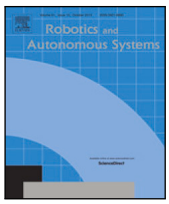
Robotics and Autonomous Systems 173 (2024) 104614

Contents lists available at ScienceDirect

Robotics and Autonomous Systems

journal homepage: www.elsevier.com/locate/robot



Multitask control of aerial manipulator robots with dynamic compensation based on numerical methods

Christian P. Carvajal [a,*], Gabriela M. Andaluz [c], Víctor H. Andaluz [b], Flavio Roberti [a], Guillermo Palacios-Navarro [c], Ricardo Carelli [a]

[a] Instituto de Automática (INAUT), Universidad Nacional de San Juan - CONICET, Av. San Martín (Oeste) 1109, San Juan, Argentina
[b] Universidad de las Fuerzas Armadas ESPE, Sangolquí, Ecuador
[c] University of Zaragoza, Department of Electronic Engineering and Communications, Zaragoza, Spain

ARTICLE INFO

Keywords:
Aerial manipulator
Motion control
Cascaded controller
Redundant system
Dynamic model
Numerical methods

ABSTRACT

This paper presents a control scheme for aerial manipulators which allows to solve different motion problems: end-effector position control, end-effector trajectory tracking control and path-following control. The scheme has two cascaded controllers: *i*) the first controller is a minimum norm controller based on numerical methods, it solves the three motion control problems just by modifying the controller references. Also, since the aerial manipulator robot is a redundant system, *i.e.*, it has extra degrees of freedom to accomplish the task, it is possible to set other control objectives in a hierarchical order. As a secondary objective of the control it is proposed to maintain a desired configuration for the robotic arm during the task. *ii*) The second cascade controller is designed to compensate the dynamics of the system which main objective is to drive the velocity errors to zero. The coupled dynamic model of the robotic system (hexarotor and robotic arm) is presented. This model is usually developed as a function of the forces and torques. However, in this work, it is written as a function of reference velocities which are usual references for these vehicles. The proposed control algorithms are given with the corresponding stability and robustness analysis. Finally, to validate the control scheme, experimental tests are performed in a partially structured environment with an aerial manipulator conformed by an aerial platform and a 3DOF robotic arm.

1. Introduction

In the last decade, the robots have been evolving due to the capabilities and functionalities that robots have developed in different fields such as in agricultural services [1,2], search and rescue of people [3,4], nuclear and exploration activities [5,6]. A robot can have total or partial autonomy, which allows it to make decisions based on the work environment. By different navigation techniques the robots are able to perform tasks such as rescue, search and localization, among other activities. These types of tasks can be simple and performed by non-complex robotic systems. However, there are applications such as object manipulation, maintenance of structures, transportation, that require the ability to move from one place to another and the ability to manipulate to reach the final goal. Among this type of robotic structures are aerial manipulators (AM), which allow performing active and passive tasks combining the manipulation characteristics of a robotic arm and the transport characteristics of an unmanned aerial vehicle (UAV).

Some of the control algorithms used to solve motion problems of aerial manipulators are based on the kinematic model and the dynamic model of the robotic system. In the literature there are several motion control techniques that involve kinematics and/or dynamics, *e.g.*, in [7] the trajectory tracking of an aerial manipulator is presented using an adaptive algorithm to compensate for the variations of the UAV dynamics, but it does not consider the dynamics of the robotic arm. Similarly, in [8] a trajectory tracking control based on an adaptive algorithm of infinite H-norm is presented, in which the robotic arm is assumed as a perturbation within the model of the aerial manipulator. In [9] it is presented a model based adaptive control algorithms for an under actuated AM composed of a conventional multirotor and a robotic arm. The authors consider the aerial platform and the robotic arm as two independent systems for modeling and control. In the same context, position control based on the sliding mode technique is presented in [10], the dynamic model of the AM supposes the mass of the UAV within the mass of the first link, which generates control problems due to change of center of gravity. To solve this, the authors implement

* Corresponding author.
E-mail address: cpcarvajal@inaut.unsj.edu.ar (C.P. Carvajal).

https://doi.org/10.1016/j.robot.2023.104614
Received 5 April 2023; Received in revised form 17 November 2023; Accepted 27 December 2023
Available online 5 January 2024

an adaptive control law. Several works in the literature [11,12] present the dynamic model of an aerial manipulator robot independently, that is, the dynamic model of the aerial platform and the dynamic model of the robotic arm are analyzed separately. Therefore, one of the contributions of this work is to obtain and validate the dynamic model of an aerial manipulator as a coupled system (aerial platform + robotic arm). Another aspect to take into account from the review carried out is that the dynamic model of robots is usually expressed in terms of the forces and torques that act on the systems [13,14]. However, in order to avoid estimating or measuring forces and torques of the robot, another contribution of this work is to obtain a dynamic model as a function of the reference velocities, facilitating the application of control actions to the real robot.

Motion control issues that commonly exist for aerial manipulators are: (*i*) position control; (*ii*) path-following control; and (*iii*) trajectory tracking control. Some of the control algorithms developed to solve motion problems are based on the kinematic model and the dynamic model of the robotic system. In this context, several works have been carried out, such as in [15] where it is presented an inverse kinematics-based controller for a dual-arm aerial manipulator, being the motion constraints handled by the kinematics Null-Space. An inverse dynamics scheme for the internal control of the dynamics is presented in [16] with a robot model having torque inputs. There are schemes with trajectory optimization for the control of aerial manipulator [17]. These optimization schemes have been applied to the separate analysis of the UAV and the robotic arm, which generates inconveniences when solving the problem of different control schemes for each system. On the other hand, more advanced control schemes have been implemented for trajectory tracking such as null space algorithms and path-following with secondary objectives [18,19], however, they do not take into account the dynamics of the system, but only use the kinematic model of the robots. These works are presented at the simulation level, without evaluating the behavior of the system in face of the changes in the inertias, present in a combined coupled system. Therefore, the analysis of the dynamics of aerial manipulator robots in the control systems are totally pertinent and the performance at an experimental level should also be investigated. For example, in [20] an inverse dynamic control is developed for an aerial manipulator with a parallel 3DOF manipulator robot. There, the authors propose the dynamic compensation only of the omnidirectional floating base robot and mention that for better precision of the end-effector, it should be considered dynamics compensation control of the complete system in a coupled way.

In the present work the coupled dynamics of the complete system is obtained and an inverse dynamics compensation controller is proposed using numerical methods. Research has been carried out for the implementation of the theory of numerical methods in robotics applications such as in [21], which performs trajectory tracking in mobile robots. The controller design method is based on linear algebra theory, and numerical estimation techniques are used to estimate the uncertainty value for each sampling time. Also in [22] a sliding mode controller using linear algebra for trajectory tracking of a mobile robot is presented. The model of aerial manipulator robots are characterized by highly nonlinear and complex characteristics and its knowledge is mandatory for the development of traditional control strategies such as Sliding Mode Control (SMC) or inverse dynamics control strategies.

For all the above mentioned, this paper presents a new type of dynamic model for this kind of robotic systems that has not yet been discussed in the literature. The dynamic model is presented in terms of reference velocities, facilitating the application of control actions to the real robot. The Euler–Lagrange methodology is used to obtain the dynamic model, after which the parameters of the model are identified with the respective validation. The dynamic model is then used for implementing dynamic compensation in cascade with a kinematic controller, which is designed to allow the three control objectives of position control, trajectory and path-following control, in an unified way, by only modifying the desired references. In addition, the stability and robustness analysis of the control system are presented. Finally, experimental tests are performed in partially structured environment, to validate and assess the performance of the proposed control scheme. Such a control scheme is the main contribution of this work.

The work is structured as follows. Section 2 presents the problem formulation. Section 3 presents the motion control problems. Section 4 presents the kinematic and dynamic models of the aerial manipulator. Section 5 describes the proposed cascade controller scheme. Section 6 presents the proposed controllers with the respective stability and robustness analysis. Finally, Section 7 presents the experimental results with their analysis.

2. Problem formulation

This section presents the multilayer scheme proposed in this work, see Fig. 1. The proposed multilayer scheme considers different objectives, *i.e.*, each layer has different functionalities and is composed of five main layers:

Task configuration layer. The main objective of this layer is to plan and define the tasks to be executed by the aerial manipulator robot. This layer is subdivided into: (*i*) Off-line configuration. In this sub-layer the initial parameters of the robot are defined; the motion problem to be solved is selected, *i.e.*, it is defined if the robot executes position control, trajectory tracking or path-following; and the secondary control objective to be solved by redundancy of the robotic system is assigned. And finally, (*ii*) On-line configuration. In this sub-layer it is possible to modify the primary control objective; and modify or add another secondary control objective.

Kinematic control layer. In this layer, a minimum-norm kinematic controller based on numerical methods is implemented in order to solve the problems of position control, trajectory tracking and path-following for an aerial manipulator robot. The controller in each sampling period receives the reference positions and velocities of the desired shape. Therefore, this controller is responsible of calculating the maneuverability velocities for the system. In addition, the control of secondary objectives is considered in this layer.

Dynamic compensation layer. Is responsible for compensating the dynamics of the aerial manipulator robot in order to reduce the velocity tracking error.

Robot layer. Represents the aerial manipulator robot consisting of a robotic arm on the underside of an unmanned aerial vehicle.

Environment layer. Represents the partially structured workspace where the aerial manipulator robot executes the task planned in the first layer.

The proposed multilayer scheme presents independence of each layer, *i.e.*, the changes within any layer do not cause structural changes in the other layers. Several nonlinear control algorithms can be tested using the same control strategy. It is important to mention that from the presented scheme a simple structure can be obtained, in other words, some layers can be increased or removed keeping the basic structure, *i.e.*, subtraction or addition of layers does not affect the rest of the layers. For example, the on-line configuration layer could be discarded in the case of trajectory tracking in a known environment and without obstacles. An adaptive dynamic control layer can also be included, for tasks at high velocities and transport of heavy loads.

3. Motion control problems

In this paper, a control scheme is proposed to solve the problem of position control, trajectory tracking and path-following for an aerial manipulator robot in a unified way. The proposed control scheme considers the desired position and velocity of the end-effector of the

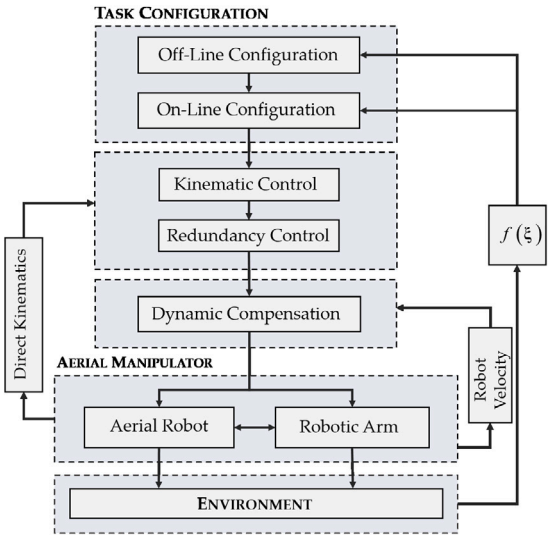


Fig. 1. Multilayer scheme.

aerial manipulator robot as reference signals. Therefore, these signals can be represented as:

$$\boldsymbol{\xi_d}(t,s,\xi) = \begin{bmatrix}\xi_{1d} & \xi_{2d} & \cdots & \xi_{md}\end{bmatrix}^\intercal \in \mathbb{R}^m \tag{1}$$

$$\mathbf{v_d}(t,s,\xi) = \begin{bmatrix}v_{1d} & v_{2d} & \cdots & v_{md}\end{bmatrix}^\intercal \in \mathbb{R}^m \tag{2}$$

where, $\boldsymbol{\xi_d} = \begin{bmatrix}\xi_{\mathbf{pd}}{}^\intercal & \xi_{od}\end{bmatrix}^\intercal$ is the desired position vector and orientation; and $\mathbf{v_d}$ represents the desired velocity vector of the end-effector of the robot at each time instant. According to the control problem, the desired position and velocity can be defined as a function of: t which represents the time variable; s represents the variable of the desired path profile; and $\boldsymbol{\xi} = \begin{bmatrix}\xi_{\mathbf{p}}{}^\intercal & \xi_o\end{bmatrix}^\intercal$ is the vector that contains the current position vector $\xi_{\mathbf{p}}$ and the current orientation $\xi_{\mathbf{o}}$ of the end-effector of the robotic system [23].

3.1. Position control

The positioning problem consists of maneuvering the robot with the objective of positioning the end-effector of the robot at a desired point with a desired orientation $\boldsymbol{\xi_d}$. Therefore, the desired position and orientation (1); and velocity (2) for position control is defined as: $\boldsymbol{\xi_d}(t,s,\xi) = \boldsymbol{\xi_d} \in \mathbb{R}^m$ and $\mathbf{v_d}(t,s,\xi) = \mathbf{0} \in \mathbb{R}^m$. The controller receives the desired constant position and orientation $\left(\boldsymbol{\xi_d} \middle| t \in [t_0, t_f]\right)$ and the desired velocity $\left(\mathbf{v_d} = \mathbf{0} \middle| t \in [t_0, t_f]\right)$. Then, the motion control problem is to determine a control law $\boldsymbol{\mu_K}(t) = f\left(\tilde{\boldsymbol{\xi}}(t)\right)$ with $\tilde{\boldsymbol{\xi}}(t) = \boldsymbol{\xi_d} - \boldsymbol{\xi}(t)$, such that $\lim_{t\to 0} \tilde{\boldsymbol{\xi}}(t) = \mathbf{0} \in \mathbb{R}^m$.

3.2. Trajectory tracking

The trajectory tracking problem of a robotic system consists of maneuvering the robot over a desired continuous smooth profile parameterized in time. Fig. 2 shows the trajectory tracking problem, where $\boldsymbol{\xi_d}(t)$ and $\dot{\boldsymbol{\xi}}_{\mathbf{d}}(t)$ represent the desired position-orientation and velocity of the end-effector at time t with respect to the inertial reference frame $\mathrm{R}_{\mathcal{F}}$. Therefore, the desired position (1) and velocity (2) for the trajectory tracking control are defined as:

$$\boldsymbol{\xi_d}(t,s,\xi) = \boldsymbol{\xi_d}(t) \quad \mathbf{v_d}(t,s,\xi) = \dot{\boldsymbol{\xi}}_{\mathbf{d}}(t) = \tfrac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{\xi_d}(t)$$

The trajectory tracking controller receives the desired trajectory and orientation $\left(\boldsymbol{\xi_d}(t) \middle| t \in [t_0, t_f]\right)$ and the desired velocity $\left(\mathbf{v_d} = \dot{\boldsymbol{\xi}}_{\mathbf{d}}(t) \middle| t \in [t_0, t_f]\right)$. Therefore, the problem of motion control is to determine a control law $\boldsymbol{\mu_K}(t) = f\left(\tilde{\boldsymbol{\xi}}(t), \dot{\boldsymbol{\xi}}_{\mathbf{d}}(t)\right)$ with $\tilde{\boldsymbol{\xi}}(t) = \boldsymbol{\xi_d}(t) - \boldsymbol{\xi}(t)$, such that $\lim_{t\to 0} \tilde{\boldsymbol{\xi}}(t) = \mathbf{0} \in \mathbb{R}^m$.

3.3. Path-following

Path-following control consists of maneuvering the robot so that the end-effector follows a desired non-parameterized path in time. Fig. 3 shows the robot path-following problem, where, $\mathbf{p}(s) = \begin{bmatrix}\xi_{px}(s) & \xi_{py}(s) & \xi_{pz}(s)\end{bmatrix}^\intercal$ represents the desired path to be followed; $\boldsymbol{\xi_d} = \begin{bmatrix}\mathbf{p_d}{}^\intercal & \xi_{od}\end{bmatrix}^\intercal = \begin{bmatrix}\xi_{xd} & \xi_{yd} & \xi_{zd} & \xi_{od}\end{bmatrix}^\intercal$ such that $\mathbf{p_d}$ represents the point closest to the current position of the end-effector; ξ_{od} is the desired orientation of the end-effector in the plane XY; $\rho = \left\|\tilde{\xi}_x, \tilde{\xi}_y, \tilde{\xi}_z\right\|$ is the position error norm; and $\mathbf{v_d}$ is the desired velocity of the end-effector on the desired path, defined with respect to the inertial reference frame $\mathrm{R}_{\mathcal{F}}$. Therefore, the desired position (1) and velocity (2) for path-following control is defined as:

$$\boldsymbol{\xi_d}(t,s,\xi) = \boldsymbol{\xi_d}(s,\xi) \quad \mathbf{v_d}(t,s,\xi) = \mathbf{v_d}(s,\xi)$$

The desired position and velocity over the path does not depend on time as an independent variable, it depends on the current position of the robotic system. $\left(\boldsymbol{\xi_d}(s,\xi) \middle| t \in [t_0, t_f]\right)$ represents the closest position of the desired path to the end-effector of the aerial manipulator with its desired orientation. Also, the desired velocity is $\left(\mathbf{v_d}(s,\xi) = \left[\mathbf{v_p}(s,\xi), \dot{\xi}_{od}(s,\xi)\right] \middle| t \in [t_0, t_f]\right)$, with $\mathbf{v_p}$ the velocity tangent to the path at the point $\mathbf{p_d}$ and with norm $\left\|\mathbf{v_p}\right\|$ which represents the desired speed at this point, and $\dot{\xi}_{od}$ is the desired rotational velocity of the end-effector. Therefore, the control objective is to design a control law for the robotic system $\boldsymbol{\mu_K}(t) = f\left(\tilde{\boldsymbol{\xi}}(t), \mathbf{v_d}(s,\xi)\right)$ with $\tilde{\boldsymbol{\xi}}(t) = \boldsymbol{\xi_d}(t) - \boldsymbol{\xi}(t)$ to guarantee the convergence the end-effector position $\boldsymbol{\xi}(\mathbf{t})$ to the desired path $\mathbf{p}(\mathbf{s})$, *i.e.*, $\lim_{t\to\infty} \tilde{\boldsymbol{\xi}}(t) = \mathbf{0} \in \mathbb{R}^m$.

4. Aerial manipulator model

Two kinds of models are used to design control strategies in robotics: (*i*) *Kinematic Model* which describes the motion of the robot end-effector as a function of the velocities of the joints of the system; and (*ii*) *Dynamic Model*, which provides a description of the relationship between the actuator torques of the robot joints and the motion of the system. On the other hand, the dynamic model is usually defined in terms of torques and forces [13,24]. In the present work, the dynamic model of the aerial manipulator is obtained as a function of the robot maneuvering velocities by including the dynamics of the servo actuators. With this dynamic model representation, it is possible to propose cascade dynamic compensation control schemes after controllers based on kinematics.

