

Equilibrium-Based Force and Torque Control for an Aerial Manipulator to Interact with a Vertical Surface

*Original*

Equilibrium-Based Force and Torque Control for an Aerial Manipulator to Interact with a Vertical Surface / Tavora, B.; Park, H.; Romano, M.; Yun, X.. - In: ROBOTICA. - ISSN 0263-5747. - 38:4(2019), pp. 582-604. [10.1017/S0263574719000870]

*Availability:*

This version is available at: 11583/2963432 since: 2022-05-12T22:47:58Z

*Publisher:*

Cambridge University Press

*Published*

DOI:10.1017/S0263574719000870

*Terms of use:*

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

*Publisher copyright*

Cambridge University Press postprint/Author's Accepted Manuscript con licenza CC

This article has been published in a revised form in [Journal] [<http://doi.org/XXX>]. This version is published under a Creative Commons CC-BY-NC-ND. No commercial re-distribution or re-use allowed. Derivative works cannot be distributed. © copyright holder.

(Article begins on next page)

# Equilibrium-Based Force and Torque Control for an Aerial Manipulator to Interact with a Vertical Surface

Bruno Tavora<sup>†</sup>, Hyeongjun Park<sup>‡\*</sup> ,  
Marcello Romano<sup>¶</sup> and Xiaoping Yun<sup>§</sup>

<sup>†</sup>*Division of Aerodynamics, Controls, and Structures, Institute of Aeronautics and Space, Brazilian Air Force, São José dos Campos, Brazil. E-mail: [brunotavorabgt@fab.mil.br](mailto:brunotavorabgt@fab.mil.br)*

<sup>‡</sup>*Department of Mechanical and Aerospace Engineering, New Mexico State University, Las Cruces, NM, USA*

<sup>¶</sup>*Department of Mechanical and Aerospace Engineering, Naval Postgraduate School, Monterey, CA, USA. E-mail: [mromano@nps.edu](mailto:mromano@nps.edu)*

<sup>§</sup>*Department of Electrical and Computer Engineering, Naval Postgraduate School, Monterey, CA, USA. E-mail: [xyun@nps.edu](mailto:xyun@nps.edu)*

(Accepted May 11, 2019. First published online: June 17, 2019)

## SUMMARY

In this paper, a force and torque controller for aerial manipulation is developed using an unmanned aerial vehicle equipped with a robotic arm to interact near or on a vertical surface such as a wall. Control of aerial manipulators interacting with the environment is a challenging task due to dynamic interactions between aerial vehicles, robotic arms, and environment. To achieve this, modeling of aerial manipulators is first investigated and presented considering interaction with the environment. Nonlinear models of generic aerial manipulators, as well as of a prototype aerial manipulator composed of a hexacopter with a three-joint robotic arm, are established. An equilibrium-based force and torque controller is developed to conduct tasks that require the aerial manipulator to exert forces and torques on a wall. Simulations and experiments validate the performance of the controller that successfully applies desired forces and torques to an object fixed on a wall while flying near the wall.

**KEYWORDS:** Aerial manipulation; Multicopter; Robotic arm; Interaction; Force and torque control.

## 1. Introduction

Applications of unmanned aerial vehicles (UAVs) are usually limited to tasks without physical interaction, for example, tracking, surveillance, mapping, and visual inspection. The use of UAVs equipped with robotic arms, capable of interacting with the environment, enables to significantly increase their capability. In recent years, a new area of research has been growing on the use of aerial manipulators for a larger variety of applications such as physical inspection, maintenance, cleaning, and collecting objects in areas of difficult access.

Several research groups have developed aerial manipulators interacting with the environment. For instance, simulations and experiments are performed in ref. [1] to study an attitude control method of a quadcopter with a three-link robotic arm. A variable parameter integral backstepping controller with feed-forward compensation for the robotic arm static torques is compared to a proportional-integral-derivative (PID) controller. In ref. [2], for a multirotor with eight rotors,

\* Corresponding author. E-mail: [hjpark@nmsu.edu](mailto:hjpark@nmsu.edu)

a 7 degree-of-freedom (DoF) arm, sensors, and processing hardware needed for outdoor positioning, an interconnection and damping assignment passivity-based control (IDA-PBC) is presented with guaranteed stability of the closed-loop system based on Lyapunov theory. Kim et al.<sup>3</sup> developed an aerial manipulator with a two-link robotic arm. Using an adaptive sliding mode controller, the aerial manipulator takes off, flies toward an object whose position is known, grasps the object, and finally releases it in another place. In ref. [4], an aerial manipulator with two robotic arms is employed to turn a valve. The multicopter trajectory and robotic arms are commanded by means of a human-machine interface with support of a motion capture system. In this case, different dynamic models are derived for free flight and for the valve turning task, where aerodynamic effects are substituted by valve turning friction terms. An overview of research on aerial manipulation is provided in ref. [5].

In recent years, noticeable progress has been seen in the research area of aerial interaction with vertical surfaces. In refs. [6–8], a small quadcopter was designed to avoid obstacles during free flight and perform physical interaction tasks. A switching logic, based on the vehicle velocity, is defined to identify if the quadcopter is free flying or interacting with the wall. The control law is based on model predictive control (MPC). For the free-flight case, the cost function in MPC includes terms to avoid obstacles, while during the physical interaction mode, hard constraints, like maximum attitude angle to prevent the vehicle from crashing into the wall, are implemented. A tilt trirotor is employed and experimentally validated to apply a force on a vertical surface in ref. [9]. Since two of the propellers tilt while the fixed body remains in the horizontal plane, the aerial vehicle can produce a greater force on the object while keeping stability. The object is pushed by means of a flat end-effector with tactile sensors, for contact detection and switching mode capability. In ref. [10], a passivity-based controller is proposed and validated in simulations and experiments for a quadcopter equipped with a manipulator for aerial inspection. After the contact with the vertical surface, the system works as a mass-spring-damped system, converging to a constant interaction force and steady-state position error. Scholten et al.<sup>11</sup> used a multicopter equipped with a manipulator to interact on a vertical surface. The hybrid control switches from free-flight to interaction mode using a force sensor output threshold. During the contact with the wall, an impedance controller is responsible for movement. A similar experimental setup is described in ref. [12] to validate an impedance controller for physical interaction.

However, an important problem that has not been addressed by the research community is the interaction of a multicopter equipped with a robotic arm and a vertical wall. There are several applications related to this problem, such as performing maintenance on a piece of a vertical equipment, cleaning a wall, and opening a door knob. There were some research studies involving interaction with objects on the ground<sup>2–4</sup> or multicopter with simple prismatic joints pushing a vertical wall,<sup>6–9</sup> however, there is no record in the literature of an aerial manipulator with a more complex robotic arm, capable of producing forces or torques about any direction while interacting with a vertical obstacle.

The objective of this research is to investigate the feasibility of controlling a multicopter equipped with a multi-joint robotic arm to interact with a vertical wall while applying a specified force or torque. An equilibrium-based force and torque controller<sup>13</sup> for a multicopter with a robotic arm is considered. The controller is developed for a generic multicopter equipped with a multi-link robotic arm to apply any force and torque combination on an object fixed on a vertical wall. For this reason, the equilibrium-based force and torque controller has potential to be used in a large range of applications. Simulations and experiments with a hexacopter equipped with a three-DoF robotic manipulator are performed to validate the controller. The controller is proven its ability by applying a regulated force on a spring-mounted setup with high level of friction.

The rest of this paper is organized as follows. In Section 2, the dynamics of a generalized aerial manipulator with a multi-link robotic arm is described, and a model is introduced for a hexacopter with a three-link robotic arm. Next, the equilibrium-based force and torque controller is presented in Section 3. In Section 4, the simulation results with a MATLAB/Simulink simulator are reported for test scenarios. In Section 5, the experimental testbed setup is presented and the results compared with the simulation results are analyzed. Finally, concluding remarks are made at the end of the paper in Section 6.

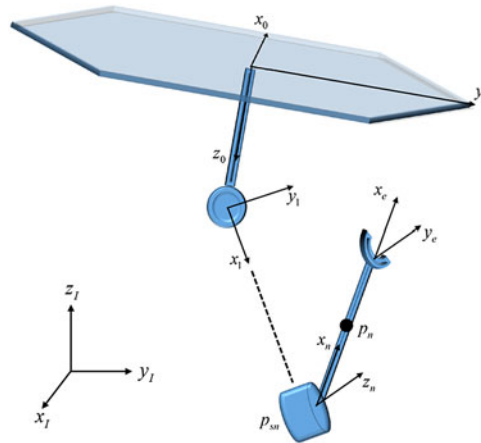


Fig. 1. Reference frames of an aerial manipulator.

## 2. Model

A mathematical model for a multicopter equipped with a robotic arm is developed in this section. A damped spring system is chosen to model interaction forces and torques on the wall. The forces and torques are then transformed by the Jacobian of the end-effector to be included in the model.

### 2.1. Aerial manipulator

A mathematical model for a generic multicopter with a robotic arm using rotational joints<sup>5,13,14</sup> is initially introduced. The dynamic equations are derived through the Euler-Lagrangian formulation.<sup>15</sup> In the equations, the torques and forces produced by the propellers, the torques produced by the revolute joint servos, the external interaction with the wall and the multicopter, and the robotic arm rigid body dynamics are considered. Here, the propeller rigid body dynamics and any other aerodynamic effects are neglected since the proposed research is concerned with tasks with no rapid-moving maneuvers.

The development of the dynamics equations follows the notations and conventions in ref. [16]. First, the position and orientation of the multicopter and the robotic links are derived in terms of the generalized coordinates of the aerial manipulator. These equations are differentiated to obtain the translational and angular velocities in terms of Jacobian matrices and the first derivatives of the generalized coordinates. Next, the resultant expression is applied to compute the kinetic and potential energy of the aerial manipulator. Finally, the energy expressions are applied to the Lagrangian to derive the dynamic equations.

A multirotor equipped with an  $n$ -link robotic arm has  $6 + n$  DoF. The vector  $\mathbf{q} = [x \ y \ z \ \phi \ \theta \ \psi \ \alpha_1 \ \dots \ \alpha_n]^T$  denotes the generalized coordinates, where  $\mathbf{p}_0 = [x \ y \ z]^T$  is the position of the multicopter's center of mass, expressed in the inertial frame.  $\Phi = [\phi \ \theta \ \psi]^T$  are the Euler angles for the multicopter orientation and  $\alpha = [\alpha_1 \ \dots \ \alpha_n]^T$  are the robotic arm joint angles. The reference frames for the multicopter body and the robot links are illustrated in Fig. 1. Let  $\mathbf{p}_{sk}$ , where  $k = 1, \dots, n$ , be the position of the  $k$ th robot joint expressed in the inertial frame,  $\mathbf{p}_k$  be the position of the  $k$ th link's center of mass expressed in the inertial frame, and  $\omega_k$  be its angular velocity expressed in the respective link's body-fixed frame. The multicopter body-fixed frame is located according to the North-East-Down convention and the Euler angles are defined by the sequence of rotations yaw-pitch-roll, about the axes  $z$ ,  $y$ , and  $x$ , respectively. The rotation matrix<sup>15</sup> denoting the orientation of the multicopter body-fixed frame with respect to the inertial frame is

$$\mathbf{R}_{i0} = \begin{bmatrix} c_\theta c_\psi & s_\phi s_\theta c_\psi - c_\phi s_\psi & c_\phi s_\theta c_\psi + s_\phi s_\psi \\ c_\theta s_\psi & s_\phi s_\theta s_\psi + c_\phi c_\psi & c_\phi s_\theta s_\psi - s_\phi c_\psi \\ -s_\theta & c_\theta s_\phi & c_\theta c_\phi \end{bmatrix}, \quad (1)$$

where  $c_\beta$  and  $s_\beta$  denote  $\cos \beta$  and  $\sin \beta$ , respectively.

