

# Modeling and Control of an Aerial Manipulator from the Perspective of its End-effector

Júnio Eduardo de Morais\* Daniel N. Cardoso\*  
Guilherme V. Raffo\*,\*\*

\* Graduate Program in Electrical Engineering, Federal University of  
Minas Gerais, MG, (e-mail: jeduardo@ufmg.br).

\*\* Department of Electronic Engineering, Federal University of Minas  
Gerais, MG (e-mail: raffo@ufmg.br)

---

**Abstract:** This work presents the dynamic model of an unmanned aerial manipulator (UAM) obtained from the perspective of the end-effector. This perspective allows designing single-layer whole-body based controllers to directly perform trajectory tracking of the pose of the end-effector. A robust linear mixed  $\mathcal{H}_2/\mathcal{H}_\infty$  controller with  $\mathcal{D}$ -stability constraints is designed and implemented in a hardware-in-the-loop framework. Numerical experiments are conducted in a high fidelity simulator to evaluate the controller's performance when the system is subjected disturbances.

*Keywords:* Unmanned aerial manipulator, Lagrangian systems, Underactuated mechanical systems, Robust control, Mixed  $\mathcal{H}_2/\mathcal{H}_\infty$  control.

---

## 1. INTRODUCTION

An Unmanned Aerial Manipulator (UAM) is a kind of mobile manipulator composed of an Unmanned Aerial Vehicle (UAV) coupled with one or more manipulator arms. These systems can encompass a wide workspace and possess the capability of interacting with the environment using the end-effector. Such features are useful for the execution of tasks like manipulation, load transportation, remote sensing, among others.

The kinematic and dynamical models of UAMs are commonly obtained from the perspective of the UAV and the control performed using two separate controllers, in which one is designed for the UAV and other for the manipulator arm, as can be seen, for example, in Heredia et al. (2014); Chaikalis et al. (2020); Chen et al. (2020); Bouzgou et al. (2020); Acosta et al. (2020); Nava et al. (2020). These separate controllers are designed based on simplified models that neglect the coupling effects between the UAV and the manipulator arm, with these effects being considered as external disturbances, which degrades the controller's performance. A solution to this problem is the design of whole-body based control laws, such as in Mello et al. (2015, 2016); Morais et al. (2020). However, the whole-body controllers designed based on models obtained from the perspective of the UAV can only set references to the pose of the UAV and the joints of the manipulator arm.

---

\* This work was in part supported by the project INCT (National Institute of Science and Technology) under the grant CNPq (Brazilian National Research Council) 465755/2014-3, FAPESP (São Paulo Research Foundation), Brazil 2014/50851-0. This work was also supported in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), Brazil (Finance Code 88887.136349/2017-00, and 001), CNPq, Brazil (grant numbers 313568/2017-0 and 426392/2016-7), and FAPEMIG, Brazil (grant number APQ-03090-17).

Accordingly, if it is intended to set references to the pose of the end-effector, an outer layer, such as a kinematic controller, must be implemented in a hierarchical control structure.

To avoid this shortcoming, and aiming to perform a robust trajectory tracking of the pose of the end-effector, this work derives the kinematic model and conducts the whole-body modeling of a UAM from the perspective of the end-effector. The dynamic model is obtained using the Euler-Lagrange formalism and, based on this model, a single layer whole-body based controller is designed through the robust linear mixed  $\mathcal{H}_2/\mathcal{H}_\infty$  control strategy considering  $\mathcal{D}$ -stability constraints. The proposed controller is implemented in an embedded computational platform using C++ programming language, and numerical experiments are conducted to evaluate its performance in a Hardware-In-the-Loop (HIL) framework using the ProVANT Simulator, which is an open-source software developed at the Federal University of Minas Gerais (UFMG) based on Robot Operating System (ROS) and Gazebo (Koenig and Howard, 2004), that provides high accuracy simulations with visual feedback<sup>1</sup>.

Summarizing, the main contributions of this paper are: (i) the kinematic and the whole-body model of a UAM, obtained from the perspective of the end-effector; (ii) the design of a single layer whole-body based controller through the linear mixed  $\mathcal{H}_2/\mathcal{H}_\infty$  control strategy with  $\mathcal{D}$ -stability constraints for a robust trajectory tracking of the pose of the end-effector of a UAM; and (iii) the implementation of this controller in an embedded system with validation by means of HIL experiments.

---

<sup>1</sup> The ProVANT Simulator is available for download at <https://github.com/Guiraffo/ProVANT-Simulator>

## 2. DYNAMIC MODELING

This section presents the kinematic modeling and derives the equations of motion of a UAM from the perspective of its end-effector.

The UAM used in this work, shown in Figure 1, is composed of a quadrotor UAV serially coupled with a planar manipulator with three revolution joints. Figures 1 and 2 show the reference frames rigidly attached to the system to compute the kinematic model, which are: the inertial reference frame  $\mathcal{F}_I$ , the end-effector frame  $\mathcal{F}_e$ , the frame  $\mathcal{F}_q$  attached to the geometric center of the quadrotor UAV, and the frames  $\mathcal{F}_{l_i} \in \{1, 2, 3\}$  attached to the joints of the manipulator.

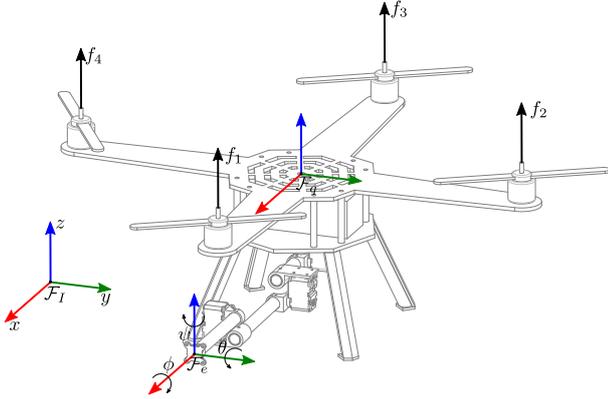


Figure 1. Reference frames rigidly attached to the UAM in order to derive the kinematic model.