4.1. Kinematics of aerial manipulators

The configuration of the aerial manipulator is defined by the vector $\mathbf{q}$ of n independent coordinates, known as generalized coordinates of the system. The vector $\mathbf{q} = \begin{bmatrix}\mathbf{q_u}{}^\intercal & \mathbf{q_a}{}^\intercal\end{bmatrix}^\intercal$ with $\mathbf{q_u}$ representing the vector of generalized coordinates of the aerial platform and $\mathbf{q_a}$ the generalized coordinate vector of the robotic arm. It is possible to see that $n = n_u + n_a$, where, n_u and n_a are the dimensions of the spaces associated with the generalized coordinates of the aerial platform and the robotic arm respectively. The configuration $\mathbf{q}$ corresponds to the configuration space of the aerial manipulator $\mathcal{N}$. The location of the end-effector of the system is defined by the vector $\boldsymbol{\xi} = \begin{bmatrix}\xi_1 & \xi_2 & \cdots & \xi_m\end{bmatrix}^\intercal$, where $\boldsymbol{\xi}$ defines the position and orientation of the end-effector into the fixed reference frame $\mathrm{R}_{\mathcal{F}}$. The m coordinates of $\boldsymbol{\xi}$ represent the dimension of the working space of the aerial manipulator denoted as $\mathcal{M}$. The end-effector location can be defined in different ways depending on the task, *i.e.*, it can consider only the end-effector position or position and orientation.

The kinematic model of an aerial manipulator gives the location of the end-effector $\boldsymbol{\xi}$ as a function of the configuration of the robotic arm $\mathbf{q_a}$ and the configuration of the aerial platform $\mathbf{q_u}$.

$$f : \mathcal{N}_u \times \mathcal{N}_a \to \mathcal{M}$$

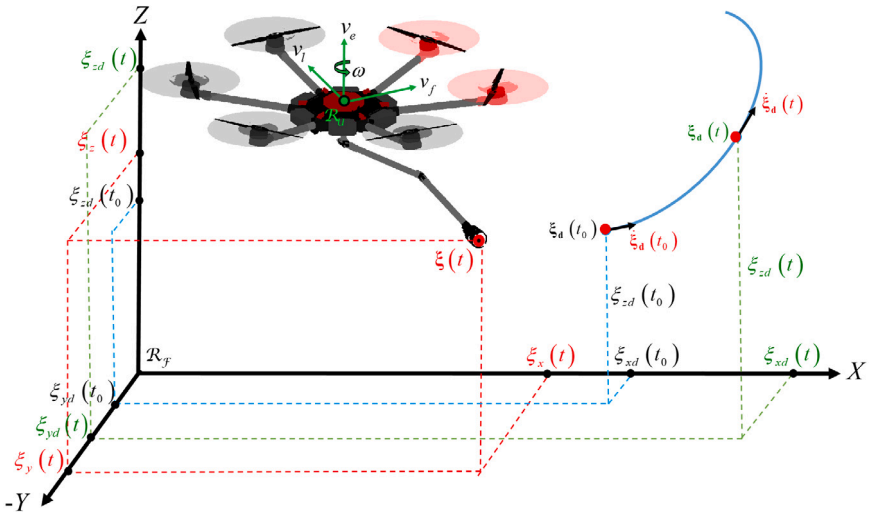


Fig. 2. Aerial manipulator trajectory tracking problem.

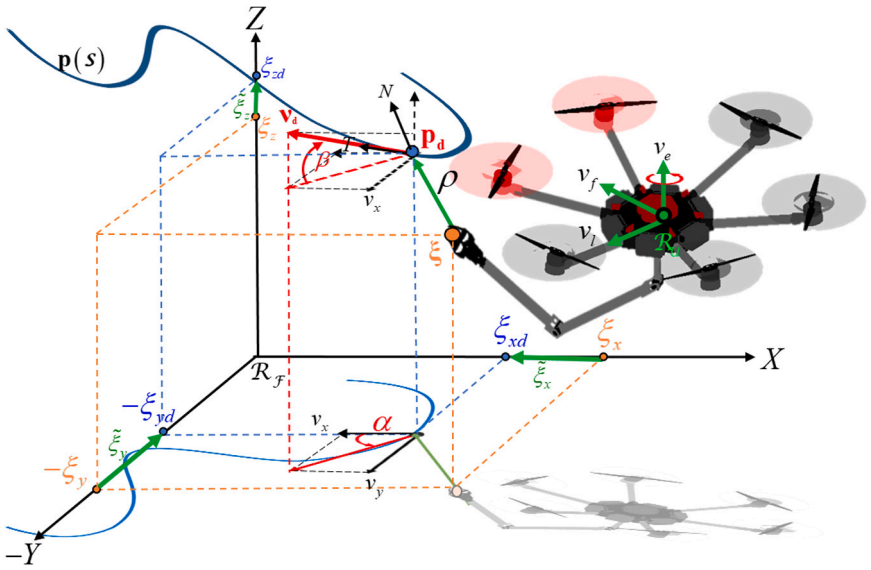


Fig. 3. Aerial manipulator path-following problem.

$$(\mathbf{q_u}, \mathbf{q_a}) \mapsto \boldsymbol{\xi} = f(\mathbf{q_u}, \mathbf{q_a})$$

where, $\mathcal{N}_u$ and $\mathcal{N}_a$ represents the configuration space of the aerial platform and the robotic arm respectively. Now, the differential kinematic model of the aerial manipulator gives the derivative of the location of the end-effector as a function of the derivatives of the configuration of the robotic arm $\mathbf{q_a}$ and the aerial platform $\mathbf{q_u}$. The result of the partial derivative of the position and orientation of the end-effector with respect to the joints of the system is:

$$\dot{\boldsymbol{\xi}}(t) = \frac{\partial f}{\partial \mathbf{q}}(\mathbf{q_u}, \mathbf{q_a})\,\dot{\mathbf{q}}(t) \tag{3}$$

where, $\dot{\boldsymbol{\xi}} = \begin{bmatrix}\dot{\xi}_1 & \dot{\xi}_2 & \cdots & \dot{\xi}_m\end{bmatrix}^\intercal$ is the velocity vector of the end-effector. Now, taking into account $\dot{\mathbf{q}}(t) = \mathbf{T}(\mathbf{q})\,\boldsymbol{\mu}(t)$ and replacing in (3), results:

$$\dot{\boldsymbol{\xi}}(t) = \mathbf{J}(\mathbf{q}(t))\,\boldsymbol{\mu}(t) \tag{4}$$

where, $\mathbf{T}(\mathbf{q}) \in \mathbb{R}$ is a matrix that includes the transformation between the velocity commands of the mobile reference frame $\mathrm{R}_\mathcal{U}$ and the velocities in the fixed reference frame $\mathrm{R}_\mathcal{F}$; $\mathbf{J}(\mathbf{q})$ is the Jacobian matrix of the system that defines the linear mapping between the vector of velocities of the aerial manipulator $\boldsymbol{\mu}(t)$ and the end-effector velocities $\dot{\boldsymbol{\xi}}(t)$.

4.2. Case study

This section describes the kinematic model of the aerial manipulator robot, which considers a coupled system, *i.e.*, aerial platform and robotic arm as a single system. The aerial manipulator consists of a hexarotor and a 3DOF robotic arm mounted on the underside of the aerial platform. Fig. 4 shows the configuration of the aerial manipulator in the operating space.

The system is composed of seven maneuverability velocities, four hexarotor velocities subject to the mobile reference frame of the robot $\mathrm{R}_\mathcal{U}$: v_f is the linear velocity for the hexarotor frontal displacement; v_l is the linear velocity for the hexarotor lateral displacement; v_e is the lineal velocity for the hexarotor elevation; $\omega = \dot{\psi}$ is the angular velocity that rotates the hexarotor around the E-axis; and three angular velocities $\dot{q}_1$, $\dot{q}_2$ and $\dot{q}_3$ which are the velocities of the robotic arm joints.

Remark 1. In the procedure for obtaining the kinematic model of the aerial manipulator robot, the system is considered to work with low

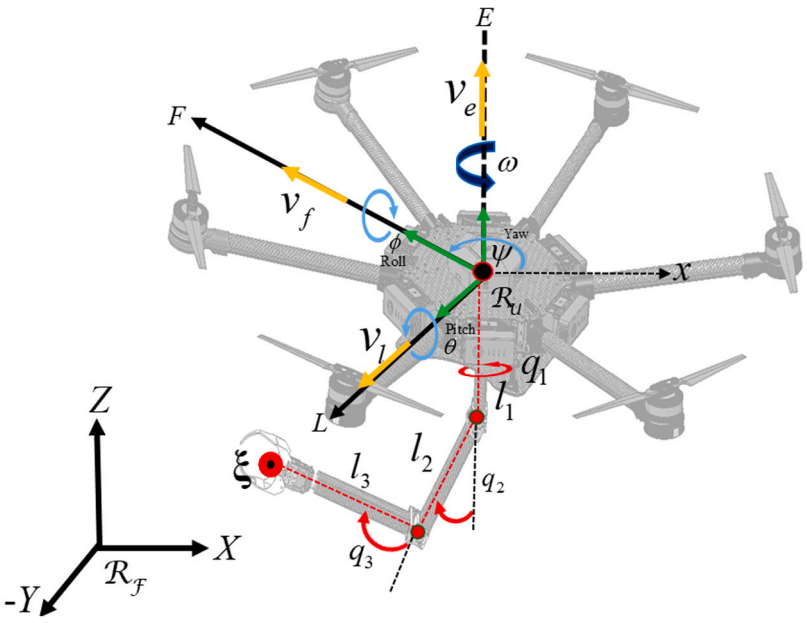


Fig. 4. Kinematic configuration of the aerial manipulator.

velocities. Therefore, the pitch and roll angles are approximately zero: $\theta \approx 0$ and $\phi \approx 0$.

The model representing the motion of the aerial manipulator in the fixed reference frame is defined as follows:

$$\dot{\boldsymbol{\xi}}(t) = \mathbf{J}(\mathbf{q})\mu(t) = \begin{cases} \dot{\xi}_x = v_f c_\psi - v_l s_\psi - (\omega + \dot{q}_1) s_{\psi q1} \left(l_2 s_{q2} + l_3 s_{q2q3}\right) + \\ \quad \cdots + \dot{q}_2 c_{\psi q1} \left(l_2 c_{q2} + l_3 c_{q2q3}\right) + \dot{q}_3 c_{\psi q1} \left(l_3 c_{q2q3}\right) \\ \dot{\xi}_y = v_f s_\psi + v_l c_\psi + (\omega + \dot{q}_1) c_{\psi q1} \left(l_2 s_{q2} + l_3 s_{q2q3}\right) + \\ \quad \cdots + \dot{q}_2 s_{\psi q1} \left(l_2 c_{q2} + l_3 c_{q2q3}\right) + \dot{q}_3 s_{\psi q1} \left(l_3 c_{q2q3}\right) \\ \dot{\xi}_z = v_e + \dot{q}_2 l_2 s_{q2} + (\dot{q}_2 + \dot{q}_3) l_3 s_{q2q3} \\ \dot{\xi}_o = \omega + \dot{q}_1 \end{cases} \tag{5}$$

where, $\dot{\boldsymbol{\xi}} = \begin{bmatrix} \dot{\xi}_x & \dot{\xi}_y & \dot{\xi}_z & \dot{\xi}_o \end{bmatrix}^\intercal$ is the vector of the end-effector velocities and the orientation of the end-effector on the XY plane of the fixed reference system $\mathrm{R}_\mathcal{F}$; $\boldsymbol{\mu} = \begin{bmatrix} v_f & v_l & v_e & \omega & \dot{q}_1 & \dot{q}_2 & \dot{q}_3 \end{bmatrix}^\intercal$ is the maneuverability velocity vector of the aerial manipulator; and $\mathbf{J}(\mathbf{q}) : \mathbb{R}^n \rightarrow \mathbb{R}^m$ is the Jacobian of the system with $n = 7$ which is the dimension of the configuration space of the aerial manipulator and $m = 4$ is the dimension of the robot workspace. In addition, $C_\alpha = \cos(\alpha)$; $C_{\alpha\beta} = \cos(\alpha + \beta)$; $S_\alpha = \sin(\alpha)$ and $S_{\alpha\beta} = \sin(\alpha + \beta)$.

4.3. Dynamic model of the aerial manipulator

The dynamic model establishes a mathematical relationship between the coordinates of the joints, their velocities, accelerations, forces and/or torques that act on the system and parameters of the robot such as: masses, length of the links, inertias and moments [25]. Fig. 5 shows a description of the parameters involved in the movement of the robot.

To obtain the dynamic model of the aerial manipulator, let us determine the velocities of each mass involved in the system, starting from the center of mass of the hexarotor to the end-effector. The position and velocity for the first mass is:

$$\mathbf{p}_1 = \begin{bmatrix} x_\mathrm{u} \\ y_\mathrm{u} \\ z_\mathrm{u} - l_1 \end{bmatrix} \quad \mathbf{v}_1 = \begin{bmatrix} \dot{x}_\mathrm{u} \\ \dot{y}_\mathrm{u} \\ \dot{z}_\mathrm{u} \end{bmatrix} \tag{6}$$

The position and velocity of the second mass is determined by:

$$\mathbf{p}_2 = \begin{bmatrix} x_\mathrm{u} + l_2 c_{\psi 1} s_2 \\ y_\mathrm{u} + l_2 s_{\psi 1} s_2 \\ z_\mathrm{u} - l_1 - l_2 c_2 \end{bmatrix} \quad \mathbf{v}_2 = \begin{bmatrix} \dot{x}_\mathrm{u} - l_2 s_{\psi 1} s_2 \left(\dot{\psi} + \dot{q}_1\right) + l_2 c_{\psi 1} c_2 \dot{q}_2 \\ \dot{y}_\mathrm{u} + l_2 c_{\psi 1} s_2 \left(\dot{\psi} + \dot{q}_1\right) + l_2 s_{\psi 1} c_2 \dot{q}_2 \\ \dot{z}_\mathrm{u} + l_2 s_2 \dot{q}_2 \end{bmatrix} \tag{7}$$

Finally, the position and velocity of the third mass is:

$$\mathbf{p}_3 = \begin{bmatrix} x_\mathrm{u} + c_{\psi 1} \left(l_2 s_2 + l_3 s_{23}\right) \\ y_\mathrm{u} + s_{\psi 1} \left(l_2 s_2 + l_3 s_{23}\right) \\ z_\mathrm{u} - l_1 - l_2 c_2 - l_3 c_{23} \end{bmatrix}$$

$$\mathbf{v}_3 = \begin{bmatrix} \dot{x}_\mathrm{u} - s_{\psi 1} \left(l_2 s_2 + l_3 s_{23}\right) \left(\dot{\psi} + \dot{q}_1\right) + c_{\psi 1} \left(\left(l_2 c_2 + l_3 c_{23}\right) \dot{q}_2 + l_3 c_{23} \dot{q}_3\right) \\ \dot{y}_\mathrm{u} + c_{\psi 1} \left(l_2 s_2 + l_3 s_{23}\right) \left(\dot{\psi} + \dot{q}_1\right) + s_{\psi 1} \left(\left(l_2 c_2 + l_3 c_{23}\right) \dot{q}_2 + l_3 c_{23} \dot{q}_3\right) \\ \dot{z}_\mathrm{u} + \left(l_2 s_2 + l_3 s_{23}\right) \dot{q}_2 + l_3 c_{23} \dot{q}_3 \end{bmatrix} \tag{8}$$

The dynamic model describes the relationship between the inputs and the motion of the robot, through the Euler–Lagrange equation [25],

$$\mathcal{L}(\mathbf{q}, \dot{\mathbf{q}}) = K(\mathbf{q}, \dot{\mathbf{q}}) - U(\mathbf{q}) \tag{9}$$

where, K is the total kinetic energy of the system, U is the potential energy of the total system. The kinetic energy of the system is determined by:

$$K = K_u + K_a \tag{10}$$

K_u is the total kinetic energy of the aerial platform and K_a is the total kinetic energy of the robotic arm. The kinetic energy of the hexarotor is composed by the kinetic energy produced during translation and the kinetic energy produced during rotation, in such a way that:

$$K_u = \frac{1}{2} \mathbf{v}_\mathbf{u}^\intercal m_{fu} \mathbf{v}_\mathbf{u} + \frac{1}{2} \boldsymbol{\omega}_\mathbf{u}^\intercal I_u \boldsymbol{\omega}_\mathbf{u} \tag{11}$$

where, m_u is the mass of the hexarotor, $\mathbf{v}_\mathbf{u}$ is the linear velocity vector of the hexarotor in the fixed system, $\boldsymbol{\omega}_\mathbf{u}$ is the angular velocity and I_u is the moment of inertia of the hexarotor. Replacing the angular velocity of the hexarotor, the kinetic energy of the platform results:

$$K_u = \frac{1}{2} \mathbf{v}_\mathbf{u}^\intercal m_u \mathbf{v}_\mathbf{u} + \frac{1}{2} I_u \dot{\psi}^2 \tag{12}$$

The kinetic energy of the hexarotor written in matrix form as a function of the generalized coordinates $\dot{\mathbf{q}}$ results as follows:

$$K_u = \frac{1}{2} \dot{\mathbf{q}}^\intercal \left(\mathbf{J}_{\mathbf{v}_\mathbf{u}}^\intercal m_u \mathbf{J}_{\mathbf{v}_\mathbf{u}} + \mathbf{J}_{\boldsymbol{\omega}_\mathbf{u}}^\intercal I_u \mathbf{J}_{\boldsymbol{\omega}_\mathbf{u}} \right) \dot{\mathbf{q}} \tag{13}$$

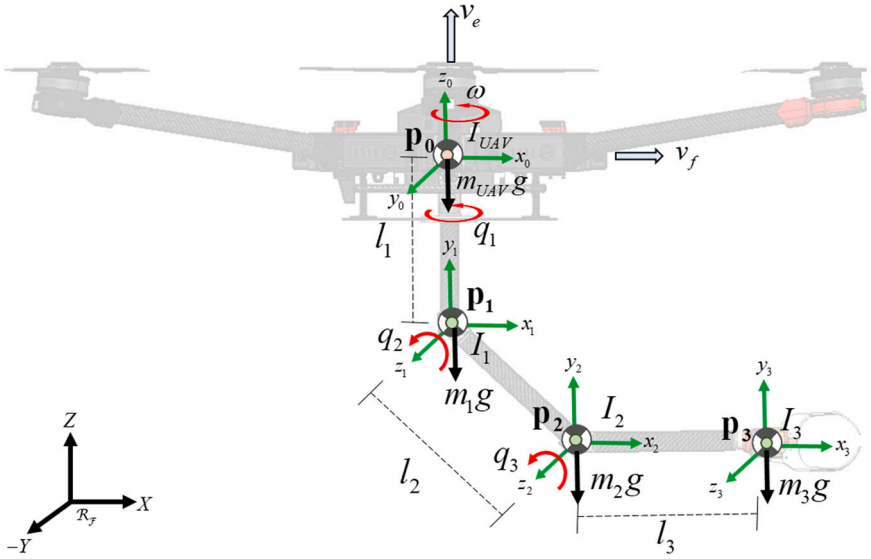


Fig. 5. Moment and mass scheme of the aerial manipulator.

