Attitude Stabilization Control of an Aerial Manipulator using a Quaternion-Based Backstepping Approach

Stefano Di Lucia Gian Diego Tipaldi Wolfram Burgard
Autonomous Intelligent Systems Group, University of Freiburg, Germany
{dilucia, tipaldi, burgard}@informatik.uni-freiburg.de

***Abstract*—Aerial manipulation is a new research area that extends the potential of aerial vehicles, allowing them to physically interact with the environment. Modeling and control of such a system is not a trivial problem due to the coupled dynamics of the aerial vehicle and the robotic manipulator. This paper presents a robust and non linear control system for a Hexacopter equipped with a two-degrees-of-freedom robotic arm. The proposed mathematical model, which is based on a quaternion representation, takes into account the changing inertia and the mass distribution, depending on the arm configuration. We show how the attitude control, using a quaternion-based backstepping, is able to react to the disturbances caused by the arm. In the backstepping design, we introduce a command filter, which contributes to reduce the computational complexity of the algorithm. Finally, we present simulation results that demonstrate the stability of our approach and provide a comparison to a standard PD control.**

I. INTRODUCTION

In recent years, both UAVs and mobile manipulation with ground vehicles have seen a growing research interest due to the numerous industrial applications. The combination of the agile mobility of an aerial vehicle with the capability of a robotic manipulator invites new applications, i.e., transportation in remote places, manipulation and construction in dangerous and inaccessible sites, and rescue operations. Moreover, it is possible to consider multiple UAVs for cooperative tasks (e.g., transportation of heavy payload or cooperative building) like proposed in the ARCAS (Aerial Robotics Cooperative Assembly System) project [1].

The inclusion of a manipulator not only contributes to increase the payload but also affects the dynamics of the system and makes the problem of stability and controllability much harder. In particular, the behavior of the robot changes due to the modification of the mass distribution and the moment of inertia depending on the configuration of the arm. Moreover, the contact forces during manipulation introduce further destabilizing effects. These challenges make this problem interesting and non-trivial.

Our main focus is the design of a robust control law for a Hexacopter endowed with a two-degrees of-freedom (2-DoF) arm. In such a system the development of a robust control law is required to compensate the torque given by the motion of the arm and its interaction with the environment. Furthermore, the control has to ensure robustness against parametric uncertainties as well as unmodeled dynamics. We design the equation of the motion of the whole system using the Newton-Euler formulation for translational and rotational dynamics of a rigid body. For modeling the attitude dynamics, we use the quaternion approach, which helps to circumvent the singularities that occur with Euler angles. It allows further improvements in complex airborne tasks that require singular configurations, like opening a door or manipulating objects on the ceiling. Moreover, a quaternion-based controller reduces the computational cost of an already complex control algorithm. To control the attitude of the robot we propose a command filtered backstepping [2] controller. The command filter is an extension of the backstepping approach that simplifies its implementation by obviating the need for analytical computation of command signal derivatives. The main contribution of our work lies in considering explicitly a time-variant inertia and its derivative in the mathematical model. This allows us to increase the robustness of our control with respect to previous control approaches and, in turn, helps in dealing with unmodeled parameters that can affect the stability of the robot during manipulation tasks. We also show with simulations that our approach is promising for applications that require precise object manipulation.

The paper is structured as follows: Section III describes dynamics and kinematics of the system, while Section IV explains the description of the control system and its stability. Finally, Section V shows simulation results of the proposed control, also in comparison with a non-linear PD control.

II. RELATED WORK

To the best of our knowledge, relatively few contributions have been made in the area of manipulation using areal vehicles. Many previous approaches focus on contact inspection [3], [4], slung load transportation [5], [6] or grasping with a 1-DoF gripper [7], [8]). The latter two approaches also used a team of robots. For example, Lindsey *et al.* [9] constructed a cubic structure with a team of quadrotors. Furthermore, Micheal *et al.* [10] manipulated a payload into a desired position with the use of cables attached to three quadrotors. These applications focus on grasping and contact tasks without realizing the complexity of a robotic arm.

Other researchers introduced a multiple degree of freedom arm, with the intent to expand the capabilities of such a system and perform more complex actions. For this purpose both helicopters and multirotors have been used. In Kondak *et al.* [11] a small scaled helicopter is equipped with a robotic arm and the interaction forces between the two subsystems are analyzed. Lippiello *et al.* [12] present an Euler-Lagrange formalism for the combined system. They further show that the Cartesian impedance control establishes a mass-damper-spring relationship between the motion of the whole system and the external forces acting on it. Kim *et al.* [13] designed a mathematical model of the combined system and developed an adaptive sliding mode controller. They also performed an experiment consisting of picking and delivering an object inside a shelf. Jimenez-Cano *et al.* [14] proposed a Newton-Euler approach for the dynamics and used Variable Parameter Integral Backstepping (VPIB) control. Their controller is able to stabilize the end-effector in a fixed position for precise aerial manipulation tasks. They showed good results both with simulations and with outdoor experiments. Caccavale *et al.* [15] proposed a two-layer adaptive controller for tracking motion references using the Euler-Lagrange formulation. Heredia *et al.* [16] developed a large-payload outdoor octocopter endowed with a 7-DoF arm. They present experiments illustrating the performance of the backstepping attitude control in challenging conditions including lateral wind. However, in contrast to our approach, all these works used an Euler angles parametrization of the system, which introduces singularities. In addition, most of them did not analyze the inertial properties of the system, that affect the robustness of the control for particular movements of the arm, e.g. when it is moving fast.

