24th Mediterranean Conference on Control and Automation (MED)
June 21-24, 2016, Athens, Greece

Modeling and Control Aspects of a UAV with an Attached Manipulator

Yiannis Stergiopoulos, Efstathios Kontouras, Konstantinos Gkountas,
Konstantinos Giannousakis and Anthony Tzes†

Abstract— **This article addresses the problem of the modeling and control of an Unmanned Aerial System (UAS) that consists of a UAV carrying a robotic manipulator. The dynamics of the UAS are derived based on the recursive Newton-Euler equations in order to handle the floating-base effect. Accordingly, the dynamics account for the force and torque transferred from the base of the manipulator to the UAV and the translational and angular velocities and accelerations from the UAV to the manipulator. The integrated nonlinear dynamic equations are symbolically derived and efficient code for execution at embedded systems is generated. An efficient nonlinear control scheme regulates the states of the UAS to a desired configuration. Simulation results are provided to evaluate the performance of the developed strategy, in comparison to the ones derived by simplifying the coupling of the UAV and manipulator dynamics by considering only the spatial coordinates of the center of mass of the manipulator.**

I. Introduction

Aerial manipulation is a research field that has attracted a significant amount of interest in the community of robotics. The main reason is that Unmanned Aerial Vehicles (UAVs) armed with a robotic manipulator demonstrate increased functionality and flexibility over their unarmed counterparts. Interesting applications include grasping, mobile manipulation, inspection of hard-to-reach structures, transportation in remote areas and collaborative working, where more than one UAVs cooperate to achieve a common goal. These tasks have their own challenges and are currently the subject of various research activities.

The modeling of a coupled UAV and robotic manipulator system is in general a difficult problem. The dynamics of both the aerial vehicle and the robotic manipulator are highly nonlinear and highly coupled with each other, thus resulting to complex mathematical models. From a control perspective, the obstacles arising are equally challenging, since the motion of the UAV affects the manipulator and vice-versa. However, several authors have proposed different approaches concerning the modeling procedure, while also developing efficient control policies for both the UAV and the robotic manipulator [1–3].

In order to derive a precise and realistic model for a UAV, it is important to consider not only the classical dynamic equations governing the translational and rotational motions of the vehicle but also the equations that describe its aerodynamic behavior; thus, further increasing the complexity of the corresponding mathematical representation [4, 5]. On the other hand, there exists extensive academic literature on the modeling of fixed-base robotic manipulators, where the gravity vector has a prespecified direction with respect to the base frame [6]. Other authors have focused on the study of floating-base manipulators, but mostly under the assumption that the environment around the robot has no gravity, as it happens, for example, in space applications [7]. In this case, the equations are significantly simplified but the subsequent modeling and control methods developed are applicable to a limited number of areas.

In this paper, we consider the trivial equations for the UAV, neglecting the rotor and blade dynamics as well as any aerodynamic effects; nevertheless, we still obtain a reliable enough model for the aerial vehicle. The floating-base effect is taken into account in terms of the recursive Newton-Euler equations that transmit the velocities and accelerations of the UAV to the manipulator and in return they compute the forces and torques applied on the UAV by the manipulator, thus allowing us to interconnect the two subsystems and derive the fully coupled dynamic model. This integrated framework is able to provide an environment for realistic simulations as well as the base for advanced model-based state-of-the-art control methods concerning the coupled system.

In Section II the mathematical models of the two individual subsystems are established, while in Section III the coupling procedure is presented with the efficiency and reliability of our methods being verified via the Gazebo simulator. In Section IV control schemes for both the UAV and the robotic manipulator are developed. Section V presents indicative simulation results validating our conceptual approach and comparisons between different controllers are held out. Finally, in Section VI we provide some concluding remarks.

II. UAS Decoupled Modeling

Based on the existing academic literature we establish the UAV and the robotic manipulator dynamics as individual systems. The UAV dynamics are derived in terms of the Newton-Euler formulation, while the manipulator dynamics are extracted through the recursive Newton-Euler equations.

Regarding notations, an orthogonal matrix ${}^{B}R_{A} \in \mathbb{R}^{3\times3}$ transforms a vector from A-frame to B-frame and satisfies the relations $\left({}^{B}R_{A}\right)^{-1} = \left({}^{B}R_{A}\right)^{\top} = {}^{A}R_{B}$, while symbols like $s\phi$ and $c\phi$ stand for $\sin\phi$ and $\cos\phi$ respectively.

A. UAV-Quadrotor Dynamics

Let us consider a quadrotor with a rigid and symmetrical structure. We introduce two distinct coordinate frames,

†The authors are with the Electrical & Computer Engineering Department, University of Patras, Rio 26500, Greece.
Corresponding author's email: tzes@ece.upatras.gr

namely the Body-Fixed Frame (B-frame) and the Earth-Fixed Frame (E-frame). For brevity, we assume that the center of mass, the aerodynamic point of pressure and the origin of the B-frame coincide. The propellers are assumed to be rigid and the thrust and drag forces are proportional to the square of the angular speed of the rotors.

