian matrix. On the other hand, Assumptions 2 - 3 are necessary for the derivation of the CoM control and orientation control, which follow the same lines as the controller in [14], where these assumptions are introduced. In particular, Assumptions 3 is necessary in order to have an upper bound for the Lyapunov function.

Considering the expressions of the intermediate control inputs, the final control law is

$$\begin{bmatrix} f\boldsymbol{e}_3 \\ \boldsymbol{\mu} \\ \boldsymbol{\tau} \end{bmatrix} = [\boldsymbol{J}_b^T \ \ \boldsymbol{Q}^T]^{-1} \boldsymbol{T}^T \left(\begin{bmatrix} \boldsymbol{O} \\ \boldsymbol{\Gamma}_{2,2} \boldsymbol{\omega}_b + \boldsymbol{\Gamma}_{2,3} \boldsymbol{\rho} \\ -\boldsymbol{\Gamma}_{2,3}^T \boldsymbol{\omega}_b \end{bmatrix} + \right.$$
$$\left. + \begin{bmatrix} \boldsymbol{f}_d^T \boldsymbol{R}_b \boldsymbol{e}_3 \boldsymbol{R}_b \boldsymbol{e}_3 \\ \boldsymbol{\Lambda}_{2,2} \dot{\boldsymbol{\omega}}_d - \boldsymbol{\Lambda}_{2,2} \bar{\boldsymbol{\Lambda}}_2^{-1} (k_R \boldsymbol{e}_R + k_\omega \tilde{\boldsymbol{\omega}}) \\ \boldsymbol{J}_3^T (\boldsymbol{\Lambda}_z \dot{\boldsymbol{z}}_d + \boldsymbol{\Gamma}_z \boldsymbol{z}_d - \boldsymbol{K}_z \tilde{\boldsymbol{z}}) \end{bmatrix} \right) , \quad (22)$$

with in particular $f = u_1 = \boldsymbol{f}_d^T \boldsymbol{R}_b \boldsymbol{e}_3$. Fig. 2 provides an overview of the control scheme. The control law leads to the closed-loop system

$$m\ddot{\tilde{\boldsymbol{x}}} + \boldsymbol{K}_x \tilde{\boldsymbol{x}} + \boldsymbol{D}_x \dot{\tilde{\boldsymbol{x}}} = \Big(\boldsymbol{R}_b \boldsymbol{e}_3 \boldsymbol{e}_3^T \boldsymbol{R}_b^T - \boldsymbol{E}_3 \Big) \boldsymbol{f}_d \ , \quad (23a)$$

$$\frac{\mathrm{d}}{\mathrm{d}t} \Big(\boldsymbol{R}_d^T \boldsymbol{R}_b \Big) = \Big(\boldsymbol{R}_d^T \boldsymbol{R}_b \Big) \hat{\tilde{\boldsymbol{\omega}}} \ , \quad (23b)$$

$$\bar{\boldsymbol{\Lambda}}_2 \dot{\tilde{\boldsymbol{\omega}}} + k_\omega \tilde{\boldsymbol{\omega}} + k_R \boldsymbol{e}_R = \boldsymbol{0} \ , \quad (23c)$$

$$\dot{\tilde{\boldsymbol{y}}} + \boldsymbol{K}_y \tilde{\boldsymbol{y}} = \tilde{\boldsymbol{z}} \quad (23d)$$

$$\boldsymbol{\Lambda}_z \dot{\tilde{\boldsymbol{z}}} + \Big(\boldsymbol{\Gamma}_z + \boldsymbol{K}_z \Big) \tilde{\boldsymbol{z}} = \boldsymbol{0} \ , \quad (23e)$$

(where $\hat{\boldsymbol{a}} \boldsymbol{b} = \boldsymbol{a} \times \boldsymbol{b}$) whose stability property are summarized in the following proposition.

Proposition 1: Given the Assumptions 1 - 3, the system (23) has a locally uniformly asymptotically stable equilibrium point in the origin, provided that $k_R, k_\omega > 0$ and $\boldsymbol{K}_x, \boldsymbol{D}_x, \boldsymbol{K}_y, \boldsymbol{K}_z$ are positive definite diagonal matrices.

