IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 24 January 2024, accepted 31 March 2024, date of publication 4 April 2024, date of current version 11 April 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3384986

RESEARCH ARTICLE

Open-Access Platform for the Simulation of Aerial Robotic Manipulators

JOSÉ VARELA-ALDÁS[1], (Member, IEEE), LUIS F. RECALDE[1], BRYAN S. GUEVARA[2], VICTOR H. ANDALUZ[3], AND DANIEL C. GANDOLFO[2]

[1]Centro de Investigaciones de Ciencias Humanas y de la Educación (CICHE), Universidad Indoamérica, Ambato 180103, Ecuador
[2]Instituto de Automática, Universidad Nacional de San Juan—CONICET, San Juan 5400, Argentina
[3]Departamento de Elóctrica y Electrónica, Universidad de las Fuerzas Armadas—ESPE, Sangolquí 171103, Ecuador

Corresponding author: José Varela-Aldás (josevarela@uti.edu.ec)

This work was supported by Universidad Indoamérica, Ecuador, under Grant INV-0019-011.

ABSTRACT Recent technological advances have brought increased attention to aerial robotic manipulators (ARMs), particularly in applications that involve physical interactions using tools such as welding and drilling, as well as in the autonomous pickup and transport of objects. However, translating control algorithms into real-world applications for aerial robotic manipulators may prove challenging, given the potential for accidents and the time-consuming nature of experiments; furthermore, the acquisition of aerial robotic manipulators could impose a substantial financial burden on universities, research centers, and companies. Therefore, this work addresses these issues by developing an open access platform to simulate aerial robotic manipulators and test control strategies. The presented simulator is based on the kinematics and dynamics of the Matrice-100 aerial platform equipped with a 3 DOF robotic arm, where the mathematical formulation was developed using the Euler-Lagrange formalism. In addition, optimization techniques were used to perform the parameter identification procedure, ensuring the development of an accurate model for the open-access platform. The simulator platform is built upon the integration of Python, the Robot Operating System (ROS), and Unity 3D. These components collaborate to describe and demonstrate the behavior of the aerial robotic manipulator during the test process of control system algorithms. Simple tests were conducted to validate the open-access simulator platform. The proposed approach ensures the evaluation, testing of control strategies, and the ability to conduct experiments before hardware implementations. Finally, the proposal was published as an open source platform available in the following Code.

INDEX TERMS Robotic simulation, aerial robotic manipulator dynamics, open source, dynamic parameters identification, control algorithms.

I. INTRODUCTION

A. MOTIVATION

In recent decades, unmanned aerial vehicles (UAVs) have received significant attention, especially from research communities and industrial companies. Many applications have been proposed under the conceptualization of passive tasks that avoid interaction with the environment, presenting potential applications such as surveillance and monitoring [1], [2], [3]; remote sensing and data collection of agricultural sectors or inaccessible areas [4], [5]; either infrastructure inspection or inspection during disasters and crises [6], [7] and for transportation and navigation purposes [8].

However, applications that involve physical interaction with the environment face restrictions and cannot be developed solely using UAVs. Consequently, research activities are increasingly focusing on interactions with aerial robotic manipulators, a trend that is gaining popularity, as noted in [9]. ARMs are integrated into aerial vehicles equipped with one or more robotic arms, enabling robust and precise manipulation tasks while enhancing manipulability capabilities. Aerial robotic manipulators have received considerable

The associate editor coordinating the review of this manuscript and approving it for publication was Cheng Qian.

attention, and many prototypes have been developed and evaluated in outdoor and indoor conditions, such as inspection of infrastructure, railways, refineries, roads and electrical lines [10], [11]; physical interaction using tools for welding and drilling [12]; and autonomous pickup and transport of objects [13], [14]. Finally, an overview of aerial manipulation applications can be found in [15].

ARMs show an exciting challenge in the control field due to the coupling dynamics and the instability generated by the UAV and the movements of the robotic arm, where many algorithms have been presented in the literature [16], [17], and [15]; nevertheless, they have revealed challenging implementations associated with high nonlinearities, underactuated properties, and expensive implementation requirements. Therefore, there is a demand for tools capable of simulating the dynamics of aerial robotic manipulator systems and facilitating the rapid implementation of control algorithms. Precise dynamics and effective communication channels are critical features in aerial robotic manipulator simulations. Primarily, an accurate mathematical representation of the system is employed to construct a sophisticated physics model, ensuring that the results obtained in the simulation platform can be applied in the physical world, thus reducing experimentation time and cost.

Simulation platforms have made crucial contributions in the robotics field [18], [19], mainly by generating benefits related to the cost of experiments, time savings, and implementing a wide variety of control algorithms in different scenarios. Furthermore, errors and bugs in control algorithms cost practically nothing on simulation platforms and allow the user to understand the system and identify possible complications under different conditions. Although there are limitations associated with the lack of a perfect representation of the physical model, simulation platforms have generated crucial advantages, especially in activities related to the robotic field [20].

B. RELATED WORKS

Nowadays, simulation platforms have developed notorious applications that mimic the behavior of complex systems. The purpose of simulations is to help and support the user during the design phase of the control algorithms and to evaluate the system's functionality. Furthermore, technological advances in the last decade have expanded the use of simulators in several areas, e.g. engineering and robotics, among others [21], [22].

A literature review shows that simulators are oriented to robotic applications, and many solutions have been presented based on external simulators and physics engines such as CoppeliaSim (previously V-REP) [23], Gazebo [24] and Mujoco [25], which provide simulation results of an extensive range of situations, where the users can develop control algorithms in a variety of environments.

However, simulation platforms presented above require either high computational capabilities or significant additional effort to learn and develop a specific project simulation. Furthermore, control algorithms for any aerial robotic manipulator developed must be evaluated through different experimental tests to verify their efficiency, stability, and robustness. Therefore, a platform for the simulation of aerial robotic manipulators that enables the test of different control algorithms without additional effort is increasingly important during the design phase in robotics applications.

C. MAIN CONTRIBUTIONS

This article presents the design and implementation of a simulation platform for aerial robotic manipulators, which provides a valuable tool to design and evaluate control algorithms and reduces time during the design stage. The platform allows researchers and developers to simulate the behavior of aerial robotic manipulators and modify system control parameters to evaluate performance in various situations. The developed platform under the Model-in-Loop (MIL) architecture can satisfy essential requirements, including system dynamics, modularity, and accessible code reuse, without additional effort associated with the simulation experiments.

The credibility of the simulator is based on the accuracy of the mathematical model used to mimic the dynamics in the system. These formulations are developed on the basis of two main principles: understanding the system behavior commonly using physics laws and the data obtained through real-world experiments, which are usually used to validate the mathematical formulations. Therefore, this work developed a kinematic and dynamic model of the aerial robotic manipulator considering a commercial aerial vehicle (Matrice-100) equipped with a 3DOF robotic arm. Additionally, commercial aerial vehicles are usually controlled through high-level reference signals such as desired angles or velocities; thus, this work formulated a reduced dynamic model in the velocity space instead of thrust and torque as a control input. The proposed mathematical formulation uses the concepts of Euler-Lagrange, parameter identification, and optimization techniques to ensure the proper behavior of the system and the perfect representation of the real aerial robotic manipulator.

The simulator platform can be easily used in other software such as Matlab [26] and Julia [27] since our simulation framework is based on Python, ROS, and Unity, which guarantees the modularity of the entire system. The widespread adoption of Python and ROS within the robotic community played a crucial role in the selection of language and middleware criteria. The simulation platform for the aerial robotic manipulator can be divided into three components: *(i) System Model:* The reduced dynamic model, developed in velocity space, was implemented using the Python language. Python is popular in the robotic community, enabling users to quickly onboard and easily reuse the model [28]. *(ii) Graphical Representation:* This aspect illustrates the behavior and evolution over time. Unity 3D graphics engine, a widely used software for developing

robotic simulators, was employed for this purpose [29], [30]. *(iii) Interface and Middleware:* For interface and middleware, ROS (Robot Operating System) was utilized. ROS is a standard in robotics software development, applicable to both real-world and simulated robots. It excels in handling communication between programs in a distributed system, ensuring modularity [31].

Control algorithms must be evaluated in different environments to verify their stability and robustness. However, this task is not trivial, particularly concerning aerial robotic manipulators and the acquisition or construction of these platforms, which can be expensive for research institutes and universities. Given the considerations related to the aerial robotic manipulator, this work validates the proposed simulation platform using a basic control algorithm based on the system model applied to the trajectory tracking control problem. Finally, the proposal has been released as an open-source platform, accessible through the following Code. This availability makes the project easily accessible and facilitates code reuse, meeting the essential requirement for simulation platforms.

II. MATHEMATICAL FORMULATIONS

The accuracy of the simulator is based on the mathematical model used to represent the dynamics of the aerial robotic manipulator. This work formulates the mathematical model using the differential kinematics and dynamics model of the aerial robotic manipulator complied with the commercial aerial vehicle (Matrice-100) equipped with a 3DOF robotic arm. In addition, this work developed a dynamic model in the reference velocity space. This choice comes from the fact that most commercial aerial vehicles are controlled using high-level reference signals, such as the desired angles or velocities.

A. AERIAL ROBOTIC MANIPULATOR MODELING

This section presents the formal notation and coordinate frames used in this work. *Notations* throughout this work, vectors are represented by bold lowercase letters $\mathbf{z} \in \mathcal{R}^n$ and the matrices are presented as uppercase letters $\mathbf{Z} \in \mathcal{R}^{n\times m}$; furthermore, the Euclidean norm can be formulated as $\|\mathbf{z}\| = \sqrt{\mathbf{z}^\intercal\, \mathbf{z}}$.

The coordinate frames used in this work are the inertial frame $\mathcal{I} : \{x_i, y_i, z_i\}$ with the vertical axis z_I pointed upward opposite to gravity, the body frame $\mathcal{B} : \{x_b, y_b, z_b\}$ attached to the center of mass of the UAV (CoM), the coordinate frame at the origin and the end effector of the robotic arm can be expressed as $\mathcal{C} : \{x_c, y_c, z_c\}$ and $\mathcal{E} : \{x_e, y_e, z_e\}$ respectively, which are introduced to describe the motion of the aerial robotic manipulator (for more details, refer to Figure 1).

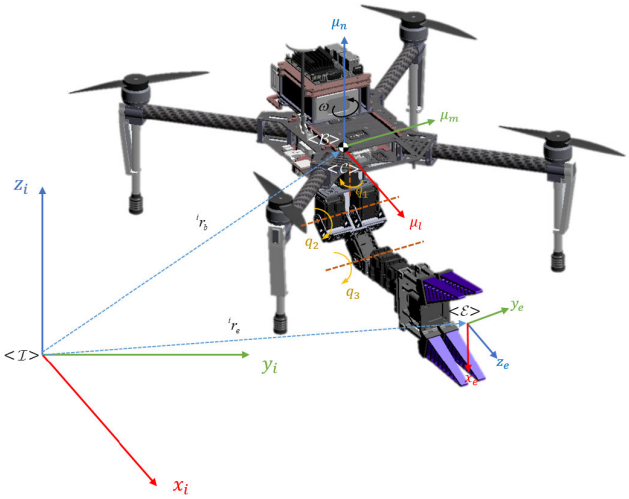

FIGURE 1. Unmanned aerial manipulators system.

1) DIFFERENTIAL KINEMATIC MODELING

This section presents the development of the differential forward kinematics of the aerial robotic manipulator, which can be formulated using the links geometry and internal states of the entire system (see Table 1), described as follows:

$$^i\mathbf{r}_e = \mathcal{K}(\mathbf{q}_u(t), \mathbf{q}_a(t)) \tag{2}$$

where the non-linear mapping $\mathcal{K}(\cdot)$ can be defined as the product of M elementary transformations $\mathcal{K}(\cdot) = \prod_{i=1}^{M}\mathbf{E}_i$ from the inertial frame $< \mathcal{I} >$ to the end effector position $< \mathcal{E} >$.

As a consequence of the low-level flight controller (attitude PID) of commercial aerial vehicles, this work assumes restricted movements of the aerial platform, taking into account the pitch (θ) and roll (ϕ) angles take values close to zero $\phi \approx 0$ and $\theta \approx 0$. The vector of generalized coordinates of the UAV is defined as $\mathbf{q}_u = \begin{bmatrix} q_x & q_y & q_z & q_\psi \end{bmatrix}^\intercal \in \mathcal{R}^4$ respect to $< \mathcal{B} >$; the vector of joint coordinates of the robotic manipulator $\mathbf{q}_a = \begin{bmatrix} q_1 & q_2 & q_3 \end{bmatrix}^\intercal \in \mathcal{R}^3$ and finally $^i\boldsymbol{r}_e = \begin{bmatrix} ^ir_x & ^ir_y & ^ir_z & ^ir_\psi \end{bmatrix}^\intercal \in \mathcal{R}^4$ is some representation of the end effector pose in the inertial coordinate frame $< \mathcal{I} >$.

Therefore, the position and orientation of the end-effector attached to the aerial robotic manipulator (see Figure 1) can

$$
\begin{aligned}
^i\dot{r}_x &= \mu_l c(q_\psi) - \mu_m s(q_\psi) - s(q_\psi q_1)(l_3 c(q_2 q_3) + l_2 c(q_2))(\omega + \dot{q}_1) - c(q_\psi q_1)(l_3 s(q_2 q_3) + l_2 s(q_2))\dot{q}_2 - l_3 c(q_\psi q_1) s(q_2 q_3)\dot{q}_3 \\
^i\dot{r}_y &= \mu_l s(q_\psi) + \mu_m c(q_\psi) - c(q_\psi q_1)(l_3 c(q_2 q_3) + l_2 c(q_2))(\omega + \dot{q}_1) - s(q_\psi q_1)(l_3 s(q_2 q_3) + l_2 s(q_2))\dot{q}_2 - l_3 s(q_\psi q_1) s(q_2 q_3)\dot{q}_3 \\
^i\dot{r}_z &= \mu_n - (l_3 c(q_2 q_3) + l_2 c(q_2))\dot{q}_2 - l_3 c(q_2 q_3)\dot{q}_3 \\
^i\dot{r}_\psi &= \dot{q}_1 + \omega
\end{aligned} \tag{1}
$$

be defined in detail as follows:

$$\begin{cases} {}^{i}r_x = q_x + c(q_\psi q_1)l_3c(q_2q_3) + c(q_\psi q_1)l_2c(q_2) \\ {}^{i}r_y = q_y + s(q_\psi q_1)l_3c(q_2q_3) + s(q_\psi q_1)l_2c(q_2) \\ {}^{i}r_z = q_z - l_1 - l_2s(q_2) - l_3s(q_2q_3) \\ {}^{i}r_\psi = q_\psi + q_1 \end{cases} \tag{3}$$

where $s = \sin(\cdot)$, $c = \cos(\cdot)$, $c(ab) = \cos(a+b)$ and $s(ab) = \sin(a+b)$.