Let  $\omega_0$  be the multicopter's angular velocity expressed in the body-fixed frame, which can be written in terms of the time derivatives of the Euler angles as in ref. [17],

$$\omega_0 = \mathbf{Q}\dot{\Phi}, \quad (2)$$

where

$$\mathbf{Q} = \begin{bmatrix} 1 & 0 & -s_\theta \\ 0 & c_\phi & s_\phi c_\theta \\ 0 & -s_\phi & c_\phi c_\theta \end{bmatrix}. \quad (3)$$

The translational and angular velocities of the multicopter body are combined into a six-dimensional vector denoted by  $\mathbf{v}_0 = [\dot{\mathbf{p}}_0^T \ \boldsymbol{\omega}_0^T]^T$ . This vector is related to the time derivative of the generalized coordinates through a Jacobian matrix as  $\mathbf{v}_0 = \mathbf{J}_0 \dot{\mathbf{q}}$ , where

$$\mathbf{J}_0 = \begin{bmatrix} \mathbf{J}_{p_0} \\ \mathbf{J}_{\omega_0} \end{bmatrix} = \begin{bmatrix} \mathbb{I}_3 & \mathbb{O}_{3 \times 3} & \mathbb{O}_{3 \times n} \\ \mathbb{O}_{3 \times 3} & \mathbf{Q} & \mathbb{O}_{3 \times n} \end{bmatrix}.$$

Here,  $\mathbb{I}_3$  is the  $3 \times 3$  identity matrix and  $\mathbb{O}_{i \times j}$  is a null matrix with  $i$  rows and  $j$  columns.

For each robotic arm link, the differential kinematics of the center of mass, orientation, and associated joint position can also be written in terms of Jacobian matrices given by

$$\mathbf{v}_k = \begin{bmatrix} \dot{\mathbf{p}}_k \\ \boldsymbol{\omega}_k \end{bmatrix} = \begin{bmatrix} \mathbf{J}_{p_k} \\ \mathbf{J}_{\omega_k} \end{bmatrix} \dot{\mathbf{q}} = \mathbf{J}_k \dot{\mathbf{q}}, \quad (4)$$

and  $\dot{\mathbf{p}}_{sk} = \mathbf{J}_{sk} \dot{\mathbf{q}}$ .

The angular velocity of each link is expressed recursively using the rotating matrix,  $\mathbf{R}_{k,k-1}$ , which denotes the orientation of the link  $k-1$  with respect to the link  $k$ . For  $k=1$ ,  $\mathbf{R}_{10}$  is the rotation matrix denoting the orientation of the multicopter body with respect to the first link. The recursive angular rate equations are given by refs. <sup>13,15</sup>

$$\boldsymbol{\omega}_k = \mathbf{R}_{k,k-1} \boldsymbol{\omega}_{k-1} + [0 \ 0 \ \dot{\alpha}_k]^T, \quad (5)$$

$$\mathbf{J}_{\omega k} = \mathbf{R}_{k,k-1} \mathbf{J}_{\omega,k-1} + \begin{bmatrix} 0 \\ \mathbb{O}_{3 \times (6+k-1)} \ 0 \ \mathbb{O}_{3 \times (n-k)} \\ 1 \end{bmatrix}. \quad (6)$$

Here, the null matrices are concatenated with the vector  $[0 \ 0 \ 1]^T$  to produce a matrix with zeros, except being equal to one for the element at the third row and the  $(k+6)$ th column. The end-effector orientation is defined relative to the orientation of the last link transformed by a constant rotation matrix  $\mathbf{R}_{en}$  as follows:

$$\mathbf{J}_{\omega e} = \mathbf{R}_{en} \mathbf{J}_{\omega n}. \quad (7)$$

Let the parameter  $\mathbf{p}_{01}$  be the displacement of the first joint relative to the center of mass of the multicopter expressed in the multicopter body-fixed frame. Hence, the kinematics of the first joint is given by

$$\mathbf{p}_{s1} = \mathbf{p}_0 + \mathbf{R}_{i0} \mathbf{p}_{01},$$

$$\dot{\mathbf{p}}_{s1} = \dot{\mathbf{p}}_0 + \mathbf{R}_{i0} \boldsymbol{\omega}_0^\times \mathbf{p}_{01}, \quad (8)$$

$$\mathbf{J}_{s1} = \mathbf{J}_{p0} - \mathbf{R}_{i0} \mathbf{p}_{01}^\times \mathbf{J}_{\omega 0}.$$

The body-fixed reference frame  $(\hat{x}_k \ \hat{y}_k \ \hat{z}_k)$  is placed at the  $k$ th joint, with  $x_k$  parallel to the link and  $z_k$  parallel to the axis of rotation of the  $k$ th joint, as shown in Fig. 1. Let  $\mathbf{R}_{ik}$  be the rotation matrix denoting the orientation of the body-fixed frame with respect to the inertial frame. Then, the position and velocity of the joints are computed recursively by means of

$$\mathbf{p}_{s,k+1} = \mathbf{p}_{sk} + \mathbf{R}_{ik} [l_k \ 0 \ 0]^T,$$

$$\dot{\mathbf{p}}_{s,k+1} = \dot{\mathbf{p}}_{sk} + \mathbf{R}_{ik} \boldsymbol{\omega}_k^\times [l_k \ 0 \ 0]^T, \quad (9)$$

$$\mathbf{J}_{s,k+1} = \mathbf{J}_{sk} - \mathbf{R}_{ik} \left( [l_k \ 0 \ 0]^T \right)^\times \mathbf{J}_{\omega k},$$

where  $l_k$  is the length of the  $k$ th link.<sup>15</sup> The position and velocity of each link at the center of mass are also expressed recursively as follows:

$$\begin{aligned} \mathbf{p}_k &= \mathbf{p}_{sk} + \mathbf{R}_{ik} [l_{ck} \ 0 \ 0]^T, \\ \dot{\mathbf{p}}_k &= \dot{\mathbf{p}}_{sk} + \mathbf{R}_{ik} \boldsymbol{\omega}_k^\times [l_{ck} \ 0 \ 0]^T, \end{aligned} \quad (10)$$

$$\mathbf{J}_{pk} = \mathbf{J}_{sk} - \mathbf{R}_{ik} \left( [l_{ck} \ 0 \ 0]^T \right)^\times \mathbf{J}_{\omega k},$$

where  $l_{ck}$  is the distance from the  $k$ th joint to the center of mass of the  $k$ th link. Similarly, the Jacobian of the end-effector is given by

$$\mathbf{J}_{pe} = \mathbf{J}_{sn} - \mathbf{R}_{in} \left( [l_e \ 0 \ 0]^T \right)^\times \mathbf{J}_{\omega n}. \quad (11)$$

Let  $\mathbf{I}_k$  and  $m_k$  be the moment of inertia matrix and the mass of the link  $k$ . The kinetic energy of the multicopter ( $k=0$ ) and the links ( $k=1, \dots, n$ )<sup>18</sup> are defined by

$$K_k = \frac{1}{2} \mathbf{v}_k^T \begin{bmatrix} m_k \mathbb{I}_3 & \mathbb{O}_{3 \times 3} \\ \mathbb{O}_{3 \times 3} & \mathbf{I}_k \end{bmatrix} \mathbf{v}_k = \frac{1}{2} \mathbf{v}_k^T \mathbf{M}_k \mathbf{v}_k. \quad (12)$$

From Eqs. (4) and (12), the expressions for the kinetic energy of each link and for the total kinetic energy are given by

$$K_k = \frac{1}{2} \dot{\mathbf{q}}^T \mathbf{J}_k^T \mathbf{M}_k \mathbf{J}_k \dot{\mathbf{q}} = \frac{1}{2} \dot{\mathbf{q}}^T \mathbf{B}_k \dot{\mathbf{q}}, \quad (13)$$

$$K = \frac{1}{2} \dot{\mathbf{q}}^T \left( \sum_{k=0}^n \mathbf{B}_k \right) \dot{\mathbf{q}} = \frac{1}{2} \dot{\mathbf{q}}^T \mathbf{B} \dot{\mathbf{q}}. \quad (14)$$

The potential energy depends on the masses and the vertical component of the center of mass position of each link as follows:

$$P_k = [0 \ 0 \ -m_k g] \mathbf{p}_k, \quad P = \sum_{k=0}^n P_k. \quad (15)$$

Substituting the kinetic and potential energies in the Lagrange equation, we have the motion equation:

$$\frac{d}{dt} \left( \frac{\partial K}{\partial \dot{q}_i} \right) - \frac{\partial K}{\partial q_i} + \frac{\partial P}{\partial q_i} = \tau_i, \quad (16)$$

where  $i = 1, \dots, n+6$ .

Combining Eqs. (14)–(16) yields the dynamic equations of the system

$$\begin{aligned} & \mathbf{B}(\mathbf{q}) \ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) \\ &= \mathbf{B}(\mathbf{q}) \ddot{\mathbf{q}} + \dot{\mathbf{B}}(\mathbf{q}) \dot{\mathbf{q}} - 0.5 \begin{bmatrix} \dot{\mathbf{q}}^T \frac{\partial \mathbf{B}}{\partial q_1} \\ \vdots \\ \dot{\mathbf{q}}^T \frac{\partial \mathbf{B}}{\partial q_{n+6}} \end{bmatrix} \dot{\mathbf{q}} + \begin{bmatrix} \frac{\partial P}{\partial q_1} \\ \vdots \\ \frac{\partial P}{\partial q_{n+6}} \end{bmatrix} \\ &= \boldsymbol{\tau} + \boldsymbol{\tau}_{\text{ext}}, \end{aligned} \quad (17)$$

where  $\boldsymbol{\tau}$  is the generalized input force vector and  $\boldsymbol{\tau}_{\text{ext}}$  is the generalized external forces vector at the joint level. The matrix  $\dot{\mathbf{B}}$  is determined by  $\dot{\mathbf{B}} = \sum_{i=1}^{n+6} \left( \frac{\partial \mathbf{B}}{\partial q_i} \dot{q}_i \right)$ .



Fig. 2. Hexacopter with a three-DoF robotic arm.