where, $\dot{\mathbf{q}} = \begin{bmatrix}\dot{x}_u & \dot{y}_u & \dot{z}_u & \dot{\psi} & \dot{q}_1 & \dot{q}_2 & \dot{q}_3\end{bmatrix}^\intercal$ is the velocity vector of the generalized coordinates of the aerial manipulator, $\mathbf{J}_{\mathbf{v_u}}$ and $\mathbf{J}_{\boldsymbol{\omega}_\mathbf{u}}$ are defined as:

$$\mathbf{J}_{\mathbf{v_u}} = \begin{bmatrix}\mathbf{I}_{3\times3} & \mathbf{0}_{3\times4}\end{bmatrix} \quad \mathbf{J}_{\boldsymbol{\omega}_\mathbf{u}} = \begin{bmatrix}0 & 0 & 0 & 1 & 0 & 0 & 0 \\ & & & \mathbf{0}_{2\times7} & & & \end{bmatrix} \tag{14}$$

Now, the kinetic energy produced by the robotic arm is composed by the kinetic energy of translation and rotation produced by each link:

$$K_\mathrm{a} = K_{a1} + K_{a2} + K_{a3} \tag{15}$$

Replacing the kinetic energies we obtain:

$$K_\mathrm{a} = \frac{1}{2}\left(\mathbf{v}_1^\intercal m_1 \mathbf{v}_1 + I_1\left(\dot{\psi} + \dot{q}_1\right)^2 + \mathbf{v}_2^\intercal m_2 \mathbf{v}_2 + I_2 \dot{q}_2^2 + \mathbf{v}_3^\intercal m_3 \mathbf{v}_3 + I_3\left(\dot{q}_2 + \dot{q}_3\right)^2\right) \tag{16}$$

Then, separating the kinetic energies produced by the masses of the robotic arm links and writing in matrix form as a function of the generalized coordinates, it results that the kinetic energy for mass one is defined by:

$$K_{a1} = \frac{1}{2}\dot{\mathbf{q}}^\intercal\left(\mathbf{J}\mathbf{v}_1^\intercal m_1 \mathbf{J}\mathbf{v}_1 + \mathbf{J}\boldsymbol{\omega}_1^\intercal I_1 \mathbf{J}\boldsymbol{\omega}_1\right)\dot{\mathbf{q}} \tag{17}$$

with,

$$\mathbf{J}\mathbf{v}_1 = \begin{bmatrix}\mathbf{I}_{3\times3} & \mathbf{0}_{3\times4}\end{bmatrix} \quad \mathbf{J}\omega_1 = \begin{bmatrix}0 & 0 & 0 & 1 & 1 & 0 & 0 \\ & & & \mathbf{0}_{2\times7} & & & \end{bmatrix} \tag{18}$$

Kinetic energy of mass 2:

$$K_{a2} = \frac{1}{2}\dot{\mathbf{q}}^\intercal\left(\mathbf{J}\mathbf{v}_2^\intercal m_2 \mathbf{J}\mathbf{v}_2 + \mathbf{J}\boldsymbol{\omega}_2^\intercal I_2 \mathbf{J}\boldsymbol{\omega}_2\right)\dot{\mathbf{q}} \tag{19}$$

with,

$$\mathbf{J}\mathbf{v}_2 = \begin{bmatrix}1 & 0 & 0 & -l_2 s_{\psi 1} c_2 & -l_2 s_{\psi 1} c_2 & -l_2 c_{\psi 1} s_2 & 0 \\ 0 & 1 & 0 & l_2 c_{\psi 1} c_2 & l_2 c_{\psi 1} c_2 & -l_2 s_{\psi 1} s_2 & 0 \\ 0 & 0 & 1 & 0 & 0 & l_2 c_2 & 0\end{bmatrix}$$
$$\mathbf{J}\boldsymbol{\omega}_2 = \begin{bmatrix}0 & 0 & 0 & 0 & 0 & 1 & 0 \\ & & & \mathbf{0}_{2\times7} & & & \end{bmatrix} \tag{20}$$

The kinetic energy of mass three is determined by:

$$K_{a3} = \frac{1}{2}\dot{\mathbf{q}}^\intercal\left(\mathbf{J}\mathbf{v}_3^\intercal m_3 \mathbf{J}\mathbf{v}_3 + \mathbf{J}\boldsymbol{\omega}_3^\intercal I_3 \mathbf{J}\boldsymbol{\omega}_3\right)\dot{\mathbf{q}} \tag{21}$$

with, (see Eq. (22) in Box I). Then, the total kinetic energy of the system is given by:

$$K = \frac{1}{2}\dot{\mathbf{q}}^\intercal\left(\mathbf{J}\mathbf{v}_\mathbf{u}^\intercal m_u \mathbf{J}\mathbf{v}_\mathbf{u} + \mathbf{J}\boldsymbol{\omega}_\mathbf{u}^\intercal I_u \mathbf{J}\boldsymbol{\omega}_\mathbf{u}\right)\dot{\mathbf{q}} + \frac{1}{2}\dot{\mathbf{q}}^\intercal\sum_{i=1}^{3}\left(\mathbf{J}\mathbf{v}_i^\intercal m_i \mathbf{J}\mathbf{v}_i + \mathbf{J}\boldsymbol{\omega}_i^\intercal I_i \mathbf{J}\boldsymbol{\omega}_i\right)\dot{\mathbf{q}} \tag{23}$$

replacing the Jacobians, masses and moments of inertia results:

$$K = \frac{1}{2}\dot{\mathbf{q}}^\intercal\bar{\mathbf{M}}(\mathbf{q})\dot{\mathbf{q}} \tag{24}$$

Finally, the potential energy is determined by:

$$U = m_u g \mathbf{z}_0 \mathbf{p}_\mathbf{u} + \sum_{i=1}^{3} m_i g \mathbf{z}_0 \mathbf{p}_i \tag{25}$$

where, the vector $\mathbf{z}_0$ is given by $\mathbf{z}_0 = \begin{bmatrix}0 & 0 & 1\end{bmatrix}^\intercal$ and g is the gravity.

Replacing the kinetic and potential energies in the Euler–Lagrange Eq. (9), it results:

$$\mathcal{L}(\mathbf{q}, \dot{\mathbf{q}}) = \frac{1}{2}\dot{\mathbf{q}}^\intercal\bar{\mathbf{M}}(\mathbf{q})\dot{\mathbf{q}} - U(\mathbf{q}) \tag{26}$$

The torques and forces of the system are determined by [25]:

$$\boldsymbol{\tau} = \frac{d}{dt}\left(\frac{\partial\mathcal{L}}{\partial\dot{\mathbf{q}}}\right) - \left(\frac{\partial\mathcal{L}}{\partial\mathbf{q}}\right) \tag{27}$$

where, $\boldsymbol{\tau} = \begin{bmatrix}\mathbf{f}^\intercal & \boldsymbol{\tau}^\intercal\end{bmatrix}^\intercal = \begin{bmatrix}f_{x_u} & f_{y_u} & f_{z_u} & \tau_\psi & \tau_1 & \tau_2 & \tau_3\end{bmatrix}^\intercal$ is the vector of torques and forces associated with the generalized coordinates of the aerial manipulator. The derivative of $\mathcal{L}(\mathbf{q}, \dot{\mathbf{q}})$ with respect to the vector $\dot{\mathbf{q}}$ is:

$$\frac{\partial\mathcal{L}(\mathbf{q}, \dot{\mathbf{q}})}{\partial\dot{\mathbf{q}}} = \frac{1}{2}\frac{\partial}{\partial\dot{\mathbf{q}}}\left(\dot{\mathbf{q}}^\intercal\bar{\mathbf{M}}(\mathbf{q})\dot{\mathbf{q}}\right) = \bar{\mathbf{M}}(\mathbf{q})\dot{\mathbf{q}} \tag{28}$$

Now, deriving with respect to time the Eq. (28) results:

$$\frac{d}{dt}\left(\frac{\partial\mathcal{L}(\mathbf{q}, \dot{\mathbf{q}})}{\partial\dot{\mathbf{q}}}\right) = \bar{\mathbf{M}}(\mathbf{q})\ddot{\mathbf{q}} + \dot{\bar{\mathbf{M}}}(\mathbf{q})\ddot{\mathbf{q}} \tag{29}$$

Finally, the derivative of $\mathcal{L}(\mathbf{q}, \dot{\mathbf{q}})$ with respect to $\mathbf{q}$:

$$\frac{\partial\mathcal{L}(\mathbf{q}, \dot{\mathbf{q}})}{\partial\mathbf{q}} = \frac{1}{2}\left(\frac{\partial}{\partial\mathbf{q}}\left(\dot{\mathbf{q}}^\intercal\bar{\mathbf{M}}(\mathbf{q})\dot{\mathbf{q}}\right)\right) - \frac{\partial}{\partial\mathbf{q}}(U(\mathbf{q})) \tag{30}$$

The equations of motion written in compact matrix form representing the dynamic model of the joint space is given by:

$$\boldsymbol{\tau} = \bar{\mathbf{M}}(\mathbf{q})\ddot{\mathbf{q}} + \bar{\mathbf{C}}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \bar{\mathbf{g}}(\mathbf{q}) \tag{31}$$

where, $\bar{\mathbf{M}}(\mathbf{q}) \in \mathbb{R}^{7\times7}$ is the mass or inertia matrix of the system, which is a positive definite symmetric matrix, $\bar{\mathbf{C}}(\mathbf{q}, \dot{\mathbf{q}}) \in \mathbb{R}^{7\times7}$ is the matrix of centripetal and Coriolis forces determined by:

$$\bar{\mathbf{C}}(\mathbf{q}, \dot{\mathbf{q}}) = \dot{\bar{\mathbf{M}}}(\mathbf{q}) - \frac{1}{2}\dot{\mathbf{q}}^\intercal\frac{\partial}{\partial\mathbf{q}}\bar{\mathbf{M}}(\mathbf{q}) \tag{32}$$

and is $\bar{\mathbf{g}}(\mathbf{q}) \in \mathbb{R}^{7\times1}$ the gravitational force vector obtained as follows:

$$\bar{\mathbf{g}}(\mathbf{q}) = -\frac{\partial}{\partial\mathbf{q}}U(\mathbf{q}) \tag{33}$$

$$\mathbf{Jv_3} = \begin{bmatrix} 1 & 0 & 0 & -s_{\psi 1}\left(l_2c_2 + l_3c_{23}\right) & -s_{\psi 1}\left(l_2c_2 + l_3c_{23}\right) & -c_{\psi 1}\left(l_2s_2 + l_3s_{23}\right) & -l_3c_{\psi 1}s_{23} \\ 0 & 1 & 0 & c_{\psi 1}\left(l_2c_2 + l_3c_{23}\right) & c_{\psi 1}\left(l_2c_2 + l_3c_{23}\right) & -l_2s_{\psi 1}s_2 & -l_3s_{\psi 1}s_{23} \\ 0 & 0 & 1 & 0 & 0 & l_2c_2 + l_3c_{23} & l_3c_{23} \end{bmatrix} \quad (22)$$

$$\mathbf{J}\omega_\mathbf{3} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ & & & \mathbf{0}_{2\times 7} & & & \end{bmatrix}$$

Box I.

The dynamic model (31) verifies all properties defined in [25]. Now, let us represent the generalized coordinate velocity vector $\dot{\mathbf{q}}$ in terms of the velocity vector of the aerial platform by the equality $\dot{\mathbf{q}} = \mathbf{T}(\mathbf{q})\,\boldsymbol{\mu}$. Where, $\mathbf{T}(\mathbf{q}) \in \mathbb{R}^{7\times 7}$ is the matrix to transform the associated velocities in the fixed frame $\mathrm{R}_\mathcal{F}$ to the mobile reference frame of the hexarotor $\mathrm{R}_\mathcal{U}$, and $\boldsymbol{\mu} = \left[v_f, v_l, v_e, \omega, \dot{q}_1, \dot{q}_2, \dot{q}_3\right]^\intercal$ is the vector of robot maneuverability velocities. The acceleration $\ddot{\mathbf{q}}$ is obtained from the derivative:

$$\ddot{\mathbf{q}} = \frac{d}{dt}\left(\mathbf{T}(\mathbf{q})\,\boldsymbol{\mu}\right) = \mathbf{T}(\mathbf{q})\,\dot{\boldsymbol{\mu}} + \dot{\mathbf{T}}(\mathbf{q})\,\boldsymbol{\mu} \quad (34)$$

The dynamic model (31) as a function of robot velocities, accelerations, forces and control torques is defined by:

$$\mathbf{f_\tau} = \left[\mathbf{T}(\mathbf{q})^{-1}\bar{\mathbf{M}}(\mathbf{q})\,\mathbf{T}(\mathbf{q})\right]\dot{\boldsymbol{\mu}} + \left[\mathbf{T}(\mathbf{q})^{-1}\left(\bar{\mathbf{C}}(\mathbf{q},\boldsymbol{\mu})\,\mathbf{T}(\mathbf{q}) + \bar{\mathbf{M}}(\mathbf{q})\,\dot{\mathbf{T}}(\mathbf{q})\right)\right]\boldsymbol{\mu} \\ \cdots + \left[\mathbf{T}(\mathbf{q})^{-1}\bar{\mathbf{g}}(\mathbf{q})\right] \quad (35)$$

where, $\mathbf{f_\tau} = \left[\mathbf{f_u}^\intercal \quad \boldsymbol{\tau_a}^\intercal\right]^\intercal$ is the vector of forces and torques of the aerial manipulator robot and is composed by $\mathbf{f_u} = \left[f_f \quad f_l \quad f_e \quad \tau_\psi\right]^\intercal$; and $\boldsymbol{\tau_a} = \left[\tau_1 \quad \tau_2 \quad \tau_3\right]^\intercal$. Now, to transform the dynamic model of the aerial manipulator as a function of reference velocities, we first obtain the transformation of the torques of the robotic arm as a function of velocities considering the electro servo actuators. The torques applied on the robotic arm joints are calculated by:

$$\tau_i = \frac{k_a}{r_a}\left(v_{t_i} - k_b\dot{q}_i\right) \quad i = 1,2,3 \quad (36)$$

where, $\dot{q}_i$ represents the angular velocity produced by each joint; k_a is the torque constant multiplied by the reduction constant; r_a is the armature resistance of the motor; k_b is the counter-electromotive constant multiplied by the reduction constant and v_{t_i} is the voltage applied to the motor. On the other hand, the motor voltages are controlled by a PD controller, which is common in commercial motors to follow reference speeds. The model of the controller is described by:

$$v_{t_i} = k_p\left(\dot{q}_{i_{ref}} - \dot{q}_i\right) - k_d\ddot{q}_i \quad (37)$$

where, $\dot{q}_{i_{ref}}$ is the reference angular velocity and $\ddot{q}_i$ is the angular acceleration produced by the motors. Replacing Eq. (37) in the equation of the motor torques (36) results:

$$\tau_i = \frac{k_a}{r_a}\left(k_p\dot{q}_{i_{ref}} - \left(k_p - k_b\right)\dot{q}_i - k_d\ddot{q}_i\right) \quad (38)$$

The torque vector of the robotic arm $\boldsymbol{\tau_a}$ is determined by:

$$\boldsymbol{\tau_a} = \mathbf{G_p}\dot{\mathbf{q}}_{\mathbf{a_{ref}}} - \left(\mathbf{G_p} + \mathbf{G_b}\right)\dot{\mathbf{q}}_\mathbf{a} - \mathbf{G_d}\ddot{\mathbf{q}}_\mathbf{a} \quad (39)$$

where, $\mathbf{G_p} = \left(\frac{k_a k_p}{r_a}\right)\mathbf{I_{3\times 3}}$; $\mathbf{G_d} = \left(\frac{k_a k_d}{r_a}\right)\mathbf{I_{3\times 3}}$ and $\mathbf{G_b} = \left(\frac{k_a k_b}{r_a}\right)\mathbf{I_{3\times 3}}$.

Now, since the hexarotor is a robot that accepts reference velocity commands, it is assumed that the internal control (attitude and altitude) of the hexarotor has PD-type controllers. The robot has four controllers that handle the three linear velocities and the rotational (yaw) speed. As a first part, we have the expression of the force generated in each direction of the robot as a function of velocities [26]. Furthermore, according to the aerodynamic properties [27,28], there is a velocity that opposes the thrust force of the rotors in each direction of the platform. Therefore, the expression of the total force exerted by the robot is given by:

$$f_i = k_{fi}\left(t_i - k_{ti}v_i\right) \quad i = f,l,e \quad (40)$$

where, t_i is the thrust of the hexarotor; k_{ti} is a constant that transforms the current velocity v_i into opposite thrust due to aerodynamic effects and k_{fi} is a drag constant that transforms into linear displacement force. Now, it is assumed that the internal control of the hexarotor has PD controllers for the thrust, therefore, the thrust in the three directions is:

$$t_i = k_p\left(v_{i_{ref}} - v_i\right) - k_d\dot{v}_i \quad (41)$$

As in (40), it is assumed that the rotational torque produced by the hexarotor is given by

$$\tau_\psi = k_\tau\left(t_\psi - k_\psi\omega\right) \quad (42)$$

And the PD-type controller for the rotational torque t_ψ of the robot is:

$$t_\psi = k_p\left(\omega_{ref} - \omega\right) - k_d\dot{\omega} \quad (43)$$

In general the forces for linear displacement and the rotation torque of the robot are defined by:

$$f_i = k_{fi}k_p v_{i_{ref}} - \left(k_{fi}k_p + k_{fi}k_t\right)v_i - k_{fi}k_d\dot{v}_i \\ \tau_\psi = k_\tau k_p\omega_{ref} - \left(k_\tau k_p + k_\tau k_\psi\right)\omega - k_\tau k_d\dot{\omega} \quad (44)$$

Then, the forces for linear displacement and the rotational torque of the robot are calculated by:

$$\mathbf{f_u} = \mathbf{G_{p1}}\mathbf{v}_{\mathbf{u_{ref}}} - \left(\mathbf{G_{p1}} + \mathbf{G_{b1}}\right)\mathbf{v_u} - \mathbf{G_{d1}}\dot{\mathbf{v}}_\mathbf{u} \quad (45)$$

where, $\mathbf{v}_{\mathbf{u_{ref}}} = \left[v_{f_{ref}}, v_{l_{ref}}, v_{e_{ref}}, \omega_{ref}\right]^\intercal$ is the hexarotor reference velocity commands vector, while the actual linear velocities vector that the hexarotor generates is $\mathbf{v_u} = \left[v_f, v_l, v_e, \omega\right]^\intercal$; and $\dot{\mathbf{v}}_\mathbf{u} = \left[\dot{v}_f, \dot{v}_l, \dot{v}_e, \dot{\omega}\right]^\intercal$ is the acceleration vector of the hexarotor. The matrices that represent the transformation of forces into references velocities are defined by: $\mathbf{G_{p1}} = diag\left(\left[k_{pf}, k_{pl}, k_{pe}, k_{p\psi}\right]\right)$, $\mathbf{G_{b1}} = diag\left(\left[k_{tf}, k_{tl}, k_{te}, k_{t\psi}\right]\right)$ and $\mathbf{G_{d1}} = diag\left(\left[k_{df}, k_{dl}, k_{de}, k_{d\psi}\right]\right)$.