The differential kinematics that relates the linear and angular velocities of the end effector to the maneuverability velocities of the aerial platform and the joint velocities of the robotic arm can be formulated taking the time derivative of the (3) resulting as in (1), shown at the bottom of the previous page, more information about the system parameters is presented in Table 1.

The differential kinematics of the entire system can be compactly written in a matrix form defined as follows:

$${}^{i}\dot{\mathbf{r}}_e = \mathbf{J}(\mathbf{q}(t))\mathbf{v}(t) \tag{4}$$

where, $\mathbf{q} = \begin{bmatrix}\mathbf{q}_u^\intercal & \mathbf{q}_a^\intercal\end{bmatrix}^\intercal \in \mathcal{R}^7$ and $\mathbf{v} = \begin{bmatrix}\mathbf{v}_u^\intercal & \mathbf{v}_a^\intercal\end{bmatrix}^\intercal \in \mathcal{R}^7$ are the generalized vector of the internal states and maneuverability velocities of the aerial robotic manipulator.

Furthermore, the velocities of the UAV are presented in $\mathbf{v}_u = \begin{bmatrix}\mu_l & \mu_m & \mu_n & \omega\end{bmatrix}^\intercal \in \mathcal{R}^4$, $\mathbf{v}_a = \begin{bmatrix}\dot{q}_1 & \dot{q}_2 & \dot{q}_3\end{bmatrix}^\intercal \in \mathcal{R}^3$ is the vector of joint coordinate velocities associated to the robotic arm; and ${}^{i}\dot{\boldsymbol{r}}_e = \begin{bmatrix}{}^{i}\dot{r}_x & {}^{i}\dot{r}_y & {}^{i}\dot{r}_z & {}^{i}\dot{r}_\psi\end{bmatrix}^\intercal \in \mathcal{R}^4$ is the end-effector pose velocity.

Finally, the Jacobian matrix defined as $\mathbf{J}(\boldsymbol{q}) = \partial\mathcal{K}(\mathbf{q})/\partial\mathbf{q} \in \mathcal{R}^{4\times7}$ allows a linear mapping between the velocities in the end effector to the generalized velocities. The Jacobian matrix is presented in the Appendix of this work.

2) GENERALIZED DYNAMIC MODEL

This section presents the dynamic model of the aerial robotic manipulator, which guarantees the proper behavior of the proposed open access platform. The dynamic model of the entire system was developed considering the Euler-Lagrange formulation. The *Lagrangian* of the system can be formulated as $\mathcal{L} = \mathcal{T} - \mathcal{U}$, which is defined by the kinetic and potential energy presented in the aerial robotic manipulator; thus, the total kinetic energy can be formulated as follows:

$$\mathcal{T} = \frac{1}{2}\dot{\mathbf{q}}^\intercal \sum_{n=1}^{4}\left[m_n\mathbf{J}_n^k(\mathbf{q})^\intercal\mathbf{J}_n^k(\mathbf{q})\right]\dot{\mathbf{q}} + ..\dot{\mathbf{q}}^\intercal \sum_{n=1}^{4}\left[\mathbf{J}_n^\omega(\mathbf{q})^\intercal\mathbf{R}_n(\mathbf{q})\,\mathbf{I}_n\,\boldsymbol{R}_n(\mathbf{q})^\intercal\mathbf{J}_n^\omega(\mathbf{q})\right]\dot{\mathbf{q}} \tag{5}$$

where the matrices $\mathbf{J}_n^k(\mathbf{q})$ and $\mathbf{J}_n^\omega(\mathbf{q})$ express the energy considering the translational and rotational velocities; m_n and $\mathbf{I}_n$ represent the mass and the inertial matrix of the entire system considering the aerial platform and each link individually; finally, $\mathbf{R}_n(\mathbf{q})$ is a rotational matrix. The kinetic energy can be formulated in a compact form as a function of the generalized coordinate $\dot{\mathbf{q}} = \begin{bmatrix}\dot{\mathbf{q}}_u^\intercal & \dot{\mathbf{q}}_a^\intercal\end{bmatrix}^\intercal \in \mathcal{R}^7$ defined as:

$$\mathcal{T} = \frac{1}{2}\dot{\mathbf{q}}^\intercal\mathbf{M}(\mathbf{q})\,\dot{\mathbf{q}} \tag{6}$$

where $\mathbf{M}(\mathbf{q}) \in \mathcal{R}^{7\times7}$ is the mass and inertia matrix which holds the following properties: *(1) strictly symmetric* and *(2) positive definite*.

On the other hand, the potential energy can be summarize considering the contribution of the aerial platform and the internal configuration of each link of the robotic arm, the formulation can be written as:

$$\mathcal{U} = \sum_{n=1}^{4}(m_n\mathbf{g}^\intercal\,{}^{i}\mathbf{r}_n) \tag{7}$$

where $\mathbf{g} = \begin{bmatrix}0 & 0 & g\end{bmatrix}^\intercal$ is the acceleration vector associated with the gravity g and ${}^{i}\mathbf{r}_n$ represents the location of the end effector considering the aerial platform and the position of the links presented in the robotic arm.

Therefore, the *Lagrangian* associated with the aerial robotic manipulator can be formulated as:

$$\mathcal{L}(\mathbf{q},\dot{\mathbf{q}}) = \frac{1}{2}\dot{\mathbf{q}}^\intercal\mathbf{M}(\mathbf{q})\dot{\mathbf{q}} - \sum_{n=1}^{4}(m_n\mathbf{g}^\intercal\,{}^{i}\mathbf{r}_n) \tag{8}$$

where this work used the Euler-Lagrange constraint $\frac{d}{dt}\left(\frac{\partial\mathcal{L}}{\partial\dot{\mathbf{q}}}\right) - \frac{\partial\mathcal{L}}{\partial\mathbf{q}} = \tau$ in order to obtain the dynamic model. This work developed the non-linear dynamic model of the aerial robotic manipulator as follows:

$$\begin{aligned} &\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}}(t) + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}}(t) + \mathbf{g}(\mathbf{q}) = {}^{i}\boldsymbol{f}(t) \\ &\mathbf{C}(\mathbf{q},\dot{\mathbf{q}}) = \dot{\mathbf{M}}(\mathbf{q}) - \frac{1}{2}\dot{\mathbf{q}}^\intercal\frac{\partial\mathbf{M}(\mathbf{q})}{\partial\mathbf{q}}; \quad \mathbf{g}(\mathbf{q}) = \frac{\partial\mathcal{U}}{\partial\mathbf{q}} \end{aligned} \tag{9}$$

where $\mathbf{C}(\mathbf{q},\dot{\mathbf{q}}) \in \mathcal{R}^{7\times7}$ is the Coriolis matrix that includes the centrifugal and centripetal forces presented in the entire system, $\mathbf{g}(\mathbf{q}) \in \mathcal{R}^7$ is the generalized gravity force vector, ${}^{i}\boldsymbol{f} = \begin{bmatrix}{}^{i}f_x & {}^{i}f_y & {}^{i}f_z & \tau_\psi & \tau_{q_1} & \tau_{q_2} & \tau_{q_3}\end{bmatrix}^\intercal \in \mathcal{R}^7$ represents generalized forces and torques applied to the aerial robotic manipulator associated to $< \mathcal{I} >$. Finally, $\ddot{\mathbf{q}}$ and $\dot{\mathbf{q}}$ are the acceleration and velocity vector of the generalized coordinates.

3) SIMPLIFIED DYNAMIC MODEL

The simplified dynamic model is presented in this section considering that commercial aerial platforms include a low-level flight controller (attitude PID) to maintain the system close to the equilibrium point with slight variations in the angles of pitch (θ) and roll (ϕ). Furthermore, the robotic manipulator incorporates DC motors with low-level PID controllers, ensuring precise angular reference tracking velocities in each joint.

As a result of the attitude controllers, both the aerial platform and the robotic arm can be controlled using linear and angular reference velocities $\boldsymbol{v_d}$. Considering this, the dynamic model of the aerial manipulator should incorporate these reference velocities as inputs to the system.

TABLE 1. Aerial robotic manipulator parameters.

Symbolic variable	Description	Symbolic variable	Description
l_1	Link 1 dimension of the robotic manipulator	l_2	Link 2 dimension robotic manipulator.
l_3	Link 3 dimension robotic manipulator	q_1	Angular displacement associated to the joint 1
q_2	Angular displacement associated to the joint 2	q_3	Angular displacement associated to the joint 3
m_{auv}	Mass associated to the aerial platform	m_1	Mass of the link 1 of the robotic manipulator
m_2	Mass of the link 2 of the robotic manipulator	m_3	Mass of the link 3 of the robotic manipulator
q_ψ	Angular rotation of the aerial platform	K_1	Conversion factor between torque and forces
b_1	Conversion factor between torque and forces	K_{ap}	Torque conversion factor in the aerial platform
K_{bp}	Electromotive factor of the aerial platform	R_{ap}	Electric resistance of the aerial platform
K_a	Torque conversion factor in the robotic arm	K_b	Electromotive factor of the robotic arm
R_b	Electric resistance of the robotic arm	(a_1, a_2, a_3, a_4)	Conversion factors of the linear to angular velocities
$(K_{p1}, K_{p2}, K_{p3}, K_{p4})$	Proportional gains associated to the aerial platform	K_p	Proportional gain associated to the robotic arm
$(K_{d1}, K_{d2}, K_{d3}, K_{d4})$	Derivative gains associated to the aerial platform	K_d	Derivative gain associated to the robotic arm
I_{uav}	Inertial value of the aerial platform		

The acceleration of generalized coordinates can be formulated as $\ddot{\mathbf{q}} = \mathbf{S}(\mathbf{q})\dot{\mathbf{v}} + \dot{\mathbf{S}}(\mathbf{q})\mathbf{v}$, where $\dot{\mathbf{v}} = \begin{bmatrix}\dot{\mathbf{v}}_u^\intercal & \dot{\mathbf{v}}_a^\intercal\end{bmatrix}^\intercal$. The dynamic model, formulated earlier, can be expressed as follows:

$$\mathbf{M}(\mathbf{q})\mathbf{S}(\mathbf{q})\dot{\mathbf{v}} + \mathbf{M}(\mathbf{q})\dot{\mathbf{S}}(\mathbf{q})\mathbf{v} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\mathbf{S}(\mathbf{q})\mathbf{v} + \mathbf{g}(\mathbf{q}) = {}^{i}\boldsymbol{f}(t) \tag{10}$$

On the other hand, the generalized force and torque of the dynamic model, denoted as ${}^{i}\boldsymbol{f}(t)$, must be formulated as a function of the reference velocities $\boldsymbol{\nu}_\mathbf{d}$. Consequently, these force and torque vectors are expressed in the body frame $< \mathcal{B} >$ attached to the UAV, represented as ${}^{b}\boldsymbol{f}(t)$, and the result is as follows:

$${}^{i}\boldsymbol{f}(t) = {}^{i}\mathbf{R}_b(\mathbf{q})\,{}^{b}\boldsymbol{f}(t) \tag{11}$$

where ${}^{i}\mathbf{R}_e(\mathbf{q})$ is a matrix that includes the rotation from the frame $< \mathcal{I} >$ to $< \mathcal{B} >$ and an identity matrix relate to the torque vector of each motor in the robotic arm.

The force and torque vector ${}^{b}\boldsymbol{f}$ expressed is the result of the torque in each motor of the aerial platform and the robotic arm expressed as $\tau = \begin{bmatrix}\tau_{w1} & \tau_{w2} & \tau_{w3} & \tau_{w4} & \tau_{q_1} & \tau_{q_2} & \tau_{q_3}\end{bmatrix}^\intercal$, which results in the following formulation:

$${}^{b}\boldsymbol{f}(t) = \mathbf{A}\tau(t) \tag{12}$$

where $\mathbf{A}$ is a matrix that allows the transformation of the internal torque of each motor presented in the entire system $\tau(t)$ to the generalized forces ${}^{b}\boldsymbol{f}(t)$.

Considering that the motors of the aerial platform and the robotic manipulator share the same features, the motor actuator model can be written in a matrix form as follows:

$$\tau = \mathbf{B}\mathbf{v}_\omega - \mathbf{P}\omega \tag{13}$$

where $\mathbf{v}_\omega = \begin{bmatrix}V_{w_1} & V_{w_2} & V_{w_3} & V_{w_4} & V_{q_1} & V_{q_2} & V_{q_3}\end{bmatrix}^\intercal$ is the vector that includes the voltage presented in each motor of the aerial platform and robotic arm; additionally, the angular velocities of each motor can be expressed in $\omega = \begin{bmatrix}\omega_{u_1} & \omega_{u_2} & \omega_{u_3} & \omega_{u_4} & \omega_{q_1} & \omega_{q_2} & \omega_{q_3}\end{bmatrix}^\intercal$. The mechanical characteristics of the DC motors within the aerial platform and the robotic arm can be effectively summarized using matrices $\mathbf{B}$ and $\mathbf{P}$.

The angular velocities of each motor ω should be formulated as a function of the generalized velocities of the entire system $\mathbf{v}$, this work assumes a linear mapping that can be written in a compact form as:

$$\mathbf{v} = \mathbf{E}\omega \tag{14}$$

where $\mathbf{E}$ includes the values that guarantees the mapping defined before. Furthermore, this work takes into account a linear mapping of the voltage required for each motor to the overall voltage produced for the reference velocities. This linear mapping can be expressed as follows:

$$\mathbf{v}_s = \mathbf{D}\mathbf{v}_\omega \tag{15}$$

where $\mathbf{v}_s = \begin{bmatrix}V_{\mu_l} & V_{\mu_m} & V_{\mu_n} & V_\omega & V_{q_1} & V_{q_2} & V_{q_3}\end{bmatrix}^\intercal$ represents the vector of voltages corresponding to the generalized velocities of the system, and $\mathbf{D}$ is a constant matrix that facilitates the mapping described earlier.