In order to derive a precise representation of the nonlinear model of the UAV, the Newton-Euler formulation has been adopted. The Newton-Euler equations describe the combined translational and rotational dynamics of a rigid body. According to this method, the dynamics of a rigid body under external forces and torques, expressed in the B-frame, may be written as

$$\begin{bmatrix} m_s\mathbb{I}_{3\times3} & \mathbb{O}_{3\times3} \\ \mathbb{O}_{3\times3} & I \end{bmatrix} \begin{bmatrix} \dot{V} \\ \dot{\omega} \end{bmatrix} + \begin{bmatrix} \omega \times V \\ \omega \times I\omega \end{bmatrix} = \begin{bmatrix} F \\ \tau \end{bmatrix}, \quad (1)$$

where F, τ, $\mathbb{I}_{3\times3}$, $\mathbb{O}_{3\times3}$, I, V, ω, m_s correspond to the total force vector acting on the center of mass, the total torque acting about the center of mass, the identity matrix, the zero matrix, the moment of inertia about the center of mass, the translational velocity, the angular velocity and the mass of the body respectively.

The square rotation matrix that transforms a vector from B-frame to E-frame is defined as

$${}^{E}R_B = \begin{bmatrix} c\theta c\psi & s\phi s\theta c\psi - c\phi s\psi & c\phi s\theta c\psi + s\phi s\psi \\ c\theta s\psi & s\phi s\theta s\psi + c\phi c\psi & c\phi s\theta s\psi - s\phi c\psi \\ -s\theta & s\phi c\theta & c\phi c\theta \end{bmatrix}.$$

According to [4, 5], the simplified nonlinear UAV model, where the hub forces, the friction forces, the rolling moments and the external disturbances are neglected, may be described by the following set of differential equations

$$\begin{bmatrix} \dot{V}_E \\ \dot{\omega}_B \end{bmatrix} = \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \\ \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} u_x U_1/m_s \\ u_y U_1/m_s \\ -g + (c\phi c\theta)U_1/m_s \\ \dot{\theta}\dot{\psi}a_1 + b_1 U_2 \\ \dot{\phi}\dot{\psi}a_2 + b_2 U_3 \\ \dot{\theta}\dot{\phi}a_3 + b_3 U_4 \end{bmatrix}, \quad (2)$$

where $\dot{V}_E = [\ddot{x} \quad \ddot{y} \quad \ddot{z}]^\top$ is the translational acceleration expressed in the E-frame and $\dot{\omega}_B = [\ddot{\phi} \quad \ddot{\theta} \quad \ddot{\psi}]^\top$ is the angular acceleration expressed in the B-frame. The fictitious control inputs $U_1, U_2, U_3, U_4 \in \mathbb{R}$ are defined as

$$\begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} = \begin{bmatrix} b(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \\ b(-\Omega_2^2 + \Omega_4^2) \\ b(\Omega_1^2 - \Omega_3^2) \\ d(-\Omega_1^2 + \Omega_2^2 - \Omega_3^2 + \Omega_4^2) \end{bmatrix} \quad (3)$$

and the individual parameters associated with each input are determined in terms of the following equations:

$$u_x = c\phi s\theta c\psi + s\phi s\psi, \quad u_y = c\phi s\theta s\psi + s\phi c\psi \quad (4)$$

$$a_1 = (I_{yy} - I_{zz})/I_{xx}, \quad a_2 = (I_{zz} - I_{xx})/I_{yy} \quad (5)$$

$$a_3 = (I_{xx} - I_{yy})/I_{zz} \quad (6)$$

$$b_1 = l_a/I_{xx}, \quad b_2 = l_a/I_{yy}, \quad b_3 = 1/I_{zz}. \quad (7)$$

The input U_1 is related to the total thrust, the inputs U_2, U_3 and U_4 are related to the rotations of the UAV, $\Omega_1, \Omega_2, \Omega_3, \Omega_4 \in \mathbb{R}$ are the rotational speeds of the propellers, m_s is the total mass of the UAV and $g = 9.81\ [m/s^2]$ is the gravitational acceleration. Parameters u_x and u_y represent the directions of thrust vector that cause the motion about the E_x and E_y axis respectively and U_1, in combination with ϕ and θ states, rule the altitude motion. The rest of the parameters involved in equations (3)-(7) are listed in Table (I).

I_{xx}	Moment of inertia of the UAV about the B_x axis
I_{yy}	Moment of inertia of the UAV about the B_y axis.
I_{zz}	Moment of inertia of the UAV about the B_z axis.
l_a	Arm length of the UAV.
b	Thrust coefficient.
d	Drag coefficient.

TABLE I: UAV model parameters.