Orsag *et al.* [17] and Korpela *et al.* [18], [19] used a Euler angle parametrization for the system and introduced a mathematical formalism that describes the variations in the moment of inertia. In their works, the authors pushed the research field towards complex aerial manipulation. In particular, they proposed a PI-D attitude control and applied Routh-Hurwitz stability criteria to stabilize a quadrotor with dual multi-degree manipulators. They performed experiments involving grab and drop of a foam block and a long cylinder as well as valve turning. Due to the slow movements of the arm, they consider an approximation of the derivative of the inertia matrix and it does not appear in the equations of motion. In contrast to them, we introduce it in the mathematical model and our derived controller is also able to react to fast movements.

Álvarez-Muñoz *et al.* [20] employed a quaternion-based parameterization for such a system. They consider a varying mass distribution but a symmetric constant inertia matrix. This allows them to simplify the inertia term in the design of the control law, as it is not considered explicitly. In contrast to this, we use time varying inertia. In this way, we improve the robustness of the proposed control, that is able to adapt to the substantial changes of the inertia moment.

III. MODEL DESIGN

This section discusses the quaternion-based mathematical model used for the controller design and analyzes the differential equations governing the whole system.

A. Coordinate frames and unit quaternion kinematics

To establish the dynamic equations of the system, we need to define the coordinate frames. Fig. 1 shows the configuration of the coordinates for the system considered in this paper, i.e., an hexacopter equipped with a 2-DoF robotic arm. The terms

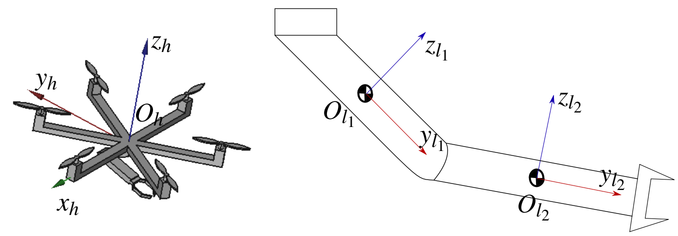


Fig. 1: Configuration of the coordinates of the system.

O_w, O_h and O_{l_i} represent world, body and links coordinate frame respectively. Rotations between the reference frames is represented by the attitude matrix:

$$R \in SO(3) = \left\{ R \in \mathbb{R}^{3\times 3} : R^T R = I, det(R) = 1 \right\}, \quad (1)$$

where I is the 3×3 identity matrix. The rotation matrix can be also expressed by the Euler-Rodrigues' formula using the quaternion parametrization:

$$R(q) = I + 2\eta[\varepsilon^\times] + 2[\varepsilon^\times]^2, \quad (2)$$

where the cross product between two vectors $\omega, v \in \mathbb{R}^3$ is represented by a matrix multiplication $[w^\times]v = \omega \times v$. Please note that a quaternion $q = [\eta \ \ \varepsilon^T]^T \in \mathbb{R}\times\mathbb{R}^3$ is a unit vector and can be represented as:

$$q = \begin{bmatrix} \cos\frac{\beta}{2} \\ \hat{k}\sin\frac{\beta}{2} \end{bmatrix} = \begin{bmatrix} \eta \\ \varepsilon \end{bmatrix}. \quad (3)$$

Is also worth noticing that quaternions q and $-q$ represent the same attitude but opposite directions of rotation axis. Let $\omega = [\omega_x \quad \omega_y \quad \omega_z]^T$ be the angular velocity vector of the hexacopter coordinate frame O_h, relative to the world reference frame O_w. The quaternion kinematics is given by:

$$\begin{bmatrix} \dot{\eta} \\ \dot{\varepsilon} \end{bmatrix} = \frac{1}{2}\begin{bmatrix} -\varepsilon^T \\ \eta I + [\varepsilon^\times] \end{bmatrix}\omega = \frac{1}{2}Q(q)\omega. \quad (4)$$

B. Dynamics and kinematics of the whole system

The rigid body mathematical model of the hexacopter has been already analyzed in several works and it is well understood. Typically, it is assumed that the mass distribution of the hexacopter is symmetric, which lead to simplified model equations. With the introduction of a manipulator the mass distribution is no longer symmetric and its movement produces a twofold effect: a shift in the center of mass and a change in the moment of inertia. In this section we present the mathematical model which considers the manipulator configuration.