Proof: Indicating with $\mathrm{tr}(A)$ the trace of a matrix A, the assertion is proved using the candidate Lyapunov function

$$V = V_1 + \frac{1}{2} \tilde{\boldsymbol{z}}^T \boldsymbol{\Lambda}_z \tilde{\boldsymbol{z}} + \tilde{\boldsymbol{y}}^T \boldsymbol{K}_y \boldsymbol{K}_z \tilde{\boldsymbol{y}} \ , \quad (24)$$

$$V_1 = \frac{1}{2} m \| \dot{\tilde{\boldsymbol{x}}} \|^2 + \frac{1}{2} \tilde{\boldsymbol{x}}^T \boldsymbol{K}_x \tilde{\boldsymbol{x}} + \gamma_1 \tilde{\boldsymbol{x}}^T \dot{\tilde{\boldsymbol{x}}} + $$
$$+ \frac{1}{2} \tilde{\boldsymbol{\omega}}^T \bar{\boldsymbol{\Lambda}}_{2,2} \tilde{\boldsymbol{\omega}} + k_R \mathrm{tr}(\boldsymbol{E}_3 - \boldsymbol{R}_d^T \boldsymbol{R}_b) + \gamma_2 \boldsymbol{e}_R^T \tilde{\boldsymbol{\omega}} \quad (25)$$

where $\gamma_1, \gamma_2 > 0$ are small constants satisfying the inequalities in [14]. The function V_1 has been provided in [14] and, using Assumptions 2 - 3, it was shown to be a strict Lyapunov function for the system (23a) - (23c), with $W_{1,m} \leq V_1 \leq W_{1,M}$, $W_{1,m}, W_{1,M}$ positive definite and $\dot{V}_1$ negative definite. Since the minimum and maximum eigenvalues of $\boldsymbol{\Lambda}_z$ are always nonnegative, then it also holds $W_m \leq V \leq W_M$, with W_m, W_M positive definite. Additionally, using the passivity property $\dot{\boldsymbol{\Lambda}}_z = \boldsymbol{\Gamma}_z + \boldsymbol{\Gamma}_z^T$ (which holds thanks to assumption 1) and the diagonal structure of the gain matrices, it follows

$$
\begin{aligned}
\dot{V} &= \dot{V}_1 - \tilde{\boldsymbol{z}}^T \boldsymbol{K}_z \tilde{\boldsymbol{z}} - 2\tilde{\boldsymbol{y}}^T \boldsymbol{K}_y \boldsymbol{K}_z \left(\boldsymbol{K}_y \tilde{\boldsymbol{y}} - \tilde{\boldsymbol{z}} \right) \\
&= \dot{V}_1 - \begin{bmatrix} \tilde{\boldsymbol{z}}^T & \tilde{\boldsymbol{y}}^T \end{bmatrix} \begin{bmatrix} \boldsymbol{K}_z & -\boldsymbol{K}_z \boldsymbol{K}_y \\ -\boldsymbol{K}_y \boldsymbol{K}_z & 2\boldsymbol{K}_y^2 \boldsymbol{K}_z \end{bmatrix} \begin{bmatrix} \tilde{\boldsymbol{z}} \\ \tilde{\boldsymbol{y}} \end{bmatrix},
\end{aligned} \quad (26)
$$

which is negative definite, as for the second term the Schur complement conditions for positive definite matrices hold, i.e. $\boldsymbol{K}_z > 0$ and $2\boldsymbol{K}_y^2 \boldsymbol{K}_z - \boldsymbol{K}_y^2 \boldsymbol{K}_z > 0$. Therefore, V is a strict (time-varying) Lyapunov function [20]. ∎

