

## Article

# LAMDA Controller Applied to the Trajectory Tracking of an Aerial Manipulator

Gabriela M. Andaluz<sup>1,2,\*</sup>, Luis Morales<sup>3</sup>, Paulo Leica<sup>3</sup>, Víctor H. Andaluz<sup>4</sup>  and Guillermo Palacios-Navarro<sup>2,\*</sup> 

- <sup>1</sup> Escuela de Ingeniería Mecatrónica, Universidad Internacional del Ecuador, Quito 170411, Ecuador  
<sup>2</sup> Department of Electronic Engineering and Communications, University of Zaragoza, 44003 Teruel, Spain  
<sup>3</sup> Departamento de Automatización y Control, Escuela Politécnica Nacional, Quito 170517, Ecuador; luis.moralesec@epn.edu.ec (L.M.); paulo.leica@epn.edu.ec (P.L.)  
<sup>4</sup> Departamento de Eléctrica y Electrónica, Universidad de las Fuerzas Armadas ESPE, Sangolquí 171103, Ecuador; vhandaluz1@espe.edu.ec  
\* Correspondence: gaandaluzor@uide.edu.ec (G.M.A.); guillermo.palacios@unizar.es (G.P.-N.)

**Abstract:** In this work, a novel LAMDA (Learning Algorithm for Multivariable Data Analysis) control strategy for trajectory tracking for an aerial manipulator is presented. Four control strategies are developed: Kinematic, Inverse Dynamics, Sliding Mode (SMC), and LAMDA. These are compared with each other in order to verify their performance to fulfill the control objective. Experimental tests were also carried out to validate the developed controllers. In addition, a study of stability has been also performed for all the controllers. The results obtained by the LAMDA controller demonstrated the good performance of the controller in the aerial manipulator robot. To the best of our knowledge, this is the first time a LAMDA controller has been applied to an aerial robotic manipulator.



**Citation:** Andaluz, G.M.; Morales, L.; Leica, P.; Andaluz, V.H.; Palacios-Navarro, G. LAMDA Controller Applied to the Trajectory Tracking of an Aerial Manipulator. *Appl. Sci.* **2021**, *11*, 5885. <https://doi.org/10.3390/app11135885>

Academic Editor: Ioannis Kostavelis

Received: 14 May 2021  
Accepted: 22 June 2021  
Published: 24 June 2021

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

**Keywords:** aerial manipulator; inverse dynamics; kinematic; LAMDA; trajectory tracking; SMC; fuzzy; multivariable

## 1. Introduction

At present, the characteristics of the aerial manipulator have been used in multiple applications, because the mobility of the hexacopter and the maneuverability of the robotic arm make it very useful [1–3]. The first works are oriented to the development of applications considering only the kinematics of the aerial manipulator, i.e., coriolis forces, mass inertia, etc., are not considered. In this context, several research studies on aerial manipulators have been presented, for example, an algorithm for the control of a contact inspection for an unmanned aerial vehicle (UAV) manipulator [4], or a trajectory-tracking control algorithm for the parallel aerial manipulator based on the Stewart platform [5]. Guayasamín et al. [6] designed a trajectory tracking control based on the Lyapunov and Sliding Mode Control (SMC) theory applied to a Kinematic model. In the same line, predictive control for trajectory tracking has been included for a two-armed aerial manipulator [7]. In the work of Yarai et al. [8], the dynamics of the robotic arm is taken as an external perturbation and a decoupling controller is proposed to guarantee that the dynamics produced by the robotic arm to the quadrotor (and vice versa) are not part of the quadrotor dynamics. Wang et al. [9] designed a Sliding Mode controller to track the trajectory of the spatial manipulator, and a linear quadratic regulator (LQR) controller of the subsystem was designed to suppress the vibration of the flexible components. A null-space-based controller has been defined for a group of aerial manipulators [10].

The previous studies only contemplate the kinematics of the aerial manipulator. However, in real applications, it is necessary to consider the mass, inertia, friction forces, perturbations, etc. These characteristics require other types of controllers to compensate the dynamics of the system. Thus, several works have been developed to solve this problem, like the work of Gkoutas and Tzes [11] who presented the modeling and control of

unmanned aerial manipulators (UAM) in a leader–follower configuration that performed a cooperative manipulation task taking into account the dynamics of the system. In the study of Morton et al. [12] the dynamics were compensated with an inverse dynamics controller for trajectory tracking.

Different control algorithms have been developed to improve the performance of aerial manipulators robots. For example, a controller based on neural networks was implemented for trajectory tracking with the dynamic model [13]. In the same context, a controller based on null space to coordinate a group of robots was presented in [14]. An adaptive position control was implemented by Liu and Huang [15] aimed at compensating for certain perturbations in the modeling of the aerial manipulator, whereas a feedback control strategy was presented by Naldi et al. [16] allowing both the position and orientation of the end-effector to follow a desired trajectory considering the manipulator dynamics. The abovementioned research considers the control of an aerial manipulator from the point of view of its dynamics. In order to carry this out, the search for new controllers that can include the dynamics of aerial manipulators robots, meet the control objectives, and improve the performance of the task is vital for researchers.