Considering a hexacopter with a 2-DoF robotic arm, let $\theta = [\theta_1, \theta_2]$ be the joint coordinates of the arm. The motion of the system can be decomposed into translational and rotational components, whereas the dynamics of the combined system can be described by the Newton-Euler formalism:

$$T : \begin{cases} \dot{p} = v \\ m_{tot}\dot{v} = -m_{tot}g + R(q)F_b \end{cases}, \quad (5)$$

$$R : \begin{cases} \dot{q} = \frac{1}{2}Q(q)\omega \\ J_{tot}(\theta)\dot{\omega} + [[\mathbf{D}_\theta J_{tot}(\theta)]\dot{\theta}]\omega = -[\omega^\times]J_{tot}(\theta)\omega + \tau_{tot} \end{cases}, \quad (6)$$

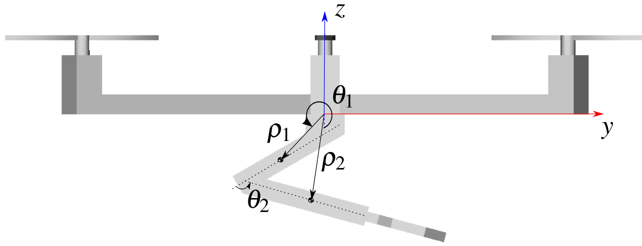


Fig. 2: Position vectors of each links w.r.t. O_h

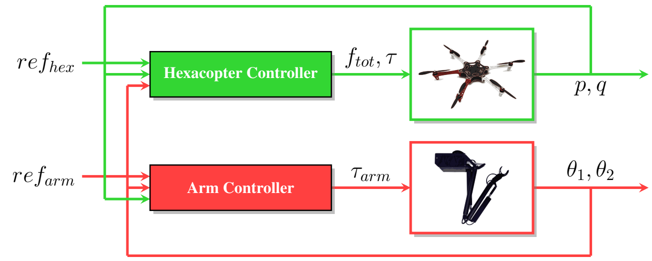


Fig. 3: Control scheme of the system.

where p and v are linear position and velocity in body frame coordinates, g is the acceleration due to gravity, $m_{tot} \triangleq m_{hexa} + m_{arm} + m_{load}$ is the total mass of the system, $R(q)$ represents the rotation matrix given in Eq. (2) and being $J_{tot}(\theta)$ the inertia matrix of the system with respect to the joint angles. $\mathbf{D}_\theta J_{tot}(\theta) \in \mathbb{R}^{3\times3\times2}$ is the differential operator of the inertia matrix with respect to the joint coordinates. As a consequence, $[[\mathbf{D}_\theta J_{tot}(\theta)]\dot{\theta}] \in \mathbb{R}^{3\times3}$ is the time derivative of $J_{tot}(\theta)$. Furthermore, $F_b \triangleq [0 \quad 0 \quad f_{tot}]^T$ and $\tau_{tot} \triangleq \tau + \tau_{arm}(\theta)$ respectively are the total thrust generated by the rotors and the total torque acting on the UAV. Both f_{tot} and τ are related to the six propeller thrusts $f_i, i = 1,\ldots,6$ via the following relation:

$$\begin{bmatrix} f_{tot} \\ \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ -\frac{d}{2} & -d & -\frac{d}{2} & \frac{d}{2} & d & \frac{d}{2} \\ -\frac{\sqrt{3}}{2}d & 0 & \frac{\sqrt{3}}{2}d & \frac{\sqrt{3}}{2}d & 0 & -\frac{\sqrt{3}}{2}d \\ -c & c & -c & c & -c & c \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \end{bmatrix}, \quad (7)$$

where d is the distance from each motor to the vehicle center of mass, $c \triangleq \gamma_d/\gamma_t$, and γ_d and γ_t are respectively the drag and thrust coefficient. τ_{arm} is the torque generated by the arm onto the hexacopter w.r.t. its center, which is displaced from the center of mass ([20]). In particular,

$$\tau_{arm} = m_{arm} g \zeta \times R(q) e_3, \quad (8)$$

where, $m_{arm} \triangleq m_{link_1} + m_{link_2}$ is the mass of the arm, $\zeta \in \mathbb{R}^3$ the position of the center of mass of the hexacopter w.r.t. the pivot point, and $\hat{e}_3 \triangleq [0 \quad 0 \quad 1]^T$. ζ can be expressed as:

$$\zeta = \frac{1}{m_t}[m_{link_1}\rho_1 + m_{link_2}\rho_2 + m_{load}\rho_{load}], \quad (9)$$

where m_t is the total mass of the manipulator $(m_{link_1} + m_{link_2})$ and the load (m_{load}). The terms ρ_i and ρ_l are the position vectors of each link and of the load w.r.t. the hexacopter reference frame. Considering Fig. 2 these position vectors are given by:

$$\rho_1 = \begin{bmatrix} 0 & l_{c1}\cos(\theta_1) & l_{c1}\sin(\theta_1) \end{bmatrix}^T \quad (10a)$$
$$\rho_2 = \begin{bmatrix} 0 & (l_1 + l_{c2})\cos(\theta_1+\theta_2) & (l_1 + l_{c2})\sin(\theta_1+\theta_2) \end{bmatrix}^T \quad (10b)$$
$$\rho_l = \begin{bmatrix} 0 & (l_1 + l_2)\cos(\theta_1+\theta_2) & (l_1 + l_2)\sin(\theta_1+\theta_2) \end{bmatrix}^T, \quad (10c)$$