Finally, the low-level controller, which ensures the reference velocity tracking of the aerial robotic manipulator, can be formulated in a compact form as follows:

$$\mathbf{v}_s = \mathbf{F}\,\mathbf{v}_\mathbf{d} - \mathbf{F}\mathbf{v} - \mathbf{H}\dot{\mathbf{v}} \tag{16}$$

where $\mathbf{v}_\mathbf{d} = \begin{bmatrix}\mu_{ld} & \mu_{md} & \mu_{nd} & \omega_d & \dot{q}_{1d} & \dot{q}_{2d} & \dot{q}_{3d}\end{bmatrix}^\intercal$ is the vector of desired maneuverability velocities, $\dot{\mathbf{v}} = \begin{bmatrix}\dot{\mu}_l & \dot{\mu}_m & \dot{\mu}_n & \dot{\omega} & \ddot{q}_1 & \ddot{q}_2 & \ddot{q}_3\end{bmatrix}^\intercal$ is the acceleration of the generalized velocities of the entire system and finally $\mathbf{F}$ and $\mathbf{H}$ are gain matrices associated with proportional and derivative error.

Under the formulations presented before, it is possible to formulate the generalized forces of the dynamic model as follows:

$${}^{i}\boldsymbol{f}(t) = {}^{i}\mathbf{R}_e\mathbf{A}(\mathbf{B}\mathbf{D}^{-1}(\mathbf{F}\,\mathbf{v}_\mathbf{d} - \mathbf{F}\boldsymbol{\nu} - \mathbf{H}\dot{\boldsymbol{\nu}}) - \mathbf{P}\mathbf{E}^{-1}\mathbf{v}) \tag{17}$$

Finally, combining (17) and (10) is possible to formulate the dynamic model of the entire system in the reference velocity space, which includes the low-level controller

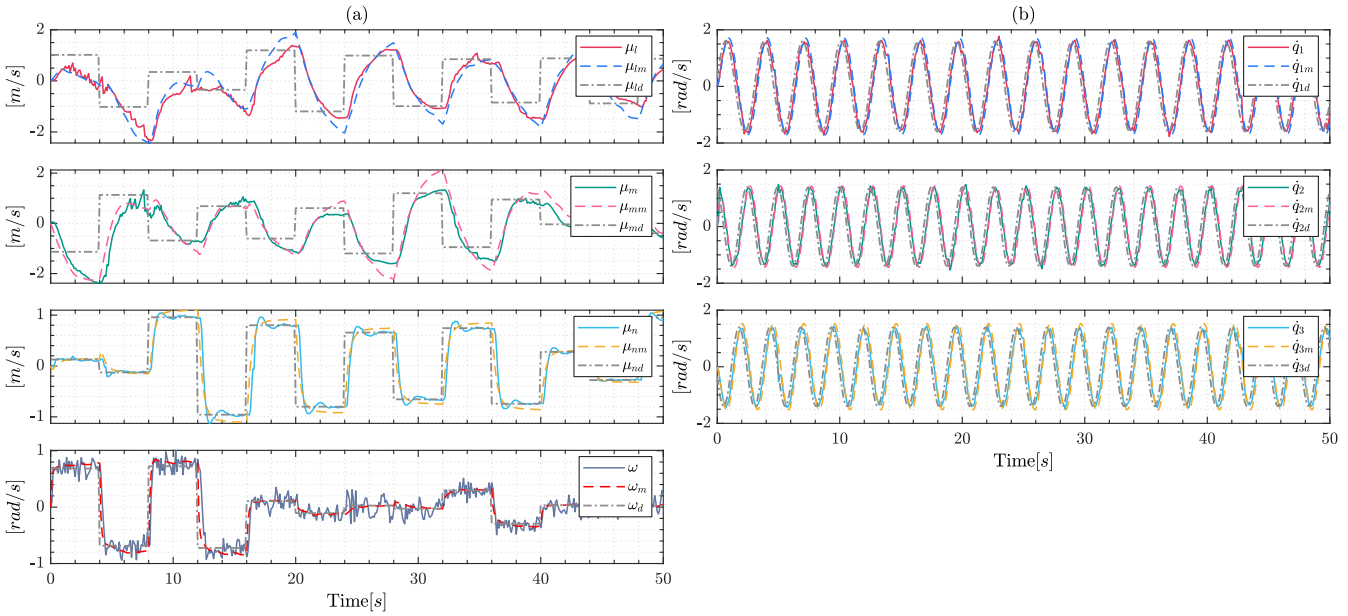

FIGURE 2. Validation procedure of the Aerial Robotic Manipulator,(a) shows the validation signals of the aerial platform and (b) validation signals associated to the robotic arm.

presented in commercial platforms; taking this into account, the formulation can be written as follows:

$$\bar{\mathbf{M}}(\mathbf{q})\dot{\mathbf{v}}(t) + \bar{\mathbf{C}}(\mathbf{q}, \dot{\mathbf{q}})\mathbf{v}(t) + \bar{\mathbf{g}}(\mathbf{q}) = \mathbf{v_d}(t) \tag{18}$$

where $\bar{\mathbf{M}}(\mathbf{q}) = ({}^i\mathbf{R}_b\mathbf{AB}(\mathbf{D})^{-1}\mathbf{F})^{-1}(\mathbf{M}(\mathbf{q})\mathbf{S} + {}^i\mathbf{R}_b\mathbf{AH}) \in \mathcal{R}^{7\times7}$ is the new mass and inertia matrix; $\bar{\mathbf{C}}(\mathbf{q}, \dot{\mathbf{q}}) = ({}^i\mathbf{R}_b\mathbf{AB}(\mathbf{D})^{-1}\mathbf{F})^{-1}(\mathbf{M}\dot{\mathbf{S}} + \mathbf{CS} - {}^i\mathbf{R}_b\mathbf{APE}^{-1}) \in \mathcal{R}^{7\times7}$ is the new Coriolis matrix and $\bar{\mathbf{g}}(\mathbf{q}) = ({}^i\mathbf{R}_b\mathbf{AB}(\mathbf{D})^{-1}\mathbf{g}(\mathbf{q})$ is the gravitational vector of the entire system. The matrices mentioned above include the mechanical, electrical and PD parameters of the aerial robotic manipulator. For further details regarding the matrices discussed, please refer to the Appendix at the end of this paper. Additionally, for a comprehensive explanation of each variable, consult Table 1.

B. AERIAL ROBOTIC MANIPULATOR DYNAMIC MODEL IDENTIFICATION AND VALIDATION

1) SYSTEM IDENTIFICATION OF THE SIMPLIFIED DYNAMIC MODEL

While the dynamic model presented earlier (18) could potentially rely on values provided by companies that develop robotic platforms, this work employs optimization techniques in the identification procedure of the dynamic model. This approach ensures the proper behavior of the proposed open-access platform.

An essential characteristic of the dynamic model formulated in (18) is its linearity representation with the vector ζ, expressed as follows:

$$\mathbf{Y}(\mathbf{q}, \dot{\mathbf{v}}, \mathbf{v})\zeta = \mathbf{v_d}(t) \tag{19}$$

where $\zeta = \begin{bmatrix}\zeta_1 & \zeta_2 & .. & \zeta_{32}\end{bmatrix}^\intercal \in \mathbf{R}^{32}$ represents a compact combination of the mechanical, electrical, and PD parameters. Finally, $\mathbf{Y}(\mathbf{q}, \dot{\mathbf{v}}, \mathbf{v})$ is a regressor matrix that is a function of the internal states $\mathbf{q}$, general velocities $\mathbf{v}$ and the acceleration of the system $\dot{\mathbf{v}}$.

This work estimates the values of the vector of unknown parameters ζ under the following formulation:

$$\ell_m(\mathbf{q}_i, \mathbf{v}_i, \dot{\mathbf{v}}_i, \mathbf{v_d}_i, \zeta) = (1-\alpha)\|\mathbf{v_d}_i - \mathbf{Y}(\mathbf{q}_i, \dot{\mathbf{v}}_i, \mathbf{v}_i)\zeta\|_2^2 + \alpha\|\zeta\|_1 \tag{20}$$

which represents the identification cost error, where i is the instant of measurement along the experimental time and $\alpha \in [0, 1)$ is an auxiliary variable that guarantees the minimum number of parameters ζ during the identification process.

The cost function to be optimized can be written as:

$$V_m(\mathbf{q}_i, \mathbf{v}_i, \dot{\mathbf{v}}_i, \mathbf{v_d}_i, \zeta) = \sum_{i=0}^{t_f} \ell_m(\mathbf{q}_i, \mathbf{v}_i, \dot{\mathbf{v}}_i, \mathbf{v_d}_i, \zeta) \tag{21}$$

where t_f is the total number of experimental snapshots used during the identification procedure, using the equation (21) this works formulates the optimization problem as follows:

$$\min_{\zeta} \quad V_m(\mathbf{q}_i, \mathbf{v}_i, \dot{\mathbf{v}}_i, \mathbf{v_d}_i, \zeta) \tag{22}$$

The formulation presented before can be solved using a wide range of optimization algorithms. In this work, we employ the well-known sequential quadratic programming (SQP), a commonly used method in most optimization solvers. The parameters that minimized the identification procedure for the aerial robotic manipulator are listed in Table 3. Further details and a physical representation of these parameters can be found in Table 2.

TABLE 2. Packed system identification parameters.

Symbolic variable	Parameters	Symbolic variable	Parameters
ζ_1	$(R_{ap}(m_1+m_2+m_3+m_{uav}-K_1K_{d1}))/(4K_1K_{ap}K_{p1})$	ζ_2	$-(K_{d2}R_{ap})/(4K_{ap}K_{p1})$
ζ_3	$(K_{d3}R_{ap})/(4K_{ap}K_{p1})$	ζ_4	$R_{ap}/(4K_1K_{ap}K_{p1})$
ζ_5	l_2m_2	ζ_6	l_2m_3
ζ_7	l_3m_3	ζ_8	K_1K_{d4}
ζ_9	$-(K_{d1}R_{ap})/(4K_{ap}K_{p2})$	ζ_{10}	$(R_{ap}(m_1+m_2+m_3+m_{uav}+K_1K_{d2}))/(4K_1K_{ap}K_{p2})$
ζ_{11}	$(K_{d3}R_{ap})/(4K_{ap}K_{p2})$	ζ_{12}	$R_{ap}/(4K_1K_{ap}K_{p2})$
ζ_{13}	$(K_{d1}R_{ap})/(4K_{ap}K_{p3})$	ζ_{14}	$(K_{d2}R_{ap})/(4K_{ap}K_{p3})$
ζ_{15}	$(R_{ap}(m_1+m_2+m_3+m_{uav}+K_1K_{d3}))/(4K_1K_{ap}K_{p3})$	ζ_{16}	$(K_{d4}R_{ap})/(4K_{ap}K_{p3})$
ζ_{17}	$-R_{ap}/(4K_1K_{ap}K_{p3})$	ζ_{18}	$R_{ap}/(4K_{ap}K_{p4}b_1)$
ζ_{19}	$K_{d1}b_1$	ζ_{20}	$K_{d2}b_1$
ζ_{21}	$(K_{d3}R_{ap})/(4K_{ap}K_{p4})$	ζ_{22}	$2I_{uav}-2K_{d4}b_1$
ζ_{23}	$l_2\zeta_5+l_2\zeta_6+l_3\zeta_7$	ζ_{24}	$R_a/(K_aK_p)$
ζ_{25}	K_d	ζ_{26}	$-(K_{bp}-4K_{p1}a_1)/(4K_{p1}a_1)$
ζ_{27}	$-(m_1+m_2+m_3+m_{uav})$	ζ_{28}	$-(K_{bp}-4K_{p2}a_1)/(4K_{p2}a_1)$
ζ_{29}	$-(K_{bp}-4K_{p3}a_3)/(4K_{p3}a_3)$	ζ_{30}	$(K_{ap}b_1(K_{bp}-4K_{p4}a_4))/(R_{ap}a_4)$
ζ_{31}	$(K_a(K_b-K_p))/R_a$	ζ_{32}	$-(K_b-K_p)/K_p$

2) SYSTEM VALIDATION OF THE SIMPLIFIED DYNAMIC MODEL

This section presents the validation results of the identified model previously formulated, wherein the mathematical formulation ensures sufficient accuracy for the open platform simulator of aerial robotic manipulators. In addition, the results of the validation procedure are shown in Figure 2, where $\mathbf{v_d} = \begin{bmatrix}\mu_{ld} & \mu_{md} & \mu_{nd} & \omega_d & \dot{q}_{1d} & \dot{q}_{2d} & \dot{q}_{3d}\end{bmatrix}^\intercal$ represents the desired velocities for the aerial platform and the robotic arm; the snapshot measurements are represented in the vector $\mathbf{v} = \begin{bmatrix}\mu_l & \mu_m & \mu_n & \omega & \dot{q}_1 & \dot{q}_2 & \dot{q}_3\end{bmatrix}^\intercal$ and finally the formulated dynamic model and its evolution during the validation is shown in the vector $\mathbf{v}_m = \begin{bmatrix}\mu_{lm} & \mu_{mm} & \mu_{nm} & \omega_m & \dot{q}_{1m} & \dot{q}_{2m} & \dot{q}_{3m}\end{bmatrix}^\intercal$.

TABLE 3. Aerial manipulator identification parameters ζ.

Parameters		
$\zeta_1 = 0.76$	$\zeta_2 = 0.10$	$\zeta_3 = 0.15$
$\zeta_4 = 38.60$	$\zeta_5 = -0.25$	$\zeta_6 = 0.25$
$\zeta_7 = 0.008$	$\zeta_8 = 0.003$	$\zeta_9 = -0.06$
$\zeta_{10} = 0.60$	$\zeta_{11} = -0.26$	$\zeta_{12} = 28.6$
$\zeta_{13} = 0.09$	$\zeta_{14} = -0.016$	$\zeta_{15} = 0.2$
$\zeta_{16} = 0.076$	$\zeta_{17} = 0.019$	$\zeta_{18} = 0.05$
$\zeta_{19} = -0.40$	$\zeta_{20} = 0.66$	$\zeta_{21} = 0.02$
$\zeta_{22} = 3.10$	$\zeta_{23} = -0.064$	$\zeta_{24} = 0.06$
$\zeta_{25} = 2.74$	$\zeta_{26} = 0.496$	$\zeta_{27} = 0.01$
$\zeta_{28} = -0.491$	$\zeta_{29} = -0.867$	$\zeta_{30} = -15.8$
$\zeta_{31} = -13.34$	$\zeta_{32} = -0.802$	

The results of the validation process demonstrate the precision of the proposed dynamic model and the ability to simulate the behavior of the real world aerial robotic manipulator; additionally, the identification cost error during the validation procedure reflected a value $V_m(\mathbf{q}_i, \mathbf{v}_i, \dot{\mathbf{v}}_i, \mathbf{v}_{\mathbf{d}i}, \zeta) = 10.8023$ demonstrating the effectiveness of the proposal, which will be used in the simulator platform.