When the manipulator end-effector interacts with a wall, torques and forces are induced on the multicopter's center of mass, and torques are induced on the joints. The external interaction forces and torques, exerted by the environment on the end-effector, are mapped to the generalized coordinate space as follows:<sup>16</sup>

$$\tau_{\text{ext}} = \begin{bmatrix} \mathbf{J}_{pe} \\ \mathbf{J}_{\omega e} \end{bmatrix}^T \begin{bmatrix} \mathbf{F}_e \\ \boldsymbol{\tau}_e \end{bmatrix} = \mathbf{J}_e^T \begin{bmatrix} \mathbf{F}_e \\ \boldsymbol{\tau}_e \end{bmatrix}. \quad (18)$$

The input forces and torques are produced by the rotors of the multicopter and the servos in the robotic arm joints. They are mapped to the generalized coordinate level<sup>19</sup> given by

$$\boldsymbol{\tau} = \begin{bmatrix} \mathbf{F}_0 \\ \boldsymbol{\tau}_\phi \\ \boldsymbol{\tau}_\alpha \end{bmatrix} = \begin{bmatrix} \mathbf{R}_{i0} & 0 & 0 \\ 0 & \mathbf{Q}^T & 0 \\ 0 & 0 & \mathbb{I}_n \end{bmatrix} \begin{bmatrix} [0 \ 0 \ -F_t]^T \\ \boldsymbol{\tau}_p \\ \boldsymbol{\tau}_\alpha \end{bmatrix}. \quad (19)$$

When a single propeller rotates, it produces a torque and a force along the vertical axis (orthogonal to the plane of rotation). In a multicopter with several propellers, those forces and torques are combined to result in a force and a torque on the center of mass of the multicopter body. In Eq. (19), the term  $F_t$  is the lifting force in the multicopter body-fixed frame produced by the propellers and  $\boldsymbol{\tau}_p$  is the resultant torque. These terms are mapped to the generalized coordinate system resulting in  $\mathbf{F}_0$ , the force expressed in terms of the inertial frame, and  $\boldsymbol{\tau}_\phi$  which is the torque associated with the multicopter Euler angles. Finally, the vector  $\boldsymbol{\tau}_\alpha$  represents the torques applied by the servos on the robotic arm.

## 2.2. Hexacopter with a three-DoF robotic arm

The equations of motion developed for a multicopter with  $n$ -DoF manipulators are applied here to a hexacopter with a three-DoF robotic arm (see Fig. 2) which is used in the subsequent simulation study and experimentation.

The orientation of the joints are shown in Fig. 3. The drawing illustrates the link coordinate systems at the initial configuration for the angles  $\alpha_1$ ,  $\alpha_2$ , and  $\alpha_3$  that are all zeros with  $x_1$ ,  $x_2$ , and  $x_3$  pointing in the same (downward) direction. With the defined link coordinate systems, the link rotation matrices<sup>15</sup> are given by

$$\mathbf{R}_{01}(\alpha_1) = \begin{bmatrix} 0 & 0 & 1 \\ -s_{\alpha_1} & -c_{\alpha_1} & 0 \\ c_{\alpha_1} & -s_{\alpha_1} & 0 \end{bmatrix}, \quad \mathbf{R}_{12}(\alpha_2) = \begin{bmatrix} c_{\alpha_2} & -s_{\alpha_2} & 0 \\ 0 & 0 & -1 \\ s_{\alpha_2} & c_{\alpha_2} & 0 \end{bmatrix},$$

$$\mathbf{R}_{23}(\alpha_3) = \begin{bmatrix} c_{\alpha_3} & -s_{\alpha_3} & 0 \\ s_{\alpha_3} & c_{\alpha_3} & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

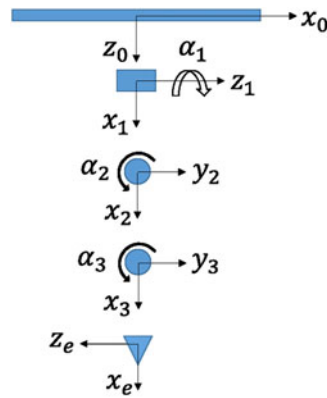


Fig. 3. Joint orientation of the robotic arm.

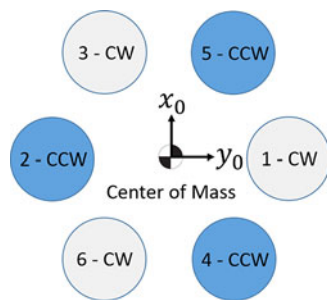


Fig. 4. Top view of the hexacopter.

The fixed frame on the end-effector is also attached to the third link as illustrated in Fig. 3. As a result, the rotation matrix between this frame and the third link coordinate system is constant and given by

$$\mathbf{R}_{3e} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}.$$

The force and torque, induced on the center of mass of the multicopter, depend on the number of propellers, their geometry, and direction of rotation (clockwise or counter-clockwise). In Fig. 4, the hexagonal geometry for the hexacopter is shown, as well as the number that identifies each propeller and their direction of rotation. Given the pulse width modulation (PWM) input to each motor, the resultant forces and torques are calculated as in ref. [20],

$$\begin{aligned} F_t &= K_F \sum_{k=1}^6 (\rho_k - \rho_0), \\ \tau_x &= \frac{l_H K_F}{2} (\rho_2 + \rho_3 + 2\rho_6 - 2\rho_1 - \rho_4 - \rho_5), \\ \tau_y &= \frac{\sqrt{3}l_H K_F}{2} (\rho_3 + \rho_5 - \rho_6 - \rho_4), \\ \tau_z &= K_\tau (\rho_2 + \rho_4 + \rho_5 - \rho_1 - \rho_3 - \rho_6), \\ \tau_p &= [\tau_x \quad \tau_y \quad \tau_z]^T, \end{aligned} \quad (20)$$

where  $K_F$ ,  $\rho_0$ , and  $K_\tau$  are the propulsion system constants,  $l_H$  is the distance between the centers of adjacent propellers, and  $\rho_k$  is the input to the motor  $k$ .



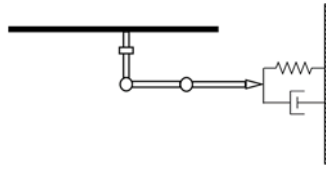


Fig. 5. Aerial manipulator interacting with a wall.

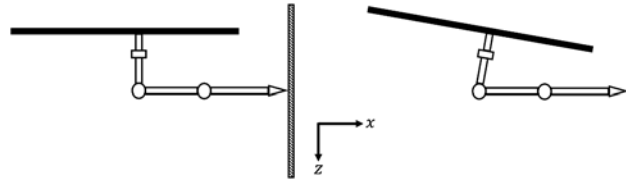


Fig. 6. Vertical surface interaction of an aerial manipulator.

### 2.3. Simulation model for wall interaction

When an aerial manipulator performs a task to interact with a wall as shown in Fig. 5, the end-effector exerts a force  $\mathbf{F}_e$  and torque  $\boldsymbol{\tau}_e$  on the wall.<sup>18</sup> The force and torque are modeled as in a damped spring system given by

$$\begin{bmatrix} \mathbf{F}_e \\ \boldsymbol{\tau}_e \end{bmatrix} = \mathbf{K}_{pw} \begin{bmatrix} \Delta \mathbf{p}_e \\ \Delta \Phi_e \end{bmatrix} + \mathbf{K}_{dw} \begin{bmatrix} \dot{\mathbf{p}}_e \\ \boldsymbol{\omega}_e \end{bmatrix}, \quad (21)$$

where  $\Delta \mathbf{p}_e$  and  $\Delta \Phi_e$  are the position and attitude error and  $\mathbf{K}_{pw}$  and  $\mathbf{K}_{dw}$  are stiffness and damping matrices, respectively. The interaction forces and torques constrain the movement of the end-effector as if it was holding a rigid object on the wall. By choosing high gain values, a rigid system is addressed, while the flexibility of the robotic arm and the manipulated object are simulated for smaller gain values.

## 3. Controller Design

Control of the end-effector position, during free flight and while the robot is picking or placing objects, has been studied in aerial manipulation.<sup>1-3</sup> In this research, a new problem to develop a controller is explored, in which an aerial manipulator is used to perform manipulation tasks on a vertical wall.

### 3.1. Equilibrium-based force and torque controller

A controller is developed here for aerial manipulators interacting with an object on a wall. Initially, an open-loop controller is designed based on the equilibrium constraint when the robotic arm interacts with the object. Next, to enhance the system stability, the feedback from an internal attitude estimator and a gyroscope sensor, both mounted on the multicopter, is included in the control law. Throughout the development, the equations are derived for a generic aerial manipulator.

For static interaction between an aerial manipulator and a wall, the following two constraints must be satisfied. First, the aerial manipulator needs to achieve an appropriate posture to exert a desired force and torque since the system is underactuated. For instance, if the desired interaction is a static force toward the wall, the vehicle is required to have a pitch angle so that a horizontal component of the thrust force is induced, which is visualized in Fig. 6. Second, in this case, the end-effector manipulates an object that is fixed on the wall. The orientation and position of the end-effector do not vary during the interaction.

The appropriate posture to achieve the desired force and torque interaction corresponds to an equilibrium condition, where the velocities and accelerations are null. Rewriting the dynamic equation (17) of the system with  $\dot{\mathbf{q}} = 0$  and  $\ddot{\mathbf{q}} = 0$  results in the equilibrium equation for the aerial interaction as

$$\mathbf{G}(\mathbf{q}) = \boldsymbol{\tau} + \boldsymbol{\tau}_{\text{ext}}. \quad (22)$$

Substituting Eqs. (18) and (19) into Eq. (22) gives

$$\begin{bmatrix} \mathbf{F}_T \\ \boldsymbol{\tau}_p \\ \boldsymbol{\tau}_a \end{bmatrix} = \begin{bmatrix} \mathbf{R}_{i0}(\mathbf{q}) & 0 & 0 \\ 0 & \mathbf{Q}^T(\mathbf{q}) & 0 \\ 0 & 0 & \mathbb{I}_n \end{bmatrix}^{-1} \left( \mathbf{G}(\mathbf{q}) - \mathbf{J}_e^T(\mathbf{q}) \begin{bmatrix} \mathbf{F}_e \\ \boldsymbol{\tau}_e \end{bmatrix} \right), \quad (23)$$

where  $\mathbf{F}_T = [0 \ 0 \ -F_t]^T$ .