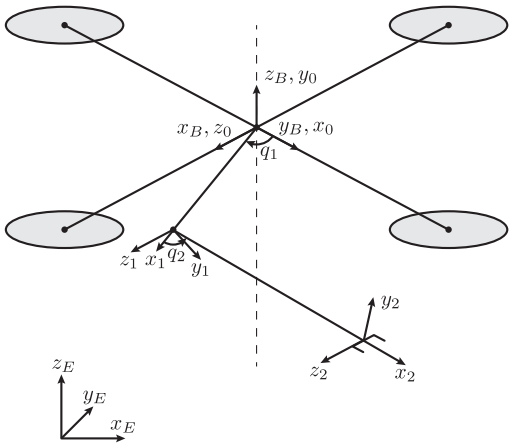

Fig. 1: Quadrotor armed with a two-link robotic manipulator.

B. Robotic Manipulator Dynamics

Generally, the dynamics of a robotic manipulator may be derived either by using the Euler-Lagrange equations or the recursive Newton-Euler equations [6]. The Euler-Lagrange equations work efficiently for the case of a trivial fixed-base robotic manipulator since the gravity vector remains unaltered with respect to the Base-Fixed Frame (O-frame). In our case, the position and the orientation of the O-frame change over time according to the movement of the UAV. As a matter of fact, the gravity vector is also changing when expressed in the O-frame.

A rather common way of addressing the study of robotic manipulators with a floating-base involves the use of the Newton-Euler equations [6]. The main idea of this method is to initialize the velocities and accelerations of the base link at a non-zero value. These velocities and accelerations are in turn transmitted from one link to another by a forward recursion and would result in additional resultant forces and torques computed by a backward recursion. In this work, we use a special kind of recursive Newton-Euler equations

known as the computationally efficient recursive Newton-Euler equations [6].

(1) Forward Recursion: Here the angular and linear velocities and accelerations of each link are calculated recursively in terms of its preceding link starting from the base to the end-effector. The initial conditions for the base link are $v_0 = v$, $\dot{v}_0 = \dot{v}$ and $\omega_0 = \Omega$, $\dot{\omega}_0 = \dot{\Omega}$. With these initializations the "forward" equations are calculated for $i = 1, 2, \ldots, n$ as

$$ {}^{i}R_{0}\omega_{i} = \begin{cases} {}^{i}R_{i-1}\left({}^{i-1}R_{0}\omega_{i-1} + z_0\dot{q}_i\right), & \text{if } i \text{ is rot.} \\ {}^{i}R_{i-1}\left({}^{i-1}R_{0}\omega_{i-1}\right), & \text{if } i \text{ is trans.} \end{cases} \tag{8} $$

$$ {}^{i}R_{0}\dot{\omega}_{i} = \begin{cases} {}^{i}R_{i-1}\left({}^{i-1}R_{0}\dot{\omega}_{i-1} + z_0\ddot{q}_i + \left({}^{i-1}R_{0}\omega_{i-1}\right) \times z_0\dot{q}_i\right), & \text{if } i \text{ is rot.} \\ {}^{i}R_{i-1}\left({}^{i-1}R_{0}\dot{\omega}_{i-1}\right), & \text{if } i \text{ is trans.} \end{cases} \tag{9} $$

$$ {}^{i}R_{0}\dot{v}_{i} = \begin{cases} \left({}^{i}R_{0}\dot{\omega}_{i}\right) \times \left({}^{i}R_{0}p_{i}^{*}\right) + \left({}^{i}R_{0}\omega_{i}\right) \times \left(\left({}^{i}R_{0}\omega_{i}\right) \times \left({}^{i}R_{0}p_{i}^{*}\right)\right) \\ +{}^{i}R_{i-1}\left({}^{i-1}R_{0}\dot{v}_{i-1}\right), & \text{if } i \text{ is rot.} \\ {}^{i}R_{i-1}\left(z_0\ddot{q}_i + {}^{i-1}R_{0}\dot{v}_{i-1}\right) + \left({}^{i}R_{0}\dot{\omega}_{i}\right) \times \left({}^{i}R_{0}p_{i}^{*}\right) \\ +2\left({}^{i}R_{0}\omega_{i}\right) \times \left({}^{i}R_{i-1}z_0\dot{q}_i\right) \\ +\left({}^{i}R_{0}\omega_{i}\right) \times \left(\left({}^{i}R_{0}\omega_{i}\right) \times \left({}^{i}R_{0}p_{i}^{*}\right)\right), & \text{if } i \text{ is trans.} \end{cases} \tag{10} $$

$$ {}^{i}R_{0}\bar{a}_{i} = \left({}^{i}R_{0}\dot{\omega}_{i}\right) \times \left({}^{i}R_{0}\bar{s}_{i}\right) + \left({}^{i}R_{0}\omega_{i}\right) \times \left(\left({}^{i}R_{0}\omega_{i}\right) \times \left({}^{i}R_{0}\bar{s}_{i}\right)\right) + {}^{i}R_{0}\dot{v}_{i}, \tag{11} $$

where $\bar{s}_i$ is the position of the center of mass of link i from the origin of the i-th local coordinate system, p_i^* is the origin of the i-th coordinate frame with respect to the $(i-1)$-th coordinate system, ω_i is the angular velocity of link i, v_i is the linear velocity of the center of mass of link i, $\bar{a}_i$ is the linear acceleration of the center of mass of link i, q_i denotes the joint variable of link i and $z_0 = [0 \quad 0 \quad 1]^\top$.