Then, expressing the system in hexarotor and robotic arm reference velocities, it results:

$$\mathbf{f_\tau} = \mathbf{G_{f_p}}\boldsymbol{\mu}_\mathbf{ref} - \mathbf{G_{f_b}}\boldsymbol{\mu} - \mathbf{G_{f_d}}\dot{\boldsymbol{\mu}} \quad (46)$$

where, $\boldsymbol{\mu}_\mathbf{ref} = \left[\mathbf{v}_{\mathbf{u_{ref}}}^\intercal \quad \dot{\mathbf{q}}_{\mathbf{a_{ref}}}^\intercal\right]^\intercal$ is the reference velocity vector of the aerial manipulator robot; $\boldsymbol{\mu} = \left[\mathbf{v_u}^\intercal \quad \dot{\mathbf{q}}_\mathbf{a}^\intercal\right]^\intercal$ is the velocity vector of the robot; $\dot{\boldsymbol{\mu}} = \left[\dot{\mathbf{v}}_\mathbf{u}^\intercal \quad \ddot{\mathbf{q}}_\mathbf{a}^\intercal\right]^\intercal$ is the acceleration vector of the robot and

$$\mathbf{G_{f_p}} = \begin{bmatrix} \mathbf{G_{p1}} & \mathbf{0_{3\times 3}} \\ \mathbf{0_{3\times 4}} & \mathbf{G_p} \end{bmatrix} \quad \mathbf{G_{f_b}} = \begin{bmatrix} \mathbf{G_{p1}} + \mathbf{G_{b1}} & \mathbf{0_{3\times 3}} \\ \mathbf{0_{3\times 4}} & \mathbf{G_p} + \mathbf{G_b} \end{bmatrix} \quad \mathbf{G_{f_d}} = \begin{bmatrix} \mathbf{G_{d1}} & \mathbf{0_{3\times 3}} \\ \mathbf{0_{3\times 4}} & \mathbf{G_d} \end{bmatrix} \quad (47)$$

Now, considering the dynamic model of the robot (35) and (46), we obtain:

$$\boldsymbol{\mu}_\mathbf{ref} = \mathbf{G}_{\mathbf{f_p}}^{-1}\left[\mathbf{T}(\mathbf{q})^{-1}\bar{\mathbf{M}}(\mathbf{q})\,\mathbf{T}(\mathbf{q}) + \mathbf{G_{f_d}}\right]\dot{\boldsymbol{\mu}} + \\ \cdots + \mathbf{G}_{\mathbf{f_p}}^{-1}\left[\mathbf{T}(\mathbf{q})^{-1}\bar{\mathbf{C}}(\mathbf{q},\boldsymbol{\mu})\,\mathbf{T}(\mathbf{q}) + \mathbf{T}(\mathbf{q})^{-1}\bar{\mathbf{M}}(\mathbf{q})\,\dot{\mathbf{T}}(\mathbf{q}) + \mathbf{G_{f_d}}\right]\boldsymbol{\mu} + \\ \cdots + \mathbf{G}_{\mathbf{f_p}}^{-1}\mathbf{T}(\mathbf{q})^{-1}\bar{\mathbf{g}}(\mathbf{q}) \quad (48)$$

Finally, the dynamic model can be written as follows:

$$\boldsymbol{\mu}_\mathbf{ref} = \mathbf{M}(\mathbf{q})\,\dot{\boldsymbol{\mu}} + \mathbf{C}(\mathbf{q},\boldsymbol{\mu})\,\boldsymbol{\mu} + \mathbf{g}(\mathbf{q}) \quad (49)$$

where,

$$\begin{cases} \mathbf{M}(\mathbf{q}) = \mathbf{G}_{\mathbf{f_p}}^{-1}\left[\mathbf{T}(\mathbf{q})^{-1}\bar{\mathbf{M}}(\mathbf{q})\mathbf{T}(\mathbf{q}) + \mathbf{G}_{\mathbf{f_d}}\right] \\ \mathbf{C}(\mathbf{q},\boldsymbol{\mu}) = \mathbf{G}_{\mathbf{f_p}}^{-1}\left[\mathbf{T}(\mathbf{q})^{-1}\bar{\mathbf{C}}(\mathbf{q},\boldsymbol{\mu})\mathbf{T}(\mathbf{q}) + \mathbf{T}(\mathbf{q})^{-1}\bar{\mathbf{M}}(\mathbf{q})\dot{\mathbf{T}}(\mathbf{q}) + \mathbf{G}_{\mathbf{f_d}}\right] \\ \mathbf{g}(\mathbf{q}) = \mathbf{G}_{\mathbf{f_p}}^{-1}\mathbf{T}(\mathbf{q})^{-1}\bar{\mathbf{g}}(\mathbf{q}) \end{cases} \tag{50}$$

Remark 2. Eq. (49) represents the dynamic model of the AM without taking into account external disturbances and unmodeled robot dynamics. The dynamic model considering perturbations and errors in the modeling is represented by $\boldsymbol{\mu}_{\mathbf{ref}} = \mathbf{M}(\mathbf{q})\dot{\boldsymbol{\mu}} + \mathbf{C}(\mathbf{q},\boldsymbol{\mu})\boldsymbol{\mu} + \mathbf{g}(\mathbf{q}) - \boldsymbol{\delta}_{\bar{\boldsymbol{\mu}}}(k)$, where, $\boldsymbol{\delta}_{\bar{\boldsymbol{\mu}}}(k) = f(\boldsymbol{\gamma}_{\varepsilon}, \boldsymbol{\gamma}_{\mu})$ is a vector depending on external disturbances $\gamma_{\varepsilon} \in \mathbb{R}^n$ and on unmodeled dynamics $\gamma_{\mu} \in \mathbb{R}^n$. However, not taken into account in (49), it is considered in the control error behavior through the robustness analysis in Eq. (81).

4.4. Parametric identification of the dynamic model

The method used for the identification is through general or black box identification, where the objective is to establish the input–output relationships of the system, without making physical interpretations on the composition of the mathematical model. In order to identify the parameters of the dynamic model of the aerial manipulator robot, it is done through the minimization of an objective function:

$$\min f(\boldsymbol{\vartheta}) \tag{51}$$

where, $\boldsymbol{\vartheta} \in \mathbb{R}^{\eta}$ is the vector of dimension η that represents the parameters of the system to be identified. The function to optimize to find the parameters is given by:

$$f(\boldsymbol{\vartheta}) = \left\|\boldsymbol{\mu}_{\mathbf{ex}} - \bar{\boldsymbol{\mu}}\right\| \tag{52}$$

such that, $\boldsymbol{\mu}_{\mathbf{ex}}$ is the reference velocity that excites the real robot; $\bar{\boldsymbol{\mu}}$ is the velocity provided by the dynamic model obtained from the real data of the robot and is calculated by:

$$\bar{\boldsymbol{\mu}} = \mathbf{M}\left(\mathbf{q}^0\right)\dot{\boldsymbol{\mu}}^0 + \mathbf{C}\left(\mathbf{q}^0, \boldsymbol{\mu}^0\right)\boldsymbol{\mu}^0 + \mathbf{g}\left(\mathbf{q}^0\right) \tag{53}$$

where, $\mathbf{q}^0$, $\boldsymbol{\mu}^0$ and $\dot{\boldsymbol{\mu}}^0$ are vectors of actual positions, velocities and accelerations obtained from the robotic system. As the velocity readings and accelerations of the robot contain noise, the acceleration $\dot{\boldsymbol{\mu}}^0$ is estimated from a first-order filter $f(s) = \frac{\lambda}{s+\lambda}$ with $\lambda > 0$.

Parameter identification is done through the Quasi-Newton optimization algorithm [29]. It is a linear search method that solves a minimization sub-problem by approximating the Hessian of the Lagrangian. When the value of $f(\boldsymbol{\vartheta})$ reaches a minimum, the search provides the vector of parameters $\boldsymbol{\vartheta}$ that fulfill the minimization of the function. The dynamic model has a total of twenty-seven parameters ($\eta = 27$), which represent nonlinear combinations of masses of links, inertia moments of the hexarotor and of the links of the robotic arm, among other parameters that make up the dynamic model.

5. Control scheme

The proposed control scheme for an aerial manipulator robot is shown in Fig. 6. The proposed controller can be applied to solve the motion control problem of positioning, trajectory tracking and path-following by appropriate selection of control references in the task configuration layer of the proposed multi-layer scheme, see Fig. 1.

1. Nonlinear kinematic control. It is a kinematic controller of minimal norm with saturation of the maneuverability commands of the aerial manipulator robot. This controller receives as inputs $\boldsymbol{\xi}_{\mathbf{d}}(k) \in \mathbb{R}^m$ and $v_d(k) \in \mathbb{R}^+$ which describe the desired position and velocity of the end-effector with respect to the inertial reference frame $\mathrm{R}_{\mathcal{I}}$; and k represents the discrete time instant. Control errors are defined as $\tilde{\boldsymbol{\xi}}(k) = \boldsymbol{\xi}_{\mathbf{d}}(k) - \boldsymbol{\xi}(k) = \Delta\boldsymbol{\xi}(k) \in \mathbb{R}^m$. Therefore, the control objective of the aerial manipulator robot can be defined as:

$$\lim_{k\to\infty}\left(\boldsymbol{\xi}_{\mathbf{d}}(k) - \boldsymbol{\xi}(k)\right) = \mathbf{0} \in \mathbb{R}^m \tag{54}$$

2. Dynamic compensation. It is a control law that compensates the dynamics of the aerial manipulator robot in order to reduce velocity tracking errors and disturbances caused by environmental conditions and dynamic effects of the system. This controller receives as inputs the vector $\boldsymbol{\mu}_{\mathbf{K}}(k) \in \mathbb{R}^n$ calculated by the nonlinear kinematic controller and generates the maneuverability commands of the aerial manipulator robot $\boldsymbol{\mu}_{\mathbf{D}}(k) \in \mathbb{R}^n$. Control errors are defined as $\tilde{\boldsymbol{\mu}}(k) = \boldsymbol{\mu}_{\mathbf{K}}(k) - \boldsymbol{\mu}(k) = \Delta\tilde{\boldsymbol{\mu}}(k) \in \mathbb{R}^n$. Therefore, the dynamic compensation control objective can be defined as:

$$\lim_{k\to\infty}\left(\boldsymbol{\mu}_{\mathbf{K}}(k) - \boldsymbol{\mu}(k)\right) = \mathbf{0} \in \mathbb{R}^n \tag{55}$$

6. Nonlinear kinematic and dynamic control

The nonlinear kinematic controller at $t = kT_0$ (where, T_0 represents the sampling period with respect to Nyquist's theorem, and $k \in \{0, 1, 2, 3, \ldots\}$ are the samples of the continuous response) receives the desired position and velocity of the robot end-effector $\boldsymbol{\xi}_{\mathbf{d}}(k) \in \mathbb{R}^m$ and $v_d(k) \in \mathbb{R}^+$ respectively. This controller generates the maneuverability velocities of the robotic system $\boldsymbol{\mu}_{\mathbf{K}}(k) \in \mathbb{R}^n$. In other words, the desired operational motion of the robot is an application of $\boldsymbol{\xi}_{\mathbf{d}}(k)$ and $v_d(k)$. Therefore, the motion control problem is to determine the maneuverability vector $\boldsymbol{\mu}_{\mathbf{K}}(k)$ to achieve the desired motion of the robotic system. The corresponding evolution of the whole robotic system is given by the vector of generalized coordinates of the aerial manipulator $\mathbf{q}(k)$. The design of the proposed control law is based on the tools of numerical methods and linear algebra properties [30]. Therefore, the kinematic model of an aerial manipulator robot (4) can be discretized through Euler's method, where it is possible to determine the state of the system at instant $(k+1)$ as:

$$\boldsymbol{\xi}(k+1) = T_0\mathbf{J}(\mathbf{q}(k))\boldsymbol{\mu}(k) + \boldsymbol{\xi}(k) \tag{56}$$

On the other hand, the use of numerical methods to calculate the evolution of systems is mainly based on the possibility of determining the state of the system at instant $(k+1)$, if the state and the control action are known at instant k. Then, a variable at instant $(k+1)$ can be substituted by the desired variable and then calculate the necessary control action to make the output system change from its current value to a desired one. To achieve this control objective, it is proposed that the behavior of the error variation is given by $\dot{\tilde{\boldsymbol{\xi}}}_{\mathbf{d}}(t) + \boldsymbol{\Gamma}\tilde{\boldsymbol{\xi}}(t) = \mathbf{0}$, where $\boldsymbol{\Gamma}$ is a square weight matrix. Then, discretizing the error equation proposed through Euler and considering the error at instant k is $\tilde{\boldsymbol{\xi}}(k) = \boldsymbol{\xi}_{\mathbf{d}}(k) - \boldsymbol{\xi}(k)$, it turns out that the state at the instant (k + 1) of the system as a function of the desired state and the control action results:

$$\boldsymbol{\xi}(k+1) = \boldsymbol{\xi}_{\mathbf{d}}(k+1) - \boldsymbol{\xi}_{\mathbf{d}}(k) + \boldsymbol{\xi}(k) + T_0\boldsymbol{\Gamma}(k)\left(\boldsymbol{\xi}_{\mathbf{d}}(k) - \boldsymbol{\xi}(k)\right) \tag{57}$$

where, $\boldsymbol{\Gamma}(k) \in \mathbb{R}^{m\times m}$ is a diagonal matrix that weights and saturates the control errors defined as $\boldsymbol{\Gamma}(k) = diag\left(\Gamma_i/\left(\sigma_i + \left|\tilde{\xi}_i(k)\right|\right)\right)$ where, $\sigma_i \in \mathbb{R}^+$ defines the saturation rate of $\tilde{\xi}_i(k)$; $T_0\Gamma_i \in \mathbb{R}^+$ represents the maximum saturation value for the ith component of the control error vector $\tilde{\boldsymbol{\xi}}(k)$. On the other hand, equating (56) and (57) we obtain the following system of linear equations:

$$\mathbf{J}(\mathbf{q}(k))\boldsymbol{\mu}(k) = \frac{1}{T_0}\left(\boldsymbol{\xi}_{\mathbf{d}}(k+1) - \boldsymbol{\xi}_{\mathbf{d}}(k)\right) + \boldsymbol{\Gamma}(k)\left(\boldsymbol{\xi}_{\mathbf{d}}(k) - \boldsymbol{\xi}(k)\right) \tag{58}$$

Now, considering that the aerial manipulator robot is a redundant system, *i.e.*, $\mathbf{J}(\mathbf{q}(k)) \in \mathbb{R}^{m\times n}$ with $m < n$ it is necessary to consider the right-hand pseudo inverse of $\mathbf{J}(\mathbf{q}(k))$. In addition, it is known that a system of linear equations is homogeneous if it can be written in the form $\mathbf{J}(\mathbf{q}(k))\boldsymbol{\mu}(k) = \mathbf{0}$. Now, if it is considered that the configuration of the robotic system is redundant, therefore, $\mathbf{J}(\mathbf{q}(k)) \in \mathbb{R}^{m\times n}$ has more unknowns than equations with range $r = n$ for every

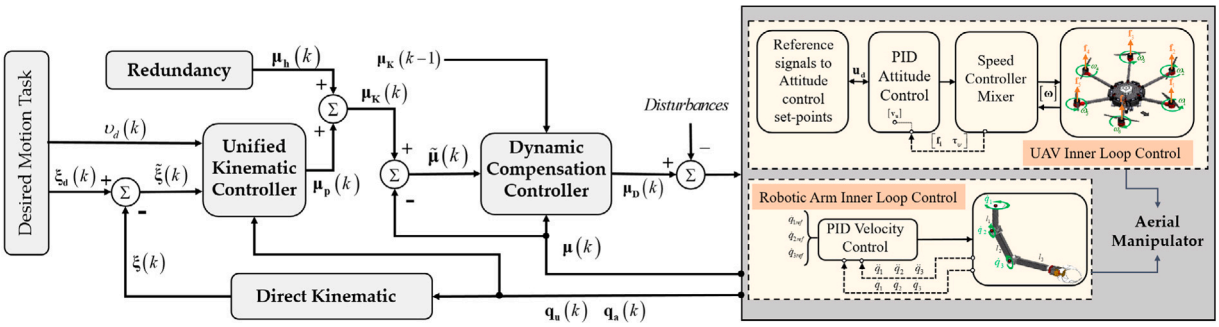


Fig. 6. Cascade control scheme.