IV. VALIDATION

The proposed control law is validated in two numerical simulations, referred to as nominal and disturbed in the following, for the system depicted in Fig. 1. While the control law used in both cases is the same, some additional effects (not included in the nominal model) are added in the disturbed case. The goal of the disturbed case is to show the capability of the controller to cope with unmodeled effects such as the additional dynamics due to the spinning parts (rotors and drive train components) and actuation dynamics (swash-plate dynamics). The helicopter is modeled as a 6-DoF rigid body, treating the other effects as a disturbance by the controller. In particular, the rotors are simulated to rotate at a constant speed, while the dynamics of the commanded thrust and torques of the helicopter are assumed to be given by a first order linear system with a rise time of about 0.1 s. Robustification of the proposed approach is relevant, but beyond the scope of the current version of the paper. The KUKA Lightweight Robot is modeled as a 7-DoF rigid robot. The inertial properties and the actuator limitations of the aerial manipulator are summarized in Table I.

A. Task and Trajectory Specification

As task it is chosen to track a desired 6-Dof end-effector pose $\boldsymbol{x}_{e,d}(t) \in \mathbb{R}^3 \times \mathrm{SO}(3)$ while keeping the third joint of the manipulator at an angle of $q_{3d} = 0$, i.e.

$$
\boldsymbol{y}_d(t) := \begin{bmatrix} \boldsymbol{x}_{ed}(t) \\ \boldsymbol{R}_{ed}(t) \\ q_{3d}(t) \end{bmatrix} \in \mathbb{R}^3 \times \mathrm{SO}(3) \times \mathbb{R} . \quad (27)
$$

As an automated planning procedure was beyond the scope of this paper, it has been chosen to decomposed the translational part of the end-effector task into a "basic" motion $\boldsymbol{x}_c(t)$ and an "agile" part $\boldsymbol{x}_a(t)$, where the first has to be realized by the whole system, while the second by the manipulator only. The choice of $\boldsymbol{x}_c(t)$ and $\boldsymbol{x}_a(t)$ is obviously

Helicopter

mass	37.6 kg
inertia	$\mathrm{diag}(1.46, 0.36, 1.46)$ kgm^2
max. thrust	980.6 N
max. torque	200 Nm
angular velocity rotors	$\pm 26\pi$ rad/s
rotor diameter	2.54 m
rotor mass	1.3 kg

KUKA Lightweight Robot

mass robot	17.2 kg
max. joint torque	
q_1, q_2	165 Nm
q_3, q_4, q_5	70 Nm
q_6, q_7	30 Nm

not unique and it should be the output of the planning phase. In particular, in the simulations, the trajectory is chosen as

$$
\boldsymbol{x}_{ed}(t) = \underbrace{\begin{bmatrix} 0.6 \\ 0.0 \\ 1.3 \end{bmatrix} + 0.5 \begin{bmatrix} -\sin(\frac{t}{2}) \\ -\cos(\frac{t}{2}) \\ \sin(\frac{t}{2}) + \frac{45}{1000}t \end{bmatrix}}_{\boldsymbol{x}_c(t)} + \underbrace{0.1 \begin{bmatrix} \sin(3t) \\ \cos(3t) \\ \cos(3t) \end{bmatrix}}_{\boldsymbol{x}_a(t)},
$$
$$ (28) $$

$$
\boldsymbol{R}_{ed}(t) = \boldsymbol{R}_1(\varphi_1(t)) \boldsymbol{R}_2(\varphi_2(t)) \boldsymbol{R}_3(\varphi_3(t)) \quad (29)
$$

$$
q_{3d}(t) \equiv 0, \quad (30)
$$

where $\boldsymbol{R}_i$ is the rotation matrix about the i-th body axis and

$$
\varphi_1(t) = 0.5 \sin(3t) , \quad \varphi_2(t) = \varphi_3(t) = 0.5 \cos(3t) . \quad (31)
$$

The reference for the helicopter heading direction is chosen as

$$
\boldsymbol{b}_{1d}(t) = \begin{bmatrix} \cos(\varphi_4(t)), \sin(\varphi_4(t)), 0 \end{bmatrix} . \quad (32)
$$

Here, $\varphi_4(t) = 0.5t$ corresponds to the global yaw angle reference for the helicopter. In order to keep the robot in a similar posture independently from the global yaw angle, the desired CoM trajectory $\boldsymbol{x}_d$ is chosen as

$$
\boldsymbol{x}_d = \boldsymbol{x}_c + \boldsymbol{R}_3(\varphi_4(t)) \boldsymbol{r}_o, \quad (33)
$$

where $\boldsymbol{r}_o = \begin{bmatrix} -0.3 & 0.0 & 0.6 \end{bmatrix}^T$ m is a constant position offset between $\boldsymbol{x}_c(t)$ and $\boldsymbol{x}_d(t)$, in order to obtain a coordinated motion of the whole system while tracking $\varphi_4(t)$.