(2) Backward Recursion: After the velocities and accelerations of the links are computed, the joint forces may be computed for each link starting from the end-effector to the base. The required "backward" equations are calculated for $i = n, n-1, \ldots, 1$ as

$$ {}^{i}R_{0}f_{i} = {}^{i}R_{i+1}\left({}^{i+1}R_{0}f_{i+1}\right) + m_i\,{}^{i}R_{0}\bar{a}_{i} \tag{12} $$

$$ \begin{aligned} {}^{i}R_{0}\eta_{i} = {}& {}^{i}R_{i+1}\left({}^{i+1}R_{0}\eta_{i+1} + \left({}^{i+1}R_{0}p_{i}^{*}\right) \times \left({}^{i+1}R_{0}f_{i+1}\right)\right) \\ & + \left({}^{i}R_{0}p_{i}^{*} + {}^{i}R_{0}\bar{s}_{i}\right) \times \left({}^{i}R_{0}F_{i}\right) \\ & + \left({}^{i}R_{0}I_{i}{}^{0}R_{i}\right)\left({}^{i}R_{0}\dot{\omega}_{i}\right) \\ & + \left({}^{i}R_{0}\omega_{i}\right) \times \left(\left({}^{i}R_{0}I_{i}{}^{0}R_{i}\right)\left({}^{i}R_{0}\omega_{i}\right)\right) \end{aligned} \tag{13} $$

$$ \tau_i = \begin{cases} \left({}^{i}R_{0}\eta_{i}\right)^{\top}\left({}^{i}R_{i-1}z_0\right) + b_i\dot{q}_i, & \text{if } i \text{ is rot.} \\ \left({}^{i}R_{0}f_{i}\right)^{\top}\left({}^{i}R_{i-1}z_0\right) + b_i\dot{q}_i, & \text{if } i \text{ is trans.} \end{cases}, \tag{14} $$

where m_i is the total mass of link i, $F_i = m_i\bar{a}_i$ is the total external force exerted on link i at the center of mass, I_i is the inertia matrix of link i about its center of mass with reference to the O-frame, f_i is the force exerted on link i by link $i-1$ at the $(i-1)$-th local coordinate frame to support link i and the links above it, η_i is the moment exerted on link i by link $i-1$ at the $(i-1)$-th local coordinate frame and b_i is the viscous damping coefficient for joint i. For the remainder of this work we neglect the damping effects and we assume that $b_i = 0$ for all $i = 1, 2, \ldots, n$. The initial conditions for the backward recursion are defined by the load affecting the end-effector of the robotic manipulator. We highlight that these equations are used to compute the joint torque by using velocities, accelerations, forces and moments expressed with respect to the local link coordinate frames.

If both recursions are executed symbolically then equation (14) provides a closed-form expression for the dynamics of an n-link manipulator. In matrix form, we may write

$$ D(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = \tau, \tag{15} $$

where $q \in \mathbb{R}^n$ is the vector of joint variables, $D(q) \in \mathbb{R}^{n\times n}$ is the generalized inertia matrix, $C(q) \in \mathbb{R}^n$ is the Coriolis matrix, $G(q) = [g_1(q), \ldots, g_n(q)]^\top$ is the gravity vector and $\tau = [\tau_1, \ldots, \tau_n]^\top$ is the vector of generalized forces; that is, the i-th element denotes either the force or the torque applied to the i-th joint of the manipulator.

III. UAS Coupled System Dynamics

The main idea behind the proposed scheme is to exploit the recursive Newton-Euler equations and specifically, to consider as initial conditions the velocities and the accelerations of the UAV. The recursions are executed symbolically and the coupled system dynamics are naturally obtained. Our results were tested for their validity via the Gazebo simulator.

A. Modeling of the Coupled System

In order to derive the coupled dynamics, we decided to use the recursive Newton-Euler equations since they provide an elegant and systematic method of integrating the entire system. If we assume that the base of the manipulator is collocated with the center of mass of the UAV then for the base link we may consider the following initial conditions

$$ v_0 = {}^{O}R_{E}V_{E}, \quad \dot{v}_0 = {}^{O}R_{E}\left(\dot{V}_{E} + [0 \quad 0 \quad g]^\top\right) \tag{16} $$

$$ \omega_0 = {}^{O}R_{B}\omega_{B}, \quad \dot{\omega}_0 = {}^{O}R_{B}\dot{\omega}_{B}, \tag{17} $$

where v_0, $\dot{v}_0$, ω_0, $\dot{\omega}_0$ are the translational and angular velocities and accelerations of the UAV with respect to the O-frame and matrices ${}^{O}R_{E}$ and ${}^{O}R_{B}$ are defined as

$$ {}^{O}R_{E} = {}^{O}R_{B}\,{}^{B}R_{E}, \quad {}^{O}R_{B} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}. \tag{18} $$