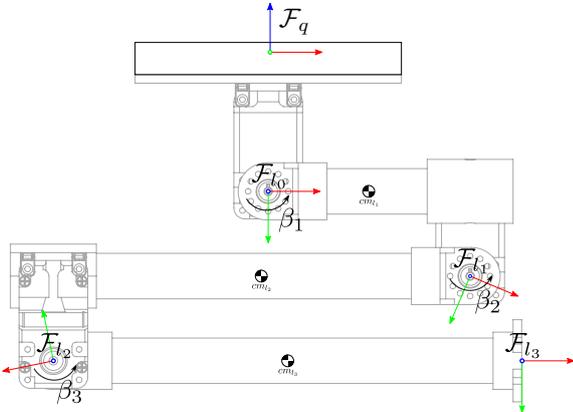


Figure 2. Reference frames attached to the planar manipulator. The z-axis of these reference frames points inward to the page.

The pose of the end-effector relative to the inertial frame,  $\mathbf{q}_e(t) = [\boldsymbol{\xi}'(t) \boldsymbol{\eta}'(t)]'$ , has six Degrees of Freedom (DOF) and is composed of its attitude  $\boldsymbol{\xi} = [\phi \ \theta \ \psi]'$ , with  $\phi$ ,  $\theta$ ,  $\psi$  being the roll, pitch and yaw Euler angles in the ZYX local axis convention, and linear position  $\boldsymbol{\eta} = [x \ y \ z]'$ . Moreover, the manipulator arm has three DOF  $\mathbf{q}_m = [\beta_1 \ \beta_2 \ \beta_3]'$ , where  $\beta_i$  denotes the angle of the  $i$ -th joint of the manipulator arm in relation with the previous joint in the kinematic chain. Therefore, the whole-body dynamic model of the UAM has nine DOF, with the generalized coordinates vector being given by  $\mathbf{q} = [\mathbf{q}'_e \ \mathbf{q}'_m]$ .

The pose of the end-effector with respect to (w.r.t.)  $\mathcal{F}_I$  is computed by the following homogeneous transformation matrix (HTM):

$$\mathbf{H}_e^I = \begin{bmatrix} \mathbf{R}_e^I & \mathbf{p}_{I,e}^I \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix}, \quad (1)$$

where  $\mathbf{R}_e^I \in SO(3)$  is the orthonormal rotation matrix,  $\mathbf{p}_{I,e}^I = \boldsymbol{\eta}$  denotes the position of the end-effector w.r.t.  $\mathcal{F}_I$ , expressed in  $\mathcal{F}_I$ , and  $\mathbf{0}_{n \times m} \in \mathbb{R}^{n \times m}$  is a matrix of zeros.

After obtaining the pose of the end-effector, the pose of the reference frames attached to manipulator joints are computed using the Denavit-Hartenberg (DH) convention (Spong et al., 2006). In the DH convention, the pose of a link w.r.t. the previous link in the kinematic chain is given by the following HTM:

$$\mathbf{H}_i^{l_{i-1}} = \begin{bmatrix} \mathbf{R}_i^{l_{i-1}} & \mathbf{p}_{l_i, l_{i-1}}^{l_{i-1}} \\ \mathbf{0} & 1 \end{bmatrix}, \quad (2)$$

with

$$\mathbf{R}_i^{l_{i-1}} = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) \cos(\alpha_i) & \sin(\theta_i) \sin(\alpha_i) \\ \sin(\theta_i) & \cos(\theta_i) \cos(\alpha_i) & -\cos(\theta_i) \sin(\alpha_i) \\ 0 & \sin(\alpha_i) & \cos(\alpha_i) \end{bmatrix},$$

$$\mathbf{p}_{l_i, l_{i-1}}^{l_{i-1}} = [a_i \cos(\theta_i) \ a_i \sin(\theta_i) \ d_i]'$$

where  $\theta_i, d_i, a_i, \alpha_i$  are the DH parameters of the  $i$ -th link. The pose of the  $i$ -th joint reference frame w.r.t.  $\mathcal{F}_I$  is then obtained as

$$\mathbf{H}_i^I = \mathbf{H}_e^I \mathbf{H}_{l_3}^e \mathbf{H}_{l_2}^{l_3} \dots \mathbf{H}_{l_{i-1}}^{l_i}, \quad (3)$$

where  $\mathbf{H}_{l_{i-1}}^{l_i} = (\mathbf{H}_{l_i}^{l_{i-1}})^{-1}$ . Finally, the pose of the quadrotor UAV w.r.t.  $\mathcal{F}_I$  is obtained by

$$\mathbf{H}_q^I = \mathbf{H}_{l_0}^I \mathbf{H}_q^{l_0}, \quad (4)$$

and

$$\mathbf{H}_{l_3}^e = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{H}_q^{l_0} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & -l_0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

### 2.1 Equations of motion

The equations of motion of the system are obtained using the Euler-Lagrange formalism in the canonical form

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \mathbf{B}(\mathbf{q})\boldsymbol{\Gamma} + \mathbf{d}, \quad (5)$$

where  $\mathbf{M}(\mathbf{q}), \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \in \mathbb{R}^{9 \times 9}$  are, respectively, the symmetric positive definite inertia matrix and the Coriolis and centripetal forces matrix,  $\mathbf{g}(\mathbf{q}), \mathbf{d} = [d_\phi \ d_\theta \ d_\psi \ d_x \ d_y \ d_z \ d_{\beta_1} \ d_{\beta_2} \ d_{\beta_3}]' \in \mathbb{R}^9$  are respectively the gravitational force vector and the disturbance vector,  $\mathbf{B}(\mathbf{q}) \in \mathbb{R}^{9 \times 7}$  is the input coupling matrix, and  $\boldsymbol{\Gamma} = [f_1 \ f_2 \ f_3 \ f_4 \ \tau_1 \ \tau_2 \ \tau_3]' \in \mathbb{R}^7$  is the control input vector where  $f_j$  denotes the force generated by the  $j$ -th quadrotor UAV propeller, and  $\tau_i$  is the torque applied to the  $i$ -th joint of the manipulator.