where l_{ci} is the distance from the joints and the center of mass of each link, and l_i is the length of each link. The moment of inertia of the whole system with respect to the hexacopter reference frame, as showed in [17] is:

$$J_{tot}(\theta) \triangleq J_{hexa} + J_{link_1}(\theta) + J_{link_2}(\theta), \quad (11)$$

where J_{hexa} is the inertia w.r.t. the hexacopter central axis and

$$J_{link_i} \triangleq R_x^T \begin{bmatrix} J_{xx} & 0 & 0 \\ 0 & J_{yy} & 0 \\ 0 & 0 & J_{zz} \end{bmatrix} R_x + m_{link_i}\tilde{\rho}_i^{\,2}, \quad (12)$$

is the inertia tensor of i-th link. In the formula, R_x transforms the moment of inertia with respect to the coordinate system of the hexacopter. The second term is obtained by straightforward application of Huygens-Steiner theorem and $\tilde{\rho}_i$ is given by Eq. (10), written in a skew matrix form.

IV. CONTROL SYSTEM DESIGN

In this section we present the control system architecture (see Fig. 3) for stabilizing a hexacopter with a moving arm. The dynamical model presented in the previous section showed the complexity of the coupled system. In particular, Eq. (5) indicates that the hexacopter is an under-actuated system and the translational dynamics are strongly coupled with the attitude dynamics. Furthermore, the movement of the arm introduces torques (Eq. (8)) and moment of inertia variations (Eq. (11)). For these reasons, the control system of the hexacopter can be divided into an outer-loop, for the position, and an inner-loop, for the attitude. The controller should be able to handle the significant disturbances caused by the additional torque of the arm, e.g., holding the position during pick and release phases. Furthermore, the arm controller should take into account the motion of the hexacopter, in order to get the end-effector to reach the desired position.

Our work mainly focuses on attitude control. In fact, such a controller ultimately needs to compensate and correct the disturbances produced by the movements and the interactions of the arm with the environment. Regarding the position controller we use a standard PID, while the attitude controller employs a command filtered and quaternion-based backstepping technique. We use the "computed torque control" for the arm joint control. We first present the design of the position control, that generates the desired attitude signals for the backstepping inner-loop. Subsequently, we design the backstepping architecture, showing the stability through Lyapunov theory with respect to mass distribution and inertia changes.

A. Position control

Considering the desired position p_d and compensating the gravity force we can write the PID controller as

$$\dot{v}_d = K_p(p_d - p) - K_d v + K_i \int (p_d - p)dt + g. \quad (13)$$

Here, $K_p \succ 0$, $K_d \succ 0$, $K_i \succ 0$ are proportional, derivative and integral gain matrix respectively. To calculate the desired attitude for the inner-loop controller, we can combine the previous expression with Eq. (5), obtaining

$$\dot{v}_d = R(q_d)\begin{bmatrix} 0 \\ 0 \\ \frac{f_{tot}}{m_{tot}} \end{bmatrix}. \tag{14}$$

The direction of the thrust is

$$\hat{f} = \frac{\dot{v}_d}{\| \dot{v}_d \|}, \tag{15}$$

where

$$\|\dot{v}_d\| = \|R(q_d)\| \left\| \begin{bmatrix} 0 \\ 0 \\ \frac{f_{tot}}{m_{tot}}. \end{bmatrix} \right\| = \begin{bmatrix} 0 \\ 0 \\ \frac{\| f_{tot} \|}{m_{tot}} \end{bmatrix}, \tag{16}$$

We can express the desired attitude as

$$\dot{q}_d = \begin{bmatrix} \cos\frac{\beta_d}{2} \\ \hat{k}_d \sin\frac{\beta_d}{2} \end{bmatrix}, \tag{17}$$

where,

$$\begin{aligned} \hat{k}_d &= \left[\hat{e}_3^{\times}\right] \hat{f}, \\ \beta_d &= atan2(\sqrt{1-\hat{f}^2}, \hat{f}). \end{aligned} \tag{18}$$

B. Command filtered backstepping attitude control

The design of the recursive backstepping algorithm and the notation used are similar to the procedure presented in Zhao *et al.* [2] with the difference of having a time dependent inertia matrix.

Step 1: We define a backstepping virtual control variable ω_c to ensure that the attitude of the vehicle tracks the desired attitude $q_d(t)$. We also set $q(t)$ to be the current attitude of the hexacopter and $\tilde{q}(t)$ its tracking error, expressed as quaternion product:

$$\tilde{q}(t) \triangleq q(t) \otimes q_d(t)^{-1}. \tag{19}$$

To simplify the notation, hereafter the variables are not explicitly shown as a function of time t.