B. Simulation

The simulation is carried on using the algorithm in [21], with a sampling time of 10^{-3} s. The controller gains are given in Table II.

During the execution of the task, the system is perturbed by external forces and torques unknown to the controller. For 4 s $\leq t \leq 5$ s (green area in the plots) a constant force of 50 N in $\boldsymbol{e}_2$ direction is applied to the helicopter, while for 9 s $\leq t \leq 10$ s (orange area in the plots) the end-effector is affected by a constant force of 40 N and a constant moment of 20 Nm along all coordinate axes. As depicted in Fig. 3 the end-effector tracking is almost not affected by the external force acting on the helicopter. This is an important feature of

TABLE II

CONTROL PARAMETERS

$\boldsymbol{K}_x$	800 N/m
$\boldsymbol{D}_x$	420 Ns/m
k_R	250 Nm
k_ω	200 Nm/rad
$\overline{\boldsymbol{\Lambda}}_2$	$\mathrm{diag}(1.46, 0.36, 1.46)\ \mathrm{kgm}^2$
$\boldsymbol{K}_y$	$\mathrm{blkdiag}(30\boldsymbol{E}_3, 10\boldsymbol{E}_3, 7)\ \mathrm{1/s}$
$\boldsymbol{K}_z$	$30\boldsymbol{E}_7\ \mathrm{Nms/rad}$
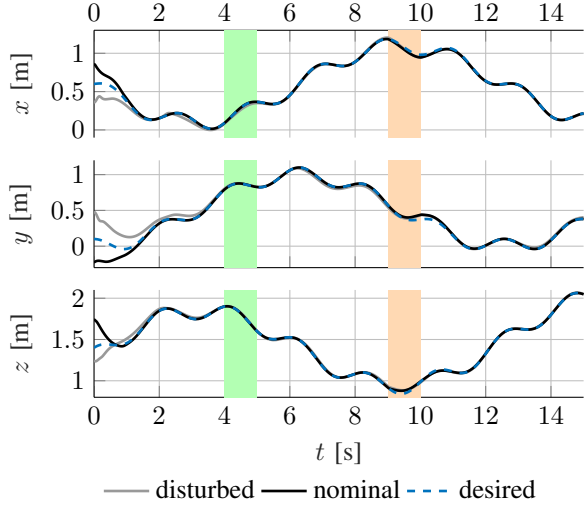
Fig. 3. Tracking performance for the end-effector along the three coordinate axis. Different initial conditions are used to distinguish the nominal and perturbed case. In the green area an external perturbation acts on the helicopter, while in the orange one the perturbation is at the end-effector.
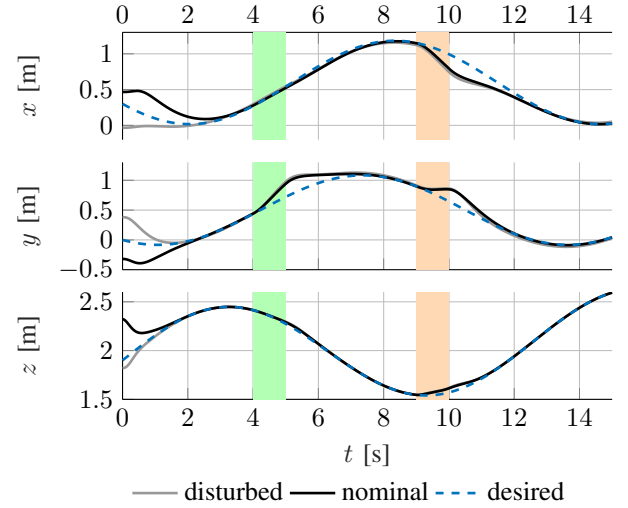