For the desired interaction, defined by  $\mathbf{F}_e$  and  $\boldsymbol{\tau}_e$ , the equilibrium condition is only achieved if the first two elements of the resultant column vector are zero, as in the left-hand side of Eq. (23). The first two rows are zero since the thrust force has no components in the  $xy$  plane of the body-fixed frame. The derivation of the attitude constraint is developed by algebraic manipulation of the first three rows of the equilibrium equation. The relevant components of  $\mathbf{G}(\mathbf{q})$  and  $\mathbf{J}_e^T(\mathbf{q})$  are

$$\mathbf{G}(\mathbf{q}) = \begin{bmatrix} 0 \\ 0 \\ -m_t g \\ \vdots \end{bmatrix}, \quad \mathbf{J}_e^T(\mathbf{q}) = \begin{bmatrix} \mathbb{I}_3 & \mathbb{O}_{3 \times 3} \\ \vdots & \vdots \end{bmatrix}. \quad (24)$$

Hence, the first three rows of Eq. (23) are equivalent to

$$\begin{bmatrix} c_\phi s_\theta c_\psi + s_\phi s_\psi \\ c_\phi s_\theta s_\psi - s_\phi c_\psi \\ c_\phi c_\theta \end{bmatrix} F_t = \begin{bmatrix} 0 \\ 0 \\ m_t g \end{bmatrix} + \begin{bmatrix} F_{ex} \\ F_{ey} \\ F_{ez} \end{bmatrix}. \quad (25)$$

If the last row is taken to express  $F_t$  in terms of  $F_{ez}$ , the first two rows can be manipulated to result in

$$\tan \theta = \frac{F_{ex} c_\psi + F_{ey} s_\psi}{F_{ez} + m_t g}, \quad \tan \phi = \frac{F_{ex} s_\psi - F_{ey} c_\psi}{F_{ez} + m_t g} c_\theta, \quad (26)$$

which are the final equations for the attitude constraint. It is interesting to note that if  $\phi$  and  $\theta$  are zero, then  $F_{ex}$  and  $F_{ey}$  are necessarily zero. This is the mathematical confirmation that the multicopter needs to tilt to an appropriate attitude to apply forces in the horizontal plane. In addition to the underactuation constraint, there is the second constraint related to the orientation of the end-effector during the force and torque interaction. The end-effector orientation  $\Phi_e(\boldsymbol{\eta})$  is determined from the multicopter orientation  $\Phi$  and the joint angles  $\boldsymbol{\alpha}$ . During interaction, the end-effector orientation is constant and denoted by  $\Phi_{ei} = \Phi_e(\boldsymbol{\eta})$ , where  $\boldsymbol{\eta} = [\Phi^T \ \boldsymbol{\alpha}^T]^T$ .<sup>15</sup>

The equilibrium constraints can be grouped into a nonlinear function  $\mathbf{f}(\boldsymbol{\eta}) = 0$  given by

$$\mathbf{f}(\boldsymbol{\eta}) = \begin{bmatrix} \Phi_e(\boldsymbol{\eta}) - \Phi_{ei} \\ \tan \theta - \frac{F_{ex} c_\psi + F_{ey} s_\psi}{F_{ez} + m_t g} \\ \tan \phi - \frac{F_{ex} s_\psi - F_{ey} c_\psi}{F_{ez} + m_t g} c_\theta \end{bmatrix} = \begin{bmatrix} \Phi_e(\boldsymbol{\eta}) - \Phi_{ei} \\ f_4(\boldsymbol{\eta}) \\ f_5(\boldsymbol{\eta}) \end{bmatrix} = 0. \quad (27)$$

Depending on the complexity of the robotic arm, Eq. (27) may not have an analytical solution. For this case, a first-order approximation is proposed to compute the solution  $\boldsymbol{\eta}_r$ ,

$$\mathbf{f}(\boldsymbol{\eta}_r) = \mathbf{f}(\boldsymbol{\eta}) + \mathbf{F}_\eta (\boldsymbol{\eta}_r - \boldsymbol{\eta}) = 0, \quad (28)$$

where  $\mathbf{F}_\eta = \frac{\partial \mathbf{f}(\boldsymbol{\eta})}{\partial \boldsymbol{\eta}}$ . Since  $\Phi_{ei}$  is a three-by-one vector, the system defined by  $\mathbf{f}(\boldsymbol{\eta})$  has five equations and  $3 + n$  variables. If an aerial manipulator has two links with  $n = 2$ ,  $\mathbf{F}_\eta$  is a  $5 \times 5$  square matrix. Equation (28) has a unique solution given by  $\boldsymbol{\eta}_r = \boldsymbol{\eta} - \mathbf{F}_\eta^{-1} \mathbf{f}(\boldsymbol{\eta})$ . If an aerial manipulator has more than two links, the system is redundant.<sup>21</sup> Redundancy can be utilized to accomplish secondary tasks, such as increasing reachability, avoiding an obstacle, avoiding singularities, or achieving a safer posture.<sup>22</sup> In this research, the focus is on accomplishing the primary task of exerting a desired force and torque on a wall. A standard solution using the pseudoinverse of  $\mathbf{F}_\eta$  is chosen as

$$\boldsymbol{\eta}_r = \boldsymbol{\eta} - \mathbf{F}_\eta^T (\mathbf{F}_\eta \mathbf{F}_\eta^T)^{-1} \mathbf{f}(\boldsymbol{\eta}). \quad (29)$$

Since Eq. (29) is a first-order approximation, the solution is valid only if  $\eta_r$  is in the neighborhood of the state variables in  $\eta$ . It is important to analyze and simulate each particular case carefully before the experiments, since the numerical convergence is not guaranteed when  $\eta$  is out of the neighborhood of the solution. A better approach is, when planning the experiment, to select a vector  $\eta_0$  that is known *a priori* to be close to the solution and compute Eq. (29) for  $\eta = \eta_0$ .

From Eqs. (27) and (28),

$$\mathbf{F}_\eta = \begin{bmatrix} \frac{\partial \Phi_e}{\partial \eta} \\ \frac{\partial f_4(\eta)}{\partial \eta} \\ \frac{\partial f_5(\eta)}{\partial \eta} \end{bmatrix}. \quad (30)$$

The partial derivative  $\frac{\partial \Phi_e}{\partial \eta}$  can be determined from the Jacobian  $\mathbf{J}_{\omega e}$ , which is known from Eq. (7). It is important to note that the end-effector orientation does not depend on the position of the multicopter. The Jacobian can be reduced to a smaller matrix  $\mathbf{J}_{oe}$ ,

$$\omega_e = \mathbf{J}_{\omega e} \dot{\mathbf{q}} = [\mathbb{O}_{3 \times 3} \quad \mathbf{J}_{oe}] [\dot{\mathbf{p}}^T \quad \dot{\eta}^T]^T = \mathbf{J}_{oe} \dot{\eta}. \quad (31)$$

Similarly to the multicopter orientation, as in Eq. (2), the end-effector angular velocities can also be mapped to the time derivatives of the Euler angles,

$$\dot{\Phi}_e = \mathbf{Q}_e^{-1} \omega_e = \mathbf{Q}_e^{-1} \mathbf{J}_{oe} \dot{\eta}, \quad (32)$$

where  $\mathbf{Q}_e$  is obtained similarly to the way  $\mathbf{Q}$  is determined from Eq. (3). Since  $\dot{\Phi}_e$  can also be written as  $\dot{\Phi}_e = \frac{\partial \Phi_e}{\partial \eta} \dot{\eta}$ , the partial derivative of the end-effector orientation is given by  $\frac{\partial \Phi_e}{\partial \eta} = \mathbf{Q}_e^{-1} \mathbf{J}_{oe}$ .

The partial derivatives of  $f_4(\eta)$  and  $f_5(\eta)$  are

$$\begin{bmatrix} \frac{\partial f_4(\eta)}{\partial \eta} \\ \frac{\partial f_5(\eta)}{\partial \eta} \end{bmatrix} = \begin{bmatrix} \frac{\partial f_4(\Phi)}{\partial \Phi} & \frac{\partial f_4(\alpha)}{\partial \alpha} \\ \frac{\partial f_5(\Phi)}{\partial \Phi} & \frac{\partial f_5(\alpha)}{\partial \alpha} \end{bmatrix}, \quad (33)$$

where the derivatives with respect to  $\alpha$  are zero, and

$$\begin{bmatrix} \frac{\partial f_4(\Phi)}{\partial \Phi} \\ \frac{\partial f_5(\Phi)}{\partial \Phi} \end{bmatrix} = \begin{bmatrix} 0 & \frac{1}{c_\theta^2} & F_1 \\ \frac{1}{c_\theta} & F_1 s_\theta & F_2 c_\theta \end{bmatrix}, \quad (34)$$

where

$$F_1 = \frac{F_{ex} s_\psi - F_{ey} c_\psi}{F_{ez} + m_t g}, \quad F_2 = -\frac{F_{ex} c_\psi + F_{ey} s_\psi}{F_{ez} + m_t g}.$$

An important step during the experiment design is to analyze for what subset of the possible values of  $\eta$ , the matrix is full rank, that is,  $\mathbf{F}_\eta \mathbf{F}_\eta^T$  is invertible. One of the reasons why  $\mathbf{F}_\eta$  would not be full rank is the impossibility to rotate the multicopter about all possible directions, due to limitations of the robotic arm. For instance, if the robotic arm has only one joint, the multicopter only rotates about one axis during manipulation. In this case, the matrix  $\mathbf{F}_\eta$  has only four columns and the rank is not greater than four. For a robotic arm with at least two DoF, where the joints are mounted orthogonally to each other,  $\mathbf{F}_\eta$  is expected to be full rank in most configurations.

To illustrate how to investigate singularities, the specific configuration considered for the simulations and experiments is analyzed.<sup>13</sup> During the experiments, the initial posture from which the multicopter is supposed to apply the forces and torques is shown in Fig. 6. For the case, the end-effector orientation and robotic arm initial angles are given by

$$\Phi_e = [0 \quad 0 \quad 0]^T, \quad \alpha = [0 \quad \pi/2 \quad 0]^T. \quad (35)$$

Here, the desired interaction forces are small when compared with the weight of the aerial manipulator. Hence, the attitude angles performed by the multicopter are also small. Equation (29) is calculated from the initial condition  $\eta_0 = [0 \quad 0 \quad 0 \quad 0 \quad \pi/2 \quad 0]^T$ , which is in the neighborhood of  $\eta_r$  where the

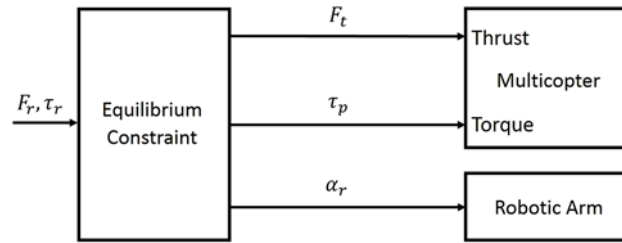


Fig. 7. Open-loop control scheme for vertical surface interaction.

first-order approximation is valid. The partial derivative of  $\mathbf{f}(\boldsymbol{\eta})$  is given by

$$\mathbf{F}_{\boldsymbol{\eta}}(\boldsymbol{\eta}_0) = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & \frac{-F_{ey}}{F_{ez}+m_t g} & 0 & 0 & 0 \\ 1 & 0 & \frac{-F_{ex}}{F_{ez}+m_t g} & 0 & 0 & 0 \end{bmatrix}, \quad (36)$$

which implies  $\det(\mathbf{F}_{\boldsymbol{\eta}}\mathbf{F}_{\boldsymbol{\eta}}^T) = 2 \neq 0$ . Since the determinant is not zero, the matrix  $\mathbf{F}_{\boldsymbol{\eta}}\mathbf{F}_{\boldsymbol{\eta}}^T$  is invertible independently of the desired interaction forces.

In summary, the equilibrium constraints are computed from Eqs. (23) and (29). First, the reference attitude and joint angles of reference are determined from Eq. (29). Once the posture of reference  $\boldsymbol{\eta}_r$  is defined, it is substituted in Eq. (23) to obtain the thrust and the attitude torques. In Fig. 7, the open-loop scheme is shown, where the block ‘Equilibrium Constraint’ represents the calculation of the system inputs based on the reference force and torque,  $\mathbf{F}_r$  and  $\boldsymbol{\tau}_r$ . This controller does not use any sensor feedback and, in practice, is not appropriate in terms of stability and convergence. Therefore, another controller is developed to integrate sensor feedbacks to this scheme.