The robot's attitude q is aligned to the desired attitude q_d whenever

$$\tilde{q} = [\tilde{\eta} \quad \tilde{\varepsilon}]^T = [\pm 1 \quad 0 \quad 0 \quad 0]^T. \tag{20}$$

The attitude dynamic error $\dot{\tilde{q}}$ is represented as:

$$\dot{\tilde{q}} = \begin{bmatrix} \dot{\tilde{\eta}} \\ \dot{\tilde{\varepsilon}} \end{bmatrix} = \frac{1}{2}\begin{bmatrix} -\tilde{\varepsilon}^T \\ \tilde{\eta} I + [\tilde{\varepsilon}^{\times}] \end{bmatrix} \tilde{\omega}, \tag{21}$$

where we define $\tilde{\omega} \triangleq \omega - \omega_c$. Considering the backstepping variable $\tilde{\varepsilon}$, its derivative is:

$$\begin{aligned} \dot{\tilde{\varepsilon}} &= \frac{1}{2} T(\tilde{q})(\omega - R(\tilde{q})\omega_d) = \\ &= \frac{1}{2} T(\tilde{q})(\omega_c + \tilde{\omega} - R(\tilde{q})\omega_d) \end{aligned} \tag{22}$$

where $T(\tilde{q}) \triangleq \tilde{\eta} I + [\tilde{\varepsilon}^{\times}]$. Computation of ω_d, the corresponding angular velocity of q_d, is defined in the next subsection. Considering the first Lyapunov function candidate

$$V_1 = \tilde{\varepsilon}^T \tilde{\varepsilon} + (1-\tilde{\eta})^2. \tag{23}$$

The time derivative of V_1 is

$$\dot{V}_1 = 2\tilde{\varepsilon}^T \dot{\tilde{\varepsilon}} - 2(1-\tilde{\eta})\dot{\tilde{\eta}} \tag{24}$$

and noticing that $\tilde{\varepsilon}^T [\tilde{\varepsilon}^{\times}] = 0$, we obtain

$$\dot{V}_1 = \tilde{\varepsilon}^T (\omega_c + \tilde{\omega} - R(\tilde{q})\omega_d). \tag{25}$$

Choosing

$$\omega_c = -K_1 \tilde{\varepsilon} + R(\tilde{q})\omega_d, \tag{26}$$

we obtain

$$\dot{V}_1 = -\tilde{\varepsilon}^T K_1 \tilde{\varepsilon} + \tilde{\varepsilon}^T \tilde{\omega}, \tag{27}$$

which is negative definite if $\tilde{\omega} = 0$. Here, $K_1 = K_1^T \in \mathbb{R}^{3\times 3} \succ 0$ is a feedback gain matrix.

Step 2: The goal is to choose a control law $\tau(t)$ such that the actual angular rate $\omega(t)$ tracks the desired angular velocity $\omega_c(t)$. To calculate the dynamics of the tracking error related to the angular velocity $\tilde{\omega}$, we use the second equation in (6). Due to brevity of space we represent $J_{tot}(\theta)$ as J_{tot}. Observing that the following equalities hold:

$$\begin{aligned} & J_{tot}(\dot{\tilde{\omega}} + \dot{\omega}_c) + [[\mathbf{D}_\theta J_{tot}]\dot{\theta}](\tilde{\omega} + \omega_c) = \\ & -(\tilde{\omega} + \omega_c) \times J_{tot}(\tilde{\omega} + \omega_c) + \tau_{tot}, \end{aligned} \tag{28}$$

$$\begin{aligned} J_{tot}\dot{\tilde{\omega}} + [[\mathbf{D}_\theta J_{tot}]\dot{\theta}]\tilde{\omega} &= -(\tilde{\omega} + \omega_c) \times J_{tot}(\tilde{\omega} + \omega_c) \\ & - J_{tot}\dot{\omega}_c + \tau_{tot} - [[\mathbf{D}_\theta J_{tot}]\dot{\theta}]\omega_c, \end{aligned} \tag{29}$$

we obtain:

$$\begin{aligned} J_{tot}\dot{\tilde{\omega}} + [[\mathbf{D}_\theta J_{tot}]\dot{\theta}]\tilde{\omega} &= \Sigma(\tilde{\omega}, \omega_c)\tilde{\omega} - [\omega_c^{\times}] J_{tot}\tilde{\omega} \\ & + \Upsilon(\omega_c, \dot{\omega}_c) + \tau + \tau_{arm} \end{aligned} \tag{30}$$

where

$$\begin{aligned} \Upsilon(\omega_c, \dot{\omega}_c) &= [(J_{tot}\omega_c)^{\times}]\omega_c - J_{tot}\dot{\omega}_c - [[\mathbf{D}_\theta J_{tot}]\dot{\theta}]\omega_c \\ \Sigma(\tilde{\omega}, \omega_c) &= [(J_{tot}\tilde{\omega})^{\times}] + [(J_{tot}\omega_c)^{\times}]. \end{aligned} \tag{31}$$