Fig. 4. Tracking performance for the CoM along the three coordinate axis. Different initial conditions are used to distinguish the nominal and perturbed case. In the green area an external perturbation acts on the helicopter, while in the orange one the perturbation is at the end-effector.
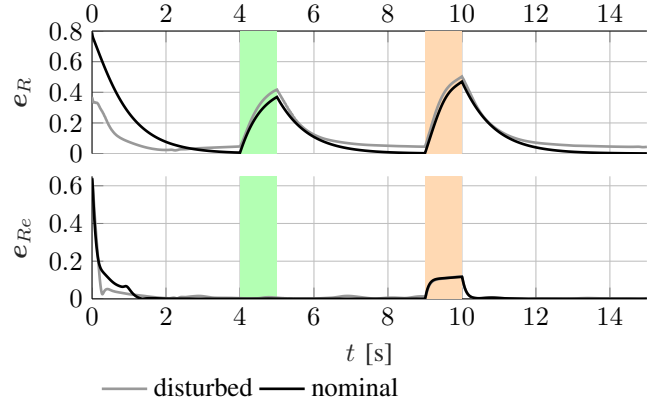

Fig. 5. Orientation error of the UAV (top plot) and of the end-effector (bottom plot). Different initial conditions are used to distinguish the nominal and perturbed case. In the green area an external perturbation acts on the helicopter, while in the orange one the perturbation is at the end-effector.

the controller, as the helicopter is the subsystem more likely to be disturbed by aerodynamic effects in real scenarios. Fig. 4 shows instead the tracking performances for the CoM, while in Fig. 5 the orientation errors for the base and end-effector are presented. In all the three figures, the results of both the perturbed case (grey line) and nominal case (black line) are reported, with different initial conditions to better visualize the two cases. Finally, in Fig. 6, all the control inputs are plotted to show the feasibility of the controller. As it can be noticed, besides a very short initial time interval in which some inputs are saturated, all the signals are within the physical limits of the system.

V. CONCLUSION

Aerial manipulation is a relatively young and active research field with still many open problems. Having a system capable of tracking a desired trajectory in task space is a basic feature, on which the execution of almost any task will rely. A solution to this problem has been presented, which closely combines the control design to a change of coordinates. The latter, based on physical principles, allows to rewrite the system in a form showing a partial decoupling between the dynamics of the UAV and the manipulator. The control design for the overall system can be then separately performed for the two subsystems. The result is a control law capable to asymptotically stabilize the underactuated aerial

manipulator, as shown theoretically and in simulations.

Extensions of the work can explore the planning phase of the desired trajectories as well as alternative stabilizing control laws in order to minimize model-based cancellations, in favor of feedforward injections.

APPENDIX

Further details on the structure of the transformed system are provided in this Appendix.

The block-diagonal structure of the transformed inertia matrix is guaranteed by the following three conditions:

$$\boldsymbol{J}_g \boldsymbol{M}^{-1} \boldsymbol{J}_{b,\omega}^T = \boldsymbol{N} \boldsymbol{M}^{-1} \boldsymbol{J}_g^T = \boldsymbol{N} \boldsymbol{M}^{-1} \boldsymbol{J}_{b,\omega}^T = \boldsymbol{0} \ . \quad (34)$$