To obtain the inertia matrix of the system, it is necessary to obtain its linear and angular velocity Jacobians respectively defined as  $\mathbf{J}_{v_i} = \partial \mathbf{p}_{I,i}^I / \partial \dot{\mathbf{q}}$  and  $\mathbf{J}_\omega = \partial \boldsymbol{\omega}_{I,i}^I / \partial \dot{\mathbf{q}}$ . The term  $\dot{\mathbf{p}}_{I,i}^I$  is the time derivative of the position of the  $i$ -th Center of Mass (CoM), and  $\boldsymbol{\omega}_{I,i}^I$  is the angular velocity of the  $i$ -th CoM w.r.t.  $\mathcal{F}_I$ , expressed in  $\mathcal{F}_I$ , and obtained from  $\mathbf{S}(\boldsymbol{\omega}_{I,i}^I) = \dot{\mathbf{R}}_i^I \mathbf{R}_i^I$ , where  $\mathbf{S}(\cdot)$  is a skew-symmetric matrix that satisfies  $\mathbf{S}(\mathbf{v})\mathbf{w} = \mathbf{v} \times \mathbf{w} \ \forall \mathbf{v}, \mathbf{w} \in \mathbb{R}^3$  (Spong

et al., 2006). Using these Jacobians, the inertia matrix is obtained as the sum of the contributions of each CoM that compose the system, as

$$\mathbf{M}(\mathbf{q}) = \sum_{i \in \{q, cm_1, cm_2, cm_3\}} (m_i \mathbf{J}'_{v_i} \mathbf{J}_{v_i} + \mathbf{J}'_{\omega_i} \mathbf{R}_i^I \mathcal{I} \mathbf{R}_i^I \mathbf{J}_{\omega_i}), \quad (6)$$

where  $m_i$  and  $\mathcal{I}_i$  are, respectively, the mass and the inertia tensor of the  $i$ -th CoM.

From the inertia matrix, it is possible to obtain the Coriolis matrix  $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$  by means of the Christoffel symbols of the first kind, which are given by

$$\mathbf{C}_{ij} = \sum_{k=1}^9 \frac{1}{2} \left( \frac{\partial \mathbf{M}_{ij}}{\partial \mathbf{q}_k} + \frac{\partial \mathbf{M}_{ik}}{\partial \mathbf{q}_j} - \frac{\partial \mathbf{M}_{jk}}{\partial \mathbf{q}_i} \right) \dot{\mathbf{q}}_k, \quad (7)$$

where  $\mathbf{C}_{i,j}$  and  $\mathbf{M}_{i,j}$  denote the elements of the  $i$ -th row and  $j$ -th column of the Coriolis and inertia matrix.

The gravitational forces vector is obtained from the potential energy of the system,  $\mathcal{U}$ , as  $\mathbf{g}(\mathbf{q}) = \partial \mathcal{U} / \partial \mathbf{q}$ , where

$$\mathcal{U} = -g \left( \sum_{i \in \{q, cm_1, cm_2, cm_3\}} m_i \mathbf{e}_3 \mathbf{p}_{I,i}^I \right), \quad (8)$$

with  $\mathbf{e}_3 = [0 \ 0 \ 1]$  and  $g$  being the acceleration of gravity.

The input coupling matrix  $\mathbf{B}(\mathbf{q})$  is given by

$$\mathbf{B}(\mathbf{q}) = \begin{bmatrix} \check{\mathbf{B}} & \mathbf{0}_{6 \times 3} \\ & \mathbf{1}_3 \end{bmatrix}, \quad (9)$$

with

$$\check{\mathbf{B}} = \begin{bmatrix} (\mathbf{J}'_{\omega_q} - \mathbf{J}'_{\omega_{i_0}}) \mathbf{R}_q^I & \mathbf{J}'_{v_q} \mathbf{R}_q^I \end{bmatrix} \mathbf{B},$$

$$\mathbf{B} = \begin{bmatrix} 0 & l \cos(\alpha_T) & 0 & -l \cos(\alpha_T) \\ -l \cos(\alpha_T) & 0 & l \cos(\alpha_T) & 0 \\ \frac{k_\tau}{k_f} \cos(\alpha_T) - \frac{k_\tau}{k_f} \cos(\alpha_T) & \frac{k_\tau}{k_f} \cos(\alpha_T) & -\frac{k_\tau}{k_f} \cos(\alpha_T) & \frac{k_\tau}{k_f} \cos(\alpha_T) \\ -\sin(\alpha_T) & 0 & \sin(\alpha_T) & 0 \\ 0 & -\sin(\alpha_T) & 0 & \sin(\alpha_T) \\ \cos(\alpha_T) & \cos(\alpha_T) & \cos(\alpha_T) & \cos(\alpha_T) \end{bmatrix},$$

where  $\mathbf{1}$  is the identity matrix,  $l$  is the distance between the propellers and the quadrotor geometric center,  $k_f$  is the propellers thrust coefficient,  $k_\tau$  is the propellers drag coefficient, and  $\alpha_T$  is a small tilting angle of the propellers towards the UAV geometric center that is introduced in order to improve controllability.