Although the robotic arm used in this research is modeled as a rigid body, it is built from plastic parts that bend under force and/or when there is backlash in servo motor gears. When the open-loop controller depicted in Fig. 7 was implemented in experiment, rapid oscillations in attitude were observed. For this reason, an attitude proportional-derivative (PD) controller is designed to damp these oscillations. Using this controller, the attitude is regulated by the propellers to guarantee convergence with stability. The PD attitude control with output to the multicopter is given by  $\boldsymbol{\tau}_{AC} = \mathbf{K}_{\Phi p}(\Phi_r - \Phi) - \mathbf{K}_{\Phi d}\boldsymbol{\omega}_0$ , where the gains  $\mathbf{K}_{\Phi p}$  and  $\mathbf{K}_{\Phi d}$  are positive definite diagonal matrices.<sup>15</sup>

The flexibility of the robotic arm is also related to another issue. The robotic arm joint angles are calculated to bring the multicopter to a desired attitude. When the arm bends, it drives the multicopter to an incorrect attitude. When it happens, the PD controller outputs a residual torque because the attitude error is not null. The incorrect attitude and the residual torque cause interaction force and torque errors. To fix the issue, an integral attitude controller is designed. With attitude feedback, the robotic arm is moved to a position for which the attitude error is zero. With these two attitude controllers, the multicopter and the robotic arm work in collaboration. The robotic arm angles are modified by an integral controller given by

$$\boldsymbol{\alpha}_{AC} = \mathbf{K}_{\Phi i} \int \Delta \boldsymbol{\alpha} dt, \quad (37)$$

where the gain  $\mathbf{K}_{\Phi i}$  is a positive definite diagonal matrix and  $\Delta \boldsymbol{\alpha}$  is the attitude error mapped to joint angle errors.

The Jacobian pseudoinverse maps the attitude error to joint angle errors, so that the robotic arm can be controlled to cancel the attitude error.<sup>13</sup> To do this, the first step is to find the differential kinematics relation between the robotic arm and the multicopter attitude by rewriting Eq. (32) as

$$\dot{\Phi}_e = \mathbf{Q}_e^{-1} \mathbf{J}_{oe} \begin{bmatrix} \dot{\Phi} \\ \dot{\boldsymbol{\alpha}} \end{bmatrix}. \quad \text{Since the end-effector does not move during interaction, the attitude and the}$$

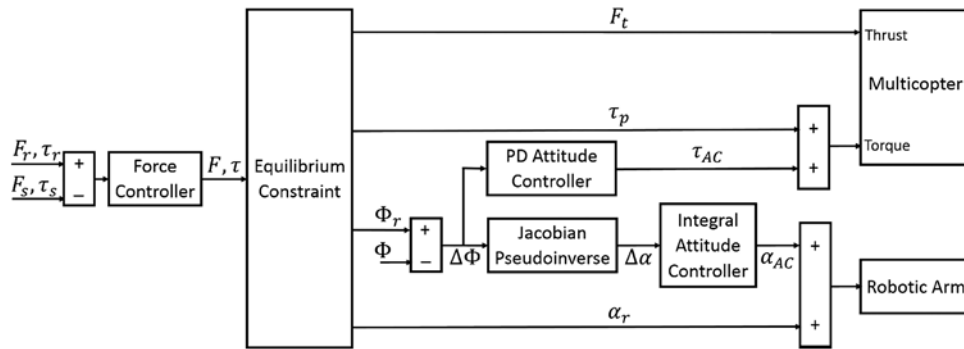


Fig. 8. Equilibrium-based force and torque controller.

robotic arm joint angles are related by

$$\mathbf{J}_{oe} \begin{bmatrix} \dot{\Phi} \\ \dot{\alpha} \end{bmatrix} = 0. \quad (38)$$

For any aerial manipulator, Eq. (38) can be rearranged to find a matrix equation that calculates  $\dot{\alpha}$  in terms of  $\dot{\Phi}$ . For redundant robotic arms, the pseudoinverse is used to choose one solution. This is illustrated by the aerial manipulator used in this research. Substituting its matrix  $\mathbf{J}_{oe}$  into Eq. (38) gives

$$\begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{\Phi} \\ \dot{\alpha} \end{bmatrix} = 0,$$

$$\text{then } \dot{\Phi} = \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = - \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix} \dot{\alpha}.$$

This equation describes how the attitude of the multicopter is affected by the robotic arm. In this case, the angle  $\psi$  does not change while the robot moves. Since the last row is unnecessary, it is removed to eliminate the singularity and then compute the pseudoinverse as follows:<sup>13</sup>

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \end{bmatrix} = - \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix} \dot{\alpha} = \mathbf{J}_{\Phi\alpha} \dot{\alpha}, \quad (39)$$

$$\dot{\alpha} = \mathbf{J}_{\Phi\alpha}^T (\mathbf{J}_{\Phi\alpha} \mathbf{J}_{\Phi\alpha}^T)^{-1} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \end{bmatrix} = - \begin{bmatrix} 1 & 0 \\ 0 & 0.5 \\ 0 & 0.5 \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \end{bmatrix}. \quad (40)$$

Finally, the column correspondent to  $\psi$  is included to result in

$$\dot{\alpha} = - \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0.5 & 0 \\ 0 & 0.5 & 0 \end{bmatrix} \dot{\Phi}, \text{ then } \Delta\alpha \approx - \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0.5 & 0 \\ 0 & 0.5 & 0 \end{bmatrix} \Delta\Phi.$$

In Fig. 8, a force and torque sensor feedback is added to correct modeling errors and external disturbances. It is especially important if high accuracy is required for a manipulation task. The force controller is implemented through a PI control given by

$$\begin{bmatrix} \mathbf{F} \\ \boldsymbol{\tau} \end{bmatrix} = \mathbf{K}_{\tau p} \left( \begin{bmatrix} \mathbf{F}_r \\ \boldsymbol{\tau}_r \end{bmatrix} - \begin{bmatrix} \mathbf{F}_s \\ \boldsymbol{\tau}_s \end{bmatrix} \right) + \mathbf{K}_{\tau i} \int \left( \begin{bmatrix} \mathbf{F}_r \\ \boldsymbol{\tau}_r \end{bmatrix} - \begin{bmatrix} \mathbf{F}_s \\ \boldsymbol{\tau}_s \end{bmatrix} \right) dt,$$

where the gains  $\mathbf{K}_{\tau p}$  and  $\mathbf{K}_{\tau i}$  are positive definite diagonal matrices.<sup>13</sup>

The proposed controller is a PID-based control combined with an equilibrium constraint calculation. Hence, there is no significant computational burden or delay for implementing this controller.

Table I. Parameters of the hexacopter, robotic arm, and control gain.

	Parameter	Value
Hexacopter	$p_{0b}$ (m)	$[-0.006 \ 0 \ 0.054]^T$
	$m_0$ (kg)	1.189
	$I_0$ (kg·m <sup>2</sup> )	$\begin{bmatrix} 0.027 & 0 & 0 \\ 0 & 0.027 & 0 \\ 0 & 0 & 0.051 \end{bmatrix}$
	$p_{01}$ (m)	$[0.054 \ 0 \ 0.073]^T$
Robotic arm	$l_1$ (m)	0.070
	$l_2$ (m)	0.106
	$l_3$ (m)	0.118
	$m_1$ (kg)	0.075
	$m_2$ (kg)	0.097
	$m_3$ (kg)	0.102
	$l_{cm1}$ (m)	0.044
	$l_{cm2}$ (m)	0.074
	$l_{cm3}$ (m)	0.070
	$I_{y,z1}$ (kg·m <sup>2</sup> )	$0.602 \times 10^{-4}$
	$I_{y,z2}$ (kg·m <sup>2</sup> )	$1.140 \times 10^{-4}$
	$I_{y,z3}$ (kg·m <sup>2</sup> )	$1.376 \times 10^{-4}$
Control gain	$\omega_{servo}$ (rad/s)	0.275
	$\mathbf{K}_{pw}$	$\begin{bmatrix} 1000\mathbb{I}_3 \text{ (N/m)} & \mathbb{O}_{3 \times 3} \\ \mathbb{O}_{3 \times 3} & 6\mathbb{I}_3 \text{ (N·m/rad)} \end{bmatrix}$
	$\mathbf{K}_{dw}$	$\begin{bmatrix} 10\mathbb{I}_3 \text{ (Ns/m)} & \mathbb{O}_{3 \times 3} \\ \mathbb{O}_{3 \times 3} & 5\mathbb{I}_3 \text{ (N·m·s/rad)} \end{bmatrix}$
	$\mathbf{K}_{\Phi p}$ (N·m/rad)	$0.9\mathbb{I}_3$
	$\mathbf{K}_{\Phi d}$ (N·m·s/rad)	$0.27\mathbb{I}_3$
	$\mathbf{K}_{\Phi i}$	$0.5\mathbb{I}_3$
	$\mathbf{K}_{\tau p}$	$\begin{bmatrix} 0.5\mathbb{I}_3 \text{ (N/m)} & \mathbb{O}_{3 \times 3} \\ \mathbb{O}_{3 \times 3} & 0.5\mathbb{I}_3 \text{ (N·m/rad)} \end{bmatrix}$
	$\mathbf{K}_{\tau i}$	$\mathbb{I}_6$

This is beneficial to real-time implementation with the many state variables for the multicopter platform and the joints of the manipulator while some optimization-based methods such as MPC would have the computational complexity issue.

#### 4. Simulation Results

In this section, the numerical simulator for the aerial manipulator and simulation results are presented. Initially, from the dynamic model, the system is written in terms of state-space equations. The dynamic model of the aerial manipulator is derived from the partial derivatives of the kinetic and potential energies. For the simulator, these partial derivatives are calculated in a recursive manner, with emphasis on computing efficiency. The parameters of the robotic arm internal controller are not known. For this reason, it is explained how to simulate the arm based on its kinematics. The interactive forces between the robotic arm and a wall are included in the simulation based on a damped spring model. Finally, some simulation cases addressing wall interaction are analyzed. Relevant system parameters and control gains are summarized in Table I.

Let the state variable vector be defined as

$$\mathbf{X} = \begin{bmatrix} \mathbf{X}_1 \\ \mathbf{X}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{q} \\ \dot{\mathbf{q}} \end{bmatrix}. \quad (41)$$

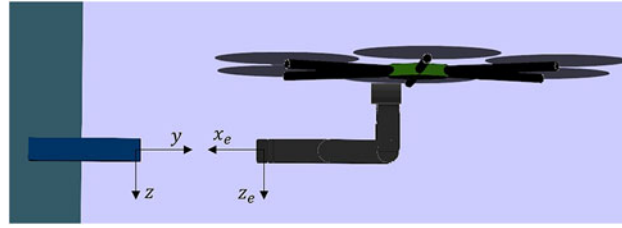


Fig. 9. Inertial frame and end-effector frame for the force and torque simulations.