While the last two are obtained by construction through the choice of $\boldsymbol{N}$, the first is physically motivated and can be derived following a procedure similar to [10]. Assume for simplicity that in (1) the torques $\boldsymbol{\mu}$ at the base are the only
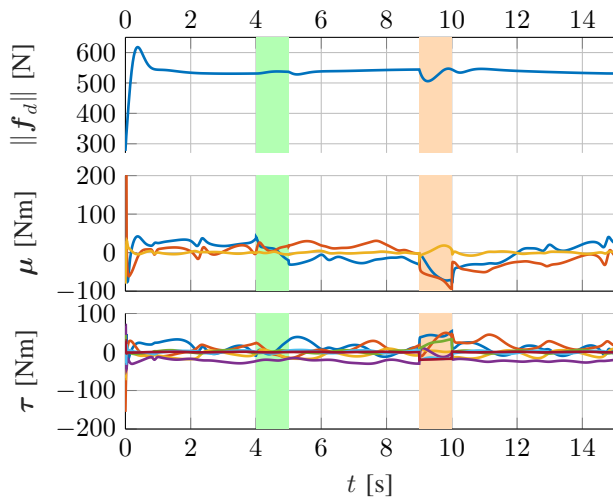
Fig. 6. Control input for the UAV (top two plots) and torques of the manipulator (bottom plot) in the nominal case. In the green area an external perturbation acts on the helicopter, while in the orange one the perturbation is at the end-effector.

input and no gravity is acting on the system. It follows from Newton's second law that the CoM acceleration must be zero. The latter can be computed using the Jacobian matrix $\boldsymbol{J}_g$ and therefore leads to

$$\boldsymbol{0} = \ddot{\boldsymbol{x}}_g = \boldsymbol{J}_g \boldsymbol{M}^{-1}\left(\boldsymbol{J}_{b,\omega}^T \boldsymbol{\mu} - \boldsymbol{C}\boldsymbol{v}\right) + \dot{\boldsymbol{J}}_g \boldsymbol{v} \ , \qquad (35)$$

in which $\dot{\boldsymbol{v}}$ has been computed using (1). Since (35) must hold for every possible choice of $\boldsymbol{v}$ and $\boldsymbol{\mu}$, then

$$\boldsymbol{J}_g \boldsymbol{M}^{-1} \boldsymbol{J}_{b,\omega}^T = \boldsymbol{0} \ , \qquad (36)$$

$$\left(\boldsymbol{J}_g \boldsymbol{M}^{-1}\boldsymbol{C} - \dot{\boldsymbol{J}}_g\right)\boldsymbol{v} = \boldsymbol{0} \ , \qquad (37)$$

where (36) is the required dynamic consistent relationship, while it can be shown that (37) appears in the transformed Coriolis matrix and it is the reason why the centrifugal and Coriolis effects disappear in the CoM dynamics after the transformation. In particular, it is always possible to choose a factorization for which the first three rows and columns of $\boldsymbol{\Gamma}$ are identically zero [22].

Instead of computing directly the transformed inertia matrix $\boldsymbol{\Lambda} = \left(\boldsymbol{T}\boldsymbol{M}^{-1}\boldsymbol{T}^T\right)^{-1}$, it is easier to evaluate its inverse. By using (34) and the block matrix expression of $\boldsymbol{T}$ in (3), the block-diagonal structure is immediately obtained . Moreover,

$$\boldsymbol{\Lambda}_1 = \left(\boldsymbol{J}_g \boldsymbol{M}^{-1}\boldsymbol{J}_g^T\right)^{-1} \ , \qquad (38)$$

which in [10] it was shown to be equal to $m\boldsymbol{E}_3$. Therein, it is also provided the expression of the transformed $\boldsymbol{g}$ that appears in (5).

Concerning $\boldsymbol{\Gamma}$, the condition $\boldsymbol{\Gamma}_{3,2} = -\boldsymbol{\Gamma}_{2,3}^T$ is a direct consequence of the passivity property $\dot{\boldsymbol{\Lambda}} = \boldsymbol{\Gamma} + \boldsymbol{\Gamma}^T$ and the block-diagonal structure of $\dot{\boldsymbol{\Lambda}}$.

REFERENCES

[1] E. Feron and E. Johnson, *Aerial robotics chapter: Springer Handbook of Robotics*. Heidelberg, Germany: Springer, 2008.

[2] M. Orsag, C. Korpela, and P. Oh, "Modeling and Control of MM-UAV: Mobile Manipulating Unmanned Aerial Vehicle," *Journal of Intelligent & Robotic Systems*, vol. 69, no. 1, pp. 227–240, 2013.