Table 1 presents the parameters of the UAM, which are obtained from the Computer Aided Design (CAD) model.

### 2.2 State space representation and linearized model

For control purposes, system (5) is represented in the state space

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{\Gamma}, \mathbf{d}) = \begin{bmatrix} \dot{\mathbf{q}} \\ \mathbf{M}(\mathbf{q})^{-1} (-\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}} - \mathbf{g}(\mathbf{q}) + \mathbf{B}(\mathbf{q}) \mathbf{\Gamma} + \mathbf{d}) \end{bmatrix}, \quad (10)$$

where  $\mathbf{x} = [\mathbf{q}' \ \dot{\mathbf{q}}']'$  is the state vector.

Then, the whole-body dynamic model (10) is linearized using first-order Taylor series expansion around the equilibrium point  $\mathbf{x}_{eq} = [0 \ -0.0698 \ \mathbf{0}_{1 \times 16}]'$ ,  $\mathbf{\Gamma}_{eq} =$

Table 1. System parameters.

Unmanned aerial manipulator parameters.				
Parameter	Value			
$l$	0.3 (m)			
$l_0$	0.075 (m)			
$k_f$	$9.510 \cdot 10^{-6}$ (N·s <sup>2</sup> /rad <sup>2</sup> )			
$k_\tau$	$1.710 \cdot 10^{-7}$ (N·m·s <sup>2</sup> /rad <sup>2</sup> )			
$\alpha_T$	5°			
$g$	9.81 (m/s <sup>2</sup> )			
$m_q$	2.24 (kg)			
$\mathcal{I}_q$	diag(0.0118, 0.0235, 0.0117) (kg·m <sup>2</sup> )			
$m_{l_1}, m_{l_2}, m_{l_3}$	0.2 (kg)			
$\mathcal{I}_{l_1}, \mathcal{I}_{l_2}, \mathcal{I}_{l_3}$	diag( $1.1 \cdot 10^{-5}$ , $1.1 \cdot 10^{-5}$ , $1.2 \cdot 10^{-5}$ ) (kg·m <sup>2</sup> )			
Denavit-Hartenberg parameters.				
link	$d_i$ (m)	$\vartheta_i$ (rad)	$a_i$ (m)	$\alpha_i$ (rad)
$l_1$	0	$\beta_1 + 0.3670$	0.0765	0
$l_2$	0	$\beta_2 + 2.5813$	0.1485	0
$l_3$	0	$\beta_3 + 2.9486$	0.1635	0

$[7.36 \ 7.48 \ 7.61 \ 7.48 \ 0.0728 \ -0.0418 \ 0.297]'$ , which is obtained by trimming the UAM in hovering, considering no disturbances affecting the system,  $\mathbf{d}_{eq} = \mathbf{0}$ . This results in the linearized model

$$\Delta \dot{\mathbf{x}} = \mathbf{A} \Delta \mathbf{x} + \mathbf{B}_\Gamma \Delta \mathbf{\Gamma} + \mathbf{B}_d \Delta \mathbf{d}, \quad (11)$$

where  $\Delta(\cdot) \triangleq (\cdot) - (\cdot)_{eq}$ , with  $\mathbf{A} \triangleq \frac{\partial \mathbf{f}(\mathbf{x}, \mathbf{\Gamma}, \mathbf{d})}{\partial \mathbf{x}} \Big|_{\substack{\mathbf{x}=\mathbf{x}_{eq} \\ \mathbf{\Gamma}=\mathbf{\Gamma}_{eq} \\ \mathbf{d}=\mathbf{d}_{eq}}}$ ,  $\mathbf{B}_\Gamma \triangleq \frac{\partial \mathbf{f}(\mathbf{x}, \mathbf{\Gamma}, \mathbf{d})}{\partial \mathbf{\Gamma}} \Big|_{\substack{\mathbf{x}=\mathbf{x}_{eq} \\ \mathbf{\Gamma}=\mathbf{\Gamma}_{eq} \\ \mathbf{d}=\mathbf{d}_{eq}}}$ , and  $\mathbf{B}_d \triangleq \frac{\partial \mathbf{f}(\mathbf{x}, \mathbf{\Gamma}, \mathbf{d})}{\partial \mathbf{d}} \Big|_{\substack{\mathbf{x}=\mathbf{x}_{eq} \\ \mathbf{\Gamma}=\mathbf{\Gamma}_{eq} \\ \mathbf{d}=\mathbf{d}_{eq}}}$ .

### 3. CONTROL DESIGN

This section presents the design of a linear mixed  $\mathcal{H}_2/\mathcal{H}_\infty$  controller with  $\mathcal{D}$ -stability constraints for trajectory tracking of the UAM end-effector.

Initially, to achieve a robust trajectory tracking with constant disturbance rejection capability, system (11) is augmented with integral actions considered for the errors of the yaw angle and position of the end-effector, and the joint angles of the manipulator arm, resulting in the new state space vector  $\Delta \bar{\mathbf{x}} = [\Delta \mathbf{x}' \ \int \Delta \psi \ dt \ \int \Delta \boldsymbol{\eta}' \ dt \ \int \Delta \mathbf{q}'_m \ dt]'$ . Then, the augmented system is represented in the closed-loop standard form