The recursive Newton-Euler equations allow the computation of the force and torque applied on the center of mass of the UAV due to the interconnection with the manipulator in terms of the equations (12)-(13). Specifically, the quantities f_1 and η_1 may be directly obtained and then expressed with respect to the O-frame as

$$ f_0 = {}^{0}R_{1}f_{1}, \quad \eta_0 = {}^{0}R_{1}\eta_{1}, \tag{19} $$

where the force f_0 and the torque η_0 satisfy the relations

$$ f_0 = f_0(V_E, \dot{V}_E, \omega_B, \dot{\omega}_B, q, \dot{q}, \ddot{q}, \phi, \theta, \psi) \tag{20} $$

$$ \eta_0 = \eta_0(V_E, \dot{V}_E, \omega_B, \dot{\omega}_B, q, \dot{q}, \ddot{q}, \phi, \theta, \psi) \tag{21} $$

and the angles ϕ, θ and ψ affect both f_0 and η_0 due to the fact that the initial conditions implicate matrix ${}^{B}R_{E}$.

Since the UAV model assumes that the external forces are expressed with respect to the E-frame, whereas the torques are expressed with respect to the B-frame, it is necessary to transform f_0 and η_0 in terms of the matrices ${}^{E}R_{O}$ and

${}^{B}R_{O}$ respectively. Consequently, the translational motion is governed by the equation

$$\dot{V}_E = \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} u_x U_1/m_s - f_{E,x} \\ u_y U_1/m_s - f_{E,y} \\ -g + (c\phi c\theta) U_1/m_s - f_{E,z} \end{bmatrix}, \qquad (22)$$

and the rotational motion is governed by the equation

$$\dot{\omega}_B = \begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} \dot{\theta}\dot{\psi} a_1 + b_1 U_2 - \eta_{B,x} \\ \dot{\phi}\dot{\psi} a_3 + b_2 U_3 - \eta_{B,y} \\ \dot{\theta}\dot{\phi} a_5 + b_3 U_4 - \eta_{B,z} \end{bmatrix}, \qquad (23)$$

where the force vector $f_E = [f_{E,x} \quad f_{E,y} \quad f_{E,z}]^\top$ and the torque vector $\eta_B = [\eta_{B,x} \quad \eta_{B,y} \quad \eta_{B,z}]^\top$ satisfy the relations

$$f_E = \left({}^{O}R_E\right)^{-1} f_0, \quad \eta_B = \left({}^{O}R_B\right)^{-1} \eta_0. \qquad (24)$$

We highlight that f and η denote the force and torque applied on the manipulator due to the interconnection with the UAV. Based on the action-reaction law we may claim that the opposite forces and torques are applied on the UAV; hence, the "$-$" signs in equations (22)-(23) are well justified.

Finally, the dynamic evolution of the robotic manipulator in the joint space may be derived in terms of the equation (14), which, inevitably, implicates quantities associated with the UAV. For a closed-form expression it is necessary to execute the "forward" and "backward" recursions, compute all f_i and η_i, explicitly determine (14) and then augment the UAV model (22)-(23) with the resulting differential equations, which may be written as

$$\begin{aligned} \tau_i &= \tau_i(V_E, \dot{V}_E, \omega_B, \dot{\omega}_B, q, \dot{q}, \ddot{q}, \phi, \theta, \psi) \\ &= \begin{cases} \left({}^{i}R_0 \eta_i\right)^\top \left({}^{i}R_{i-1} z_0\right), & \text{if } i \text{ is rot.} \\ \left({}^{i}R_0 f_i\right)^\top \left({}^{i}R_{i-1} z_0\right), & \text{if } i \text{ is tran.} \end{cases}, \end{aligned} \qquad (25)$$

In total we require $6+n$ differential equations to describe the entire system. These equations are highly coupled but the second order terms may be separated.

B. Gazebo Verification

Due to the inherent complexity that characterizes our system and in order to ensure the validity of our results, we compared them with the results obtained from the Gazebo simulator. Gazebo is a simulator that is able reproduce the dynamic environments a robot may interact with. The robots are composed by rigid bodies connected with joints. Every part of the robot has mass, velocity, friction and numerous other attributes that allow them to behave realistically [8].

To this end, Gazebo uses one of its four supported physics engines, namely ODE, Bullet, Simbody and DART. The ODE engine, which was adopted for the purposes of this work, is an open source, high performance library for simulating rigid body dynamics. It is fully featured, stable, mature and platform independent with an easy-to-use C/C++ API. It has advanced joint types and integrated collision detection with friction [9]. ODE is useful for simulating vehicles and it is currently used in many simulation tools.

To assess our model, using the same inputs we should get the same maneuver in both simulations. The simulations in Gazebo were based on "RotorS", which is a UAV simulator framework [10]. We selected a quadrotor from its collection, attached a serial two link manipulator to it and the results were almost identical to those produced by the coupled system dynamics which we had already derived.

IV. Control Synthesis

Based on the coupled dynamic model of the UAV with the manipulator attached on its base, a control policy is followed that takes into account the effect from each subsystem to the other; that is, how the UAV accelerations and velocities affect the motion of the arm, and reversibly how the forces and moments that act on the UAV base from the manipulator affect the behavior of the UAV.