Then, the system in the state space is written as  $\dot{\mathbf{X}} = \begin{bmatrix} \mathbf{X}_2 \\ \boldsymbol{\sigma} \end{bmatrix}$ , where  $\boldsymbol{\sigma} = [\sigma_p \ \sigma_\phi \ \sigma_\alpha]^T$  is determined in terms of the state variables and the inputs, based on the dynamic equation

$$\begin{bmatrix} B_{pp} & B_{p\phi} & B_{p\alpha} \\ B_{\phi p} & B_{\phi\phi} & B_{\phi\alpha} \\ B_{\alpha p} & B_{\alpha\phi} & B_{\alpha\alpha} \end{bmatrix} \begin{bmatrix} \sigma_p \\ \sigma_\phi \\ \sigma_\alpha \end{bmatrix} + \mathbf{C}\mathbf{X}_2 + \mathbf{G} \\ = \begin{bmatrix} \mathbf{R}_{i0} & 0 & 0 \\ 0 & \mathbf{Q}^T & 0 \\ 0 & 0 & \mathbb{I}_n \end{bmatrix} \begin{bmatrix} \mathbf{F}_T \\ \boldsymbol{\tau}_p \\ \boldsymbol{\tau}_\alpha \end{bmatrix} + \boldsymbol{\tau}_{\text{ext}}. \quad (42)$$

The wall interaction is modeled as a damped spring system. When the end-effector is not in contact with the wall, the interaction forces and torques are set to be zero.<sup>11</sup> When the end-effector touches the wall and holds the object, the interaction forces and torques are included in the simulation. A threshold is defined relative to the wall position to designate the contact transition given by

$$\boldsymbol{\tau}_{\text{ext}} = \begin{cases} 0, & x_e < x_w - \Delta x_{\text{max}} \\ \mathbf{J}^T \begin{bmatrix} \mathbf{F}_e \\ \boldsymbol{\tau}_e \end{bmatrix}, & x_e \geq x_w - \Delta x_{\text{max}} \end{cases}$$

where  $x_w$  is the position of the plane that determines the wall in  $x$  axis,  $x_e$  is the end-effector's  $x$  coordinate, and  $\Delta x_{\text{max}}$  is the maximum wall deformation resulted from the interaction forces during manipulation task, respectively. The interaction force  $\mathbf{F}_e$  and torque  $\boldsymbol{\tau}_e$  are from Eq. (21).

The simulations are performed to evaluate the equilibrium-based force and torque controller. The experimental setup is reproduced in the simulation environment. The reference force and torque vector is given by

$$\boldsymbol{\tau}_r = [F_{rx} \ F_{ry} \ F_{rz} \ \tau_{rx} \ \tau_{ry} \ \tau_{rz}]^T.$$

In the first simulation cases, a force or a torque is applied in only one direction, to study each component individually. In this paper, a part (lateral and longitudinal forces and roll and pitch torques) of the simulation cases is presented. Another simulation case is implemented where components of forces and torques are applied simultaneously.

The simulation configuration is illustrated in Fig. 9. The torques are expressed in terms of the end-effector frame. The external interaction forces are expressed with respect to the inertial frame. The vertical surface belongs to a plane orthogonal to the  $y$  axis.

#### 4.1. Force interaction

The lateral force of 1 N is implemented by setting the reference force and torque  $\boldsymbol{\tau}_r = [1 \ 0 \ 0 \ 0 \ 0 \ 0]^T$ . The aerial manipulator applies a lateral force on the object on the wall by rolling the multicopter body. The lateral force applied by the thrusters on the multicopter body is transmitted through the robotic arm to the end-effector, which applies a lateral force on the object. In Fig. 10, the forces and torques for the lateral force simulation are plotted. The aerial manipulator successfully applies the 1 N force along the  $x$  axis. The vertical interaction torque  $\tau_z$  is not canceled and the other interaction forces and torques converge to zero. When the multicopter rolls to apply the lateral force, the aerial manipulator behaves as a wrench. A vertical interaction torque is consequently applied on

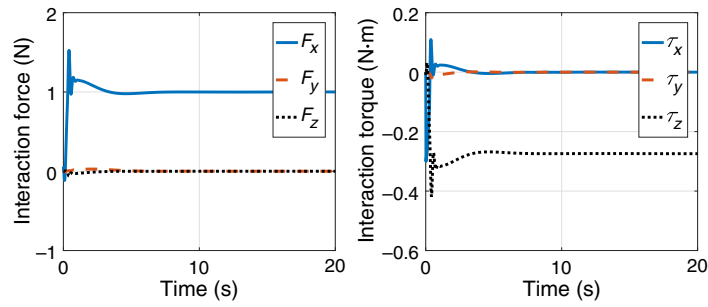


Fig. 10. Forces and torques for the lateral force case.

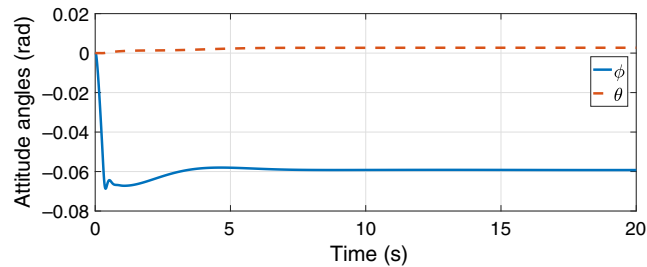


Fig. 11. Attitude of the multicopter for the lateral force case.

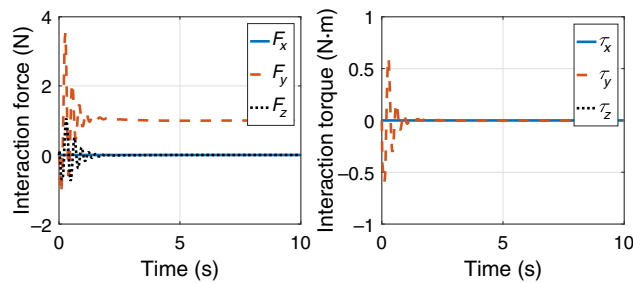


Fig. 12. Interaction forces and torque for the longitudinal force case.

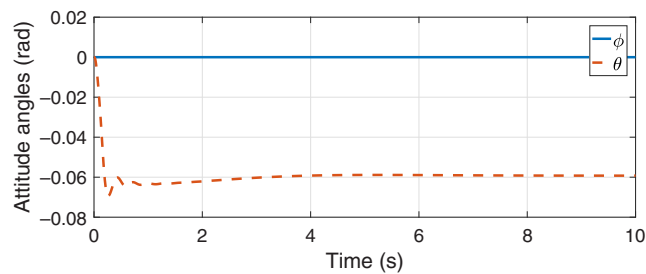


Fig. 13. Attitude angles for the longitudinal force case.

the object. In Fig. 11, the attitude roll angle where the multicopter rotates to apply the lateral force is shown.

The longitudinal force of 1 N is implemented by setting the reference force and torque  $\tau_r = [0 \ 1 \ 0 \ 0 \ 0 \ 0]^T$ . In this case, the multicopter performs a pitch movement, the thrust force is projected toward the wall, and the aerial manipulator pushes the object on the wall. In Fig. 12, the interaction forces and torques are plotted. The force  $F_y$  converges to 1 N, while the other components converge to zero, as expected. The pitch angle by which the multicopter rotates to apply the desired force is shown in Fig. 13.



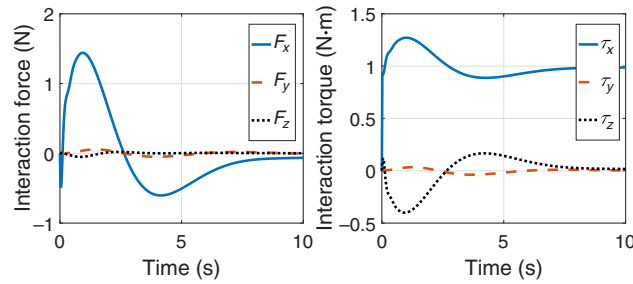


Fig. 14. Forces and torques for the roll torque interaction case.

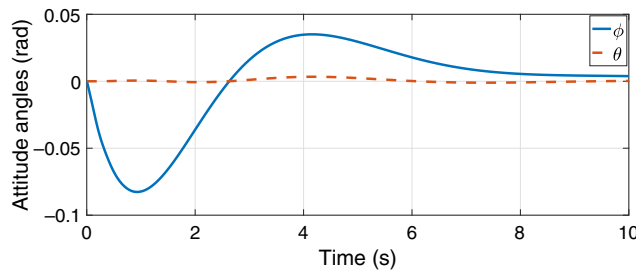


Fig. 15. Attitude angles for the roll torque interaction case.

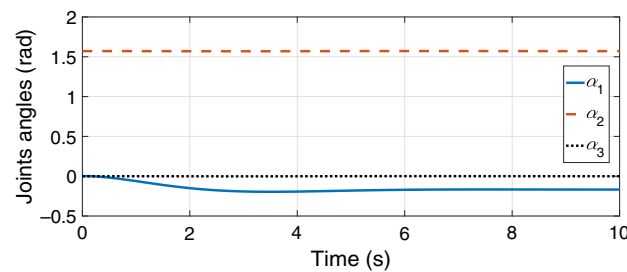


Fig. 16. Robotic arm angles for the roll torque interaction case.

#### 4.2. Torque interaction

The reference force and torque vector for the desired roll torque of 1 N·m is given by  $\tau_r = [0 \ 0 \ 0 \ 1 \ 0 \ 0]^T$ . To apply a torque on the object on the wall, the aerial manipulator applies a roll torque on the multicopter body. This torque is induced on the object through the robotic arm. In Fig. 14, the forces and torques for the roll torque interaction are shown. Since the wall external forces and torques are modeled as a damped spring system, the aerial manipulator rotates in compliance with the spring when the roll torque is applied on the object. After the multicopter rotates, the aerial manipulator interaction controller corrects the attitude roll angle, as shown in Fig. 15. The first joint of the robotic arm rotates to compensate the attitude error, as shown in Fig. 16. The interaction controller ability to correct the attitude angles is important since the experimental robotic arm is not ideally rigid.

In order to study the pitch torque interaction, the reference vector for the desired pitch torque of 1 N·m is set to  $\tau_r = [0 \ 0 \ 0 \ 0 \ 1 \ 0]^T$ . The aerial manipulator interaction controller applies a pitch torque on the multicopter body. This torque is induced on the object through the robotic arm. In Fig. 17, the interaction forces and torques are plotted and the attitude angles are shown in Fig. 18. The robotic arm corrects the attitude angles by rotating the second and third joints, as plotted in Fig. 19. While the attitude angles are corrected, the pitch torque converges to 1 N·m and the other forces and torques converge to zero, as expected.

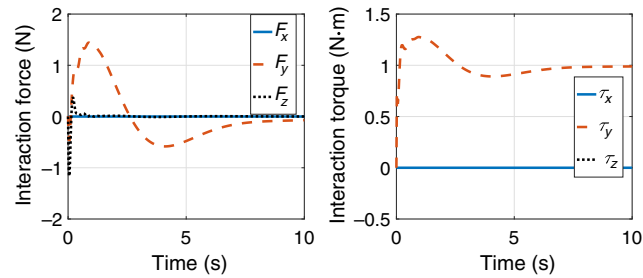


Fig. 17. Forces and torques for the pitch torque interaction case.

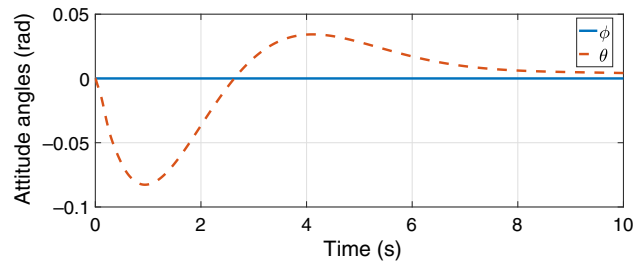


Fig. 18. Attitude angles for the pitch torque interaction case.