$$\mathcal{P}_1 : \begin{cases} \Delta \dot{\bar{\mathbf{x}}} = \underbrace{\begin{bmatrix} \mathbf{A} & \mathbf{0}_{18 \times 7} \\ \mathbf{0}_{7 \times 2} & \mathbf{1}_7 \end{bmatrix}}_{\mathbf{A}} \Delta \bar{\mathbf{x}} + \underbrace{\begin{bmatrix} \mathbf{B}_\Gamma \end{bmatrix}}_{\mathbf{B}_\Gamma} \Delta \mathbf{\Gamma} + \underbrace{\begin{bmatrix} \mathbf{B}_d \end{bmatrix}}_{\mathbf{B}_d} \Delta \mathbf{d}, \\ \Delta \mathbf{\Gamma}(t) = \mathbf{K} \Delta \bar{\mathbf{x}}(t), \\ \mathbf{z} = \mathbf{C} \bar{\mathbf{x}} + \mathbf{D}_\Gamma \Delta \mathbf{\Gamma} + \mathbf{D}_d \Delta \mathbf{d}, \end{cases} \quad (12)$$

where  $\mathbf{z}$  is the output vector,  $\mathbf{K}$  is the state feedback gain, and  $\mathbf{C}$ ,  $\mathbf{D}_\Gamma$ , and  $\mathbf{D}_d$  are tuning matrices with appropriate dimension.

The  $\mathcal{H}_2$  controller aims to minimize the  $\mathcal{H}_2$ -norm of the closed-loop system (12) and is defined as (Scherer et al., 1997)

$$\|\mathbf{H}(s)\|_2^2 = \frac{1}{2\pi} \int_{-\infty}^{\infty} \text{tr}\{\mathbf{H}^*(j\omega) \mathbf{H}(j\omega)\} d\omega, \quad (13)$$

where  $\mathbf{H}(s)$  is the system transfer function from the disturbance vector,  $\Delta \mathbf{d}$ , to the output vector,  $\mathbf{z}$ , and  $\text{tr}(\cdot)$  is the trace operator. The minimization of the  $\mathcal{H}_2$ -norm is useful to handle measurement noise, random disturbances and other stochastic aspects (Scherer et al., 1997). The  $\mathcal{H}_\infty$  controller aims to minimize the maximum gain that the closed-loop system (12) gives to a disturbance signal and is defined as (Scherer et al., 1997)

$$\|\mathbf{H}_{\delta z}(s)\|_\infty = \max_z |\mathbf{H}(j\omega)| = \gamma. \quad (14)$$

In this work, in order to achieve robustness capability, a mixed  $\mathcal{H}_2/\mathcal{H}_\infty$  controller is designed in order to minimize the  $\mathcal{H}_2$ -norm (13), while satisfying the  $\mathcal{H}_\infty$ -norm (14) for a given value of  $\gamma$ . The solution of the linear mixed  $\mathcal{H}_2/\mathcal{H}_\infty$  control problem is found by solving the following optimization problem (Cardoso et al., 2021):

$$\min_{\mathbf{N}, \mathbf{Y}, \mathbf{Q}} \text{tr}(\mathbf{N}), \quad (15)$$

$$\text{s.t.} \begin{cases} \mathbf{Q} > 0, \\ \begin{bmatrix} \mathbf{N} & & \mathbf{C}\mathbf{Q} + \mathbf{D}_\Gamma \mathbf{Y} \\ \mathbf{Q}\mathbf{C}' + \mathbf{Y}'\mathbf{D}'_\Gamma & \mathbf{Q} & \\ \overline{\mathbf{A}}\mathbf{Q} + \mathbf{Q}\overline{\mathbf{A}}' + \overline{\mathbf{B}}_\Gamma \mathbf{Y} + \mathbf{Y}'\overline{\mathbf{B}}'_\Gamma & \overline{\mathbf{B}}_d & \mathbf{Q}\mathbf{C}' + \mathbf{Y}'\mathbf{D}'_\Gamma \\ & \overline{\mathbf{B}}'_d & -\gamma \mathbf{1} & \mathbf{D}'_d \\ \mathbf{C}\mathbf{Q} + \mathbf{D}_\Gamma \mathbf{Y} & \mathbf{D}_d & -\gamma \mathbf{1} & \end{bmatrix} < 0, \end{cases}$$

where  $\gamma > \gamma^*$ , with  $\gamma^*$  being the optimal  $\mathcal{H}_\infty$  attenuation level, and  $\mathbf{N}$ ,  $\mathbf{Y}$  and  $\mathbf{Q}$  are decision variables of the optimization problem.

Additionally, to prevent excessive oscillatory behavior and allow better tuning of the system, the poles of the closed-loop system are restricted to a specified region of the complex plane, feature accomplished by including  $\mathcal{D}$ -stability constraints to the optimization problem (15), which is performed according to the following lemma.

*Lemma 1.* (Scherer et al., 1997) The poles of the closed-loop system (12) lie in the region  $\mathcal{R} = \{z \in \mathbb{C} | \mathbf{L}_i + z\mathbf{S}_i + \overline{z}\mathbf{S}'_i < 0\}$  of the complex plane, which is defined by the matrices  $\mathbf{L}_i$  and  $\mathbf{S}_i$ , where  $\overline{z}$  indicates the conjugate of  $z$ , if the following LMI is satisfied:

$$\mathbf{L}_i \otimes \mathbf{Q} + \mathbf{S}_i \otimes (\overline{\mathbf{A}}\mathbf{Q} + \overline{\mathbf{B}}_\Gamma \mathbf{Y}) + \mathbf{S}'_i \otimes (\mathbf{Q}\overline{\mathbf{A}}' + \mathbf{Y}'\overline{\mathbf{B}}'_\Gamma) < 0, \quad (16)$$

where  $\otimes$  denotes the Kronecker product.

After solving the optimization problem (15) added with the LMI constraint (16), the state feedback gain  $\mathbf{K}$  is obtained as  $\mathbf{K} = \mathbf{Y}\mathbf{Q}^{-1}$ .

#### 4. EXPERIMENT

This section presents the results of a numerical experiment performed to evaluate the proposed mixed  $\mathcal{H}_2/\mathcal{H}_\infty$  controller performance.