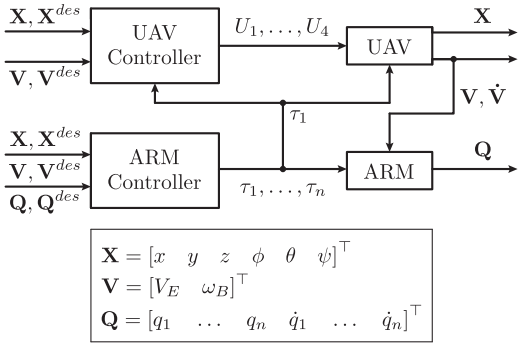

Fig. 2: Block diagram of the closed-loop coupled system.

A. Manipulator Control

The majority of the control policies concerning a manipulator maneuver rely on the Euler-Lagrange dynamic model. Given full knowledge of the dynamic parameters of the robotic arm, feedback linearization techniques may be applied in order to nullify the nonlinear terms and rebuild the model in a linear form, appropriate for joint trajectory control. This controller, well known as computed torque, may be modified in terms of our framework by considering the manipulator equations as they are obtained via the fully coupled system dynamics. Specifically, we may write that

$$\begin{aligned} \tau = {} & C^{gen}(q, \dot{q})\dot{q} + G^{gen}(q) \\ & - D^{gen}(q)\left(-\ddot{q}^{des} + K_d\left(\dot{q} - \dot{q}^{des}\right) + K_p\left(q - q^{des}\right)\right), \end{aligned} \qquad (26)$$

where $(\cdot)^{des}$ are the desired trajectories for the angles and their derivatives and K_p, K_d are diagonal matrices of positive gains. Parameters $C^{gen}(q, \dot{q})$ and $G^{gen}(q)$ denote the generic expressions for the Coriolis and gravity terms associated with the entire manipulator and are provided by the last n differential equations of the complete symbolic model (22)-(25). Finally, parameter $D^{gen}(q)$ stands for the entire generalized inertia matrix.

The main advantage of this controller is the ability to achieve perfect trajectory tracking in the joint control space. On the other hand, the main drawback is the computational complexity required not only for the off-line computation

but also for the real-time evaluation. It is also necessary to remark that the computational complexity increases significantly for manipulators with a higher number of degrees of freedom.

Alternatively, one may ignore the Coriolis and centrifugal terms, $C_i^{gen}(q, \dot{q})$, and cancel only those gravity terms that associate with a hover UAV pose, in order to partially amend for the steady-state configuration error, followed by PD control on the joint angles errors and their derivatives, resulting in a control law of the form

$$\tau = G(q) - K_d \left(\dot{q} - \dot{q}^{des}\right) - K_p \left(q - q^{des}\right). \quad (27)$$

However, when dealing with floating-base manipulators, the gravity terms in the standard Euler-Lagrange models assume fixed base, and thus the gravitational acceleration vector has fixed orientation with respect to the O-frame.

B. UAV Control

One of the most common ways of controlling a UAV is based on the use of PD controllers that regulate both the attitude and the altitude dynamics. In our framework, we consider four PD controllers, namely one for the altitude dynamics concerning parameter z and three for the attitude dynamics that regulate the roll (ϕ), pitch (θ) and yaw (ψ) angles.

Specifically, the four individual controllers governing the motion dynamics are of the form

$$U_1 = \frac{m_t}{c\phi c\theta}\left(g - k_{d,z}\left(\dot{z} - \dot{z}^{des}\right) - k_{p,z}\left(z - z^{des}\right)\right) \quad (28)$$

$$U_2 = \frac{I_{xx}}{l_a}\left(-k_{d,\phi}\left(\dot{\phi} - \dot{\phi}^{des}\right) - k_{p,\phi}\left(\phi - \phi^{des}\right)\right) \quad (29)$$

$$U_3 = \frac{I_{yy}}{l_a}\left(-k_{d,\theta}\left(\dot{\theta} - \dot{\theta}^{des}\right) - k_{p,\theta}\left(\theta - \theta^{des}\right)\right) \quad (30)$$

$$U_4 = I_{zz}\left(-k_{d,\psi}\left(\dot{\psi} - \dot{\psi}^{des}\right) - k_{p,\psi}\left(\psi - \psi^{des}\right)\right) \quad (31)$$

and $m_t = m_s + \sum_{i=1}^{n} m_i$, where m_i denotes the mass of the i-th link of the robotic manipulator. In order to achieve x-y positioning control and to compensate the torque τ_1 due to the interconnection with the manipulator, we modify the aforementioned relations as