Considering the second Lyapunov function candidate

$$V_2 = V_1 + \frac{1}{2}\tilde{\omega}^T J_{tot}(\theta)\tilde{\omega} \tag{32}$$

its time derivative is

$$\dot{V}_2 = -\tilde{\varepsilon}^T K_1 \tilde{\varepsilon} + \tilde{\varepsilon}^T \tilde{\omega} + \tilde{\omega}^T J_{tot}\dot{\tilde{\omega}} + \frac{1}{2}\tilde{\omega}^T [[\mathbf{D}_\theta J_{tot}]\dot{\theta}]\tilde{\omega}. \tag{33}$$

adding and subtracting $\frac{1}{2}\tilde{\omega}^T [[\mathbf{D}_\theta J_{tot}(\theta)]\dot{\theta}]\tilde{\omega}$ we obtain:

$$\begin{aligned} \dot{V}_2 = & -\tilde{\varepsilon}^T K_1 \tilde{\varepsilon} + \tilde{\varepsilon}^T \tilde{\omega} + \tilde{\omega}^T (J_{tot}\dot{\tilde{\omega}} + [[\mathbf{D}_\theta J_{tot}]\dot{\theta}] \\ & - \frac{1}{2}[[\mathbf{D}_\theta J_{tot}]\dot{\theta}]\tilde{\omega}). \end{aligned} \tag{34}$$

Finally, using Eq. (30),

$$\begin{aligned} \dot{V}_2 = & -\tilde{\varepsilon}^T K_1 \tilde{\varepsilon} + \tilde{\varepsilon}^T \tilde{\omega} + \tilde{\omega}^T (\Sigma\tilde{\omega} - [\omega_c^{\times}] J_{tot}\tilde{\omega} + \Upsilon + \\ & + \tau + \tau_{arm} - \frac{1}{2}[[\mathbf{D}_\theta J_{tot}]\dot{\theta}]\tilde{\omega}). \end{aligned} \tag{35}$$

choosing the control law

$$\tau = -K_2\tilde{\omega} - \tilde{\varepsilon} - \Upsilon - \tau_{arm} + [\omega_c^{\times}] J_{tot}\tilde{\omega} + \frac{1}{2}[[\mathbf{D}_\theta J_{tot}]\dot{\theta}]\tilde{\omega} \tag{36}$$

and noticing that $\tilde{\omega}^T \Sigma \tilde{\omega} = 0$, we obtain

$$\dot{V}_2 = -\tilde{\varepsilon}^T K_1 \tilde{\varepsilon} - \tilde{\omega}^T K_2 \tilde{\omega} \tag{37}$$

which is a negative definite function of $\tilde{\varepsilon}$ and $\tilde{\omega}$. Here, $K_2 = K_2^T \in \mathbb{R}^{3\times 3} \succ 0$ is a feedback gain matrix. This means that, under the control law proposed in Eq. (36), the aerial manipulator tracks the desired attitude. It is worth noting that the proposed control includes feed forward terms in order to compensate for the torque of the arm τ_{arm} and it also takes into account $J_{tot}(\theta)$.

C. Command filtered backstepping

In the previous subsection we designed a backstepping control law by using virtual control signals and their derivatives. The analytical computation of the command derivative $\dot{\omega}_c$ increases the complexity of the algorithm, making on-board implementation difficult. The introduction of a command filter in the backstepping design obviates the need of an analytical derivation, making computation more feasible. For an in-depth analysis of this mathematical tool we refer to Farrel *et al.* [21]. Furthermore, in order to maintain the unit norm property for quaternion we introduce a second order quaternion filter that computes the derivative of the command quaternion $\dot{q}_d$ and its angular acceleration $\dot{\omega}_d$. Considering ω_n the natural frequency and $\xi \in (0,1]$ the damping ratio, the command filter is designed as:

$$\begin{bmatrix}\omega_c\\ \dot{\omega}_c\end{bmatrix} = \begin{bmatrix}x_1\\ x_2\end{bmatrix} \Longrightarrow \begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = -\omega_n(x_1 - u) - 2\xi\omega_n x_2 \end{cases} \tag{38}$$

where $u = \omega_c^o = -K_1\tilde{\varepsilon} + R(\tilde{q})\omega_d$. In particular, considering the desired output signal ω_c^o as the input of the filter, the latter ensures that the error $\omega_c - \omega_c^o$ goes to 0. Regarding the quaternion command filter, it is designed as

$$\begin{aligned} \dot{q}_d &= \frac{1}{2}Q(q_d)\omega_d \\ \dot{\omega}_d &= -\omega_n^2\tilde{\varepsilon} - 2\xi\omega_n\omega_d. \end{aligned} \tag{39}$$

It ensures that:

$$\tilde{q}_d = q_d \otimes (q_d^o)^{-1} = [\pm 1 \quad 0 \quad 0 \quad 0]^T. \tag{40}$$

V. EXPERIMENTS

For testing the efficacy of our controller, we performed a set of simulations in a MATLAB environment. For the recursive Newton-Euler dynamics model of the manipulators we use the Robotics Toolbox [22]. The simulator implements the whole quaternion-based non-linear model of the system and it also includes an animation. Furthermore, to make it more realistic we add the noise due to the sensors, aerodynamic friction of the air and we apply bounds on the control action. The simulation parameters are as follow: $m_h = 3$ Kg, $m_{l_i} = 0.35$ Kg, $d = 0.60$ m, $l_i = 0.30$ m. The initial conditions are zero except for the configuration of the arm $\theta_1 = -80°$, $\theta_2 = 164°$ and for the vertical position of the robot $p_z = 0.25$ m.