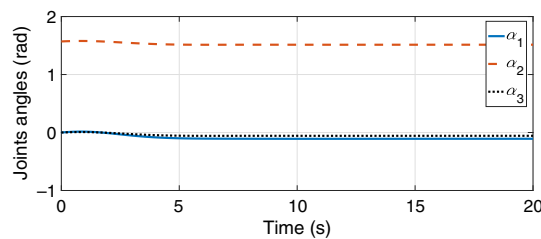


Fig. 19. Joints angles for the pitch torque case.

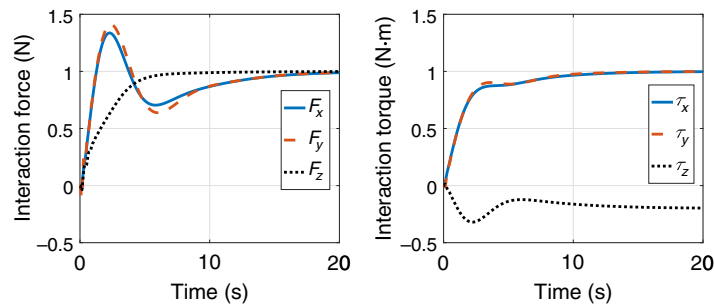


Fig. 20. Forces and torques for the force and torque combined interaction case.

#### 4.3. Force and torque combined interaction

In this simulation case, the closed-loop controller is tested for a condition where torques and forces are applied simultaneously. The reference interaction vector is set to  $\tau_r = [1 \ 1 \ 1 \ 1 \ 1 \ -0.2]^T$ . The yaw torque value is chosen to avoid PWM saturation. All the forces and torques converge to their reference levels, as shown in Fig. 20.

### 5. Experimental Results

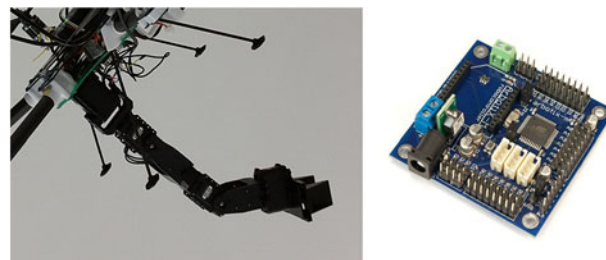
In this section, the equipment, laboratory facilities, and experiments to validate the proposed controller and simulation results are presented. A hexacopter with a microcontroller Pixhawk is described, as well as a three-DoF robotic arm and a motion capture system. It is explained how those items are integrated to build the experimental setup and how a spring-mounted device was used for

Table II. Parts of the hexacopter.

Part	Description	Weight (g)
Electric motor	T-Motor KV 750 MT2212	55
Propeller E-prop	$254 \times 120 \text{ mm}^2$	12
LiPo battery and power regulator	Thunder power 1800 mAh	269
Structure weight	–	350
Total weight	–	1189



Fig. 21. Hexacopter.

Fig. 22. Robotic arm (left) mounted on the hexacopter and Arbotix-M controller<sup>24</sup> (right).

the interaction force experiments. Finally, the experiments for the validation of the equilibrium-based force and torque controller are presented.

### 5.1. Experimental setup

The experiments were conducted with a hexacopter fabricated at the Naval Postgraduate School,<sup>20</sup> shown in Fig. 21. It is a multicopter with six motors/propellers and a primary power source provided through a tether by a ground power supply. A secondary power source is provided by a backup LiPo battery. The hexacopter is equipped with T-Motor KV750 electric motors and E-prop carbon propellers. The parts of the hexacopter are summarized in Table II.

The hexacopter is equipped with a microcontroller Pixhawk. It is an open-architecture microcontroller provided with accelerometer, gyroscope, magnetometer, and barometric altimeter sensors. The microcontroller processor has a 168 MHz ARM Cortex M4F CPU. There are some options of open-source firmware provided by a large community of users and developers. In this research, a Pixhawk Support Package with MATLAB Embedded Coder developed by Mathworks<sup>23</sup> is used. This tool generates and compiles C code from a Simulink file and uploads the program to Pixhawk.

The aerial manipulator is equipped with a three-DoF robotic arm mounted on the bottom of the multicopter. The robotic arm has four Dynamixel AX-12A servos. Three of them rotate to control the revolute joint angles, and the last one opens and closes the end-effector. The configuration of the robotic arm is illustrated in Fig. 3. The robotic arm shown in Fig. 22 is controlled by an Arbotix-M board, which is an Arduino-compatible device. Each Dynamixel servo has a bank of registers that are either read or written by the Arbotix-M board through a serial port. The bank of registers record the parameters associated with the motion of the servo. By writing into the registers, the controller

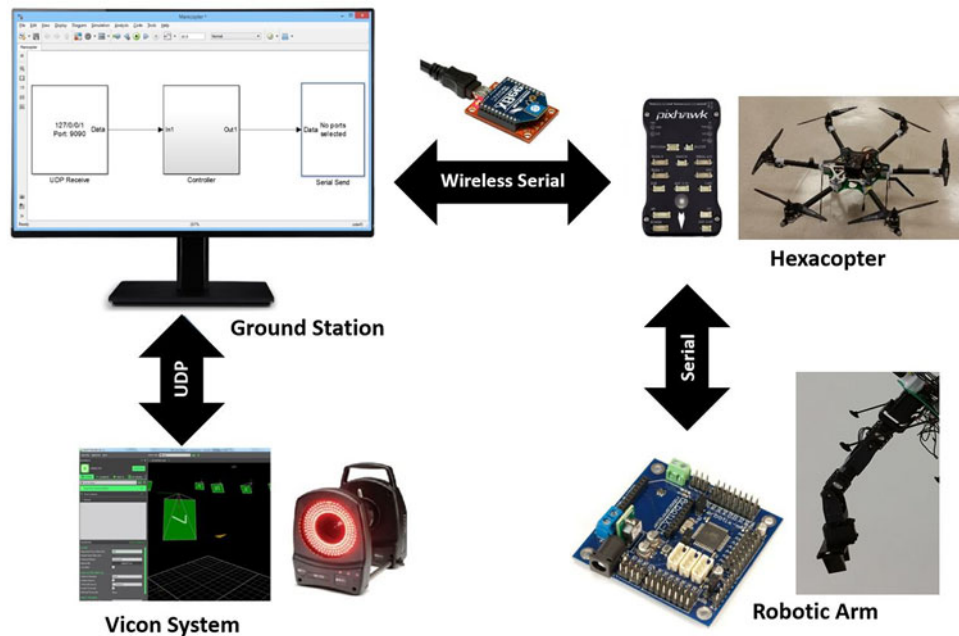


Fig. 23. Experimental setup.

defines parameters like the servo goal position and desired moving speed. By reading the registers, the controller obtains information related to the status of the servo, such as position and velocity feedback. Another serial port connects the Arbotix-M board to the multicopter microcontroller. The Arbotix-M board is programmed to pass commands from the multicopter to the robotic arm and also send feedback from the robotic arm to the multicopter. Each servo has an internal controller to convert the motion commands into torques.

The laboratory flight testbed equipped with a Vicom motion capture system is surrounded by ten cameras able to detect infrared reflective markers fixed on the multicopter and robotic arm. The software Tracker, installed in the Vicom server, computes the position of each marker by triangulation. Once the position of each marker is known, the software computes the position and the orientation of the multicopter.

The experimental setup is illustrated in Fig. 23. The Vicom server and a ground station are connected to the same network. A MATLAB script running in the Vicom server transmits the position and the orientation of the multicopter, through user datagram protocol (UDP) connection, to the ground station. The ground station and the microcontroller communicate through serial ports with XBee antennas. A serial cable connects the microcontroller Pixhawk to the robot controller Arbotix-M.

Some experiments to validate the equilibrium-based force and torque controller were executed with a spring-mounted device on a wall. As shown in Fig. 24, the end-effector holds an object that slides along a rail. By pushing the sliding object, the robotic arm compresses the spring toward the fixed object on the left side. The reflective markers are detected by the Vicom cameras and the position of the sliding device is sent through UDP communication to the ground station. The compression of the spring is directly related to the position of the sliding object. The force applied on the spring is obtained by multiplying the spring constant by the spring compression. With this experimental setup, the ground station controller has feedback of the force applied on the spring by the aerial manipulator.

To validate the equilibrium-based force and torque controller, an experiment was designed with the spring system to prove that the controller is able to regulate a specified force on the object. Other experiments were performed without force or torque feedback to validate the controller when the aerial manipulator is required to apply a specified torque. In these experiment cases, our interest is the performance of the proposed controller to generate desired forces after the end-effector holds the object near the wall. The initial flight status of the experiments were thus conducted by progressively increasing the thrust force until the multicopter could fly without the suspension line while the end-effector grasped the sliding object. The multicopter then started generating forces and torques from its initial flight mode. In Fig. 25, the aerial manipulator flies while applying a longitudinal force toward the wall.

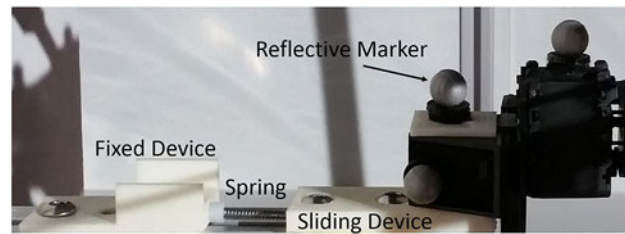


Fig. 24. Spring-mounted device for interaction force experiments.

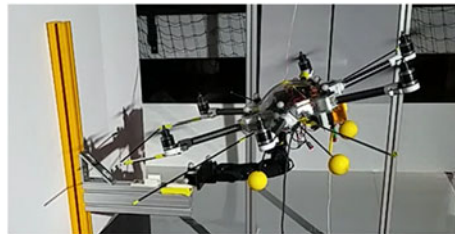


Fig. 25. Aerial manipulator force and torque interaction experiment.

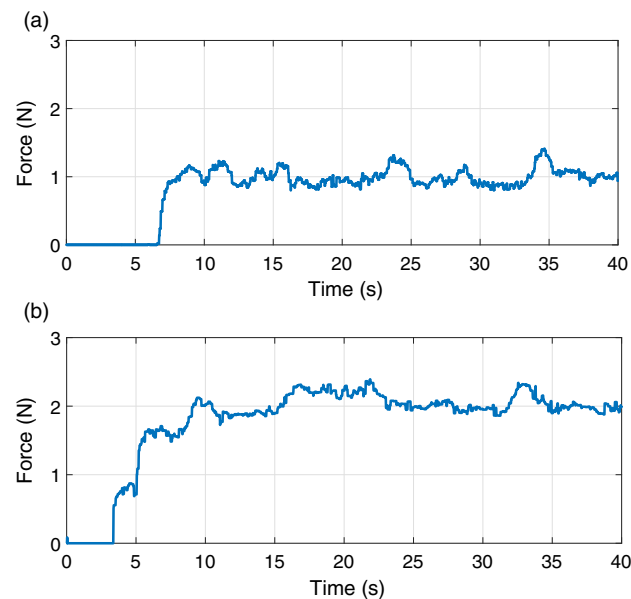


Fig. 26. Interaction force experiment for reference forces. (a) Reference force  $F_y = 1$  N. (b) Reference force  $F_y = 2$  N.