$\left(\frac{1}{T_0}\left(\boldsymbol{\xi}_{\mathbf{d}}(k+1)-\boldsymbol{\xi}_{\mathbf{d}}(k)\right)+\boldsymbol{\Gamma}(k)\left(\boldsymbol{\xi}_{\mathbf{d}}(k)-\boldsymbol{\xi}(k)\right)\right)$ and taking into account that the homogeneous equation has a non-trivial solution, the system could have infinite solutions. In this case, consider that (58) is consistent for $\left(\frac{1}{T_0}\left(\boldsymbol{\xi}_{\mathbf{d}}(k+1)-\boldsymbol{\xi}_{\mathbf{d}}(k)\right)+\boldsymbol{\Gamma}(k)\left(\boldsymbol{\xi}_{\mathbf{d}}(k)-\boldsymbol{\xi}(k)\right)\right)$ and being that $\boldsymbol{\mu}_{\mathbf{p}}(k)\in\mathbb{R}^n$ is a particular solution, the solution of (58) is the set of all vectors of the form:

$$\boldsymbol{\mu}_{\mathbf{K}}(k)=\boldsymbol{\mu}_{\mathbf{h}}(k)+\boldsymbol{\mu}_{\mathbf{p}}(k) \tag{59}$$

where, $\boldsymbol{\mu}_{\mathbf{h}}(k)$ is any solution of the homogeneous system $\mathbf{J}(\mathbf{q}(k))\,\boldsymbol{\mu}(k)=\mathbf{0}$. A viable solution method is to formulate the problem as a constrained linear optimization problem $\frac{1}{2}\|\boldsymbol{\mu}(k)\|_2^2=\min$, obtaining as a particular solution:

$$\boldsymbol{\mu}_{\mathbf{p}}(k)=\mathbf{J}(\mathbf{q}(k))^{\dagger}\left(\frac{\boldsymbol{\xi}_{\mathbf{d}}(k+1)-\boldsymbol{\xi}_{\mathbf{d}}(k)}{T_0}\right)+\mathbf{J}(\mathbf{q}(k))^{\dagger}\left(\boldsymbol{\Gamma}(k)\left(\boldsymbol{\xi}_{\mathbf{d}}(k)-\boldsymbol{\xi}(k)\right)\right) \tag{60}$$

where, $\mathbf{J}(\mathbf{q}(k))^{\dagger}=\mathbf{J}(\mathbf{q}(k))^{\intercal}\left(\mathbf{J}(\mathbf{q}(k))\mathbf{J}(\mathbf{q}(k))^{\intercal}\right)^{-1}$ is the right-hand pseudo inverse matrix of $\mathbf{J}(\mathbf{q}(k))$. The second term of (60) allows to perform positioning tasks, while the first term allows to perform velocity tracking. Therefore, Eq. (60) can be unified in such a way that it allows to perform the two tasks only by modifying the velocity parameters and is defined by:

$$\boldsymbol{\mu}_{\mathbf{p}}(k)=\mathbf{J}(\mathbf{q}(k))^{\dagger}\left(\boldsymbol{\nu}_{\mathbf{d}}(k)+\boldsymbol{\Gamma}(k)\left(\boldsymbol{\xi}_{\mathbf{d}}(k)-\boldsymbol{\xi}(k)\right)\right) \tag{61}$$

where, $\boldsymbol{\nu}_{\mathbf{d}}(k)$ represents desired tracking velocity. Now, $\boldsymbol{\mu}_{\mathbf{h}}(k)$ as a homogeneous solution of (58) is the set of solutions where $\mathbf{J}(\mathbf{q}(k))\,\boldsymbol{\mu}(k)=\mathbf{0}$. Therefore, for the optimization problem it is necessary to define the cost function as $\min\left(\frac{1}{2}\|\boldsymbol{\mu}-\boldsymbol{\mu}_0\|_2^2\right)$ with $\boldsymbol{\mu}_0\in\mathbb{R}^n$ thus obtaining the homogeneous solution:

$$\boldsymbol{\mu}_{\mathbf{h}}(k)=\left(\mathbf{I}_{n\times n}-\mathbf{J}(\mathbf{q}(k))^{\dagger}\mathbf{J}(\mathbf{q}(k))\right)\boldsymbol{\mu}_{\mathbf{0}}(k) \tag{62}$$

where, $\left(\mathbf{I}_{n\times n}-\mathbf{J}(\mathbf{q}(k))^{\dagger}\mathbf{J}(\mathbf{q}(k))\right)$ is the orthogonal projection operator in the null space of $\mathbf{J}(\mathbf{q}(k))$ and $\boldsymbol{\mu}_{\mathbf{0}}(k)$ is the velocity vector that affects only the internal structure of the system and does not affect the position of the end-effector of the aerial manipulator. Finally, considering (61) and (62) in (58), the following control law is proposed:

$$\begin{aligned}\boldsymbol{\mu}_{\mathbf{K}}(k)=&\underbrace{\mathbf{J}(\mathbf{q}(k))^{\dagger}\left(\boldsymbol{\nu}_{\mathbf{d}}(k)+\boldsymbol{\Gamma}(k)\left(\boldsymbol{\xi}_{\mathbf{d}}(k)-\boldsymbol{\xi}(k)\right)\right)}_{\boldsymbol{\mu}_{\mathbf{p}}(k)}+\\&\cdots+\underbrace{\left(\mathbf{I}_{n\times n}-\mathbf{J}(\mathbf{q}(k))^{\dagger}\mathbf{J}(\mathbf{q}(k))\right)\boldsymbol{\mu}_0(k)}_{\boldsymbol{\mu}_{\mathbf{h}}(k)}\end{aligned} \tag{63}$$

In the proposed control law (63) the first term on the right describes the main task of the end-effector which minimizes the term $\frac{1}{2}\|\boldsymbol{\mu}(k)\|_2^2$. While the second term defines the movement of the internal configuration of the aerial manipulator robot, therefore, this type of system allows to establish several secondary control objectives that do not affect the main task of the end-effector. The redundancy of aerial manipulator robots can be effectively used to achieve additional features such as: avoidance of obstacles in the workspace and singular configurations, safe configurations for the robotic arm or for optimization of various motion criteria.

Table 1
Evolution of the control error $\tilde{\xi}_i(k)$ at time instant k.

k	$\tilde{\xi}_i(k+1)$	$\left(1-T_0\Gamma_{ii}\right)\tilde{\xi}_i(k)$
1	$\tilde{\xi}_i(2)$	$\left(1-T_0\Gamma_{ii}\right)\tilde{\xi}_i(1)$
2	$\tilde{\xi}_i(3)$	$\left(1-T_0\Gamma_{ii}\right)\tilde{\xi}_i(2)=\left(1-T_0\Gamma_{ii}\right)^2\tilde{\xi}_i(1)$
3	$\tilde{\xi}_i(4)$	$\left(1-T_0\Gamma_{ii}\right)\tilde{\xi}_i(3)=\left(1-T_0\Gamma_{ii}\right)^3\tilde{\xi}_i(1)$
⋮	⋮	⋮
n	$\tilde{\xi}_i(n+1)$	$\left(1-T_0\Gamma_{ii}\right)\tilde{\xi}_i(n)=\left(1-T_0\Gamma_{ii}\right)^n\tilde{\xi}_i(1)$

6.1. Stability analysis

Now, to demonstrate the stability of the proposed control law (63), perfect velocity tracking is considered, *i.e.*, $\boldsymbol{\mu}_{\mathbf{K}}(k)\equiv\boldsymbol{\mu}(k)$. Therefore, the closed-loop equation is obtained by equating Eq. (63) with (56) which belongs to the discretized kinematic model of the system:

$$\begin{aligned}\frac{1}{T_0}\left(\boldsymbol{\xi}(k+1)-\boldsymbol{\xi}(k)\right)=&\mathbf{J}(\mathbf{q}(k))\left(\mathbf{J}(\mathbf{q}(k))^{\dagger}\left(\nu_d(k)+\boldsymbol{\Gamma}(k)\left(\xi_d(k)-\boldsymbol{\xi}(k)\right)\right)\right.\\&\left.\cdots+\left(\mathbf{I}_{n\times n}\right)-\mathbf{J}(\mathbf{q}(k))^{\dagger}\mathbf{J}(\mathbf{q}(k))\,\boldsymbol{\mu}_{\mathbf{0}}(k)\right)\end{aligned} \tag{64}$$

Considering that $\mathbf{J}(\mathbf{q}(k))\mathbf{J}(\mathbf{q}(k))^{\dagger}=\mathbf{I}\in\mathbb{R}^{m\times m}$ and after simplifying it results:

$$\boldsymbol{\xi}(k+1)-\boldsymbol{\xi}(k)=T_0\boldsymbol{\nu}_{\mathbf{d}}(k)+T_0\boldsymbol{\Gamma}(k)\left(\boldsymbol{\xi}_{\mathbf{d}}(k)-\boldsymbol{\xi}(k)\right) \tag{65}$$

In the following, it is studied the convergence of the unified kinematic controller to any of the basic motion control objectives.

6.1.1. Case 1 (Trajectory tracking problem)

In order to analyze the evolution of the control error for the trajectory tracking problem in the closed-loop equation (65) we consider that $\boldsymbol{\nu}_{\mathbf{d}}(k)=\frac{1}{T_0}\left(\boldsymbol{\xi}_{\mathbf{d}}(k+1)-\boldsymbol{\xi}_{\mathbf{d}}(k)\right)$, therefore:

$$\tilde{\boldsymbol{\xi}}(k+1)=\left(\mathbf{I}_{m\times m}-T_0\boldsymbol{\Gamma}\right)\tilde{\boldsymbol{\xi}}(k) \tag{66}$$

Table 1 represents the evolution of the ith control error of Eq. (66) for trajectory tracking problem.

If $n\to\infty$, the control error $\tilde{\xi}_i(n+1)$ will be $\tilde{\xi}_i(\infty)=\left(1-T_0\Gamma_{ii}\right)^{\infty}\tilde{\xi}_i(1)$, therefore, if $0<T_0\Gamma_{ii}<1$ then the control error results in $\tilde{\xi}_i(\infty)=0$ with $k\to\infty$. Using an analysis similar to that of control error $\tilde{\xi}_i(k)$ we can conclude that:

$$\lim_{k\to\infty}\tilde{\boldsymbol{\xi}}\left(kT_0\right)=\mathbf{0}\in\mathbb{R}^m$$

with $0<T_0\max\left(diag\left(\boldsymbol{\Gamma}\right)\right)<1$ when $k\to\infty$ the ith control error of the aerial manipulator robot is asymptotically stable satisfying the control objective.

6.1.2. Case 2 (Position control)

The stability analysis for the position control problem is similar to the trajectory tracking problem, with the difference that $\boldsymbol{\xi_d}(k+1) = \boldsymbol{\xi_d}(k)$. Therefore, it can be directly concluded that:

$$\lim_{k\to\infty} \tilde{\boldsymbol{\xi}}\left(kT_0\right) = \mathbf{0} \in \mathbb{R}^m$$

6.1.3. Case 3 (Path-following control)

It is analyzed the evolution of the control error for the path-following problem in the closed-loop equation (65), with $\xi_{\mathbf{d}}(t)$ the closest point on the path to the end-effector with a desired angular position $\xi_{od}(t)$ and $\boldsymbol{\nu}_{\mathbf{d}}(t) = \begin{bmatrix}\mathbf{v_p}(t)^\intercal & \dot{\xi}_{od}(t)\end{bmatrix}^\intercal$, where, $\mathbf{v_p}(t)$ is the desired velocity in the tangential direction to the closest point of the end-effector.

Stability for orientation path-following. From (65) let us only consider the equation for the orientation, such that:

$$\xi_o(k+1) - \xi_o(k) = T_0\dot{\xi}_{od}(k) + T_0\Gamma_o\left(\xi_{od}(k) - \xi_o(k)\right) \tag{67}$$

The desired velocity of the end-effector is defined as $\dot{\xi}_{od}(k) = \frac{\xi_{od}(k+1)-\xi_{od}(k)}{T_0}$ where $\xi_{od}(k)$ is the desired orientation at the closest point $\mathbf{p_d}(k)$ at instant k which implies that the closed-loop equation for the orientation results in the following equation $\tilde{\xi}_o(k+1) = \left(1-\Gamma_o\right)\tilde{\xi}_o(k)$. Using the analysis similar to (66) results in $\tilde{\xi}_o(\infty) = \left(1-T_0\Gamma_o\right)^\infty\tilde{\xi}_o(1)$ if $0 < T_0\Gamma_o < 1$ leads to the $\lim_{k\to\infty}\tilde{\xi}_o\left(kT_0\right) = 0$. Therefore, the control error for the orientation is asymptotically stable.

Stability for position path-following. Now, from (65) only the equation for the position of the end-effector is considered, such that $\boldsymbol{\xi}(k) = \begin{bmatrix}\xi_x(k) & \xi_y(k) & \xi_z(k)\end{bmatrix}^\intercal$ and $\mathbf{p_d}$ as the desired vector representing the closest point to the end-effector of the aerial manipulator, then we obtain that:

$$\boldsymbol{\xi}(k+1) - \boldsymbol{\xi}(k) = T_0\mathbf{v_p}(k) + T_0\boldsymbol{\Gamma}\left(\mathbf{p_d}(k) - \boldsymbol{\xi}(k)\right) \tag{68}$$

Note that $\mathbf{v_p}(k)$ is a tangential velocity in $\mathbf{p_d}(k)$ that does not come from a derivative with respect to time and has a desired speed $\left\|\mathbf{v_p}\right\|$. Now, rewriting (68) in the following way:

$$\frac{\boldsymbol{\xi}(k+1) - \boldsymbol{\xi}(k)}{T_0} = \mathbf{v_p}(k) + \boldsymbol{\Gamma}\left(\mathbf{p_d}(k) - \boldsymbol{\xi}(k)\right) \tag{69}$$

and considering Euler's derivation method, it follows that $\boldsymbol{\xi}(k+1) - \boldsymbol{\xi}(k) = T_0\dot{\boldsymbol{\xi}}(k)$ and the position error with respect to the closest point to the end-effector $\tilde{\boldsymbol{\xi}}(k) = \left(\mathbf{p_d}(k) - \boldsymbol{\xi}(k)\right)$, Eq. (69) reduces to:

$$\dot{\tilde{\boldsymbol{\xi}}}(k) = \mathbf{v_p}(k) + \boldsymbol{\Gamma}\left(\tilde{\boldsymbol{\xi}}(k)\right) \tag{70}$$

Now, the position vector of the end-effector $\boldsymbol{\xi}(k)$ according to Frenet's coordinates [31,32] has two components: $\boldsymbol{\xi}_{\mathrm{T}}(k)$ which is the tangential component (T) and $\boldsymbol{\xi}_{\mathrm{N}}(k)$ is the Normal component (N) (see Fig. 3). The velocity in the tangential component is $\dot{\boldsymbol{\xi}}_{\mathrm{T}}(k) = \mathbf{v_p}(k)$, which results the desired velocity tangential to the path; while the velocity in the normal component is:

$$\dot{\boldsymbol{\xi}}_{\mathrm{N}}(k) = \boldsymbol{\Gamma}\left(\tilde{\boldsymbol{\xi}}(k)\right) \tag{71}$$

which means a normal velocity to the tangent of the path at which the end-effector approaches the path. According to Frenet's coordinates the velocity $\dot{\boldsymbol{\xi}}_{\mathrm{N}}(k)$ has direction to the closest point if the end-effector $\mathbf{p_d}$ on the path. Then, when the end-effector moves in that direction, the error decreases. Therefore, the velocity of motion in that direction is the decreasing velocity of error written as $\dot{\boldsymbol{\xi}}_{\mathrm{N}}(k) = -\dot{\tilde{\boldsymbol{\xi}}}(k)$ Now, substituting in Eq. (71) this decreasing velocity results that:

$$\dot{\tilde{\boldsymbol{\xi}}}(k) + \boldsymbol{\Gamma}\left(\tilde{\boldsymbol{\xi}}(k)\right) = \mathbf{0} \tag{72}$$

Discretizing (72) for T_0 results $\dot{\tilde{\boldsymbol{\xi}}}(k) = \frac{\tilde{\boldsymbol{\xi}}(k+1)-\tilde{\boldsymbol{\xi}}(k)}{T_0}$ and rewriting we have:

$$\frac{\tilde{\boldsymbol{\xi}}(k+1) - \tilde{\boldsymbol{\xi}}(k)}{T_0} + \boldsymbol{\Gamma}\tilde{\boldsymbol{\xi}}(k) = \mathbf{0} \tag{73}$$

Then, taking the error $\tilde{\boldsymbol{\xi}}(k+1)$ results:

$$\tilde{\boldsymbol{\xi}}(k+1) = \left(\mathbf{I} - T_0\boldsymbol{\Gamma}\right)\tilde{\boldsymbol{\xi}}(k) \tag{74}$$

Analyzing the error for the ith value of Eq. (74) similar to (66) it results that $\tilde{\xi}_i(n+1) = \left(1-T_0\Gamma_{ii}\right)^n\tilde{\xi}_i(1)$. Then, if $n\to\infty$ the control error $\tilde{\xi}_i(n+1)$ will be equal to $\tilde{\xi}_i(\infty) = \left(1-T_0\Gamma_{ii}\right)^\infty\tilde{\xi}_i(1)$. Therefore, if $0 < T_0\Gamma_{ii} < 1$ the control error $\tilde{\xi}_i(\infty) = 0$ with $k\to\infty$. Performing an analysis similar to that of the control error $\tilde{\xi}_i(k)$ it is concluded that:

$$\lim_{k\to\infty}\tilde{\boldsymbol{\xi}}(k) = \mathbf{0} \in \mathbb{R}^3$$

with $0 < T_0\max\left(diag\left(\boldsymbol{\Gamma}\right)\right) < 1$ when $k\to\infty$ the ith control error of the aerial manipulator robot behaves as asymptotically stable, achieving the path-following control objective.

Remark 3. It is important to mention that the sampling period T_0 is given in seconds and is generally always less than 1 s in order to obtain better real-time data feedback.

Remark 4. In order to accomplish the objective of path-following control, the desired velocity tangent to the path is [19]:

$$\mathbf{v_p}(t) = \begin{bmatrix}\left\|\mathbf{v_p}\right\|\cos(\alpha)\cos(\beta)\\ \left\|\mathbf{v_p}\right\|\sin(\alpha)\cos(\beta)\\ \left\|\mathbf{v_p}\right\|\sin(\beta)\end{bmatrix} \tag{75}$$

where, $\left\|\mathbf{v_p}\right\|$ is the desired scalar velocity on the path at the closest point of the end-effector. While the α and β angles are determined by:

$$\alpha = \tan^{-1}\left(\frac{\frac{d\xi_{\mathrm{yd}}}{ds}}{\frac{d\xi_{\mathrm{xd}}}{ds}}\right) \quad \beta = \tan^{-1}\left(\frac{\frac{d\xi_{\mathrm{zd}}}{ds}}{\left\|\frac{d\xi_{\mathrm{xd}}}{ds}, \frac{d\xi_{\mathrm{yd}}}{ds}\right\|}\right) \tag{76}$$

6.2. Dynamic compensation

The cascade dynamic compensation controller receives the velocities generated by the kinematic controller, and generates new maneuvering velocities to the aerial manipulator robot. The desired velocity tracking of the robot is an application of $\boldsymbol{\mu_K}(k)$; while the control problem is to determine the maneuverability vector $\boldsymbol{\mu_D}(k)$ to accomplish the desired velocity of the robotic system. The velocity of the robotic system is given by the vector $\boldsymbol{\mu}(k)$. Therefore, if there is no perfect velocity tracking, the velocity error is defined as $\left(\tilde{\boldsymbol{\mu}}(k) = \boldsymbol{\mu_K}(k) - \boldsymbol{\mu}(k)\right)$. The proposed control law for the dynamic compensation is based on the discretization through Euler's method, therefore, the dynamic model of the robotic system obtained in (49) is:

$$\boldsymbol{\mu_{ref}}(k) = \mathbf{M}(\mathbf{q}(k))\left(\frac{\boldsymbol{\mu}(k) - \boldsymbol{\mu}(k-1)}{T_0}\right) + \mathbf{C}(\mathbf{q}(k), \boldsymbol{\mu}(k))\boldsymbol{\mu}(k) + \mathbf{g}(\mathbf{q}(k)) \tag{77}$$

Now, in order to determine the control action necessary to make the output go from its current value to the desired value, the following expression is considered:

$$\boldsymbol{\mu_D}(k) = \mathbf{M}(\mathbf{q}(k))\left(\frac{\boldsymbol{\mu_K}(k) + \boldsymbol{\Omega}\left(\boldsymbol{\mu_K}(k-1) - \boldsymbol{\mu}(k-1)\right) - \boldsymbol{\mu_K}(k-1)}{T_0}\right) + \cdots + \mathbf{C}(\mathbf{q}(k), \boldsymbol{\mu}(k))\boldsymbol{\mu}(k) + \mathbf{g}(\mathbf{q}(k)) \tag{78}$$

Here, $\boldsymbol{\Omega}(\cdot)$ is a diagonal matrix composed of sigmoid functions that weighs and saturates the control errors $\tilde{\boldsymbol{\mu}}(k-1) = \boldsymbol{\mu_K}(k-1) - \boldsymbol{\mu}(k-1)$ defined as $\boldsymbol{\Omega} = diag\left(\Omega_i/\left(\sigma_i + \left|\tilde{\mu}_i(k-1)\right|\right)\right)$, where, $\sigma_i \in \mathbb{R}^+$ defines the slope of the saturation function; $\Omega_i \in \mathbb{R}^+$ represents the maximum and minimum saturation values of the diagonal matrix $\boldsymbol{\Omega}(\cdot)$ weighing the ith component of the control error vector.