III. SIMULATION PLATFORM

Although aerial robotic manipulator platforms are widely used in a wide range of applications, they are still considered expensive systems, even for universities and research institutes. Due to the high prices of these systems, control algorithms must be evaluated in simulation to avoid accidents related to the experimental results. This work proposes a simulation platform for aerial robotic manipulators that allows the evaluation of control algorithms offering advantages to the users, such as evaluation and testing a wide range of control strategies, reducing time related to the development stage and the possibility of performing experiments before the hardware implementation.

The open access simulation platform was developed under the MIL architecture, which includes the identified system dynamics, modularity, and reuse of accessible code without additional effort for the platform users. The proposal was implemented using Python, ROS framework and Unity 3D considering the following parts of the entire system: *(i) system model* the mathematical formulation of the aerial robotic manipulator was implemented using the Python language, allowing users to get on board quickly and easily reuse; *(ii) Graphical interaction and interprocess communication* was implemented using the Unity 3D graphics engine, which guarantees compatibility with different 3D model formats, low latency communication, and the versatility to interact with various software and devices; furthermore, this work utilized ROS to facilitate communication between Unity 3D and Python, ensuring exact time synchronization. The internal information of the open platform simulation can be easily exchanged between different frameworks and languages such as Matlab, Julia, and C++ due to the capabilities that ROS offers to manage communications.

A. SIMULATED DIFFERENTIAL KINEMATICS AND DYNAMICS

The simulated differential kinematics and dynamics of the aerial robotic manipulator was developed using the model in

the loop (MIL) formulation, where this work build a ROS node which include the identified dynamic model presented in Section (II) in order to publish the robot's states to the ROS framework. Therefore, this work combines the mathematical models presented in equations (4) and (18) to generate a generalized representation of the entire system, resulting:

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{v_d}(t))$$
$$\mathbf{f}(\mathbf{x}, \mathbf{v_d}) = \begin{bmatrix} \mathbf{J}(\mathbf{q})\mathbf{v} \\ \mathbf{J}_u(\mathbf{q})\mathbf{v} \\ \bar{\mathbf{M}}^{-1}(\mathbf{q})\left(\mathbf{v_d} - \bar{\mathbf{C}}(\mathbf{q}, \dot{\mathbf{q}})\mathbf{v} - \bar{\mathbf{g}}(\mathbf{q})\right) \end{bmatrix} \tag{23}$$

where $\mathbf{x} = \begin{bmatrix} {}^i\mathbf{r}_e & {}^i\mathbf{r}_b & \mathbf{q_a} & \mathbf{v} \end{bmatrix} \in \mathcal{R}^{18}$ is the generalized vector of the entire system and $v_\mathbf{d} \in \mathcal{R}^7$ is the input vector of the system which represents the reference velocities for the aerial platform and the robotic manipulator. Additionally, the formulation presented before uses the matrix $\mathbf{J}_u$ in order to include the states of the aerial platform ${}^i\mathbf{r}_b$ and the angular displacement of each joint $\mathbf{q_a}$.

This work uses a discrete version of the equation presented in (23), which was developed using the fourth-order Runge-Kutta integrator, defined as follows:

$$\mathbf{x}_{k+1} = \mathbf{f}_k(\mathbf{x}_k, \mathbf{v_d}_k) \tag{24}$$

the formulation presented before guarantee the stable behavior of the simulation through the experiments.

The discrete formulation is available in the following repository GitHub providing a comprehensive resource for those interested in developing aerial robotic manipulators.

B. GRAPHICAL INTERACTION AND INTER-PROCESS COMMUNICATION

This section presents the description of each step in developing the aerial robotic manipulator visualization interface, where this work sets the graphical interaction and the inter-process communication section in multiple steps; more details are presented in Figure 3. The first step *(a) CAD modeling* consists in designing the aerial manipulator using CAD software, which ensures an accurate representation of the real-world system in the simulator platform. After completing the design, the polygon count of the system's elements is optimized using Autodesk 3ds Max. The final model is imported into Unity3D, where it can be used for simulation and visualization proposals.

The second step, denoted as *(b) in Unity3D*, involves the development of the virtual environment within the simulator platform. This ensures the visualization of the entire system during the testing of control algorithms. This step also encompasses the integration of all necessary sub-modules essential for the visualizer to connect with the MIL scheme, including the ROSbridge server. Finally, this work generated the respective binary files for the visualizer to be executed on a computer under Linux architectures.

Furthermore, this work developed the interface shown in Figure 4, which allows users to configure the ROS framework and set the topics for publishing and subscribing information `\aerial_manipulator\odom` and `\aerial_manipulator\joints`. This user interface provides an intuitive way to configure the simulator and set up the communication between ROS and external systems. Taking this into account, users can establish the data flow between *(d) MIL Scheme* and *(b) Unity3D*.

The implementation of the controller discussed in Section IV is incorporated into sub-module (c) named *Scientific Programming Software*. In this case, Matlab is employed to showcase the versatility of the proposed open-access simulator. This section generates a node directly from Matlab, ensuring control over the entire system. Users can easily modify the design parameters within the controller and verify the system's behavior under these modifications.

The simulated kinematics and dynamics of the aerial robotic manipulator, as discussed in Section III-A, are integrated into *(d) Submodule MIL scheme*. In this submodule, a ROS node in Python language is generated as part of this work. This node incorporates all the relevant information pertaining to the entire system, including parameters such as sample time t_s, initial conditions of the simulator, and the execution time of the system.

As shown in Figure 3, communication between submodules was established using ROS, which allowed for proper synchronization and the exchange of information between different software and languages. Therefore, the visualizer inside Unity3D framework provides a powerful platform for testing the simulated robot's behavior, where developers can efficiently design and test robotic systems in a simulated environment, reducing the time and cost required for physical testing.

IV. CONTROL ALGORITHM FORMULATION

In this section, the controller formulation based on the differential kinematics of the entire system (4) is presented. The control formulation tries to achieve the desired position of the end effector generating the maneuverability velocities for the entire system.

This work uses the concepts of non-linear control in order to generate the adequate law that guarantees the stability of the system along the desired pose defined by the users.

This work defines a Lyapunov $V({}^i\boldsymbol{r}_e) : \mathcal{R}^4 \to \mathcal{R}_{\geq 0}$, which includes the control error generated between the desired reference ${}^i\boldsymbol{r}_e^d$ and the position of the end effector, and can be expanded as follows.

$$V({}^i\boldsymbol{r}_e) = \frac{1}{2}\, {}^i\tilde{\boldsymbol{r}}_e^{\mathsf{T}}\, \mathrm{o}\, {}^i\tilde{\boldsymbol{r}}_e \tag{25}$$

where ${}^i\tilde{\boldsymbol{r}}_e = \left[\, {}^i\boldsymbol{r}_e^d - {}^i\boldsymbol{r}_e \right]^{\mathsf{T}}$ and o is a positive definite matrix. Applying the derivative in (25) the candidate Lyapunov function can be written as follows:

$$\dot{V}({}^i\boldsymbol{r}_e) = {}^i\tilde{\boldsymbol{r}}_e^{\mathsf{T}}\, \mathrm{o}\, {}^i\dot{\tilde{\boldsymbol{r}}}_e \tag{26}$$

where ${}^i\dot{\tilde{\boldsymbol{r}}}_e = {}^i\dot{\boldsymbol{r}}_e^d - \mathbf{J}(\mathbf{q})\mathbf{v}$. According to the Lyapunov stability criterion and concepts of nonlinear control, this work

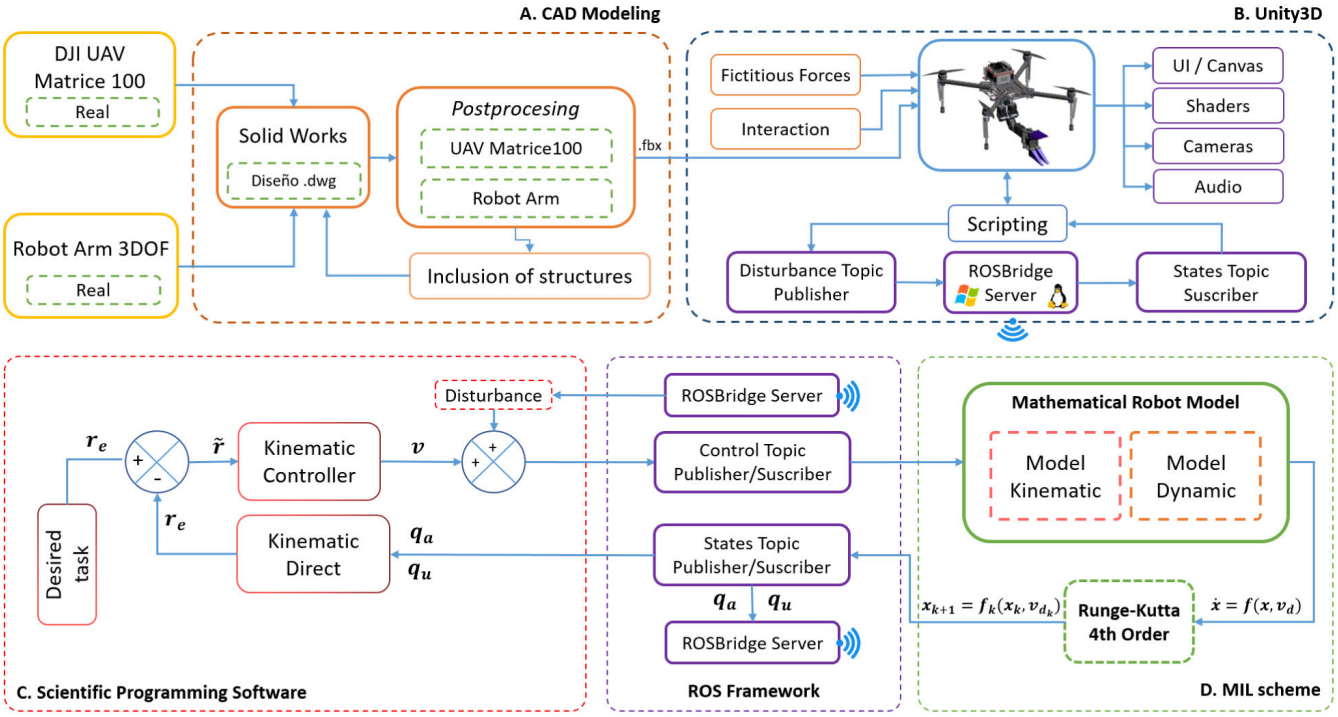

FIGURE 3. Open-access Simulator Scheme.

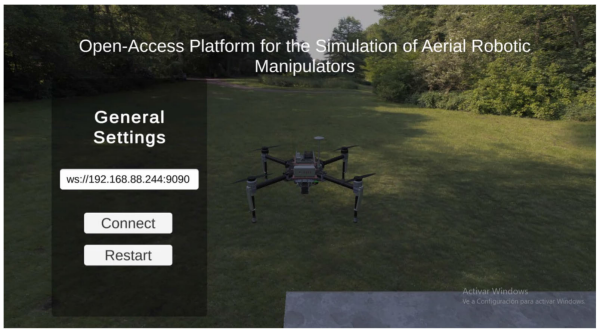

FIGURE 4. Open access simulator user interface.

formulated a control law that guarantees that $\dot{V}({}^{i}\boldsymbol{r}_e) \leq 0$, the control law can be written as follows:

$$\boldsymbol{v}_c = \mathbf{J}^{\dagger}(\mathbf{q})({}^{i}\dot{\boldsymbol{r}}_e^{d} + \mathbf{K}_2 \tanh(\mathbf{K}_2^{-1}\mathbf{K}_1 {}^{i}\tilde{\boldsymbol{r}}_e)) \tag{27}$$

where $\tanh(\cdot)$ is a saturation function, $\mathbf{K}_1 > 0$ is a diagonal matrix that defines the slopes of the saturation function, $\mathbf{K}_2 > 0$ is a diagonal matrix that represents the maximum and minimum velocities generated by the controller; additionally, this work assumes that $\boldsymbol{v}_c \equiv \boldsymbol{v}$. Under the formulation presented before the derivative of the Lyapunov function can be defined negative and written as follows:

$$\dot{V}({}^{i}\boldsymbol{r}_e) = -{}^{i}\dot{\tilde{\boldsymbol{r}}}_e^{\mathsf{T}} \mathbf{o} \mathbf{K}_2 \tanh(\mathbf{K}_2^{-1}\mathbf{K}_1 {}^{i}\tilde{\boldsymbol{r}}_e) \tag{28}$$

which guarantees either the stability of the controller or the entire system as well. The controller was used to demonstrate the behavior of the open access platform for aerial robotic manipulators, and the results of these experiments are presented in the next section.

V. SIMULATION RESULTS

This section presents the results obtained through the proposed open access simulator for the aerial robotic manipulator. In addition, it briefly introduces how users can access information within the simulator to develop additional formulations for control algorithms. The hardware used during simulation experiments using the simulator is described as follows: Intel processor i7-7700HQ, CPU 2.80GHz × 8, and Graphics GeForce GTX 1050. The performance of the simulation platform, especially the system dynamics, is not strictly related to the hardware characteristics. However, to obtain smooth animation, it is highly recommended to include specialized graphics hardware.

The node controller developed in Matlab has access to the internal states of the simulator throw topics `\aerial_manipulator\odom` and `\aerial_manipulator\joints`, which are available in the ROS framework with a publication rate of 20 *Hz*. On the other hand, the control node updates the desired reference velocities at 10 *Hz* to modify the position of the end effector. Communication between the controller node and the simulator is possible through the following topics within the ROS framework `\aerial_manipulator\cmd_vel` and `\aerial_manipulator\joints_ref`. More details of the communication structure are presented in Figure 5.