[3] K. Kondak, F. Huber, M. Schwarzbach, M. Laiacker, D. Sommer, M. Bejar, and A. Ollero, "Aerial manipulation robot composed of an autonomous helicopter and a 7 degrees of freedom industrial manipulator," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, Hong Kong, China, May 2014, pp. 2107–2112.

[4] J. Á. Acosta, M. I. Sánchez, and A. Ollero, "Robust control of underactuated aerial manipulators via IDA-PBC," in *IEEE Conf. on Decision and Control*, Los Angeles, USA, Dec. 2014, pp. 673–678.

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

[6] E. Cataldi, G. Muscio, M. Á. Trujillo, Y. Rodriguez, F. Pierri, G. Antonelli, F. Caccavale, A. Viguria, S. Chiaverini, and A. Ollero, "Impedance control of an aerial-manipulator: Preliminary results," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, Daejeon, South Korea, Oct. 2016, pp. 3848–3853.

[7] M. Tognon, B. Yüksel, G. Buondonno, and A. Franchi, "Dynamic decentralized control for protocentric aerial manipulators," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, Singapore, Singapore, May 2017, pp. 6375–6380.

[8] M. J. Kim, K. Kondak, and C. Ott, "A stabilizing controller for regulation of UAV with manipulator," *IEEE Robotics and Automation Letters (RA-L)*, vol. 22, no. 5, pp. 974–986, 2018.

[9] C. Ott, *Cartesian Impedance Control of Redundant and Flexible-Joint Robots*, ser. Springer Tracts in Advanced Robotics. Berlin: Springer-Verlag, 2008.

[10] G. Garofalo, B. Henze, J. Englsberger, and C. Ott, "On the inertially decoupled structure of the floating base robot dynamics," in *8th Vienna International Conference on Mathematical Modelling (2015)*, Vienna, Austria, Feb. 2015, pp. 322–327.

[11] A. M. Giordano, G. Garofalo, M. de Stefano, C. Ott, and A. Albu-Schäffer, "Dynamics and control of a free-floating space robot in presence of nonzero linear and angular momenta," in *IEEE Conf. on Decision and Control*, Las Vegas, USA, Dec. 2016, pp. 7527–7534.

[12] G. Hirzinger, N. Sporer, A. Albu-Schäffer, M. Hähnle, R. Krenn, A. Pascucci, and M. Schedl, "DLR's torque-controlled light weight robot III - are we reaching the technological limits now?" in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, Washington, USA, May 2002, pp. 1710–1716.

[13] M. Takegaki and S. Arimoto, "A new feedback method for dynamic control of manipulators," *Journal of Dynamic Systems, Measurement, and Control*, vol. 103, no. 2, pp. 119–125, 1981.

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

[15] A. Roza and M. Maggiore, "Path following controller for a quadrotor helicopter," in *2012 American Control Conference (ACC)*, Montreal, Canada, June 2012, pp. 4655–4660.

[16] P. D. Monte and B. Lohmann, "Trajectory tracking control for a quadrotor helicopter based on backstepping using a decoupling quaternion parametrization," in *21st Mediterranean Conference on Control and Automation*, Chania, Greece, June 2013, pp. 507–512.

[17] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: Modelling, Planning and Control*. Springer Publishing Company, Incorporated, 2008.

[18] E. Papadopoulos and S. Dubowsky, "Dynamic singularities in free-floating space manipulators," *ASME Journal of dynamic systems, measurement, and control*, vol. 115, no. 1, pp. 44–52, 1993.

[19] K. Nanos and E. Papadopoulos, "Avoiding dynamic singularities in cartesian motions of free-floating manipulators," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 51, no. 3, pp. 2305–2318, 2015.

[20] H. K. Khalil, *Nonlinear Systems*. New Jersey: Prentice Hall, 2002.

[21] G. Garofalo, C. Ott, and A. Albu-Schäffer, "On the closed form computation of the dynamic matrices and their differentiations," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, Tokyo, Japan, Nov. 2013, pp. 2364–2359.

[22] G. D. Niemeyer and J.-J. E. Slotine, "Performance in adaptive manipulator control," *Int. Journal of Robotics Research*, vol. 10, no. 2, pp. 149–161, 1991.