Table 2
Evolution of the control error $\tilde{\mu}_i(k)$ at time instant k.

k	$\tilde{\mu}_i(k)$	$(\mathbf{I}_{ii}-\boldsymbol{\Omega}_{ii})\,\tilde{\mu}_i(k-1)$
1	$\tilde{\mu}_i(1)$	$(\mathbf{I}_{ii}-\boldsymbol{\Omega}_{ii})\,\tilde{\mu}_i(0)$
2	$\tilde{\mu}_i(2)$	$(\mathbf{I}_{ii}-\boldsymbol{\Omega}_{ii})\,\tilde{\mu}_i(1) = (\mathbf{I}_{ii}-\boldsymbol{\Omega}_{ii})^2\,\tilde{\mu}_i(0)$
3	$\tilde{\mu}_i(3)$	$(\mathbf{I}_{ii}-\boldsymbol{\Omega}_{ii})\,\tilde{\mu}_i(2) = (\mathbf{I}_{ii}-\boldsymbol{\Omega}_{ii})^3\,\tilde{\mu}_i(0)$
⋮	⋮	⋮
n	$\tilde{\mu}_i(n)$	$(\mathbf{I}_{ii}-\boldsymbol{\Omega}_{ii})\,\tilde{\mu}_i(n-1) = (\mathbf{I}_{ii}-\boldsymbol{\Omega}_{ii})^n\,\tilde{\mu}_i(0)$

Table 3
Evolution of the velocity error vector $\tilde{\mu}(k)$ at time instant k.

k	$\tilde{\mu}(k)$	$(\mathbf{I}-\boldsymbol{\Omega})\,\tilde{\boldsymbol{\mu}}(k-1)+T_0\mathbf{M}(k)^{-1}\boldsymbol{\delta}_{\tilde{\mu}}(k)$
1	$\tilde{\mu}(1)$	$(\mathbf{I}-\boldsymbol{\Omega})\,\tilde{\boldsymbol{\mu}}(0)+T_0\mathbf{M}(1)^{-1}\boldsymbol{\delta}_{\tilde{\mu}}(1)$
2	$\tilde{\mu}(2)$	$(\mathbf{I}-\boldsymbol{\Omega})^2\,\tilde{\boldsymbol{\mu}}(0)+T_0\left((\mathbf{I}-\boldsymbol{\Omega})\,\mathbf{M}(1)^{-1}\boldsymbol{\delta}_{\tilde{\mu}}(1)+\mathbf{M}(2)^{-1}\boldsymbol{\delta}_{\tilde{\mu}}(2)\right)$
⋮	⋮	⋮
n	$\tilde{\mu}(n)$	$\underbrace{(\mathbf{I}-\boldsymbol{\Omega})^n\,\tilde{\boldsymbol{\mu}}(0)}_{\boldsymbol{\Delta}_{\tilde{\mu}}(n)}+\underbrace{T_o\sum_{i=1}^{n}(\mathbf{I}-\boldsymbol{\Omega})^{i-1}\mathbf{M}(n+1-i)^{-1}\boldsymbol{\delta}_{\tilde{\mu}}(n+1-i)}_{\boldsymbol{\Delta}_{\delta}(n)}$

6.2.1. Stability analysis

In order to demonstrate the stability of the proposed control law, it is considered that there are no modeling errors or environmental perturbations $\boldsymbol{\delta}_{\mu}(k)=\mathbf{0}$, in other words, $\boldsymbol{\mu}_{\mathbf{D}}(k)\equiv\boldsymbol{\mu}_{\mathbf{ref}}(k)$. Therefore, the closed-loop equation can be obtained by equating (78) with (77):

$$\boldsymbol{\mu}(k)-\boldsymbol{\mu}(k-1)=\boldsymbol{\mu}_{\mathbf{K}}(k)+\boldsymbol{\Omega}\left(\boldsymbol{\mu}_{\mathbf{K}}(k-1)-\boldsymbol{\mu}(k-1)\right)-\boldsymbol{\mu}_{\mathbf{K}}(k-1) \tag{79}$$

Considering that $\tilde{\boldsymbol{\mu}}(k-1)=\boldsymbol{\mu}_{\mathbf{K}}(k-1)-\boldsymbol{\mu}(k-1)$, it is possible to rewrite Eq. (79) as:

$$\tilde{\boldsymbol{\mu}}(k)=\left(\mathbf{I}_{7\mathrm{x}7}-\boldsymbol{\Omega}\right)\tilde{\boldsymbol{\mu}}(k-1) \tag{80}$$

Table 2 represents the evolution of the ith control error of the closed-loop equation (80).

If $n\to\infty$, the control error $\tilde{\mu}_i(n)$ in an instant of time will be $\tilde{\mu}_i(\infty)=(\mathbf{I}_{ii}-\Omega_{ii})^{\infty}\tilde{\mu}_i(0)$. Therefore, if $0<\Omega_{ii}<1$ the control error $\tilde{\mu}_i(\infty)=0$ with $k\to\infty$. Similar to the control error $\tilde{\mu}(k)$ it can be concluded that:

$$\lim_{k\to\infty}\tilde{\boldsymbol{\mu}}(k)=\mathbf{0}\in\mathbb{R}^n$$

with $0<\max(diag(\boldsymbol{\Omega}))<1$ when $k\to\infty$, in other words the ith control error of the aerial manipulator robot is asymptotically stable, thus satisfying the control objective of dynamic compensation.

6.2.2. Robustness analysis

For the robustness analysis it is considered that there are disturbances and/or modeling errors, therefore:

$$\boldsymbol{\mu}_{\mathbf{ref}}(k)=\boldsymbol{\mu}_{\mathbf{D}}(k)-\boldsymbol{\delta}_{\tilde{\mu}}(k)\in\mathbb{R}^n \tag{81}$$

where, $\boldsymbol{\delta}_{\tilde{\mathbf{u}}}(k)$ is the vector representing the perturbations and errors in the modeling. This vector is assumed to be bounded, that is $\left\|\boldsymbol{\delta}_{\tilde{\mu}}(k)\right\|\leq k_{\delta}\ \forall k$, where $k_{\delta}\in\mathbb{R}^+$. Therefore, in order to obtain the closed-loop equation, Eqs. (81) and (77) are combined:

$$\tilde{\mu}(k)=\left(\mathbf{I}_{7\mathrm{x}7}-\boldsymbol{\Omega}\right)\tilde{\boldsymbol{\mu}}(k-1)+T_0\mathbf{M}(k)^{-1}\boldsymbol{\delta}_{\tilde{\mu}}(k) \tag{82}$$

The evolution of the velocity error of Eq. (82) is analyzed in Table 3.

Now, the norm of the ith value of the velocity error $\tilde{\boldsymbol{\mu}}(n)$ can be written as:

$$\|\tilde{\boldsymbol{\mu}}(n)\|=\left\|\boldsymbol{\Delta}_{\tilde{\mu}}(n)+\boldsymbol{\Delta}_{\delta}(n)\right\|\leq\left\|\boldsymbol{\Delta}_{\tilde{\mu}}(n)\right\|+\left\|\boldsymbol{\Delta}_{\delta}(n)\right\| \tag{83}$$

where, $\left\|\boldsymbol{\Delta}_{\tilde{\mu}}(n)\right\|=\left\|(\mathbf{I}-\boldsymbol{\Omega})^n\tilde{\mu}(0)\right\|$ and $\left\|\boldsymbol{\Delta}_{\delta}(n)\right\|$ is given by.

$$\left\|\boldsymbol{\Delta}_{\delta}(n)\right\|=\leq T_o\sum_{i=1}^{n}k_{\Omega}^{i-1}\left\|(\mathbf{M}(n+1-i))^{-1}\delta_{\tilde{u}}(n+1-i)\right\| \tag{84}$$

where, $k_{\Omega}=\lambda_{\max}(\mathbf{I}-\boldsymbol{\Omega})$ and $\lambda_{\max}(\cdot)$ represents the maximum eigenvalue of a matrix. Taking into account the properties of the matrix $\mathbf{M}(\mathbf{q})\in\mathbb{R}^+$, the following constants are considered: $k_M=\sqrt{\lambda_{\max}\left(\mathbf{M}(n)^{-\intercal}\mathbf{M}(n)^{-1}\right)}\ \forall n$ and k_{δ} that represents the maximum singular value of $\mathbf{M}(n)^{-1}$ and the boundary value of the error $\delta_{\tilde{u}}(n)$, respectively. Rewriting (84) results:

$$\left\|\boldsymbol{\Delta}_{\boldsymbol{\delta}}(n)\right\|\leq T_o\sum_{i=1}^{n}k_{\Omega}^{i-1}k_Mk_{\delta} \tag{85}$$

Now, we obtain a geometric series [33], where, $a=k_Mk_{\delta}$ and $r=k_{\Omega}$. Thus, the series converges if and only if $|r|<1$, such that the terms decrease to close to zero in the limit. Indeed, as $0<\max(diag(\boldsymbol{\Omega}))<1$, then, $0<\lambda_{\max}(\mathbf{I}-\boldsymbol{\Omega})<1$ and consequently the series converges. The sum of the first n-terms of the geometric series is given by:

$$\sum_{i=1}^{n}k_{\Omega}^{i-1}k_Mk_{\delta}=k_Mk_{\delta}\left(1-k_{\Omega}{}^n\right)\left(1-k_{\Omega}\right)^{-1} \tag{86}$$

and the convergence of the series is:

$$\lim_{n\to\infty}\sum_{i=1}^{n}k_{\Omega}^{i-1}k_Mk_{\delta}=k_Mk_{\delta}\left(1-k_{\Omega}\right)^{-1} \tag{87}$$

Therefore, if $n\to\infty$ Eq. (85) results:

$$\left\|\lim_{n\to\infty}\boldsymbol{\Delta}_{\delta}(n)\right\|\leq T_0k_Mk_{\delta}\left(1-k_{\Omega}\right)^{-1} \tag{88}$$

Now, rewriting (81) and considering (86), the velocity error value at each sampling instant k is: $\|\tilde{\boldsymbol{\mu}}(k)\|\leq\left\|\boldsymbol{\Delta}_{\tilde{\mu}}(k)\right\|+T_0k_Mk_{\delta}\left(1-k_{\Omega}{}^k\right)\left(1-k_{\Omega}\right)^{-1}$.

Finally, the behavior of the velocity error when $k\to\infty$ is given by $\left\|\lim_{k\to\infty}\tilde{\boldsymbol{\mu}}(k)\right\|=\left\|\lim_{k\to\infty}\boldsymbol{\Delta}_{\tilde{\boldsymbol{\mu}}}(k)\right\|+\left\|\lim_{k\to\infty}\boldsymbol{\Delta}_{\delta}(k)\right\|$. First, let us calculate $\lim_{k\to\infty}\boldsymbol{\Delta}_{\tilde{\boldsymbol{\mu}}}(k)=\lim_{k\to\infty}(\mathbf{I}-\boldsymbol{\Omega})^k\tilde{\mu}(0)$. As $0<\max(diag(\boldsymbol{\Omega}))<1$, it results that $(\mathbf{I}-\boldsymbol{\Omega})^k\tilde{\mu}(0)\to0$ when $k\to\infty$. Therefore, $\left\|\boldsymbol{\Delta}_{\tilde{\mu}}(k)\right\|=\left\|(I-\boldsymbol{\Omega})^k\tilde{\mu}(0)\right\|\to0$. And taking into account (88), the velocity error $\tilde{\boldsymbol{\mu}}(k)$ when $k\to\infty$ results:

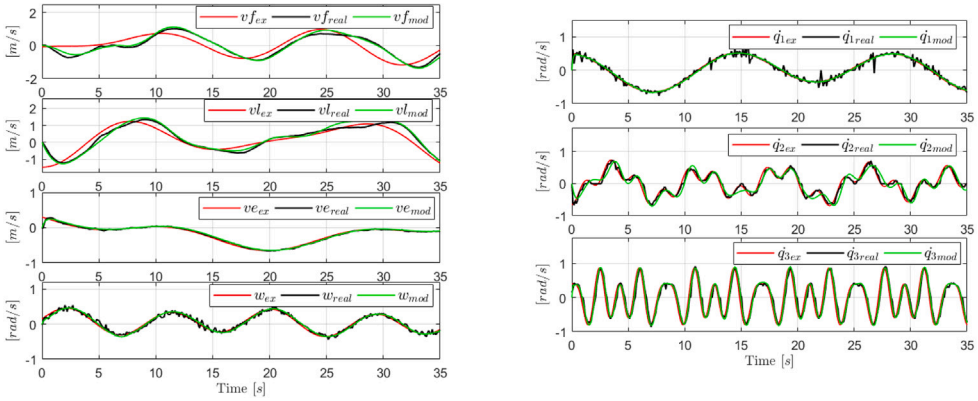
$$\|\tilde{\boldsymbol{\mu}}(k)\|\leq T_0k_Mk_{\delta}\left(1-k_{\Omega}\right)^{-1} \tag{89}$$

This means that, in presence of disturbances and/or modeling errors, the velocity error $\tilde{\mu}(k)$ is bounded, moreover, the maximum value of the velocity error can be estimated by Eq. (89) if k_M and k_{δ} are known. In case there were no disturbances or errors in the model, the error $\tilde{\mu}(k)\to\mathbf{0}$ when $k\to\infty$ and the system verifies to be asymptotically stable.

7. Experimental results

This section presents the experimental results of: (*i*) validation of the dynamic model of the aerial manipulator; (*ii*) experimental validation for position control; (*iii*) experimental validation for trajectory tracking; and (*iv*) experimental validation of the controller for path-following. Fig. 7 shows the robotic system used for the identification of the parameters of the dynamic model and for the experimental tests [19]. The aerial manipulator robot consists of: (*i*) DJI Matrice 600Pro hexarotor, it has a total weight including batteries of 9.5 kg and the maximum speed reached by the platform is 18 m/s without wind. The aerial platform through libraries (SDK) provides the global and local position data, global velocities, pitch, yaw and roll angles that are indispensable to feed back the control loop and calculate the position of the end-effector; (*ii*) A robotic arm of three degrees of freedom, the joints of the robotic arm are constituted by ROBOTIC DYNAMIXEL (MX-106) actuators. This type of motors provides the necessary data to feedback the control loop (position, angular velocity and angular acceleration); and (*iii*) An Intel computer (Intel® NUC 11 Enthusiast) which is in charge of commanding the system. It interconnects each part of the robot to send and receive information from the aerial

Fig. 7. Aerial manipulator robot used for the experimental tests.



(a) Hexarotor velocities. (b) Robotic arm velocities.

Fig. 8. Validation of the dynamic model, the subscript "*ex*" represents the excitation velocity to the robotic system; "*real*" represents the actual velocity produced by the robot; and "*mod*" is the velocity obtained from the dynamic model of the aerial manipulator.

platform and the robotic arm, in other words, it sends the control reference commands and obtains the feeding of the robot speeds and positions. The software that performs the numerical computation to execute the control algorithm is Matlab.

For the experiments the sampling period used is $T_0 = 0.1$ s, during which the robot is able to receive and send the position and velocity data to feedback the control loop. The experiments were carried out in a partially structured environment, where the environmental conditions (wind speed) at the time of the experimental tests can be seen at [34].

Remark 5. As the system is redundant, a secondary task is defined to avoid undesired configurations of the robotic arm during experiments and facilitate the task with the end-effector. The velocity vector in the control law $\boldsymbol{\mu_0}(k)$ that affects only the configuration of the robotic arm is defined by:

$$\boldsymbol{\mu_0}(k) = \mathbf{K}_2 \tanh\left(\begin{bmatrix}0 & 0 & 0 & 0 & \tilde{q}_1(k) & \tilde{q}_2(k) & \tilde{q}_3(k)\end{bmatrix}\right) \tag{90}$$

where, $\tilde{q}_i(k) = q_{id} - q_i(k)$ with $i = 1, 2, 3$ is the error of the robotic arm joints; q_{id} is the desired angular position value for each joint of the robotic arm; $\mathbf{K}_2 \in \mathbb{R}^{7\times7}$ is a positive diagonal matrix that weights the joint errors; and $\tanh(\cdot)$ is an odd function that saturates the joint errors. Therefore, the desired values for the joints q_{id} are defined in order to maximize the manipulability index $\mathrm{w} = \sqrt{\det(\mathbf{J}\mathbf{J}^{\intercal})}$ [35] as a secondary control objective.