The numerical experiment consists in a HIL simulation, in which the proposed controller is implemented in a Raspberry Pi 4 Single Board Computer (SBC) connected via a 1Gbps Ethernet network to a simulation server running on the ProVANT Simulator. The simulation server provides the desired reference signals and the current state vector of the UAM to the embedded controller that must execute the control law and respond to the server with the desired control inputs. Figure 3 is an overview of this experimental setup.

In this work, the simulation server is a general purpose computer equipped with an Intel Core i7 7500U processor, 16GB of RAM, and an NVidia GTX920MX GPU.

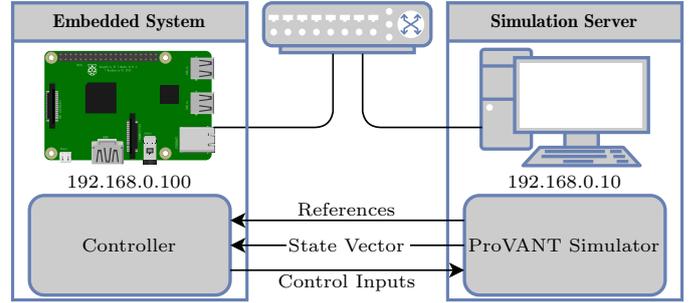


Figure 3. Illustration of the connection between the embedded system and the simulation server, the software running in each system component, and the data transmitted between the softwares.

In order to generate a reference trajectory that allows the performance evaluation of several activities commonly present in tasks performed by UAMs, a mission composed of the following stretches was designed: (i) the UAM starts displaced from the desired trajectory, at position  $x = 0.43565$ ,  $y = 2.7$ ,  $z = 0.167$ , and with the remaining states equal to zero; (ii) the UAM goes through an intermediate point in the position  $x = y = 1.5$ , and  $z = 0.75$ ; (iii) the UAM goes to the target position  $x = 2.23$ ,  $y = 2.7$ ,  $z = 0.73$  and hovers while extending its manipulator arm; (iv) the UAM retracts the manipulator arm; and (v) the UAM returns to the starting position and lands.

The waypoints of the mission phases are then used as target points in an offline path planner implemented using the MoveIt<sup>2</sup> software with a modified version of the Rapidly exploring Random Tree (RRT) algorithm, denoted RRT\* with a subsequent smoothing phase. The resulting trajectory is stored in a Comma Separated Values (CSV) file, and used to provide the references for the system during the simulation.

Figure 4 shows an image of the simulation environment in the Gazebo simulator, which illustrates the UAM at the initial position, the obstacle that is colored in transparent gray to allow an unobstructed view of the simulation, and a yellow cylinder located at the target position reached in phase (iii) of the experiment. It is important to note that the color of the obstacle does not influence in the collision detection of the simulation, and thus must still be avoided by the UAM regardless of its color.

The model of the UAM used in Gazebo is exported from the CAD model to a Unified Robot Description Format (URDF). The URDF contains the robot's kinematic model represented in a tree structure with links connected by joints. The files also contain the physical parameters such as mass, moments of inertia, and position of the CoM of each link, allowing the dynamical model simulation using the Open Dynamics Engine (ODE) physics software (Hsu and Peters, 2014). The URDF also includes tridimensional

<sup>2</sup> MoveIt is an open-source software implemented in ROS that provides motion planning algorithms, collision checking, and other utilities for robots (Coleman et al., 2014).

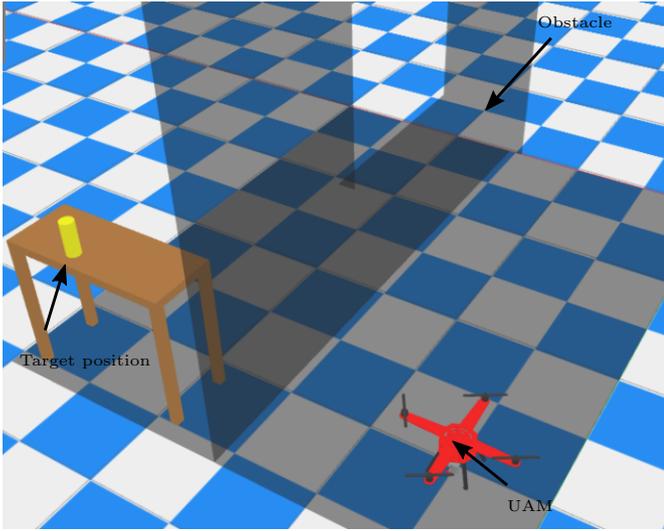


Figure 4. Illustration of the simulation environment in Gazebo.

meshes of each link, which is used by Gazebo to provide visual feedback and check for collisions with other objects.

In order to perform the numerical experiment, the mixed  $\mathcal{H}_2/\mathcal{H}_\infty$  controller is tuned with  $\gamma = 25$ ,  $\mathbf{C} = \mathbf{1}_{25}$ ,  $\mathbf{D}_\Gamma = \mathbf{O}_{25 \times 7}$ ,  $\mathbf{D}_d = \mathbf{O}_{25 \times 9}$ , and the poles of the closed-loop system are constrained to lie inside a horizontal strip with radius of 20, to the left of the half plane with origin at -1 and to the right of the half plane with origin at -1000. These constraints are employed by adding the LMI (16) to the optimization problem (15), considering the following combinations of  $\mathbf{L}_i$  and  $\mathbf{S}_i$  matrices, for  $i \in \{1, 2, 3\}$ :

$$\mathbf{L}_1 = \begin{bmatrix} -20 + 2i & 0 \\ 0 & -20 \end{bmatrix}, \quad \mathbf{S}_1 = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}, \quad (17)$$

$$\mathbf{L}_2 = 2 + i, \quad \mathbf{S}_2 = 1, \quad (18)$$

$$\mathbf{L}_3 = -2000 + i, \quad \mathbf{S}_3 = -1. \quad (19)$$

The solution of the optimization problem was found with  $\text{tr}(\mathbf{N}) = 111.73$ .