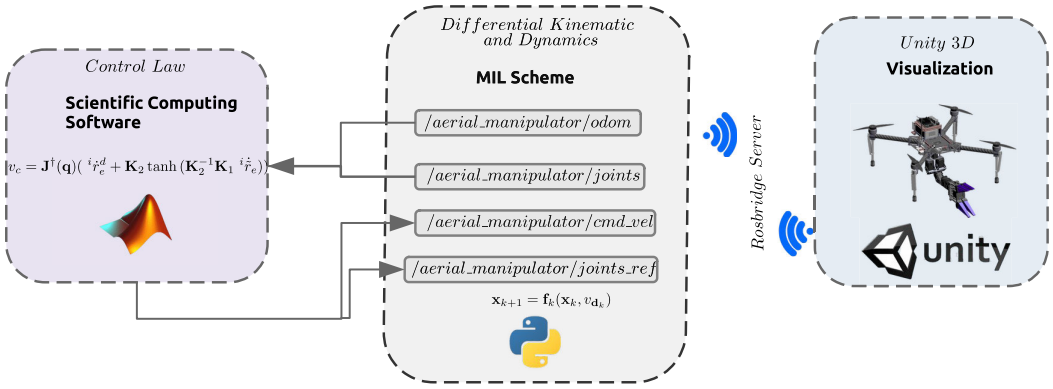

FIGURE 5. Inter-process communication.

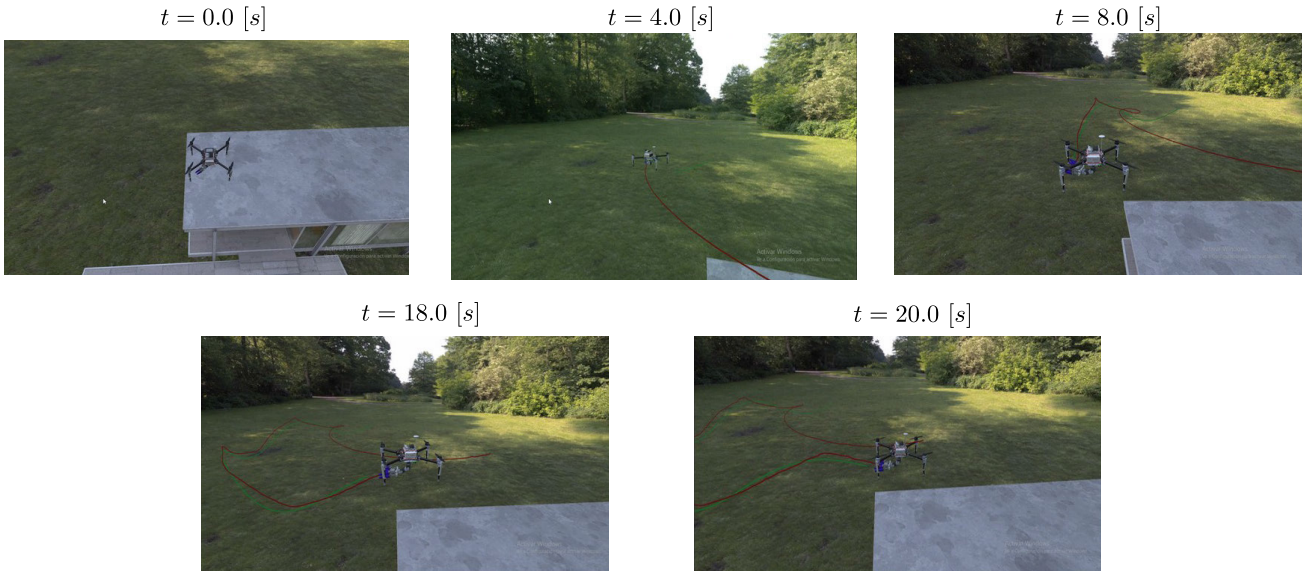

FIGURE 6. Simulation results using the open-access platform for aerial robotic manipulators.

Based on the considerations presented above, the results of the proposed controller using the open access simulation platform are presented in Figure 7(a), where the end effector follows the desired reference trajectory ${}^{i}\dot{\mathbf{r}}_e^d$, the location of the end effector is represented as ${}^{i}\mathbf{r}_e$ and the aerial platform is represented by ${}^{i}\mathbf{r}_b$. Furthermore, the control error vector is presented in Figure 7(b), which is defined by ${}^{i}\tilde{\mathbf{r}}_e = {}^{i}\mathbf{r}_e^d - {}^{i}\mathbf{r}_e$ considering that $({}^{i}\tilde{r}_x,\ {}^{i}\tilde{r}_y,\ {}^{i}\tilde{r}_z)$ is the error on each axis.

The visualization results using the proposed open access simulator are presented in Figure 6, where the proposal can solve the problems associated with the implementation of control algorithms in real-world robotic systems, guaranteeing users the development of new research proposals aimed at autonomous or semi-autonomous control in the area of aerial robotic manipulators.

The proposed open-access simulator showcases its ability to control ARMs through various experimental tests, verifying the stability and robustness of the controller. In numerous instances, the acquisition of aerial robotics manipulators poses a significant financial burden for universities and research institutes. Moreover, experimental tests are deemed risky, not only for the user but also for the entire system. However, the developed platform fulfills essential requirements, including system dynamics, modularity, and accessibility to code reuse without additional effort, features that are often lacking in many simulator systems.

Finally, due to the system dynamics identification procedure, the controller proposed in Section IV cannot control the system perfectly, which guarantees how the simulator can be used to verify the performance of different control laws. This work presented a video in which the user can verify the open access simulator proposal for the aerial robotic manipulator in the following link Video.

VI. DISCUSSION

The parameters identified in a robotic system are important not only to guarantee an accurate behavior of the simulator but also to ensure robust control algorithms, especially in

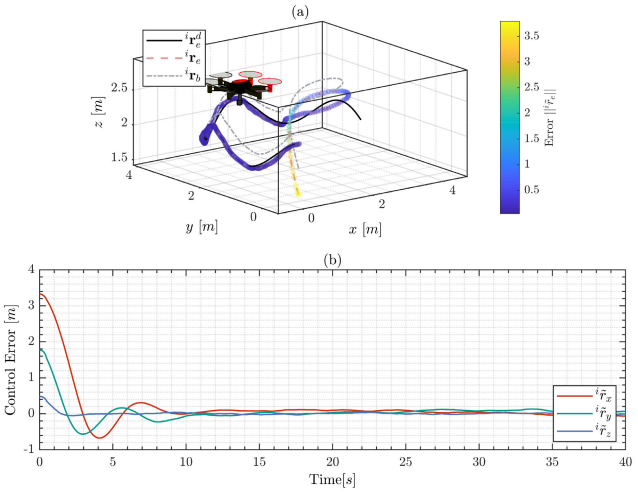

FIGURE 7. Simulation results using the open-access simulator and the proposed control law, (a) shows the evolution of system under the proposed controller and (b) presents the control error during the experiment.

the robotic field, where systems are exposed to external forces such as loads or disturbances. In this context, manipulator robots have generated interest by proposing new algorithms for the accurate identification of dynamic parameters [32]. Similarly, for the field of UAVs, there are proposals for the non-linear identification of parameters for aerial platform vehicles [33]. Recently, the combination of manipulator robots and UAVs has been used in many areas that allow robust and accurate manipulation tasks. Although few proposed models still consider the dynamics of the aerial manipulator, the dynamic models were developed only through a physics engine simulator [34]. However, our dynamic model has been validated against a real aerial manipulator, ensuring an accurate response similar to real-world scenarios.

Dynamic models are widely used to develop simulators, allowing the user to validate new control algorithms [35]. In general, the dynamic of robots is integrated into the physics engine, and researchers have compared various models to achieve optimal performance. However, there are still many factors in the area of simulator development that require further investigation, as mentioned in [18]. Specifically, in the field of UAVs, several simulators are available to evaluate flight controllers in both simple and complex tasks, where the simulator provides realistic environments that consider various aspects, including dynamics, aerodynamics, multi-robot systems, communications, and more [36], [37], [38]. However, many of these simulators have limited access for researchers, imposing usage costs or restrictions on the base models incorporated within the simulator. In contrast, our simulator offers free access to the model and allows the evaluation of control algorithms in the aerial manipulator within a realistic environment.

Furthermore, the literature related to aerial manipulator simulators is scarce; for example, in the study presented in [39], a simulator is used to provide users with realistic haptic feedback during human-aerial manipulator interaction activities. Forces exchanged during the interaction are measured and then applied to a dynamically simulated aerial manipulator. In contrast to our proposal, this research is focused solely on this specific application and does not provide a detailed explanation of the dynamic model that governs the behavior of the aerial manipulator. With this justification, our work offers the scientific community the opportunity to evaluate aerial manipulation algorithms using different methods; additionally, the proposal guarantees modularity and access to code reuse without additional effort, where many simulator systems lack these features. Finally, it is important to note the limitations of our simulator, as it does not consider the aerodynamics of the system, which can be crucial in the design of certain applications.

VII. CONCLUSION

This paper describes an open-access simulator for aerial robotic manipulators based on the Model-in-Loop (MIL) architecture. On the other hand, the precision of the proposed simulator is based on the mathematical model of the entire system, which was developed using the concepts of Euler-Lagrange, parameter identification, and optimization techniques applied to the commercial aerial vehicle (Matrice-100) equipped with a 3DOF robotic arm. The simulator platform was developed based on Python, ROS and Unity, which guarantees the modularity of the entire system, shows the behavior and evolution over time, and manages the communication between different languages in a distributed system such as Matlab and Julia. This work validated the proposed simulation platform using a basic control algorithm applied to the trajectory tracking control problem, which guarantees that the platform allows users to simulate the behavior of aerial robotic manipulators and modify the control parameters of the system to evaluate performance in different situations. Finally, the proposal was released as an open-source platform, making the project accessible and easy to reuse code, which are essential requirements for simulation platforms. Future work includes the integration of collision detection and contact forces into the simulation platform, which can guarantee manipulation experiments with more advanced robotics applications.

APPENDIX

This section presents the extra-material of differential kinematics and dynamics of the aerial manipulator. Taking this into account the Jacobian matrix can be formulated as follow:

$$\mathbf{J} = \begin{bmatrix} c(q_\psi) & -s(q_\psi) & 0 & \mathbf{J}^{(1,4)} & \mathbf{J}^{(1,5)} & \mathbf{J}^{(1,6)} & \mathbf{J}^{(1,7)} \\ s(q_\psi) & c(q_\psi) & 0 & \mathbf{J}^{(2,4)} & \mathbf{J}^{(2,5)} & \mathbf{J}^{(2,6)} & \mathbf{J}^{(2,7)} \\ 0 & 0 & 1 & 0 & 0 & \mathbf{J}^{(3,6)} & \mathbf{J}^{(3,7)} \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 \end{bmatrix} \tag{A-1}$$

where:

$$\begin{aligned}
\mathbf{J}^{(1,4)} &= -s(q_\psi q_1)(l_3 c(q_2 q_3) + l_2 c(q_2)) \\
\mathbf{J}^{(1,5)} &= \mathbf{J}^{(1,4)} \\
\mathbf{J}^{(1,6)} &= -c(q_\psi q_1)(l_3 s(q_2 q_3) + l_2\, s(q_2)) \\
\mathbf{J}^{(1,7)} &= -l_3 c(q_\psi q_1) s(q_2 q_3) \\
\mathbf{J}^{(2,4)} &= c(q_\psi q_1)(l_3 c(q_2 q_3) + l_2 c(q_2)) \\
\mathbf{J}^{(2,5)} &= \mathbf{J}^{(2,4)} \\
\mathbf{J}^{(2,6)} &= -s(q_\psi q_1)(l_3 s(q_2 q_3) + l_2 s(q_2)) \\
\mathbf{J}^{(2,7)} &= -l_3 s(q_\psi q_1) s(q_2 q_3) \\
\mathbf{J}^{(3,6)} &= -l_3 c(q_2 q_3) - l_2 c(q_2) \\
\mathbf{J}^{(3,7)} &= -l_3 c(q_2 q_3)
\end{aligned}$$

The acceleration of the generalized coordinates used the matrix $\mathbf{S}(\mathbf{q})$ and its derivative $\dot{\mathbf{S}}(\mathbf{q})$ defined as:

$$\mathbf{S}(\mathbf{q}) = \begin{bmatrix} c(q_\psi) & -s(q_\psi) & 0 & 0 & 0 & 0 & 0 \\ s(q_\psi) & c(q_\psi) & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{(A-2)}$$

$$\dot{\mathbf{S}}(\mathbf{q}) = \begin{bmatrix} -\omega s(q_\psi) & -\omega c(q_\psi) & 0 & 0 & 0 & 0 & 0 \\ \omega c(q_\psi) & -\omega c(q_\psi) & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad \text{(A-3)}$$

The matrix $\mathbf{A}$ allows the mapping between the force and torque vector and the vector τ; the matrix can be written as follows:

$$\mathbf{A} = \begin{bmatrix} -K_1 & -K_1 & -K_1 & -K_1 & 0 & 0 & 0 \\ -K_1 & K_1 & K_1 & -K_1 & 0 & 0 & 0 \\ K_1 & K_1 & K_1 & K_1 & 0 & 0 & 0 \\ b_1 & b_1 & b_1 & b_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{(A-4)}$$

The actuator model of the entire system can be written in a compact form using the matrices $\mathbf{B}$ and $\mathbf{P}$, which are defined as follows:

$$\mathbf{B} = \begin{bmatrix} \frac{K_{ap}}{R_{ap}} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{K_{ap}}{R_{ap}} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{K_{ap}}{R_{ap}} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{K_{ap}}{R_{ap}} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{K_a}{R_a} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{K_a}{R_a} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{K_a}{R_a} \end{bmatrix} \quad \text{(A-5)}$$

$$\mathbf{P} = \begin{bmatrix} \frac{K_{ap}K_{bp}}{R_{ap}} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{K_{ap}K_{bp}}{R_{ap}} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{K_{ap}K_{bp}}{R_{ap}} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{K_{ap}K_{bp}}{R_{ap}} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{K_aK_b}{R_a} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{K_aK_b}{R_a} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{K_aK_b}{R_a} \end{bmatrix} \quad \text{(A-6)}$$

The generalized velocity of the entire system $\boldsymbol{\nu}$ can be formulated as a function of the angular velocities of each motor ω using the the matrix $\mathbf{E}$ defined as follows:

$$\mathbf{E} = \begin{bmatrix} -a_1 & -a_1 & a_1 & a_1 & 0 & 0 & 0 \\ -a_2 & a_2 & a_2 & -a_2 & 0 & 0 & 0 \\ a_3 & a_3 & a_3 & a_3 & 0 & 0 & 0 \\ a_4 & -a_4 & a_4 & -a_4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{(A-7)}$$