7.1. Dynamic model validation

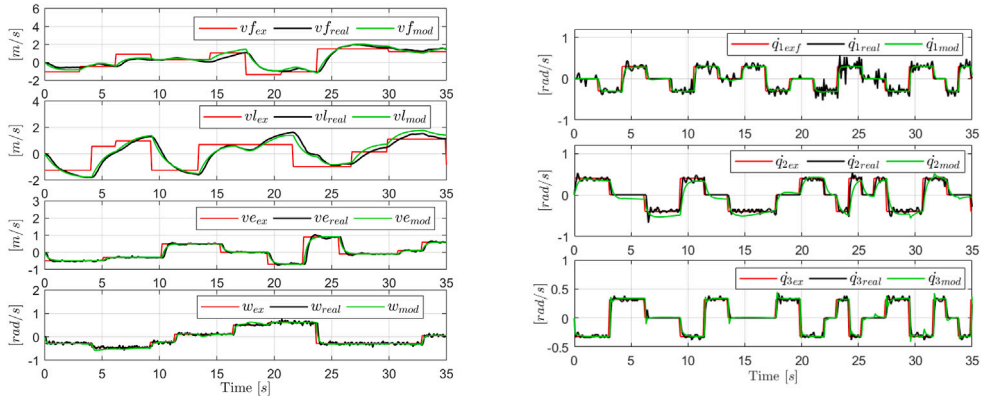
The validation of the dynamic model allows to observe that the identification is acceptable, comparing the real reference inputs injected into the robot, with the real velocities it provides and the velocities obtained from the dynamic model. Two reference velocity inputs are considered to observe how the dynamic model responds. Fig. 8 shows the behavior of the model with the first type of reference signals. These signals are different from the ones used for parameter identification.

Now, Fig. 9 shows the validation of the dynamic model with step-type reference inputs for the aerial platform and for the robotic arm joints. The similarity between the real and the model signals validates the identified dynamic model.

7.2. Position control experiment

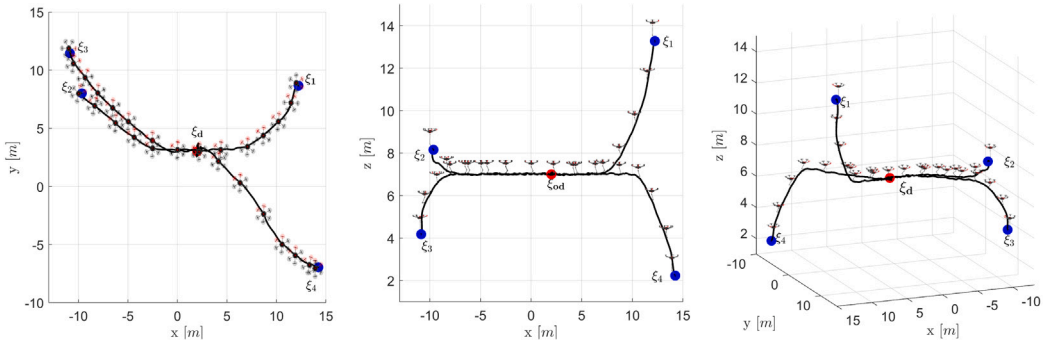
To validate the position control, multiple experiments are proposed. The experiments consist of having the end-effector of the aerial manipulator converge to a desired position, starting from various initial conditions. Table 4 shows the conditions established for the experiment, while Fig. 10 shows the behavior of the aerial manipulator in various performed experiments. The video recreated in Matlab with real data obtained during the experiment is available at [36].

Fig. 11(a) shows the evolution of the end-effector position in XYZ space starting at the initial condition of $\mathbf{q_4}(t_0)$, while the behavior of



(a) Hexarotor velocities. (b) Robotic arm velocities.

Fig. 9. Validation of the dynamic model with step-type reference inputs.



(a) View in the XY plane. (b) View in the XZ plane (c) View in the XYZ Space

Fig. 10. Movement of the overhead mobile manipulator to position the end-effector in the desired position.

Table 4
Position controller parameters.

Desired position $\xi_{\mathbf{d}}$	$\xi_{xd} = 2$, $\xi_{yd} = 3$, $\xi_{zd} = 2$ [m], $\xi_{od} = 0.5$ [rad]
Hexarotor and robotic arm initial conditions $\mathbf{q}(t_0) = \left[\mathbf{q_u}(t_0)^\intercal, \mathbf{q_a}(t_0)^\intercal\right]^\intercal$	$\mathbf{q}_1(t_0) = \left[12, 9, 14, 0, \frac{\pi}{4}, 0, \frac{\pi}{3}\right]^\intercal$ $\mathbf{q}_2(t_0) = \left[-10, 8, 9, \frac{\pi}{4}, -\frac{\pi}{4}, 0, \frac{\pi}{3}\right]^\intercal$, $\mathbf{q}_3(t_0) = \left[-10, 12, 5, -\frac{\pi}{6}, -\frac{\pi}{4}, 0, \frac{\pi}{3}\right]^\intercal$ and $\mathbf{q}_4(t_0) = \left[14, -7, 3, \frac{\pi}{3}, -\frac{\pi}{4}, 0, \frac{\pi}{3}\right]^\intercal$
Control gains	$\Gamma_1 = 0.5$, $\Gamma_2 = 0.5$, $\Gamma_3 = 0.1$, $\Gamma_4 = 0.1$, $\sigma_i = 1 \quad i = 1, 2, 3, 4$ and $\mathbf{K}_2 = diag\left([0, 0, 0, 0, 1.5, 1.75, 1.75]\right)$
Desired joint positions	$q_{1d} = 0$, $q_{2d} = -\frac{\pi}{3}$ and $q_{3d} = \frac{3}{4}\pi$ [rad]

Table 5
Trajectory tracking controller parameters.

Desired trajectory and orientation $\xi_{\mathbf{d}}(t)$	$\xi_{xd}(t) = 4\sin(0.35t) + 2$, $\xi_{yd}(t) = 8\sin(0.175t) - 1$, $\xi_{zd}(t) = 0.5\sin(0.525t) + 7$ and $\xi_{od}(t) = \arctan\left(\dot{\xi}_{yd}(t)/\dot{\xi}_{xd}(t)\right)$
Hexarotor and robotic arm initial conditions $\mathbf{q}(t_0)$	$x_u(t_0) = 4$, $y_u(t_0) = 2$, $z_u(t_0) = 9$ [m]. $\psi(t_0) = \frac{\pi}{3}$, $q_1(t_0) = -\frac{\pi}{6}$, $q_2(t_0) = 0$ and $q_3(t_0) = \frac{\pi}{2}$ [rad]
Control gains $\boldsymbol{\mu}_{\mathbf{K}}$	$\Gamma_1 = 0.8$, $\Gamma_2 = 0.8$, $\Gamma_3 = 1.5$, $\Gamma_4 = 2$, $\sigma_i = 1 \quad i = 1, 2, 3, 4$ and $\mathbf{K}_2 = diag\left([0, 0, 0, 0, 1.5, 1.75, 2.75]\right)$
Control gains $\boldsymbol{\mu}_{\mathbf{D}}$	$\Omega_1 = 0.85$, $\Omega_2 = 0.85$, $\Omega_3 = 0.8$, $\Omega_4 = 0.64$, $\Omega_5 = 0.8$, $\Omega_6 = 0.8$ and $\Omega_7 = 0.8$
Desired joint positions	$q_{1d} = 0$, $q_{2d} = -\frac{\pi}{3}$ and $q_{3d} = \frac{3}{2}\pi$ [rad]

the position control errors can be seen in Fig. 11(b). We can see how the errors converge close to zero when the end-effector is adjusted to the desired point.

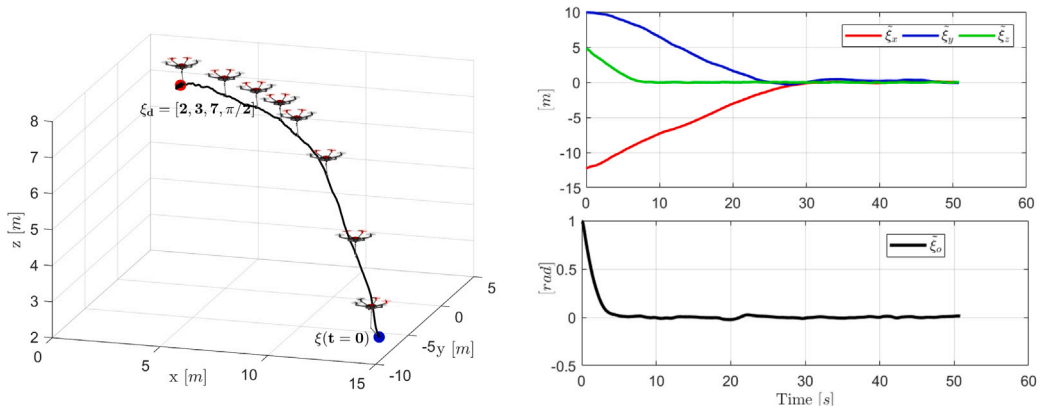
The evolution of the robotic arm joints is presented in Fig. 12. It can be seen how each joint converges to the desired values by the control action in the null space.

7.3. Trajectory tracking experiment

For the validation of the trajectory tracking controller with dynamic compensation, an experimental test is performed which consists of the end-effector following a trajectory without dynamics compensation (only the kinematic control action applied to the robot), and subsequently the cascaded dynamics compensation is activated. Table 5 shows the parameters established for the experiment, while Fig. 13 shows the evolution of the end-effector on the plane and in 3D space. It is evident the effect of the dynamic compensation in reducing the trajectory tracking error by seeing how the end-effector fits better to the desired trajectory (green line). The video recreated in Matlab with real data obtained during the experiment is available at [37].

The behavior of the control errors is shown in Fig. 14. As can be seen in the first part of the experiment, the control errors are higher with a Root Mean Square Error of RMSE = 0.636 m for only kinematic control. While from $t = 75$ s with the dynamic compensation the RMSE = 0.205 m, therefore, it is verified that the error is smaller and that the dynamic compensation improves the trajectory tracking of the end-effector.

The objective of the dynamic cascade compensation consists in reducing the velocity errors to zero so that the end-effector fits better on



(a) Evolution of the end-effector position during the experiment for $\mathbf{q_4}\,(t_0)$.

(b) End-effector position control errors and hexarotor orientation error for $\mathbf{q_4}\,(t_0)$.

Fig. 11. Robot movement in XYZ space and control errors of the experiment.

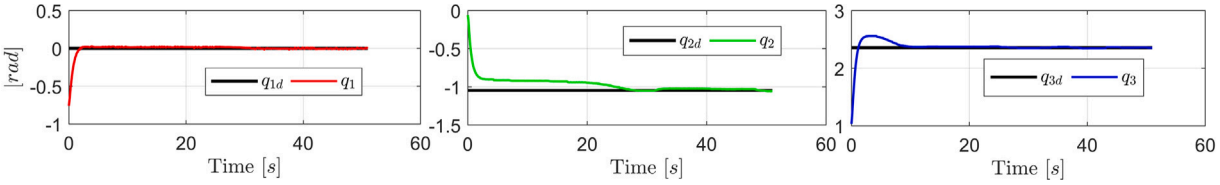
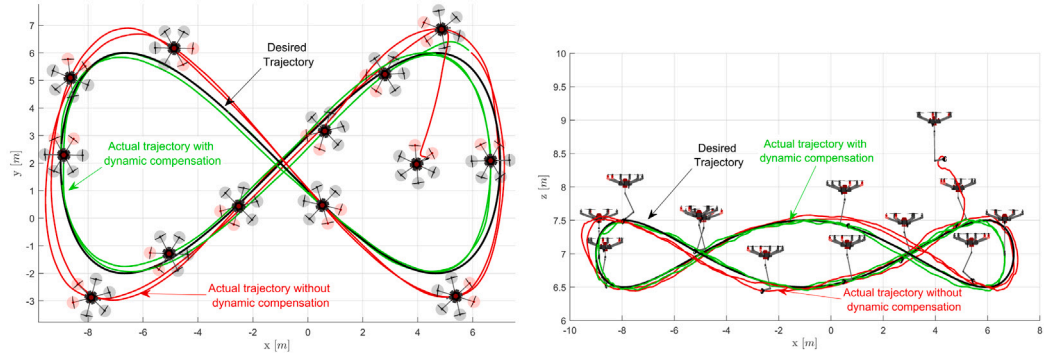


Fig. 12. Evolution of the joints of the robotic arm for the experiment with $\mathbf{q_4}\,(t_0)$.



(a) Motion executed as seen in the XY plane.

(b) Motion executed as seen in the XZ plane.

Fig. 13. Evolution of the end-effector during the trajectory tracking experiment. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

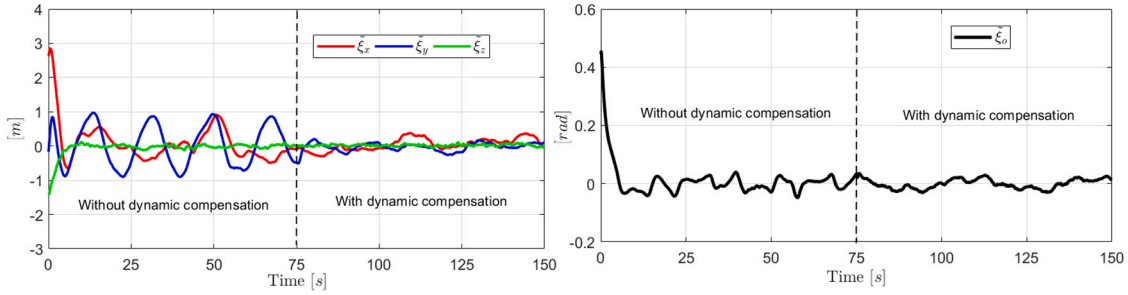


Fig. 14. Position error and orientation error of the trajectory tracking experiment.

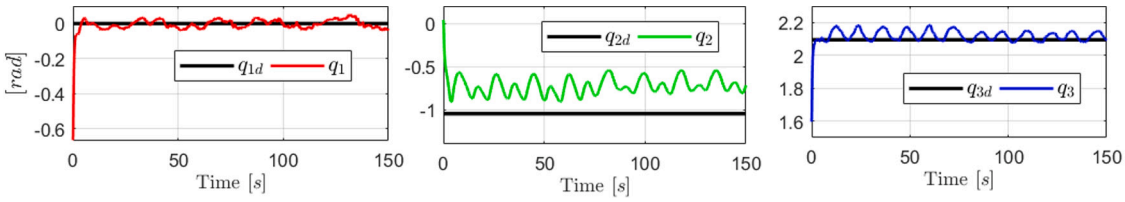
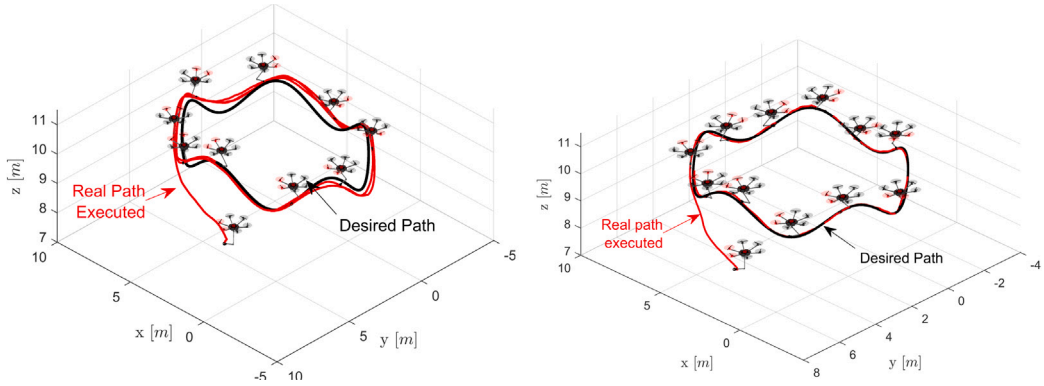


Fig. 15. Evolution of the joints of the robotic arm for trajectory tracking experiment.



(a) Experiment without dynamic compensation. (b) Experiment with dynamic compensation.

Fig. 16. Evolution of the end-effector on the desired path in the experiments executed.

Table 6
Path-following controller parameters.

Desired path $\mathbf{p}(s)$ and desired orientation for the end-effector	$\xi_{xd}(s) = 8\sin(12s) + 3$, $\xi_{yd}(s) = 6\cos(12s) + 12$, $\xi_{zd}(s) = 0.5\cos(60s) + 10$ and $\xi_{od}(s) = \arctan\left(\frac{d\xi_{yd}(s)}{ds} \Big/ \frac{d\xi_{xd}(s)}{ds}\right) \Big\\| \mathbf{v_p} \Big\\|$
Desired speed	$\\| \mathbf{v_p} \\| = 1.5$ [m/s]
Hexarotor and robotic arm initial conditions $\mathbf{q}(t_0)$	$x_u(t_0) = 3$, $y_u(t_0) = 2$, $z_u(t_0) = 8$ [m]. $\psi(t_0) = \frac{\pi}{3}$, $q_1(t_0) = -\frac{\pi}{9}$, $q_2(t_0) = 0$ and $q_3(t_0) = \frac{\pi}{2}$ [rad]
Control gains $\boldsymbol{\mu}_{\mathbf{K}}$	$\Gamma_1 = 1$, $\Gamma_2 = 1$, $\Gamma_3 = 1.5$, $\Gamma_4 = 1.8$, $\sigma_i = 1 \quad i = 1,2,3,4$ and $\mathbf{K}_2 = diag([0,0,0,0,1.5,2,4])$
Control gains $\boldsymbol{\mu}_{\mathbf{D}}$	$\Omega_1 = 0.85$, $\Omega_2 = 0.85$, $\Omega_3 = 0.8$, $\Omega_4 = 0.64$, $\Omega_5 = 0.8$, $\Omega_6 = 0.8$ and $\Omega_7 = 0.8$
Desired joint positions	$q_{1d} = 0$, $q_{2d} = -\frac{\pi}{3}$ and $q_{3d} = \frac{3}{2}\pi$ [rad]

the desired trajectory. Fig. 15 shows the velocity errors of the hexarotor and the robotic arm joints during the experiment.

7.4. Path-following experiment

The experiment to validate the path-following controller consists of defining a non-time parameterized desired path for the end-effector. Again, both experiments, with and without dynamic compensation have been done. Table 6 presents the parameters of the experiment. Fig. 16(a) shows the evolution of the end-effector during the path-following experiment without dynamic compensation and 16(b) shows the evolution of the end-effector during the path-following experiment with dynamic compensation. The videos recreated in Matlab with real data obtained during the experiments are available at [38,39].

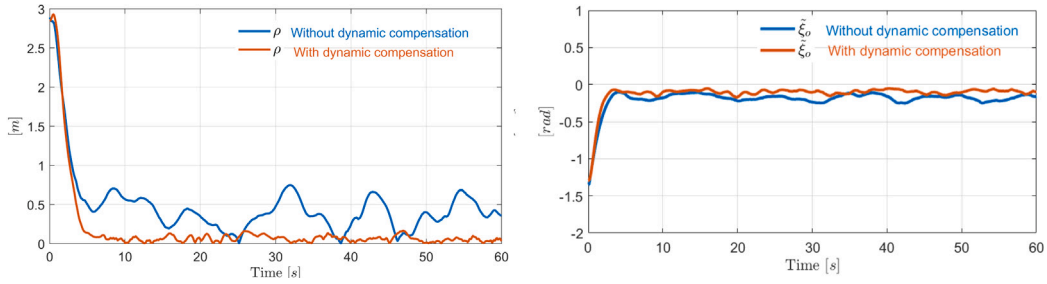
The $\mathbb{R}^2$ norm of the control error $\rho = \left\| \tilde{\xi}_x, \tilde{\xi}_y, \tilde{\xi}_z \right\|$ of the two path-following experiments are presented in Fig. 17(a), which are very close to zero. Small oscillations are observed which are produced by the environmental conditions during the experimentation, resulting a value of RMSE = 0.175 m for experiment with dynamic compensation. On the other hand, a value of RMSE = 0.4817 m was obtained for the case of the control scheme without dynamic compensation, *i.e.*, only path-following kinematic control. The orientation error of the path-following experiments are presented in Fig. 17(b), it is evident how the orientation error with dynamic compensation is smaller than the error without dynamic compensation.