During the experiment, the following external disturbances (see (5)) are applied to the system after 22.5 seconds:  $d_\phi = d_\theta = d_\psi = 0.25$  [Nm],  $d_x = d_y = 1$  [N],  $d_z = -8$  [N],  $d_{\beta_1} = d_{\beta_2} = d_{\beta_3} = -1$  [Nm]. The results<sup>3</sup> of the experiment are shown in Figures 5 and 6. Table 2 shows the minimum, average and maximum time necessary for computation of the control law.

During the first phase of the experiment, the reference trajectory is modified so that the end-effector starts vertically displaced from the desired trajectory, and the end-effector quickly converges to the desired trajectory in a few seconds, with a small overshoot. The UAM is able to track the desired trajectory, passing by the waypoints of phase (ii), and arriving at (iii) with a very small tracking error. In phase (iii), while extending the manipulator arm, the CoM of the system is displaced by the required motions, but the system remains stable and the tracking error remains small.

<sup>3</sup> A video recording of the experiments is available in [https://youtu.be/\\_uxSyj9Rx0o](https://youtu.be/_uxSyj9Rx0o).

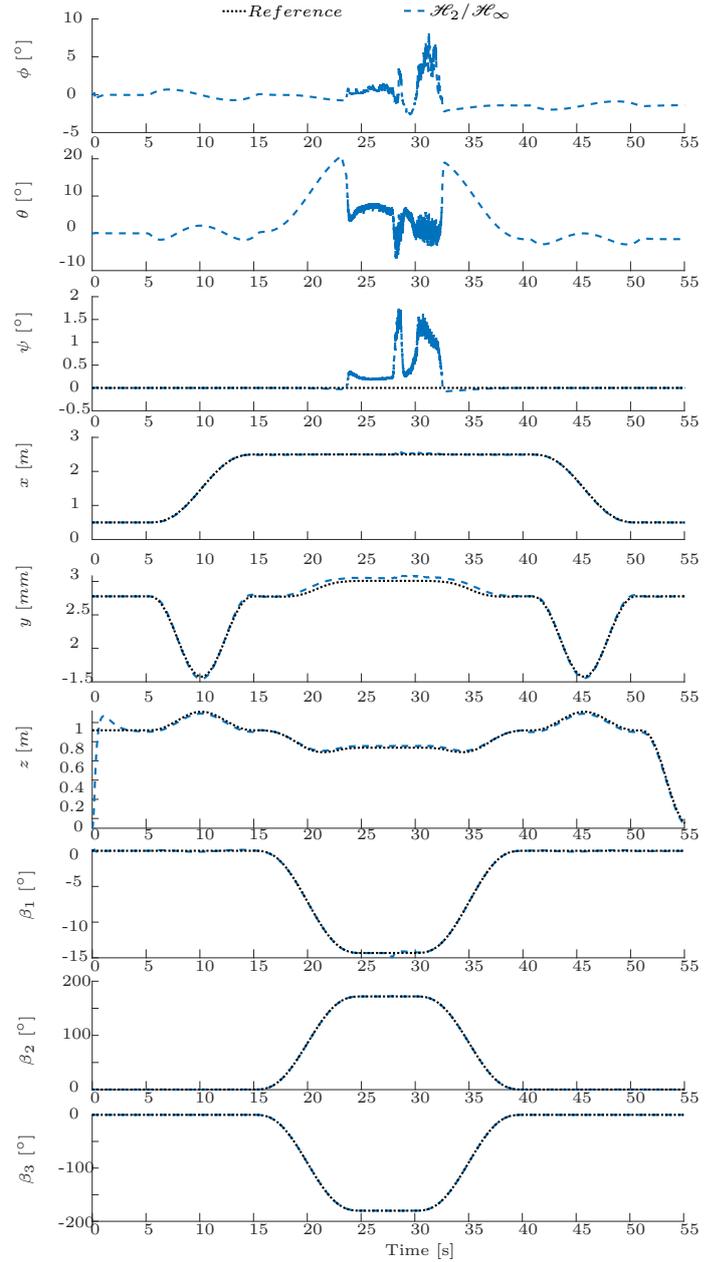


Figure 5. Roll, pitch and yaw angles, translational position, and angles of the manipulator arm during the experiment.

At the beginning of phase (iv), at around 22.5 seconds, the disturbance signals begin to affect the system and are quickly counteracted, the states of the system converge to the trajectory after a few transient behaviour. During phases (iv) and (v), the tracking error remains small. Therefore, we conclude that the proposed control strategy is able to perform robust trajectory tracking and fulfill the objectives of the proposed experiment.