Figures 4 to 7 show the simulations results. In the first simulation, the hexacopter hovers while the arm moves from a totally extended horizontal position to a fully extended vertical position. The goal is to verify the variations of the moment of inertia, shown in Fig. 4. The motion of the arm causes opposite effects on J_{xx}, J_{yy}, which increase, and J_{zz}, which decreases.

In the second simulation, the goal is to grab with the end-effector an object from a known position. Once the hexacopter

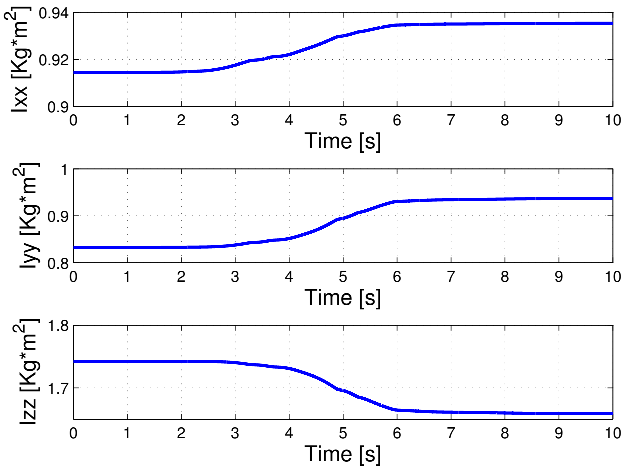
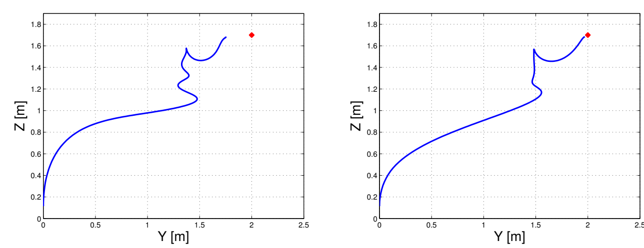


Fig. 4: Variations of the moment of inertia.

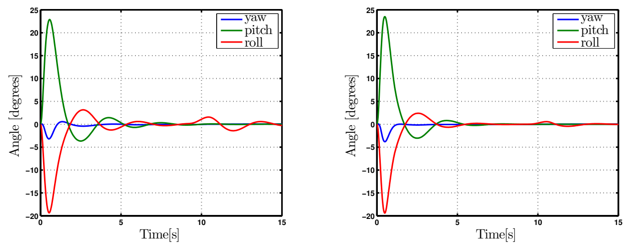


(a) (b)

Fig. 5: End-effector position with PD (a) and backstepping attitude control (b).

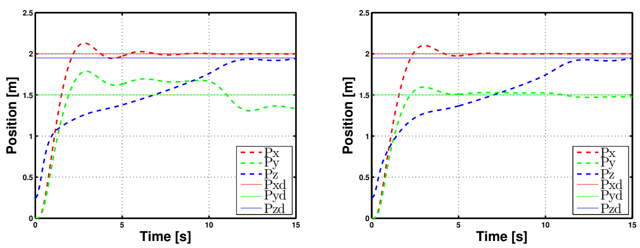
reaches the designed position, the arm moves towards the object and grabs it. We show a comparison between two different attitude controllers: a quaternion based non-linear PD, that does not account for the disturbances caused by the arm, and the command filtered backstepping. Fig. 5 shows the trajectory of the end-effector in the vertical plane (Y-Z) and the position of the object. The final position of the end-effector at the end of the arm trajectory using the PD controller (Fig. 5a) has a distance of about 18 cm respect to the designed position, against the about 1 cm of the proposed controller (Fig. 5b). Therefore, the use of the backstepping technique performs the task better, whereas the PD control is not even able to grab the object.

Finally, Fig. 6 shows the roll, pitch and yaw attitude angles, while Fig. 7 shows the linear position of the hexacopter (dashed line) respect to the reference values. Note that although we consider a quaternion parametrization, we use Euler angles to represent the attitude in the graphs, in order to make them more intuitive. It can be seen that when the arm starts to move, around 9 sec, its movement causes an oscillation to the hexacopter around the x axes, that the PD is not able to compensate. This results in a wrong position of the center of mass of the hexacopter along the y axis, causing a discrepancy between the position of the object and the final position of the end-effector. It is clear that even small oscillations can compromise the success of the task. Our approach reduces



(a) (b)

Fig. 6: Angular position with PD (a) and backstepping attitude control (b).



(a) (b)

Fig. 7: Linear position with PD (a) and backstepping attitude control (b).

the oscillation, thereby allowing the end-effector to reach the desired position.

VI. CONCLUSIONS

In this paper we presented a mathematical model that characterizes the coupled interaction between a Hexacopter and its 2-DoF robotic arm using a non-linear and robust control algorithm. We employed a quaternion-based representation to deal with singularities and, unlike previous works, we explicitly considered the derivative term of the moment of inertia. In addition, we utilized a command filtered implementation of the backstepping approach to reduce the complexity of the algorithm. We showed that with our approach the controller is able to compensate for the changes that occur in the system during the movements of the arm. In simulation experiments we demonstrated the robustness of our control strategy and an increased performance in comparison with a standard non-linear PD controller. In future work we plan to evaluate our algorithm also with a real robotic platform and to generalize our approach so that it is able to estimate the mass the robot is carrying. In addition, we want to investigate the designed control scheme in the context of tasks that imply singularities in the pose of the robot.