The linear mapping between the voltage require in each motor and the general voltage produced for the reference velocities can be implemented using the matrix $\mathbf{D}$, which can be written as follows:

$$\mathbf{D} = \frac{1}{4}\begin{bmatrix} -1 & -1 & 1 & 1 & 0 & 0 & 0 \\ -1 & 1 & 1 & -1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & -1 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 4 \end{bmatrix} \quad \text{(A-8)}$$

The gains of the reference velocities controller can be compacted written using the matrices $\mathbf{F}$ and $\mathbf{H}$, which can

be defined as:

$$\mathbf{F} = \begin{bmatrix} K_{p1} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & K_{p2} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & K_{p3} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & K_{p4} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & K_p & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & K_p & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & K_p \end{bmatrix} \quad \text{(A-9)}$$

$$\mathbf{H} = \begin{bmatrix} K_{d1} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & K_{d2} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & K_{d3} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & K_{d4} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & K_d & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & K_d & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & K_d \end{bmatrix} \quad \text{(A-10)}$$

The mass and inertia matrix of the simplified dynamics can be written as in (A-11), shown at the bottom of the page, where

$$\begin{aligned}
\bar{\mathbf{M}}^{(1,1)} &= \zeta_1 \\
\bar{\mathbf{M}}^{(1,2)} &= -\zeta_2 \\
\bar{\mathbf{M}}^{(1,3)} &= \zeta_3 \\
\bar{\mathbf{M}}^{(1,4)} &= -\zeta_4(c(q_2)s(q_1)\zeta_5 - \zeta_8 + c(q_2)s(q_1)\zeta_6 \\
&\quad + c(q_2)c(q_3)c(q_1)\zeta_7 - s(q_1)s(q_2)s(q_3)\zeta_7) \\
\bar{\mathbf{M}}^{(1,5)} &= -s(q_1)\zeta_4(c(q_2)\zeta_5 + c(q_2)\zeta_6 + c(q_2q_3)\zeta_7) \\
\bar{\mathbf{M}}^{(1,6)} &= -c(q_1)\zeta_4(s(q_2)\zeta_5 + s(q_2)\zeta_6 + s(q_2q_3)\zeta_7) \\
\bar{\mathbf{M}}^{(1,7)} &= -c(q_1)s(q_2q_3)\zeta_4\zeta_7 \\
\bar{\mathbf{M}}^{(2,1)} &= -\zeta_9 \\
\bar{\mathbf{M}}^{(2,2)} &= \zeta_{10} \\
\bar{\mathbf{M}}^{(2,3)} &= \zeta_{11} \\
\bar{\mathbf{M}}^{(2,4)} &= \zeta_{12}(c(q_1)c(q_2)\zeta_5 - \zeta_8 + c(q_1)c(q_2)\zeta_6 \\
&\quad + c(q_1)c(q_2)c(q_3)\zeta_7 - c(q_1)s(q_2)s(q_3)\zeta_7) \\
\bar{\mathbf{M}}^{(2,5)} &= c(q_1)\zeta_{12}(c(q_2)\zeta_5 + c(q_2)\zeta_6 + c(q_2q_3)\zeta_7) \\
\bar{\mathbf{M}}^{(2,6)} &= -s(q_1)\zeta_{12}(s(q_2)\zeta_5 + s(q_2)\zeta_6 + s(q_2q_3)\zeta_7) \\
\bar{\mathbf{M}}^{(2,7)} &= -s(q_1)s(q_2q_3)\zeta_7\zeta_{12} \\
\bar{\mathbf{M}}^{(3,1)} &= \zeta_{13} \\
\bar{\mathbf{M}}^{(3,2)} &= \zeta_{14} \\
\bar{\mathbf{M}}^{(3,3)} &= \zeta_{15} \\
\bar{\mathbf{M}}^{(3,4)} &= \zeta_{16} \\
\bar{\mathbf{M}}^{(3,6)} &= -\zeta_{17}(s(q_2)\zeta_5 + c(q_2)\zeta_6 + c(q_2q_3)\zeta_7) \\
\bar{\mathbf{M}}^{(3,7)} &= -c(q_2q_3)\zeta_7\zeta_{17} \\
\bar{\mathbf{M}}^{(4,1)} &= \zeta_{18}(\zeta_{19} - s(q_1)(c(q_2)\zeta_5 + c(q_2)\zeta_6 \\
&\quad + c(q_2q_3)\zeta_7)) \\
\bar{\mathbf{M}}^{(4,2)} &= -\zeta_{18}(\zeta_{20} - c(q_1)(c(q_2)\zeta_5 + c(q_2)\zeta_6 \\
&\quad + c(q_2q_3)\zeta_7)) \\
\bar{\mathbf{M}}^{(4,3)} &= \zeta_{21} \\
\bar{\mathbf{M}}^{(4,4)} &= (\zeta_{18}(\zeta_{22} + \zeta_{23} + 2\,c(q_3)l_3\zeta_6 + c(2q_2)l_2\zeta_5 \\
&\quad + c(2q_2)l_2\zeta_6 + 2\,c(2q_2q_3)l_3\zeta_6 + c(2q_22\,q_3)l_3\zeta_7))/2 \\
\bar{\mathbf{M}}^{(4,5)} &= (\zeta_{18}(\zeta_{23} + 2\,c(q_3)l_3\zeta_6 + c(2q_2)l_2\zeta_5 \\
&\quad + c(2q_2)l_2\zeta_6 + 2\,c(2q_2q_3)l_3\zeta_6 + c(2q_22\,q_3)l_3\zeta_7))/2 \\
\bar{\mathbf{M}}^{(5,1)} &= -s(q_1)\zeta_{24}(c(q_2)\zeta_5 + c(q_1)\zeta_6 + c(q_2q_3)\zeta_7) \\
\bar{\mathbf{M}}^{(5,2)} &= c(q_1)\zeta_{24}(c(q_2)\zeta_5 + c(q_1)\zeta_6 + c(q_2q_3)\zeta_7) \\
\bar{\mathbf{M}}^{(5,4)} &= (\zeta_{24}(\zeta_{23} + 2\,c(q_2)l_3\zeta_6 + c(2q_2)l_2\zeta_5 \\
&\quad + c(2q_2)l_2\zeta_6 + 2\,c(2q_2q_3)l_3\zeta_6 + c(2q_22\,q_3)l_3\zeta_7))/2 \\
\bar{\mathbf{M}}^{(5,5)} &= (\zeta_{24}(\zeta_{23} + 2\zeta_{25} + 2\,c(q_3)l_3\zeta_6 + c(2q_2)l_2\zeta_5 \\
&\quad + c(2q_2)l_2\zeta_6 + 2\,c(2q_2q_3)l_3\zeta_6 + c(2q_22\,q_3)l_3\zeta_7))/2 \\
\bar{\mathbf{M}}^{(6,1)} &= -c(q_1)\zeta_{24}(s(q_2)\zeta_5 + s(q_2)\zeta_6 + s(q_2q_3)\zeta_7) \\
\bar{\mathbf{M}}^{(6,2)} &= -s(q_1)\zeta_{24}(s(q_2)\zeta_5 + s(q_2)\zeta_6 + s(q_2q_3)\zeta_7) \\
\bar{\mathbf{M}}^{(6,3)} &= -\zeta_{24}(c(q_2)\zeta_5 + c(q_2)\zeta_6 + c(q_2q_3)\zeta_7) \\
\bar{\mathbf{M}}^{(6,6)} &= \zeta_{24}(\zeta_{23} + \zeta_{25} + 2\,c(q_3)l_3\zeta_6) \\
\bar{\mathbf{M}}^{(6,7)} &= \zeta_7\zeta_{24}(l_3 + c(q_3)l_2) \\
\bar{\mathbf{M}}^{(7,1)} &= -c(q_1)s(q_2q_3)\zeta_7\zeta_{24} \\
\bar{\mathbf{M}}^{(7,2)} &= -s(q_1)s(q_2q_3)\zeta_7\zeta_{24} \\
\bar{\mathbf{M}}^{(7,3)} &= -c(q_2q_3)\zeta_7\zeta_{24} \\
\bar{\mathbf{M}}^{(7,6)} &= \zeta_7\zeta_{24}(l_3 + c(q_3)l_2) \\
\bar{\mathbf{M}}^{(7,7)} &= \zeta_{24}(\zeta_{25} + l_3\zeta_7)
\end{aligned}$$

The Coriolis matrix can be compacted written as follows:

$$\bar{\mathbf{C}} = \begin{bmatrix} \bar{\mathbf{C}}^{(1,1)} & \bar{\mathbf{C}}^{(1,2)} & 0 & \bar{\mathbf{C}}^{(1,4)} & \bar{\mathbf{C}}^{(1,5)} & \bar{\mathbf{C}}^{(1,6)} & \bar{\mathbf{C}}^{(1,7)} \\ \bar{\mathbf{C}}^{(2,1)} & \bar{\mathbf{C}}^{(2,2)} & 0 & \bar{\mathbf{C}}^{(2,4)} & \bar{\mathbf{C}}^{(2,5)} & \bar{\mathbf{C}}^{(2,6)} & \bar{\mathbf{C}}^{(2,7)} \\ 0 & 0 & \bar{\mathbf{C}}^{(3,3)} & 0 & 0 & \bar{\mathbf{C}}^{(3,6)} & \bar{\mathbf{C}}^{(3,7)} \\ \bar{\mathbf{C}}^{(4,1)} & \bar{\mathbf{C}}^{(4,2)} & 0 & \bar{\mathbf{C}}^{(4,4)} & \bar{\mathbf{C}}^{(4,5)} & \bar{\mathbf{C}}^{(4,6)} & \bar{\mathbf{C}}^{(4,7)} \\ \bar{\mathbf{C}}^{(5,1)} & \bar{\mathbf{C}}^{(5,2)} & 0 & \bar{\mathbf{C}}^{(5,4)} & \bar{\mathbf{C}}^{(5,5)} & \bar{\mathbf{C}}^{(5,6)} & \bar{\mathbf{C}}^{(5,7)} \\ \bar{\mathbf{C}}^{(6,1)} & \bar{\mathbf{C}}^{(6,2)} & 0 & \bar{\mathbf{C}}^{(6,4)} & \bar{\mathbf{C}}^{(6,5)} & \bar{\mathbf{C}}^{(6,6)} & \bar{\mathbf{C}}^{(6,7)} \\ \bar{\mathbf{C}}^{(7,1)} & \bar{\mathbf{C}}^{(7,2)} & 0 & \bar{\mathbf{C}}^{(7,4)} & \bar{\mathbf{C}}^{(7,5)} & \bar{\mathbf{C}}^{(7,6)} & \bar{\mathbf{C}}^{(7,7)} \end{bmatrix} \quad \text{(A-12)}$$

$$\bar{\mathbf{M}}(\mathbf{q}) = \begin{bmatrix} \bar{\mathbf{M}}^{(1,1)} & \bar{\mathbf{M}}^{(1,2)} & \bar{\mathbf{M}}^{(1,3)} & \bar{\mathbf{M}}^{(1,4)} & \bar{\mathbf{M}}^{(1,5)} & \bar{\mathbf{M}}^{(1,6)} & \bar{\mathbf{M}}^{(1,7)} \\ \bar{\mathbf{M}}^{(2,1)} & \bar{\mathbf{M}}^{(2,2)} & \bar{\mathbf{M}}^{(2,3)} & \bar{\mathbf{M}}^{(2,4)} & \bar{\mathbf{M}}^{(2,5)} & \bar{\mathbf{M}}^{(2,6)} & \bar{\mathbf{M}}^{(2,7)} \\ \bar{\mathbf{M}}^{(3,1)} & \bar{\mathbf{M}}^{(3,2)} & \bar{\mathbf{M}}^{(3,3)} & \bar{\mathbf{M}}^{(3,4)} & 0 & \bar{\mathbf{M}}^{(3,6)} & \bar{\mathbf{M}}^{(3,7)} \\ \bar{\mathbf{M}}^{(4,1)} & \bar{\mathbf{M}}^{(4,2)} & \bar{\mathbf{M}}^{(4,3)} & \bar{\mathbf{M}}^{(4,4)} & \bar{\mathbf{M}}^{(4,5)} & 0 & 0 \\ \bar{\mathbf{M}}^{(5,1)} & \bar{\mathbf{M}}^{(5,2)} & 0 & \bar{\mathbf{M}}^{(5,4)} & \bar{\mathbf{M}}^{(5,5)} & 0 & 0 \\ \bar{\mathbf{M}}^{(6,1)} & \bar{\mathbf{M}}^{(6,2)} & \bar{\mathbf{M}}^{(6,3)} & 0 & 0 & \bar{\mathbf{M}}^{(6,6)} & \bar{\mathbf{M}}^{(6,7)} \\ \bar{\mathbf{M}}^{(7,1)} & \bar{\mathbf{M}}^{(7,2)} & \bar{\mathbf{M}}^{(7,3)} & 0 & 0 & \bar{\mathbf{M}}^{(7,6)} & \bar{\mathbf{M}}^{(7,7)} \end{bmatrix} \quad \text{(A-11)}$$