Due to the linear nature of the control law, the computational cost of the applied control inputs is very small as illustrated in Table 2, showing good results during the experiment. However, the whole-body system of the UAM obtained from the perspective of the end-effector is a very complex, nonlinear, highly coupled dynamical system. Due to this dynamical coupling, compromises are required

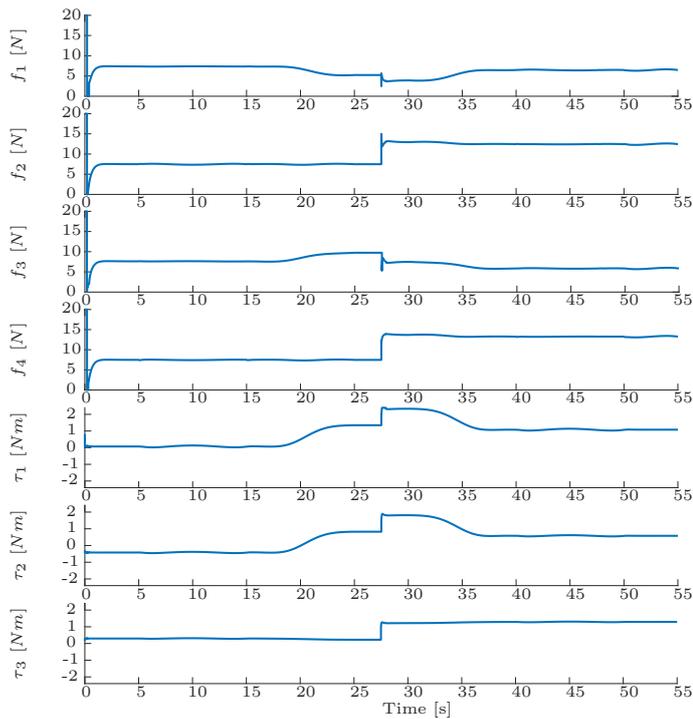


Figure 6. Applied control inputs during the experiment.

Table 2. Minimum, Average, and Maximum Control law execution time.

Minimum	Average	Maximum
1.21 ms	1.47 ms	1.56 ms

during the controller tuning, rendering this process very difficult.

## 5. CONCLUSION

This work obtained the kinematic and whole-body dynamical models of a UAM from the perspective of the end-effector, and designed a linear mixed  $\mathcal{H}_2/\mathcal{H}_\infty$  controller for a robust trajectory tracking of the pose of the UAM end-effector. The linear mixed  $\mathcal{H}_2/\mathcal{H}_\infty$  controller was implemented in C++ programming language, embedded into a Raspberry Pi 4 SBC, and simulated in a HIL framework using the ProVANT Simulator. The numerical experiments demonstrated that the controller achieved good trajectory tracking and computational performance, and successfully attenuated the effects of the applied disturbances. The proposed whole-body model allowed the design of a single-layer controller for trajectory tracking of the pose of the end-effector, and showed potential to formulate, in a future work, a unified control and path-planning approach with the incorporation of workspace constraints in order to avoid collisions. Future works include the application of scaling techniques to the system in order to allow easier control tuning, the formulation of robust nonlinear control strategies that consider constraints on the UAM workspace, allowing collision avoidance, and finally, we intend to perform real flight experiments.

## REFERENCES

Acosta, J., de Cos, C.R., and Ollero, A. (2020). Accurate control of Aerial Manipulators outdoors. A reliable and

self-coordinated nonlinear approach. *Aerospace Science and Technology*, 99, 105731.

Bouzgou, K., Benchikh, L., Nouveliere, L., Bestaoui, Y., and Ahmed-Foitih, Z. (2020). PD sliding mode controller based decoupled aerial manipulation. In *ICINCO 2020 - Proceedings of the 17th International Conference on Informatics in Control, Automation and Robotics*, 484–489.

Cardoso, D.N., Esteban, S., and Raffo, G.V. (2021). A new robust adaptive mixing control for trajectory tracking with improved forward flight of a tilt-rotor UAV. *ISA Transactions*, 110, 86–104.

Chaikalis, D., Khorrami, F., and Tzes, A. (2020). Adaptive Control Approaches for an Unmanned Aerial Manipulation System. *2020 International Conference on Unmanned Aircraft Systems, ICUAS 2020*, 498–503.

Chen, Y., Zhan, W., He, B., Lin, L., Miao, Z., Yuan, X., and Wang, Y. (2020). Robust Control for Unmanned Aerial Manipulator under Disturbances. *IEEE Access*, 8, 129869–129877.

Coleman, D., Sucas, I., Chitta, S., and Correll, N. (2014). Reducing the Barrier to Entry of Complex Robotic Software: a MoveIt! Case Study. *Journal of Software Engineering for Robotics*, 1(1), 1–14.

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

Hsu, J.M. and Peters, S.C. (2014). Extending Open Dynamics Engine for the DARPA Virtual Robotics Challenge. In *International Conference on Simulation, Modeling, and Programming for Autonomous Robots*, 37–48. Springer, Bergamo.

Koenig, N. and Howard, A. (2004). Design and use paradigms for gazebo, an open-source multi-robot simulator. In *IEEE International Conference on Intelligent Robots and Systems*, volume 3, 2149–2154.

Mello, L.S., Raffo, G.V., and Adorno (2016). Robust whole-body control of an unmanned aerial manipulator. In *2016 European Control Conference, ECC 2016*, 702–707. IEEE, Aalborg.

Mello, L.S., Raffo, G.V., and Adorno, B.V. (2015). Whole-body modeling and control of an unmanned aerial manipulator. In *XII Simpósio Brasileiro de Automação Inteligente (SBAI)*, 1–6. Sociedade Brasileira de Automática.

Morais, J.E., Neri, D., and Raffo, G.V. (2020). Robust optimal nonlinear control strategies for an aerial manipulator. In *Congresso Brasileiro de Automática - CBA*, 1–8. Sociedade Brasileira de Automática.

Nava, G., Sablé, Q., Tognon, M., Pucci, D., and Franchi, A. (2020). Direct Force Feedback Control and Online Multi-Task Optimization for Aerial Manipulators. *IEEE Robotics and Automation Letters*, 5(2), 331–338.

Scherer, C., Gahinet, P., and Chilali, M. (1997). Multiobjective output-feedback control via LMI optimization. *IEEE Transactions on Automatic Control*, 42(7), 896–911.

Spong, M.W., Hutchinson, S., and Vidyasagar, M. (2006). *Robot Modeling and Control*. Jhon Wiley & Sons, inc.