ACKNOWLEDGMENT

This work has been partly supported by Graduate School of Robotics - University of Freiburg, Germany.

REFERENCES

[1] "ARCAS project," http://www.arcas-project.eu/, [Accessed 15-04-15].

[2] S. Zhao, W. Dong, and J. A. Farrell, "Quaternion-based trajectory tracking control of vtol-uavs using command filtered backstepping," in *American Control Conference, 2013*. IEEE, 2013, pp. 1018–1023.

[3] M. Fumagalli, R. Naldi, A. Macchelli, R. Carloni, S. Stramigioli, and L. Marconi, "Modeling and control of a flying robot for contact inspection," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*. IEEE, 2012, pp. 3532–3537.

[4] J. L. Scholten, M. Fumagalli, S. Stramigioli, and R. Carloni, "Interaction control of an uav endowed with a manipulator," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE, 2013, pp. 4910–4915.

[5] M. Bernard and K. Kondak, "Generic slung load transportation system using small size helicopters," in *Robotics and Automation, 2009. ICRA'09. International Conference on*. IEEE, 2009, pp. 3258–3264.

[6] M. Bisgaard, A. la Cour-Harbo, and J. D. Bendtsen, "Adaptive control system for autonomous helicopter slung load operations," *Control Engineering Practice*, vol. 18, no. 7, pp. 800–811, 2010.

[7] P. E. Pounds, D. R. Bersak, and A. M. Dollar, "Grasping from the air: Hovering capture and load stability," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE, 2011, pp. 2491–2498.

[8] D. Mellinger, Q. Lindsey, M. Shomin, and V. Kumar, "Design, modeling, estimation and control for aerial grasping and manipulation," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*. IEEE, 2011, pp. 2668–2673.

[9] Q. Lindsey, D. Mellinger, and V. Kumar, "Construction with quadrotor teams," *Autonomous Robots*, vol. 33, no. 3, pp. 323–336, 2012.

[10] N. Michael, J. Fink, and V. Kumar, "Cooperative manipulation and transportation with aerial robots," *Autonomous Robots*, vol. 30, no. 1, pp. 73–86, 2011.

[11] K. Kondak, K. Krieger, A. Albu-Schaeffer, M. Schwarzbach, M. Laiacker, I. Maza, A. Rodriguez-Castano, and A. Ollero, "Closed-loop behavior of an autonomous helicopter equipped with a robotic arm for aerial manipulation tasks," *International Journal of Advanced Robotic Systems*, vol. 10, no. 145, 2013.

[12] V. Lippiello and F. Ruggiero, "Cartesian impedance control of a uav with a robotic arm," in *10th International IFAC Symposium on Robot Control*, 2012, pp. 704–709.

[13] S. Kim, S. Choi, and H. J. Kim, "Aerial manipulation using a quadrotor with a two dof robotic arm," in *Intelligent Robots and Systems (IROS), International Conference on*. IEEE, 2013, pp. 4990–4995.

[14] A. Jimenez-Cano, J. Martin, G. Heredia, A. Ollero, and R. Cano, "Control of an aerial robot with multi-link arm for assembly tasks," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE, 2013, pp. 4916–4921.

[15] F. Caccavale, G. Giglio, G. Muscio, and F. Pierri, "Adaptive control for uavs equipped with a robotic arm," *Jeb*, vol. 3, p. I3, 2014.

[16] G. Heredia, A. Jimenez-Cano, I. Sanchez, D. Llorente, V. Vega, J. Braga, J. Acosta, and A. Ollero, "Control of a multirotor outdoor aerial manipulator," in *Intelligent Robots and Systems (IROS 2014), International Conference on*. IEEE, 2014, pp. 3417–3422.

[17] M. Orsag, C. Korpela, M. Pekala, and P. Oh, "Stability control in aerial manipulation," in *American Control Conference (ACC), 2013*. IEEE, 2013, pp. 5581–5586.

[18] C. Korpela, M. Orsag, M. Pekala, and P. Oh, "Dynamic stability of a mobile manipulating unmanned aerial vehicle," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE, 2013, pp. 4922–4927.

[19] C. Korpela, M. Orsag, and P. Oh, "Towards valve turning using a dual-arm aerial manipulator," in *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*. IEEE, 2014, pp. 3411–3416.

[20] J. Álvarez-Muñoz, N. Marchand, F. Guerrero-Castellanos, S. Durand, and A. Lopez-Luna, "Improving control of quadrotors carrying a manipulator arm," in *XVI Congreso Latinoamericano de Control Automático (CLCA 2014)*, 2014, pp. 6–p.

[21] J. A. Farrell, M. Polycarpou, M. Sharma, and W. Dong, "Command filtered backstepping," *Automatic Control, IEEE Transactions on*, vol. 54, no. 6, pp. 1391–1395, 2009.

[22] P. Corke, *Robotics, vision and control: fundamental algorithms in MATLAB*. Springer Science & Business Media, 2011, vol. 73.