First, experiments were performed to study the force interaction. The aerial manipulator objective during the experiments with the spring-mounted device was to control the force applied on the spring. In Fig. 26(a) and (b), the forces applied on the spring for two different reference forces are plotted. In all the experiments, the equilibrium-based force and torque controller successfully regulated the forces on the spring. Friction was observed between the sliding object and the rail when the sliding object moved on the surface of the rail. In some experiments, the aerial manipulator needed more time to converge and maintain the desired force, but the convergence was always achieved. Controlling a spring force in only one direction may seem easier than controlling simultaneously a six-dimensional force torque vector with a force and torque sensor; however, the spring device was mounted on a rail with some friction.

In Fig. 27, the attitude pitch angle is plotted for the same experiment in Fig. 26(b). The projection of the thrust force on the horizontal plane toward the wall is determined by the pitch angle. For a negative pitch angle, the multicopter moves the nose down and the aerial manipulator pushes the

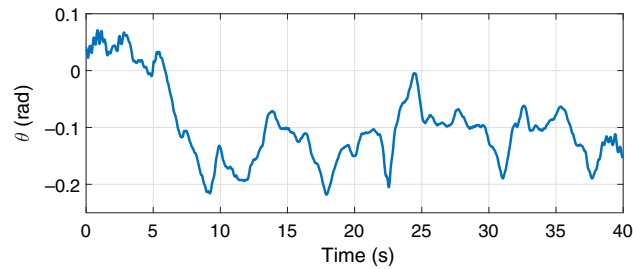


Fig. 27. Hexacopter pitch angle for the experiment with reference  $F_y = 2$  N.

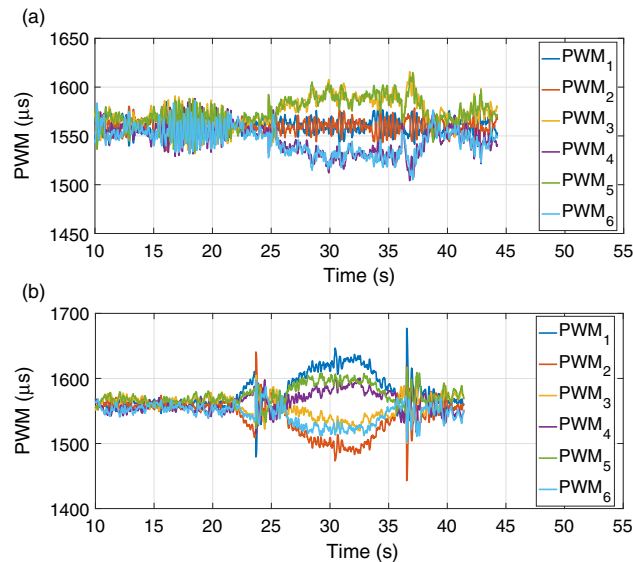


Fig. 28. PWM history for interaction torque experiments with reference torques. (a) Reference torque  $\tau_y = 0.5$  N·m. (b) Reference torque  $\tau_x = 1$  N·m.

wall. By comparing the force plot with the attitude plot, the correlation between the attitude and the interaction force is perceived.

Additional experiments were performed to validate the equilibrium-based force and torque controller for torque interactions. Although the experiments are not configured with any kind of torque feedback, the recorded data could be analyzed to prove that the aerial manipulator is capable of applying torques. In Fig. 28, experiments for pitch torque and roll torque are shown. In both experiments, the reference torque are set to be zero in the first 20 s approximately. After that, the reference torque is modified. When the reference torque changes, the PWM inputs also change, indicating that an additional torque is applied by the rotors and transmitted to the object through the robotic arm. In the plot in Fig. 28(b), the multicopter oscillates during the transition and the PWM inputs are also modified by the attitude controller to damp those oscillations.

## 6. Conclusion

In this paper, the feasibility of employing a multicopter UAV equipped with a robotic arm for manipulation tasks on vertical surfaces was studied. The main objective of this research was to develop a force and torque controller for wall interaction. A mathematical model for a multicopter with an  $n$ -link robotic arm was derived through Lagrangian formalism. The mathematical model was applied to the experimental hexacopter equipped with a three DoF robotic arm. In an aerial manipulator, the dynamics of the robotic arm and the multicopter are coupled. For this reason, an aerial manipulator controller offers a better performance if the system is modeled and controlled as a whole.

An equilibrium-based force and torque controller was developed to regulate the forces and torques that the aerial manipulator applies on an object fixed on a wall. The equilibrium-based force and torque controller computes the multicopter attitude and propeller inputs to induce the desired force



and torque on the wall. Through simulations, the equilibrium-based force and torque controller successfully regulated torques and forces. Simulations also validated the controller for a manipulation task with forces and torques applied simultaneously in all directions. The experiments were carried out with a spring-mounted device near a wall. The spring-mounted device was designed to validate the use of the developed force and torque controller for force regulation. The experimental results demonstrate that the force and torque controller effectively regulates the force applied on the spring while manipulating near the wall.

In this research, we neglected the propeller rigid body dynamics and any other aerodynamic effects by considering slow-moving maneuvers of the multicopter. However, in practice, aerodynamic forces and other modeling uncertainties, such as the near-wall effect, may affect the interaction forces. The propeller rigid body dynamics would be significant when the multicopter rotates fast about any of the body-frame horizontal axes. In the future work, we will consider to include the propeller rigid body dynamics and the near-wall effect analysis. In addition, comparison with other existing results will be implemented. The proposed controller will be applied to other potential applications including cleaning, painting, and rescue missions.

### Acknowledgments

The authors sincerely thank Dr. Kevin Jones at the US Naval Postgraduate School for developing the hexacopter and experimental setup reflected in this paper.

### References

1. A. Jimenez-Cano, J. Martin, G. Heredia, A. Ollero and R. Cano, "Control of an Aerial Robot with Multi-link Arm for Assembly Tasks," *IEEE International Conference on Robotics and Automation (ICRA)*, Karlsruhe, Germany (2013) pp. 4916–4921.
2. 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," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Chicago, Illinois, USA (2014) pp. 3417–3422.
3. S. Kim, S. Choi and H. J. Kim, "Aerial Manipulation Using a Quadrotor with a Two DoF Robotic Arm," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Tokyo, Japan (2013) pp. 4990–4995.
4. M. Orsag, C. Korpela, S. Bogdan and P. Oh, "Valve Turning Using a Dual-Arm Aerial Manipulator," *International Conference on Unmanned Aircraft Systems (ICUAS)*, Orlando, FL, USA (2014) pp. 836–841.
5. K. Kondak, A. Ollero, I. Maza, K. Krieger, A. Albu-Schaeffer, M. Schwarzbach and M. Laiacker, "Unmanned Aerial Systems Physically Interacting with the Environment: Load Transportation, Deployment, and Aerial Manipulation," In: *Handbook of Unmanned Aerial Vehicles* (Springer, Dordrecht, Netherlands, 2015) pp. 2755–2785.
6. K. Alexis, G. Darivianakis, M. Burri and R. Siegwart, "Aerial robotic contact-based inspection: planning and control," *Auton. Rob.* **40**(4), 631–655 (2016).
7. K. Alexis, C. Huerzeler and R. Siegwart, "Hybrid Modeling and Control of a Coaxial Unmanned Rotorcraft Interacting with its Environment Through Contact," *IEEE International Conference on Robotics and Automation (ICRA)*, Karlsruhe, Germany (2013) pp. 5417–5424.
8. G. Darivianakis, K. Alexis, M. Burri and R. Siegwart, "Hybrid Predictive Control for Aerial Robotic Physical Interaction Towards Inspection Operations," *IEEE International Conference on Robotics and Automation (ICRA)*, Hong Kong (2014) pp. 53–58.
9. C. Papachristos, K. Alexis and A. Tzes, "Efficient Force Exertion for Aerial Robotic Manipulation: Exploiting the Thrust-Vectoring Authority of a Tri-tiltrotor UAV," *IEEE International Conference on Robotics and Automation (ICRA)*, Hong Kong (2014) pp. 4500–4505.
10. M. Fumagalli, R. Naldi, A. Macchelli, R. Carloni, S. Stramigioli and L. Marconi, "Modeling and Control of a Flying Robot for Contact Inspection," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Algarve, Portugal (2012) pp. 3532–3537.
11. J. Scholten, M. Fumagalli, S. Stramigioli and R. Carloni, "Interaction Control of an UAV Endowed with a Manipulator," *2013 IEEE International Conference on Robotics and Automation (ICRA)*, Hong Kong (IEEE, 2013) pp. 4910–4915.
12. M. Fumagalli and R. Carloni, "A Modified Impedance Control for Physical Interaction of UAVs," *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Tokyo, Japan (IEEE, 2013) pp. 1979–1984.
13. B. Tavora, Feasibility Study of an Aerial Manipulator Interacting with a Vertical Wall, *Ph.D. Thesis* (Naval Postgraduate School, 2017).
14. H. Lee, S. Kim and H. J. Kim, "Control of an Aerial Manipulator Using On-line Parameter Estimator for an Unknown Payload," *2015 IEEE International Conference on Automation Science and Engineering (CASE)*, Gothenburg, Sweden (IEEE, 2015) pp. 316–321.
15. B. Siciliano and O. Khatib, *Springer Handbook of Robotics* (Springer, Heidelberg, Germany, 2016).

16. B. Siciliano, L. Sciavicco, L. Villani and G. Oriolo, *Robotics: Modelling, Planning and Control* (Springer Science & Business Media, London, UK, 2010).
17. B. Wie, *Space Vehicle Dynamics and Control* (AIAA, Reston, VA, USA, 1998).
18. F. Forte, R. Naldi, A. Macchelli and L. Marconi, "Impedance Control of an Aerial Manipulator," *American Control Conference (ACC)*, Montréal (IEEE, 2012) pp. 3839–3844.
19. V. Lippiello and F. Ruggiero, "Cartesian impedance control of a UAV with a robotic arm," *IFAC Proc. Vol.* **45**(22), 704–709 (2012).
20. E. Capello, H. Park, B. Tavora, G. Guglieri and M. Romano, "Modeling and Experimental Parameter Identification of a Multicopter via a Compound Pendulum Test Rig," *Workshop on Research, Education, and Development of Unmanned Aerial Systems (RED-UAS)* (2015) pp. 308–317.
21. K. Baizid, G. Giglio, F. Pierri, M. A. Trujillo, G. Antonelli, F. Caccavale, A. Viguria, S. Chiaverini and A. Ollero, "Behavioral control of unmanned aerial vehicle manipulator systems," *Auton. Rob.* **41**(5), 1203–1220 (2017).
22. V. Lippiello and F. Ruggiero, "Exploiting Redundancy in Cartesian Impedance Control of UAVs Equipped with a Robotic Arm," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Algarve, Portugal (2012) pp. 3768–3773.
23. Pilot Engineering Group, Pixhawk pilot support package user guide, MathWorks (2015).
24. M. Ferguson, "ARBOTIX ROBOCONTROLLER-A powerful new 3rd party controller for Robotis Bioloid AX-12+ Dynamixel servos!," *Robot-Congers* **20**, 64 (2009).