$$\bar{U}_1 = U_1 \quad (32)$$

$$\bar{U}_2 = U_2 + \nu_y c\psi - \nu_x s\psi + T_{1,x} \quad (33)$$

$$\bar{U}_3 = U_3 - \nu_x c\psi - \nu_y s\psi + T_{1,y} \quad (34)$$

$$\bar{U}_4 = U_4 + T_{1,z}, \quad (35)$$

with the additional control parameters ν_x and ν_y defined as

$$\nu_x = k_{d,x}\left(\dot{x} - \dot{x}^{des}\right) + k_{p,x}\left(x - x^{des}\right) \quad (36)$$

$$\nu_y = k_{d,y}\left(\dot{y} - \dot{y}^{des}\right) + k_{p,y}\left(y - y^{des}\right). \quad (37)$$

and the additional torques $T_{1,x}$, $T_{1,y}$, $T_{1,z}$ defined as

$$\begin{bmatrix} T_{1,x} \\ T_{1,y} \\ T_{1,z} \end{bmatrix} = \tau_1 \begin{bmatrix} 1/l_a & 0 & 0 \\ 0 & 1/l_a & 0 \\ 0 & 0 & 1 \end{bmatrix} {}^{B}R_O \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (38)$$

It is important to state that equation (38) is valid only when the first degree of freedom of the manipulator is rotational. However, a similar result may be obtained for the prismatic case as well.

V. Simulation Studies

For the purposes of this work we decided to study the aerial manipulation system illustrated in Fig. 1. The system consists of a slightly modified "Pelican" quadrotor, which is a UAV model implemented within the "RotorS" simulator, with a two-link robotic arm attached on it. The quadrotor mass is given as $m_s = 1\ [kg]$, the arm length is considered as $l_a = 0.21\ [m]$ and the moments of inertia are defined as $I_{xx} = I_{yy} = 0.01\ [kg \cdot m^2]$ and $I_{zz} = 0.02\ [kg \cdot m^2]$. As for the robotic manipulator, we consider that the links are identical with masses $m_1 = m_2 = 0.1\ [kg]$ and lengths $l_1 = l_2 = 0.1\ [m]$, whereas the moments of inertia about their joints are given as $I_1 = I_2 = \frac{1}{3} m_1 l_1^2\ [kg \cdot m^2]$.

In the sequel, we are interested in comparing the efficiency of three different manipulator controllers, while also considering two distinct dynamical scenarios for the UAS. The first controller implements the computed torque technique. The second is a model-less gravity compensator working in conjunction with a PD controller that succeeds in canceling only the gravity terms that correspond to a hovering pose of the UAV. Finally, the third controller is a trivial PD compensator. In each case, the UAV controller is determined by the equations (28)-(38) and the pay-load of the manipulator is equal to zero.

A. Locked Manipulator

Let us consider Figs. 3-5. The computed torque controller guarantees the best system response over all the other controllers since it produces identically zero tracking errors of the joint variables. The model-less gravity compensator, based on the hover terms only, results in steady-state manipulator angle errors, whereas the PD compensator leads to the worst system behavior, demonstrating the largest steady-state manipulator angle deviations. The validity of the proposed control scheme (28)-(38) that takes into account the torque applied on the quadrotor due to the interconnection with the manipulator is also verified. Indeed, the UAV achieves a satisfactory tracking of the desired trajectory, regardless of the changed center of mass of the overall UAS.

B. Manipulator Maneuver

Let us consider Figs. 6-8. The computed torque controller again outperforms every other controller since it produces minimum tracking errors of the joint variables. The model-less gravity compensator, based on the hover terms only, succeeds in tracking the desired trajectory concerning the manipulator joint variables but introduces greater steady-state manipulator angle errors. Finally, the PD compensator, being a model-less controller, provides satisfactory tracking in the overall system, but demonstrates the largest steady-state manipulator angle deviations. Again, the UAV control appears to work efficiently even in the case when the manipulator executes a dynamic motion. The reactive torque due to the interconnection with the manipulator is compensated and the UAV succeeds in executing the circular reference trajectory.

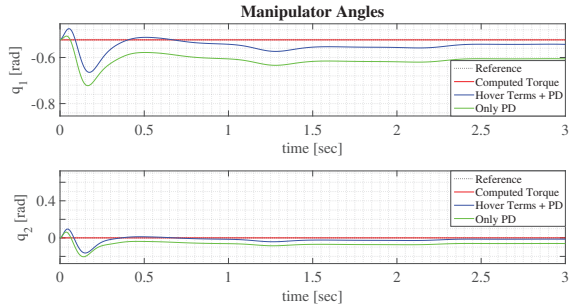

Fig. 3: Reference and actual trajectories of the joint variables of the manipulator - locked manipulator scenario.

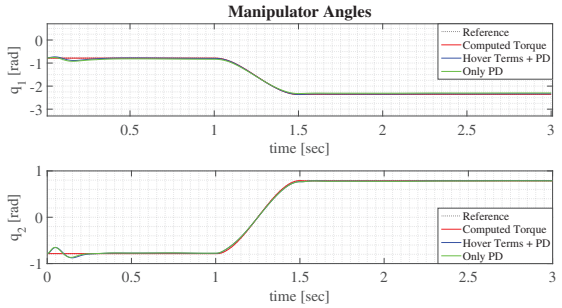

Fig. 6: Reference and actual trajectories of the joint variables of the manipulator - manipulator maneuver scenario.