Artificial intelligence (AI) approaches help to improve the behavior of the controllers. Among others, the fuzzy logic allows obtaining good results when the plant model is inaccurate or is not known in detail, which usually arises in the field of robotics. One of these methods is LAMDA (Learning Algorithm for Multivariable Data Analysis), which has been designed originally for classification and clustering tasks [17–20]. For example, a LAMDA fuzzy classification technique has been implemented to interpret the behavior of a drinking water treatment plant [21]. Recently, the use of the LAMDA features (in the detection of functional states of systems) has been proposed to make it work as a controller. For example, a class-based LAMDA controller was proposed and validated in the regulation of humidity and temperature of a complex Heating, Ventilation, and Air-Conditioning (HVAC) system in [22], obtaining excellent performance when considering that the mathematical model of the plant was variable or inexact. In [23], an adaptive LAMDA has been proposed and formalized for the control and modeling of systems, modifying the LAMDA structure with the addition of layers operating similar to neural networks, but with the advantage of having a fixed number of layers whose calibration is not trivial. This controller has proven to be adequate where the system dynamics are uncertain and complex. Finally, previous work [24] focuses on the formalization of a LAMDA algorithm for control based on the fundamentals of the Lyapunov theory and the Sliding-Mode Control (SMC), to guarantee stability and robustness of the overall system. The method is called LAMDA-SMC (LSMC) and takes advantage of LAMDA features to design a chattering-free controller. LSMC has been tested in the field of control of chemical processes, demonstrating that it is stable in the control of systems with model uncertainties and under disturbances. The results obtained by this LAMDA controller have been good in non-robotic aerial applications.

- The originality of this article lies in the validation of the robust LAMDA controller in aerial robotic systems whose dynamics are partially known or inaccurate. The main features of the proposed method are the following:
- The controller is based on concepts derived from LAMDA theory: Classes or functional states have a number of fixed layers (intrinsic feature). Therefore, the design is more straightforward than methods where the number of internal layers must be calibrated.
- The proposed controller uses the class criteria, which allows a quick convergence on the desired output without requiring a learning method that increases computational time.
- The design of a controller where the continuous and discontinuous control actions of SMC schemes are computed using the LAMDA method.
- A comparative analysis between the LAMDA method that does not use the aerial manipulator robot model is carried out with other well-known controller, which

depend on model. Thus, obtaining a good performance of the controller without considering of the dynamics system.

Based on the abovementioned features, the contribution of this article is the design of an intelligent controller based on the LAMDA fuzzy algorithm for the trajectory tracking of an aerial manipulator robot and validating the approach in this field of study. Four control algorithms will be designed: (i) Kinematic, (ii) Inverse Dynamics, (iii) SMC Sliding Mode, and (iv) LAMDA. Finally, experimental tests will be carried out for the four controllers on a real aerial manipulator robot with three degrees of freedom for trajectory-tracking purposes.

The paper is organized as follows: Section 2 presents the kinematic and dynamic robot model, whereas Section 3 describes the design of the four controllers as well as their stability. Experimental tests and analysis of the results are given in Section 4. Finally, discussion and main conclusions are presented in Section 5.

## 2. Aerial Manipulator Robot Model

The aerial manipulator robot consists of a robotic arm on a hexacopter. These robots can perform the most common missions of robotic systems that require both navigation and manipulation capabilities. In the current work, an aerial manipulator robot consisting of a hexacopter and a robotic arm is considered.

### 2.1. Kinematic Model

The aerial manipulator robot consists of a robotic arm located on hexacopter, as shown in Figure 1. In the configuration of the aerial manipulator robot the position and orientation of all its points with respect to an inertial reference system  $\mathcal{R}(\mathcal{X}, \mathcal{Y}, \mathcal{Z})$  is known. The configuration of the aerial manipulator robot is defined by a vector  $q(t)$  of  $n$  independent coordinates, which are called the generalized coordinates of the aerial manipulator robot, defined as  $q(t) = [q_a \ q_Q]$  where  $q_a$  are the velocities of hexacopter, and  $q_Q$  are velocities of the robotic arm. The configuration of  $q(t)$  is an element of the configuration space of the aerial manipulator robot represented by  $\mathcal{N}$ . The position of the aerial manipulator robot end-effector  $\zeta(t) = [\zeta_1 \ \zeta_2 \ \dots \ \zeta_m]$  is defined by an  $m$ -dimensional vector representing the orientation and position of the robot's point of interest with respect to  $\{\mathcal{R}\}$ . The position of the end-effector with respect to  $\{\mathcal{R}\}$  defines the operating space of the aerial manipulator robot, denoted by  $\mathcal{M}$ .