$$\bar{\mathbf{C}}^{(1,1)} = \zeta_{26}$$
$$\bar{\mathbf{C}}^{(1,2)} = -\omega\zeta_4\zeta_27$$
$$\begin{aligned}\bar{\mathbf{C}}^{(1,4)} = {} & \zeta_4(s(q_1)s(q_2)\dot{q}_2\zeta_5 - c(q_1)c(q_2)\dot{q}_1\zeta_6 - c(q_1)c(q_2)\omega\zeta_5 - c(q_1)c(q_2)\omega\zeta_6 - c(q_1)c(q_2)\dot{q}_1\zeta_5 + s(q_1)s(q_2)\dot{q}_2\zeta_6 \\ & -c(q_1)c(q_2)c(q_3)\dot{q}_1\zeta_7 - c(q_1)c(q_2)c(q_3)\omega\zeta_7 + c(q_1)s(q_2)s(q_3)\dot{q}_1\zeta_7 + c(q_2)s(q_1)s(q_3)\dot{q}_2\zeta_7 + c(q_3)s(q_1)s(q_2)\dot{q}_2\zeta_7 \\ & + c(q_2)s(q_1)s(q_3)\dot{q}_3\zeta_7 + c(q_3)s(q_1)s(q_2)\dot{q}_3\zeta_7 + c(q_1))s(q_2)s(q_3)\omega\zeta_7)\end{aligned}$$
$$\begin{aligned}\bar{\mathbf{C}}^{(1,5)} = {} & \zeta_4(s(q_1)s(q_2)\dot{q}_2\zeta_5 - c(q_1)c(q_2)\dot{q}_1\zeta_6 - c(q_1)c(q_2)\omega\zeta_5 - c(q_1)c(q_2)\omega\zeta_6 - c(q_1)c(q_2)\dot{q}_1\zeta_5 + sq_1s(q_2)\dot{q}_2\zeta_6 \\ & - c(q_1)c(q_2)c(q_3)\dot{q}_1\zeta_7 - c(q_1)c(q_2)c(q_3)\omega\zeta_7 + c(q_1)s(q_2)s(q_3)\dot{q}_1\zeta_7 + c(q_2)s(q_1)s(q_3)\dot{q}_2\zeta_7 + c(q_3)s(q_1)s(q_2)\dot{q}_2\zeta_7 \\ & + c(q_2)s(q_1)s(q_3)\dot{q}_3\zeta_7 + c(q_3)s(q_1)s(q_2)\dot{q}_3\zeta_7 + c(q_1)s(q_2)s(q_3)\omega\zeta_7)\end{aligned}$$
$$\begin{aligned}\bar{\mathbf{C}}^{(1,6)} = {} & \zeta_4(s(q_1)s(q_2)\dot{q}_1\zeta_5 - c(q_1)c(q_2)\dot{q}_2\zeta_6 - c(q_1)c(q_2)\dot{q}_2\zeta_5 + s(q_1)s(q_2)\dot{q}_1\zeta_6 + s(q_1)s(q_2)\omega\zeta_5 + s(q_1)s(q_2)\omega\zeta_6 \\ & - c(q_1)c(q_2)c(q_3)\dot{q}_2\zeta_7 - c(q_1)c(q_2)c(q_3)\dot{q}_3\zeta_7 + c(q_2)s(q_1)s(q_3)\dot{q}_1\zeta_7 + c(q_3)s(q_1)s(q_2)\dot{q}_1\zeta_7 \\ & + c(q_1)s(q_2)s(q_3)\dot{q}_2\zeta_7 + c(q_1)s(q_2)s(q_3)\dot{q}_3\zeta_7 + c(q_2)s(q_1)s(q_3)\omega\zeta_7 + c(q_3)s(q_1)s(q_2)\omega\zeta_7)\end{aligned}$$
$$\begin{aligned}\bar{\mathbf{C}}^{(1,7)} = {} & \zeta_4\zeta_7(c(q_2)s(q_1)s(q_3)\dot{q}_1 - c(q_1)c(q_2)c(q_3)\dot{q}_3 - c(q_1)c(q_2)c(q_3)\dot{q}_2q + c(q_3)s(q_1)s(q_2)\dot{q}_1 + c(q_1)s(q_2)s(q_3)\dot{q}_2 \\ & + c(q_1)s(q_2)s(q_3)\dot{q}_3 + c(q_2)s(q_1)s(q_3)\omega + c(q_3)s(q_1)s(q_2)\omega)\end{aligned}$$
$$\bar{\mathbf{C}}^{(2,1)} = \omega\zeta_{12}\zeta_{27}$$
$$\bar{\mathbf{C}}^{(2,2)} = -\zeta_{28}$$
$$\begin{aligned}\bar{\mathbf{C}}^{(2,4)} = {} & -\zeta_{12}(c(q_1)s(q_1)\dot{q}_1\zeta_5 + c(q_1)s(q_2)\dot{q}_2\zeta_5 + c(q_2)s(q_1)\dot{q}_1\zeta_6 + c(q_1)s(q_2)\dot{q}_2\zeta_6 \\ & + c(q_2)s(q_1)\omega\zeta_5 + c(q_2)s(q_1)\omega\zeta_6 + c(q_2)c(q_3)s(q_1)\dot{q}_1\zeta_7 + c(q_1)c(q_2)s(q_3)\dot{q}_2\zeta_7 + c(q_1)c(q_3)s(q_2)\dot{q}_2\zeta_7 \\ & + c(q_1)c(q_2)s(q_3)\dot{q}_3\zeta_7 + c(q_1)c(q_3)s(q_2)\dot{q}_3\zeta_7 + c(q_2)c(q_3)s(q_1)\omega\zeta_7 - s(q_1)s(q_2)s(q_3)\dot{q}_1\zeta_7 - s(q_1)s(q_2)s(q_3)\omega\zeta_7)\end{aligned}$$
$$\begin{aligned}\bar{\mathbf{C}}^{(2,5)} = {} & -\zeta_12(c(q_2)s(q_1)\dot{q}_1\zeta_5 + c(q_1)s(q_2)\dot{q}_2\zeta_5 + c(q_2)s(q_1)\dot{q}_1\zeta_6 + c(q_1)s(q_2)\dot{q}_2\zeta_6 + c(q_2)s(q_1)\omega\zeta_5 + c(q_2)s(q_1)\omega\zeta_6 \\ & + c(q_2)c(q_3)s(q_1)\dot{q}_1\zeta_7 + c(q_1)c(q_2)s(q_3)\dot{q}_2\zeta_7 + c(q_1)c(q_3)s(q_2)\dot{q}_2\zeta_7 + c(q_1)c(q_2)s(q_3)\dot{q}_3\zeta_7 + c(q_1)c(q_3)s(q_2)\dot{q}_3\zeta_7 \\ & + c(q_1)c(q_3)s(q_1)\omega\zeta_7 - s(q_1)s(q_2)s(q_3)\dot{q}_1\zeta_7 - s(q_1)s(q_2)s(q_3)\omega\zeta_7)\end{aligned}$$
$$\begin{aligned}\bar{\mathbf{C}}^{(2,6)} = {} & -\zeta_{12}(c(q_1)s(q_2)\dot{q}_1\zeta_5 + c(q_1)s(q_2)\dot{q}_1\zeta_6 + c(q_2)s(q_1)\dot{q}_2\zeta_5 + c(q_2)s(q_1)\dot{q}_2\zeta_6 + c(q_1)s(q_2)\omega\zeta_5 + c(q_1)s(q_2)\omega\zeta_6 \\ & + c(q_1)c(q_2)s(q_3)\dot{q}_1\zeta_7 + c(q_1)c(q_3)s(q_2)\dot{q}_1\zeta_7 + c(q_2)c(q_3)s(q_1)\dot{q}_2\zeta_7 + c(q_2)c(q_3)s(q_1)\dot{q}_3\zeta_7 + c(q_1)c(q_2)s(q_3)\omega\zeta_7 \\ & + c(q_1)c(q_3)s(q_2)\omega\zeta_7 - s(q_1)s(q_2)s(q_3)\dot{q}_2\zeta_7 - s(q_1)s(q_2)s(q_3)\dot{q}_3\zeta_7)\end{aligned}$$
$$\begin{aligned}\bar{\mathbf{C}}^{(2,7)} = {} & -\zeta_7\zeta_{12}(c(q_1)c(q_2)s(q_3)\dot{q}_1 + c(q_1)c(q_3)s(q_2)\dot{q}_1 + c(q_2)c(q_3)s(q_1)\dot{q}_2 + c(q_2)c(q_3)s(q_1)\dot{q}_3 + c(q_1)c(q_2)s(q_3)\omega \\ & + c(q_1)c(q_3)s(q_2)\omega - s(q_1)s(q_2)s(q_3)\dot{q}_2 - s(q_1)s(q_2)s(q_3)\dot{q}_3)\end{aligned}$$
$$\bar{\mathbf{C}}^{(3,3)} = -\zeta_{29}$$
$$\bar{\mathbf{C}}^{(3,6)} = \zeta_{17}(\dot{q}_2(s(q_2)\zeta_5 + s(q_2)\zeta_6 + s(q_2q_3)\zeta_7) + s(q_2q_3)\dot{q}_3\zeta_7)$$
$$\bar{\mathbf{C}}^{(3,7)} = s(q_2q_3)\zeta_7\zeta_{17}(\dot{q}_2 + \dot{q}_3)$$
$$\bar{\mathbf{C}}^{(4,1)} = c(q_1)\omega\zeta_{18}(c(q_2)\zeta_5 + c(q_2)\zeta_6 + c(q_2q_3)\zeta_7)$$
$$\bar{\mathbf{C}}^{(4,2)} = s(q_1)\omega\zeta_{18}(c(q_2)\zeta_5 + c(q_2)\zeta_6 + c(q_2q_3)\zeta_7)$$
$$\begin{aligned}\bar{\mathbf{C}}^{(4,4)} = {} & -\zeta_{18}(\zeta_{30} + (s(q_3)l_3\dot{q}_3\zeta_6)/2 + (s(2q_2)l_2\dot{q}_2\zeta_5)/2 + (s(2q_2)l_2\dot{q}_2\zeta_6)/2 + s(2q_2q_3)l_3\dot{q}_2\zeta_6 + (s(2q_2q_3)l_3\dot{q}_3\zeta_6)/2 \\ & + (s(2q_22\ q_3)l_3\dot{q}_2\zeta_7)/2 + (s(2q_22\ q_3)l_3\dot{q}_3\zeta_7)/2)\end{aligned}$$
$$\begin{aligned}\bar{\mathbf{C}}^{(4,5)} = {} & -(\zeta_{18}(s(q_3)l_3\dot{q}_3\zeta_6 + s(2q_2)l_2\dot{q}_2\zeta_5 + s(2q_2)l_2\dot{q}_2\zeta_6 + 2\ s(2q_2q_3)l_3\dot{q}_2\zeta_6 + s(2q_2q_3)l_3\dot{q}_3\zeta_6 \\ & + s(2q_22\ q_3)l_3\dot{q}_2\zeta_7 + s(2q_22\ q_3)l_3\dot{q}_3\zeta_7))/2\end{aligned}$$
$$\bar{\mathbf{C}}^{(4,6)} = -(\zeta_{18}(\dot{q}_1 + \omega)(s(2q_2)l_2\zeta_5 + s(2q_2)l_2\zeta_6 + 2\ s(2q_2q_3)l_3\zeta_6 + s(2q_22\ q_3)l_3\zeta_7))/2$$
$$\bar{\mathbf{C}}^{(4,7)} = -(\zeta_7\zeta_{18}(\dot{q}_1 + \omega)(s(q_3)l_2 + s(2q_2q_3)l_2 + s(2q_22\ q_3)l_3))/2$$
$$\bar{\mathbf{C}}^{(5,1)} = c(q_1)\omega\zeta_{24}(c(q_2)\zeta_5 + c(q_2)\zeta_6 + c(q_2q_3)\zeta_7)$$
$$\bar{\mathbf{C}}^{(5,2)} = s(q_1)\omega\zeta_{24}(c(q_2)\zeta_5 + c(q_2)\zeta_6 + c(q_2q_3)\zeta_7)$$
$$\begin{aligned}\bar{\mathbf{C}}^{(5,4)} = {} & -(\zeta_{24}(s(q_3)l_3\dot{q}_3\zeta_6 + s(2q_2)l_2\dot{q}_2\zeta_5 + s(2q_2)l_2\dot{q}_2\zeta_6 + 2\ s(2q_2q_3)l_3\dot{q}_2\zeta_6 \\ & + s(2q_2q_3)l_3\dot{q}_3\zeta_6 + s(2q_22\ q_3)l_3\dot{q}_2\zeta_7 + s(2q_22\ q_3)l_3\dot{q}_3\zeta_7))/2\end{aligned}$$

$$\bar{\mathbf{C}}^{(5,5)} = -\zeta_{24}(\zeta_{31} + (s(q_3)l_3\dot{q}_3\zeta_6)/2 + (s(2q_2)l_2\dot{q}_2\zeta_5)/2 + (s(2q_2)l_2\dot{q}_2\zeta_6)/2$$
$$+ s(2q_2q_3)l_3\dot{q}_2\zeta_6 + (s(2q_2q_3)l_3\dot{q}_3\zeta_6)/2 + (s(2q_2 2\ q_3)l_3\dot{q}_2\zeta_7)/2 + (s(2q_2 2\ q_3)l_3\dot{q}_3\zeta_7)/2)$$
$$\bar{\mathbf{C}}^{(5,6)} = -(\zeta_{24}(\dot{q}_1 + \omega)(s(2q_2)l_2\zeta_5 + s(2q_2)l_2\zeta_6 + 2\ s(2q_2q_3)l_3\zeta_6 + s(2q_2 2\ q_3)l_3\zeta_7))/2$$
$$\bar{\mathbf{C}}^{(5,7)} = -(\zeta_7\zeta_{24}(\dot{q}_1 + \omega)(s(q_3)l_2 + s(2q_2q_3)l_2 + s(2q_2 2\ q_3)l_3))/2$$
$$\bar{\mathbf{C}}^{(6,1)} = -s(q_1)\omega\zeta_{24}(s(q_2)\zeta_5 + s(q_2)\zeta_6 + s(q_2q_2)\zeta_7)$$
$$\bar{\mathbf{C}}^{(6,2)} = c(q_1)\omega\zeta_{24}(s(q_2)\zeta_5 + s(q_2)\zeta_6 + s(q_2q_3)\zeta_7)$$
$$\bar{\mathbf{C}}^{(6,4)} = (\zeta_{24}(\dot{q}_1 + \omega)(s(2q_2)l_2\zeta_5 + s(2q_2)l_2\zeta_6 + 2\ s(2q_2q_3)l_3\zeta_6 + s(2q_2 2\ q_3)l_3\zeta_7))/2$$
$$\bar{\mathbf{C}}^{(6,5)} = (\zeta_{24}(\dot{q}_1 + \omega)(s(2q_2)l_2\zeta_5 + s(2q_2)l_2\zeta_6 + 2\ s(2q_2q_3)l_3\zeta_6 + s(2q_2 2\ q_3)l_3\zeta_7))/2$$
$$\bar{\mathbf{C}}^{(6,6)} = -\zeta_{24}(\zeta_{31} + s(q_3)l_3\dot{q}_3\zeta_6)$$
$$\bar{\mathbf{C}}^{(6,7)} = -s(q_3)l_3\zeta_6\zeta_{24}(\dot{q}_2 + \dot{q}_3)$$
$$\bar{\mathbf{C}}^{(7,1)} = -s(q_1)s(q_2q_3)\omega\zeta_7\zeta_{24}$$
$$\bar{\mathbf{C}}^{(7,2)} = c(q_1)s(q_2q_3)\omega\zeta_7\zeta_{24}$$
$$\bar{\mathbf{C}}^{(7,4)} = (\zeta_7\zeta_{24}(\dot{q}_1 + \omega)(s(q_3)l_2 + s(2q_2q_3)l_2 + s(2q_2 2\ q_3)l_3))/2$$
$$\bar{\mathbf{C}}^{(7,5)} = (\zeta_7\zeta_{24}(\dot{q}_1 + \omega)(s(q_3)l_2 + s(2q_2q_3)l_2 + s(2q_2 2\ q_3)l_3))/2$$
$$\bar{\mathbf{C}}^{(7,6)} = s(q_3)l_3\dot{q}_2\zeta_6\zeta_{24}$$
$$\bar{\mathbf{C}}^{(7,7)} = -\zeta_{32}$$

more information about these quantities are, as shown in the equation at the bottom of the previous page and top of the page.