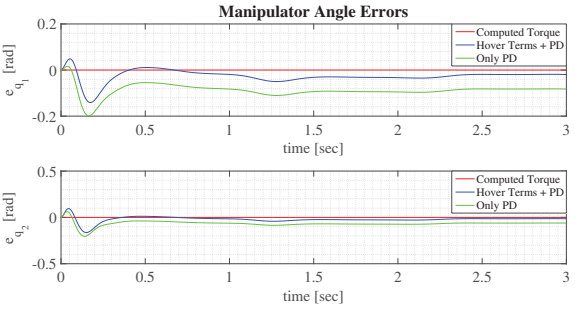

Fig. 4: Tracking errors of the joint variables of the manipulator - locked manipulator scenario.

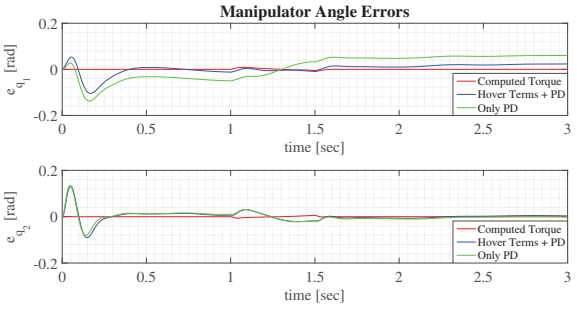

Fig. 7: Tracking errors of the joint variables of the manipulator - manipulator maneuver scenario.

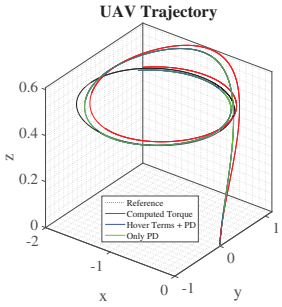

Fig. 5: Reference and actual trajectory of the UAV - locked manipulator scenario.

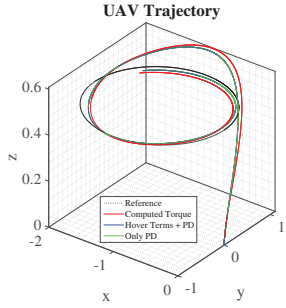

Fig. 8: Reference and actual trajectory of the UAV - manipulator maneuver scenario.

VI. Conclusions

This article concerns the modeling and control of a system that consists of a UAV with an attached manipulator. A systematic procedure that allows us to derive the fully coupled dynamic equations is presented and control schemes for both the UAV and the manipulator are developed. Simulation studies further delineate the efficiency of our methods.

Acknowledgments

This work has received funding from the European Union Horizon 2020 Research and Innovation Programme under the Grant Agreement No. 644128, AEROWORKS.

References

[1] K. Suseong, C. Seungwon, and H. J. Kim, Aerial manipulation using a quadrotor with a two DOF robotic arm, in Intelligent Robots and Systems (IROS), Tokyo, Japan, 2013, pp. 4990-4995.

[2] D. Bazylev, A. Margun, K. Zimenko and A. Kremlev, UAV equipped with a robotic manipulator, 22nd Mediterranean Conference of Control and Automation (MED), Palermo, Italy, 2014, pp. 1177-1182.

[3] S. Kannan, M. Alma, M. A. Olivares-Mendez and H. Voos, Adaptive Control of Aerial Manipulation Vehicle, International Conference on Control System, Computing and Engineering (ICCSCE), Batu Ferringhi, Malaysia, 2014, pp. 273-278.

[4] K. Alexis, G. Nikolakopoulos, and A. Tzes, Switching Model Predictive Attitude Control for a Quadrotor Helicopter subject to Atmospheric Disturbances, In Control Engineering Practice, vol. 19, no. 10, pp. 1195-1207, June, 2011.

[5] K. Alexis, G. Nikolakopoulos and A. Tzes, Model Predictive Quadrotor Control: Attitude, Altitude, and Position Experimental Studies, In IET-J. of Control Theory & Applications, vol. 6, no. 12, pp. 1812-1927, 2012.

[6] C. S. G. Lee, R. Gonzalez, K. S. Fu, Control, Sensing, Vision and Intelligence, McGraw-Hill Education, 1987.

[7] E. Papadopoulos and S. Dubowsky, On the nature of control algorithms for free-floating space manipulators, IEEE Transactions on Robotics and Automation, vol. 7(6), pp. 750-758, 1991.

[8] N. Koenig, A. Howard, Design and Use Paradigms for Gazebo, An Open-Source Multi-Robot Simulator, International Conference on Intelligent Robots and Systems, Sendai, Japan, pp.2149-2154, 2004.

[9] http://www.ode.org/

[10] F. Furrer, M. Burri, M. Achtelik, R. Siegwart, Robot Operating System (ROS): The Complete Reference (Volume 1), A Modular Gazebo MAV Simulator Framework, Springer International Publishing, pp. 595-625, 2016.