Fig. 18(a) shows the behavior of the velocity errors of hexarotor and Fig. 18(b) shows the velocity errors of the robotic arm during the path-following experiments without dynamic compensation (blue) and with dynamic compensation (red). The velocity errors of the robot with dynamic compensation are smaller than the errors without dynamic compensation, therefore, it is verified that the dynamic compensation fulfills its objective.

In the position control, trajectory tracking and path-following experiments, small oscillations can be observed in the control errors produced by the wind in the test environment or by dynamics not taken into account in the model, therefore, control errors remain bounded as foreseen in Section 6.2.2.

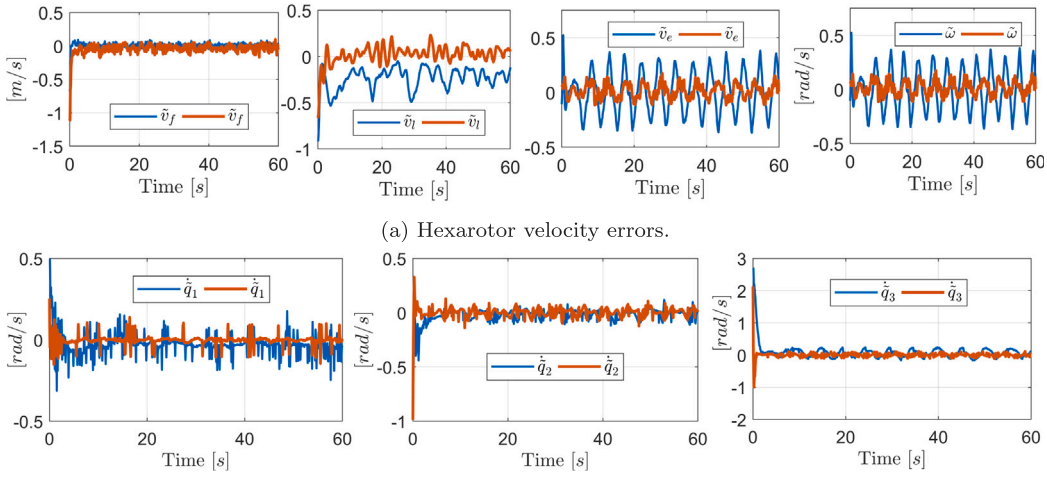
8. Conclusions

The proposed unified controller by numerical methods solves the three motion control problems of an aerial manipulator: position, trajectory tracking and path-following. Only with the suitable selection of the desired tracking velocity, the three tasks can be executed with the same control law algorithm. Furthermore, the dynamic model presented in this work has been defined in terms of reference velocities, which makes it easy to establish control structures to compensate for the system dynamics. For future work, adaptive dynamic compensation control is proposed, with the objective of transporting loads and tuning the parameters of the dynamic model according to the load handled by the robot.



(a) Control error norm ρ. (b) Control error $\tilde{\xi}_o$.

Fig. 17. Position error norm and orientation error produced in the path-following experiments.



(a) Hexarotor velocity errors.

(b) Robotic arm velocity errors.

Fig. 18. Velocity errors of the aerial manipulator during path-following experiments. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

CRediT authorship contribution statement

Christian P. Carvajal: Conceptualization, Formal analysis, Investigation, Software, Writing – original draft, Writing – review & editing, Validation, Data curation, Methodology. **Gabriela M. Andaluz:** Investigation, Conceptualization. **Víctor H. Andaluz:** Conceptualization, Investigation, Supervision, Writing – review & editing, Visualization, Funding acquisition, Validation. **Flavio Roberti:** Conceptualization, Investigation, Methodology, Supervision, Writing – review & editing, Validation. **Guillermo Palacios-Navarro:** Conceptualization, Supervision. **Ricardo Carelli:** Conceptualization, Investigation, Methodology, Supervision, Writing – review & editing, Validation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

The authors would like to thank the German Academic Exchange Service, also known by its German acronym (DAAD) for Scholarship Award in third Country Programme Latin America, the Universidad de las Fuerzas Armadas ESPE, Ecuador for the financing given to research, development, and innovation, through the Advanced Control of Unmanned Aerial Vehicles research project in Ecuador; the ARSI research group, and the Instituto de Automática of the Universidad Nacional de San Juan and CONICET in Argentina.

References

[1] J. Iqbal, R. Xu, S. Sun, C. Li, Simulation of an autonomous mobile robot for LiDAR-based in-field phenotyping and navigation, Robotics 9 (2) (2020) 46, http://dx.doi.org/10.3390/robotics9020046, Number: 2 Publisher: Multidisciplinary Digital Publishing Institute.

[2] M. Hussain, S.H.A. Naqvi, S.H. Khan, M. Farhan, An intelligent autonomous robotic system for precision farming, in: 2020 3rd International Conference on Intelligent Autonomous Systems (ICoIAS), 2020, pp. 133–139, http://dx.doi.org/10.1109/ICoIAS49312.2020.9081844.

[3] K. Choutri, L. Mohand, L. Dala, Design of search and rescue system using autonomous multi-UAVs, Intell. Decis. Technol. 14 (4) (2021) 553–564, http://dx.doi.org/10.3233/IDT-190138.

[4] J.P. Queralta, J. Taipalmaa, B. Can Pullinen, V.K. Sarker, T. Nguyen Gia, H. Tenhunen, M. Gabbouj, J. Raitoharju, T. Westerlund, Collaborative multi-robot search and rescue: Planning, coordination, perception, and active vision, IEEE Access 8 (2020) 191617–191643, http://dx.doi.org/10.1109/ACCESS.2020.3030190, Conference Name: IEEE Access.

[5] R. Gonzalez, K. Iagnemma, Slippage estimation and compensation for planetary exploration rovers. State of the art and future challenges, J. Field Robotics 35 (4) (2018) 564–577, http://dx.doi.org/10.1002/rob.21761.
[6] H.M. La, T.H. Dinh, N.H. Pham, Q.P. Ha, A.Q. Pham, Automated robotic monitoring and inspection of steel structures and bridges, Robotica 37 (5) (2019) 947–967, http://dx.doi.org/10.1017/S0263574717000601, Publisher: Cambridge University Press.
[7] D. Chaikalis, F. Khorrami, A. Tzes, Adaptive control approaches for an unmanned aerial manipulation system, in: 2020 International Conference on Unmanned Aircraft Systems (ICUAS), 2020, pp. 498–503, http://dx.doi.org/10.1109/ICUAS48674.2020.9213920, ISSN: 2575-7296.
[8] G. Zhang, Y. He, B. Dai, F. Gu, J. Han, G. Liu, Robust control of an aerial manipulator based on a variable inertia parameters model, IEEE Trans. Ind. Electron. 67 (11) (2020) 9515–9525, http://dx.doi.org/10.1109/TIE.2019.2956414, Conference Name: IEEE Transactions on Industrial Electronics.
[9] M. Jafarinasab, S. Sirouspour, E. Dyer, Model-based motion control of a robotic manipulator with a flying multirotor base, IEEE/ASME Trans. Mechatronics 24 (5) (2019) 2328–2340, http://dx.doi.org/10.1109/TMECH.2019.2936760, Conference Name: IEEE/ASME Transactions on Mechatronics.
[10] Y. Chen, W. Zhan, B. He, L. Lin, Z. Miao, X. Yuan, Y. Wang, Robust control for unmanned aerial manipulator under disturbances, IEEE Access 8 (2020) http://dx.doi.org/10.1109/ACCESS.2020.3008971, Conference Name: IEEE Access.
[11] G. Zhang, Y. He, B. Dai, F. Gu, L. Yang, J. Han, G. Liu, Aerial grasping of an object in the strong wind: Robust control of an aerial manipulator, Appl. Sci. 9 (11) (2019) 2230, http://dx.doi.org/10.3390/app9112230, Number: 11 Publisher: Multidisciplinary Digital Publishing Institute.
[12] G. Boucher, T. Laliberté, C. Gosselin, Mechanical design of a low-impedance 6 degree-of-freedom displacement sensor for intuitive physical human–robot interaction, J. Mech. Robot. 13 (2) (2021) http://dx.doi.org/10.1115/1.4049191.
[13] X. Meng, Y. He, F. Gu, Q. Li, J. Han, Dynamics modeling and simulation analysis for rotorcraft aerial manipulator system, in: 2017 36th Chinese Control Conference (CCC), 2017, pp. 1156–1161, http://dx.doi.org/10.23919/ChiCC.2017.8027504, ISSN: 1934-1768.
[14] E. Commission, AEROARMS, 2015, URL: https://aeroarms-project.eu/.
[15] E. Cataldi, F. Real, A. Suarez, P. Di Lillo, F. Pierri, G. Antonelli, F. Caccavale, G. Heredia, A. Ollero, Set-based inverse kinematics control of an anthropomorphic dual arm aerial manipulator, in: 2019 International Conference on Robotics and Automation (ICRA), 2019, pp. 2960–2966, http://dx.doi.org/10.1109/ICRA.2019.8793470, ISSN: 2577-087X.
[16] K. Morton, A. McFadyen, F. Gonzalez, Generalized trajectory control for tree-structured aerial manipulators, in: 2018 International Conference on Unmanned Aircraft Systems (ICUAS), 2018, pp. 947–956, http://dx.doi.org/10.1109/ICUAS.2018.8453446, ISSN: 2575-7296.
[17] K. Morton, A. McFadyen, L.F. Gonzalez Toro, Feasible polynomial trajectory planning for aerial manipulation, in: 2020 IEEE Aerospace Conference, 2020, pp. 1–9, http://dx.doi.org/10.1109/AERO47225.2020.9172501, ISSN: 1095-323X.
[18] O. Camacho, P. Leica, J. Antamba, J. Quinonez, Null-space based control applied to a formation of aerial manipulators in congested environment, in: 2019 International Conference on Information Systems and Computer Science (INCISCOS), IEEE, Quito, Ecuador, 2019, pp. 244–250, http://dx.doi.org/10.1109/INCISCOS49368.2019.00046.
[19] C.P. Carvajal, V.H. Andaluz, F. Roberti, R. Carelli, Path-following control for aerial manipulators robots with priority on energy saving, Control Eng. Pract. 131 (2023) 105401, http://dx.doi.org/10.1016/j.conengprac.2022.105401.
[20] K. Bodie, M. Tognon, R. Siegwart, Dynamic end effector tracking with an omnidirectional parallel aerial manipulator, IEEE Robot. Autom. Lett. 6 (4) (2021) 8165–8172, http://dx.doi.org/10.1109/LRA.2021.3101864.
[21] G.J.E. Scaglia, M.E. Serrano, S.A. Godoy, F. Rossomando, Linear algebra-based controller for trajectory tracking in mobile robots with additive uncertainties estimation, IMA J. Math. Control Inform. 37 (2) (2020) 607–624, http://dx.doi.org/10.1093/imamci/dnz016.
[22] P. Proano, L. Capito, A. Rosales, O. Camacho, Sliding mode control: Implementation like PID for trajectory-tracking for mobile robots, in: 2015 Asia-Pacific Conference on Computer Aided System Engineering, IEEE, Quito, 2015, pp. 220–225, http://dx.doi.org/10.1109/APCASE.2015.46.
[23] V. Andaluz, F. Roberti, J.M. Toibero, R. Carelli, Adaptive unified motion control of mobile manipulators, Control Eng. Pract. 20 (12) (2012) 1337–1352, http://dx.doi.org/10.1016/j.conengprac.2012.07.008.
[24] Z. Li, S.S. Ge, Fundamentals in Modeling and Control of Mobile Manipulators, Google-Books-ID: Sgu0DLV2HqsC, CRC Press, 2013.
[25] L. Sciavicco, B. Siciliano, Modelling and Control of Robot Manipulators, in: Advanced Textbooks in Control and Signal Processing, Springer, 2000, http://dx.doi.org/10.1007/978-1-4471-0449-0.
[26] Q. Quan, Introduction to multicopter design and control, 2017, http://dx.doi.org/10.1007/978-981-10-3382-7.
[27] M. Kaushik, Theoretical and experimental aerodynamics, 2019, http://dx.doi.org/10.1007/978-981-13-1678-4.
[28] J.D. Anderson, Fundamentals of Aerodynamics, sixth ed., in: McGraw-Hill Series in Aeronautical and Aerospace Engineering, McGraw Hill Education, New York, NY, 2017.
[29] D.F. Shanno, Conditioning of quasi-Newton methods for function minimization, Math. Comp. 24 (111) (1970) 647–656, http://dx.doi.org/10.1090/S0025-5718-1970-0274029-X.
[30] Y.F. Atchadé, J.S. Rosenthal, On adaptive Markov chain Monte Carlo algorithms, Bernoulli 11 (5) (2005) 815–828, Publisher: International Statistical Institute (ISI) and Bernoulli Society for Mathematical Statistics and Probability.
[31] K.D. Do, J. Pan, State- and output-feedback robust path-following controllers for underactuated ships using Serret–Frenet frame, Ocean Eng. 31 (5) (2004) 587–613, http://dx.doi.org/10.1016/j.oceaneng.2003.08.006.
[32] M.S. Underwood, K.-P. Marzlin, Fermi–frenet coordinates for spacelike curves, Internat. J. Modern Phys. A 25 (06) (2010) 1147–1154, http://dx.doi.org/10.1142/S0217751X10047841, Publisher: World Scientific Publishing Co..
[33] G. Arfken, H. Weber, F. Harris, Mathematical Methods for Physicists Elsevier, sixth ed., Boston, 2005, pp. 322–323.
[34] W. Spark, El tiempo histórico en Ambato en 16 de junio de 2021 (Ecuador) - weather spark, 2021, URL: https://es.weatherspark.com/h/d/20027/2021/6/16/Tiempo-histC3B3rico-elmiC3A9rcoles-16-de-junio-de-2021-en-Ambato-Ecuador#Figures-WindSpeed.
[35] T. Yoshikawa, Manipulabilidad de mecanismos robóticos, Int. J. Robot. Res. 4 (2) (1985) 3–9, http://dx.doi.org/10.1177/027836498500400201, Publisher: SAGE Publications Ltd STM.
[36] C. Carvajal, Position control (aerial mobile manipulator), 2022, URL: https://youtu.be/vc94n0Z9ik0.
[37] C. Carvajal, Trajectory tracking control (aerial mobile manipulator), 2022, URL: https://youtu.be/gjhfo3iJpfA.
[38] C. Carvajal, Path following control of an aerial manipulator without dynamic compensation, 2022, URL: https://youtu.be/hEXQcfkgtWI.
[39] C. Carvajal, Path following control of an aerial manipulator with dynamic compensation, 2022, URL: https://youtu.be/8n2QSbf1lmc.

Christian P. Carvajal was born in Ambato, Ecuador. He received the degree in Electronics and Instrumentation Engineer from the University of the Armed Forces ESPE in 2015, Quito-Ecuador. He is Scholarship holder of the German Institute for German Academic Exchange, DAAD, to obtain a Ph.D. in Control Systems Engineering. He is currently Ph.D. candidate with the topic "Coordinated Control of Aerial Manipulators" at the Institute of Automatics of the National University of San Juan, Argentina. He has participated in research projects and he have publications indexed in the SCOPUS database. His research interests are Control Systems, Robotic, Cooperative Control and Virtual Reality.

Gabriela M. Andaluz received the engineering degree in Electronics and Control from Escuela Politécnica Nacional, Quito, Ecuador, in 2011, and the Master's degree in Control Systems and Industrial Automation from the Escuela Superior Politécnica del Chimborazo, Riobamba, Ecuador, in 2018. She is student of the Ph.D. in Electronic Engineering at the Universtity of Zaragoza, Spain. She is currently a full-time Occasional Professor at the Universidad Internacional del Ecuador. Her areas of research interest are Autonomous Control and Tele-Operated applied Mobile Robots, Intelligent Control, Automation.

Víctor Hugo Andaluz Ortiz is an Electronics and Control Engineer from the National Polytechnic School; he was a scholarship holder of the German Institute for Academic Exchange, DAAD, obtaining his Ph.D. in Control Systems Engineering, after studying at the Institute of Automatics of the National University of San Juan, Argentina and at the Institute of Real-Time Systems of the Leibniz University of Hannover-Germany. Currently, he is a professor at the University of the Armed Forces ESPE. His areas of interest are: Robotics, automation, control systems, virtual and augmented reality.

Flavio Roberti was born in Buenos Aires, Argentina. He graduated in Electronic Engineering from the National University of San Juan, Argentina in 2003; on March, 2009 he obtained the Ph.D. degree in Control Systems Engineering at the Instituto de Automática, National University of San Juan. He is currently Full Professor at the National University of San Juan and Researcher of the National Council for Scientific and Technical Research (CONICET). His research interests are on Robotics, Visual servoing and Human–robot interaction.

Guillermo Palacios-Navarro received a B.S, M.S. ('96) and PhD. ('09) degrees in Telecommunication Engineering from University of Zaragoza – Spain. Following experience as a software engineer in several firms, he joined Teruel Polytechnic School of Engineering in 1998. Since then, he has been with the department of Electronic Engineering and Communications, University of Zaragoza, where he is currently an Associate Professor. Dr. Palacios was a visiting faculty in the Newman Laboratory for Biomechanics and Human Rehabilitation at the Massachusetts Institute of Technology (MIT). He has authored more than one hundred articles in scientific journals and international conferences and has also served as guest editor in several special issues in indexed journals. Dr. Palacios' research interests include signal and image processing, communication systems and virtual reality systems.

Ricardo Carelli was born in San Juan, Argentina. He received the graduate degree in electromechanical engineering from the National University of San Juan, San Juan, Argentina, in 1977, and the Ph.D degree in electrical engineering from the National University of Mexico, Mexico City, Mexico, in 1989. He is currently an Emeritus Professor at the National University of San Juan, Argentina, and a Senior Researcher under contract with the National Council for Scientific an Technical Research (CONICET), Argentina. His research interests include robotics, manufacturing systems, adaptive control, and artificial intelligence applied to automatic control. Prof. Carelli is a member of the Argentine Association of Automatic Control–International Federation of Automatic Control.