The gravitational vector of the entire system can be defined as follows: where:

$$\bar{\mathbf{G}} = \begin{bmatrix} 0 \\ 0 \\ \bar{\mathbf{G}}^{(3,1)} \\ 0 \\ 0 \\ \bar{\mathbf{G}}^{(6,1)} \\ \bar{\mathbf{G}}^{(7,1)} \end{bmatrix}$$

$$\bar{\mathbf{G}}^{(3,1)} = g\zeta_{17}\zeta_{27}$$
$$\bar{\mathbf{G}}^{(6,1)} = -g\zeta_{24}(\cos q_2\zeta_5 + \cos(q_2)\zeta_6 + \cos(q_2 + q_3)\zeta_7)$$
$$\bar{\mathbf{G}}^{(7,1)} = \cos(q_2 + q_3)g\zeta_7\zeta_{24} \qquad \text{(A-13)}$$

ACKNOWLEDGMENT

The authors would like to thank the CICHE Research Center, SISAu Research Group, ARSI Research Group, and Intituto de Automatica (INAUT), for the support and development of this work. The results of this work are part of the project "Tecnologías de la Industria 4.0 en Educación, Salud, Empresa e Industria" developed by Universidad Indoamérica and the project "Autonomous Control of Aerial Manipulator Robots" developed by Universidad de las Fuerzas Armadas (ESPE).

REFERENCES

[1] Y. Chen, Q. Dong, X. Shang, Z. Wu, and J. Wang, "Multi-UAV autonomous path planning in reconnaissance missions considering incomplete information: A reinforcement learning method," *Drones*, vol. 7, no. 1, p. 10, Dec. 2022. [Online]. Available: https://www.mdpi.com/2504-446X/7/1/10

[2] A. Ramachandran and A. K. Sangaiah, "A review on object detection in unmanned aerial vehicle surveillance," *Int. J. Cogn. Comput. Eng.*, vol. 2, pp. 215–228, Jun. 2021.

[3] X. Li and A. V. Savkin, "Networked unmanned aerial vehicles for surveillance and monitoring: A survey," *Future Internet*, vol. 13, no. 7, p. 174, Jul. 2021. [Online]. Available: https://www.mdpi.com/1999-5903/13/7/174

[4] B. H. Y. Alsalam, K. Morton, D. Campbell, and F. Gonzalez, "Autonomous UAV with vision based on-board decision making for remote sensing and precision agriculture," in *Proc. IEEE Aerosp. Conf.*, Mar. 2017, pp. 1–12.

[5] F. Fraundorfer, L. Heng, D. Honegger, G. H. Lee, L. Meier, P. Tanskanen, and M. Pollefeys, "Vision-based autonomous mapping and exploration using a quadrotor MAV," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2012, pp. 4557–4564.

[6] F. Kendoul, "Survey of advances in guidance, navigation, and control of unmanned rotorcraft systems," *J. Field Robot.*, vol. 29, no. 2, pp. 315–378, Mar. 2012. [Online]. Available: https://onlinelibrary.wiley.com/doi/full/10.1002/rob.20414

[7] J. Scherer, S. Yahyanejad, S. Hayat, E. Yanmaz, T. Andre, A. Khan, V. Vukadinovic, C. Bettstetter, H. Hellwagner, and B. Rinner, "An autonomous multi-UAV system for search and rescue," in *Proc. 1st Workshop Micro Aerial Vehicle Netw., Syst., Appl. Civilian Use*. New York, NY, USA: Association for Computing Machinery, May 2015, pp. 33–38, doi: 10.1145/2750675.2750683.

[8] M. Blösch, S. Weiss, D. Scaramuzza, and R. Siegwart, "Vision based MAV navigation in unknown and unstructured environments," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2010, pp. 21–28.

[9] H. Bonyan Khamseh, F. Janabi-Sharifi, and A. Abdessameud, "Aerial manipulation—A literature survey," *Robot. Auto. Syst.*, vol. 107, pp. 221–235, Sep. 2018.

[10] K. Bodie, M. Brunner, M. Pantic, S. Walser, P. Pfändler, U. Angst, R. Siegwart, J. Nieto, F. Im Breisgau, J. Ee, K. Bodie, M. Brunner, M. Pantic, S. Walser, P. Pfändler, U. Angst, and R. Siegwart, "An omnidirectional aerial manipulation platform for contact-based inspection," *Robotics, Sci. Syst. XV*, vol. 15, Jun. 2019. [Online]. Available: http://www.roboticsproceedings.org/rss15/ and https://www.roboticsproceedings.org/rss15/p19.html

[11] T. Ikeda, S. Yasui, M. Fujihara, K. Ohara, S. Ashizawa, A. Ichikawa, A. Okino, T. Oomichi, and T. Fukuda, "Wall contact by octo-rotor UAV with one DoF manipulator for bridge inspection," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2017, pp. 5122–5127.

[12] C. Papachristos, K. Alexis, and A. Tzes, "Technical activities execution with a TiltRotor UAS employing explicit model predictive control," *IFAC Proc. Volumes*, vol. 47, no. 3, pp. 11036–11042, Jan. 2014.

[13] F. Augugliaro, S. Lupashin, M. Hamer, C. Male, M. Hehn, M. W. Mueller, J. S. Willmann, F. Gramazio, M. Kohler, and R. D'Andrea, "The flight assembled architecture installation: Cooperative contruction with flying machines," *IEEE Control Syst.*, vol. 34, no. 4, pp. 46–64, 2014.

[14] C. C. Kessens, J. Thomas, J. P. Desai, and V. Kumar, "Versatile aerial grasping using self-sealing suction," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2016, pp. 3249–3254.

[15] F. Ruggiero, V. Lippiello, and A. Ollero, "Aerial manipulation: A literature review," *IEEE Robot. Autom. Lett.*, vol. 3, no. 3, pp. 1957–1964, Jul. 2018.

[16] A. Mohiuddin, T. Tarek, Y. Zweiri, and D. Gan, "A survey of single and multi-UAV aerial manipulation," *Unmanned Syst.*, vol. 8, no. 2, pp. 119–147, Apr. 2020, doi: 10.1142/s2301385020500089.

[17] X. Meng, Y. He, and J. Han, "Survey on aerial manipulator: System, modeling, and control," *Robotica*, vol. 38, no. 7, pp. 1288–1317, Jul. 2020. [Online]. Available: https://www.cambridge.org/core/journals/robotica/article/abs/survey-on-aerial-manipulator-system-modeling-and-control/5FDD2404D65EF73477CF10CA62B69720

[18] J. Yoon, B. Son, and D. Lee, "Comparative study of physics engines for robot simulation with mechanical interaction," *Appl. Sci.*, vol. 13, no. 2, p. 680, Jan. 2023. [Online]. Available: https://www.mdpi.com/2076-3417/13/2/680/htm https://www.mdpi.com/2076-3417/13/2/680

[19] T. Erez, Y. Tassa, and E. Todorov, "Simulation tools for model-based robotics: Comparison of Bullet, Havok, MuJoCo, ODE and PhysX," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2015, pp. 4397–4404.

[20] J. Collins, S. Chand, A. Vanderkop, and D. Howard, "A review of physics simulators for robotic applications," *IEEE Access*, vol. 9, pp. 51416–51431, 2021.

[21] V. Román-Ibáñez, F. Pujol-López, H. Mora-Mora, M. Pertegal-Felices, and A. Jimeno-Morenilla, "A low-cost immersive virtual reality system for teaching robotic manipulators programming," *Sustainability*, vol. 10, no. 4, p. 1102, Apr. 2018. [Online]. Available: https://www.mdpi.com/2071-1050/10/4/1102

[22] J. Song, K. Hur, J. Lee, H. Lee, J. Lee, S. Jung, J. Shin, and H. Kim, "Hardware-in-the-loop simulation using real-time hybrid-simulator for dynamic performance test of power electronics equipment in large power system," *Energies*, vol. 13, no. 15, p. 3955, Aug. 2020. [Online]. Available: https://www.mdpi.com/1996-1073/13/15/3955

[23] E. Rohmer, S. P. N. Singh, and M. Freese, "V-REP: A versatile and scalable robot simulation framework," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Nov. 2013, pp. 1321–1326.

[24] N. Koenig and A. Howard, "Design and use paradigms for Gazebo, an open-source multi-robot simulator," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2004, pp. 2149–2154.

[25] E. Todorov, T. Erez, and Y. Tassa, "MuJoCo: A physics engine for model-based control," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2012, pp. 5026–5033.

[26] *Version 7.10.0 (R2010a)*, MathWorks Inc., Natick, MA, USA, 2010.

[27] J. Bezanson, S. Karpinski, V. B. Shah, and A. Edelman, "Julia: A fast dynamic language for technical computing," 2012, *arXiv:1209.5145*.

[28] P. Corke and J. Haviland, "Not your grandmother's toolbox—The robotics toolbox reinvented for Python," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2021, pp. 11357–11363.

[29] A. Konrad, "Simulation of mobile robots with unity and ROS: A case-study and a comparison with Gazebo," M.S. thesis, Division Production Syst., Univ. West, Trollhättan, Sweden, 2019.

[30] Y. Liu, G. Novotny, N. Smirnov, W. Morales-Alvarez, and C. Olaverri-Monreal, "Mobile delivery robots: Mixed reality-based simulation relying on ROS and unity 3D," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Oct. 2020, pp. 15–20.

[31] L. Su, G. Qiu, W. Tang, and M. Chen, "A ROS based open source simulation environment for robotics beginners," in *Proc. 6th Int. Conf. Robot. Autom. Eng. (ICRAE)*, Nov. 2021, pp. 286–291.

[32] J. Cheng, S. Bi, and C. Yuan, "Dynamic parameters identification method of 6-DOF industrial robot based on quaternion," *Mathematics*, vol. 10, no. 9, p. 1513, May 2022.

[33] W. Liu, X. Huo, J. Liu, and L. Wang, "Parameter identification for a quadrotor helicopter using multivariable extremum seeking algorithm," *Int. J. Control, Autom. Syst.*, vol. 16, no. 4, pp. 1951–1961, Aug. 2018.

[34] P. Kremer, J. L. Sanchez-Lopez, and H. Voos, "A hybrid modelling approach for aerial manipulators," *J. Intell. Robotic Syst.*, vol. 105, no. 4, p. 74, Aug. 2022.

[35] A. Pandey, V. S. Panwar, M. E. Hasan, and D. R. Parhi, "V-REP-based navigation of automated wheeled robot between obstacles using PSO-tuned feedforward neural network," *J. Comput. Design Eng.*, vol. 7, no. 4, pp. 427–434, 2020.

[36] A. Mairaj, A. I. Baba, and A. Y. Javaid, "Application specific drone simulators: Recent advances and challenges," *Simul. Model. Pract. Theory*, vol. 94, pp. 100–117, Jul. 2019.

[37] B. Davoudi, E. Taheri, K. Duraisamy, B. Jayaraman, and I. Kolmanovsky, "Quad-rotor flight simulation in realistic atmospheric conditions," *AIAA J.*, vol. 58, no. 5, pp. 1992–2004, May 2020.

[38] F. Fabra, C. T. Calafate, J. C. Cano, and P. Manzoni, "ArduSim: Accurate and real-time multicopter simulation," *Simul. Model. Pract. Theory*, vol. 87, pp. 170–190, Sep. 2018.

[39] E. Cuniato, J. Cacace, M. Selvaggio, F. Ruggiero, and V. Lippiello, "A hardware-in-the-loop simulator for physical human-aerial manipulator cooperation," in *Proc. 20th Int. Conf. Adv. Robot. (ICAR)*, Dec. 2021, pp. 830–835.

JOSÉ VARELA-ALDÁS (Member, IEEE) is currently pursuing the Ph.D. degree in electronic engineering with the University of Zaragoza, Spain. He is also an Associate Professor with Universidad Indoamérica, teaching the following subjects: robotics, electrical engineering, and electricity and industrial electronics. His research interests include control systems, robotics, the IoT, and virtual reality. He was the Winner of Best Young Researcher at IEEE Ecuador, in 2023. In 2024, he will serve as the President for the Robotics and Automation Society at IEEE Ecuador.

LUIS F. RECALDE received the Graduate degree in mechatronics engineering from the University of the Armed Forces (ESPE), in 2021. He is currently pursuing the master's degree in control systems engineering with the National University of San Juan, Argentina. He is currently a Researcher with CICHE, Universidad Indoamerica. Over the past few years, he has actively merged control theory with machine learning. His research interests include nonlinear model predictive control (NMPC) and reinforcement learning (RL).

BRYAN S. GUEVARA received the Graduate degree in mechatronic engineering from the University of the Armed Forces (ESPE), in 2018, and the master's degree in control systems engineering from the National University of San Juan, in 2024, where he is currently pursuing the Ph.D. degree in control systems engineering. He was supported by the DAAD Scholarship (German Academic Exchange Service) through the Funding Program: Third Country Programme Latin America, in 2022, for the Ph.D. degree. His research interests include aerial robotics, dynamic systems modeling, and optimal control.

VICTOR H. ANDALUZ received the joint Ph.D. degree in control systems engineering from the Institute of Automatics, National University of San Juan, Argentina, and Institute of Real-Time Systems, Leibniz University Hannover, Germany. He is currently an Electronics and Control Engineer with the National Polytechnic School. He was a Scholarship Holder of German Institute for Academic Exchange, DAAD. He is also a Professor with the University of the Armed Forces (ESPE). His research interests include robotics, automation, control systems, virtual, and augmented reality.

DANIEL C. GANDOLFO received the Graduate degree in electronic engineering and the Ph.D. degree in control systems engineering from the National University of San Juan (UNSJ), Argentina, in 2006 and 2014, respectively. He has been an Automation Engineer in the industry, until 2009. Currently, he is a Researcher with Argentinean National Council for Scientific Research (CONICET) and an Associate Professor with the Institute of Automatics, UNSJ-CONICET, Argentina. His research interests include algorithms for management energy systems and optimal control strategies with application in unmanned aerial vehicles (UAV).