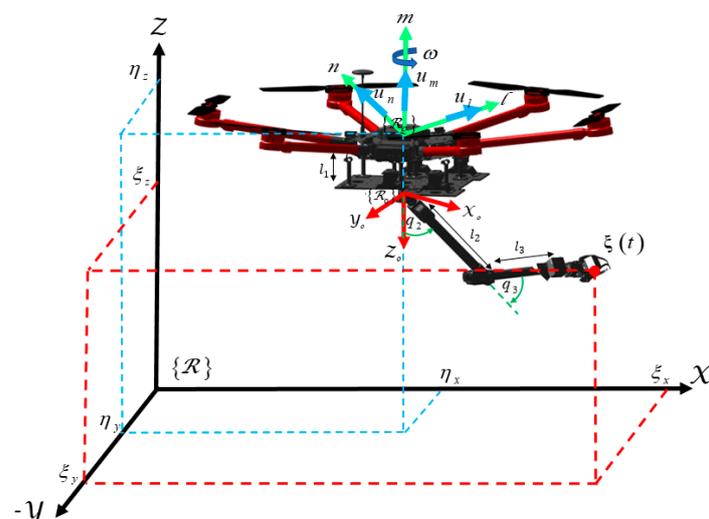


Figure 1. Aerial manipulator robot schematic.

The direct kinematics of the aerial manipulator robot defines the position and orientation of the end-effector  $\zeta(t)$  as a function of the configuration of the robotic arm and the

hexacopter (or its operational coordinates as functions of the generalized coordinates of the robotic arm and the operational coordinates of the hexacopter).

$$f : \mathcal{N}_a \times \mathcal{N}_Q \rightarrow \mathcal{M} \tag{1}$$

$$(q_a(t), q_Q(t)) \rightarrow \zeta(t) = f(q_a, q_Q) \tag{2}$$

The kinematic model of the aerial manipulator robot sets the derivative of the end-effector position as a function of the derivative of the configuration of the robotic arm and the hexacopter (or their operational velocities as functions of their generalized velocities).

$$\dot{\zeta}(t) = \frac{\partial(q_a, q_Q)}{\partial q} \mu(t) \tag{3}$$

where  $\dot{\zeta}(t) = [\dot{\zeta}_1 \ \dot{\zeta}_2 \ \dots \ \dot{\zeta}_m] \in R^m$  represents the velocity vector of the end-effector with respect to the inertial reference system  $\{\mathcal{R}\}$ . Let  $\mu(t) = [\dot{q}_a \ \dot{u}]$  be the maneuverability vector of the aerial manipulator robot. Considering  $J(q) = \frac{\partial f}{\partial q} T(q)$  in the previous equation, one has:

$$\dot{\zeta}(t) = J(q)\mu(t) \tag{4}$$

where the Jacobian matrix  $J(q_a, q_Q)$  defines the linear mapping between the aerial manipulator robot maneuverability vector  $\mu(t)$  and the end-effector velocity vector  $\dot{\zeta}(t)$ .  $T(q)$  is the transformation matrix relating the velocities of  $\dot{q}(t)$  and the velocities of the aerial manipulator robot  $\mu(t)$ , in such a way that:  $\dot{q}(t) = T(q)\mu(t)$ .

It is essential to emphasize that the dimension of the operational space  $m$  is smaller than the aerial manipulator robot degree of freedom  $n$  ( $m < n$ ), so the aerial manipulator robot is called a redundant system.

### 2.2. Dynamic Model

The dynamics of a system, as reviewed in the literature, is an important contribution in the implementation of control algorithms. Therefore, in this section, we present the dynamics of the aerial manipulator robot, by applying the Euler–Lagrange analysis, where

$$L = E_c - E_p \tag{5}$$

where  $L$  is the Lagrangean,  $E_c$  is the kinetic energy, and  $E_p$  is the potential energy, and where  $E_c$  y  $E_p$  are functions of the masses of the links of robotic arm, mass of the hexacopter, lengths and dimensions, angular positions, and velocities of the aerial manipulator robot. It is known that the Lagrange equation is given by:

$$f = \frac{d}{dt} \left( \frac{dL}{dq(t)} \right) - \frac{dL}{dq(t)} \tag{6}$$

By developing Equation (6), we obtain  $f$ , representing the equation of the dynamics of the aerial manipulator robot, shown in Equation (7).

$$f = M(q(t))\ddot{q}(t) + C(q(t), \dot{q}(t))\dot{q}(t) + G(q(t)) \tag{7}$$

$M(q(t))$  is the inertia matrix;  $C(q(t), \dot{q}(t))$  is the matrix of centrifugal and centripetal forces;  $G(q(t))$  is the gravity vector; and  $q(t)$  is a generalized coordinate of the aerial manipulator robot. Finally, Equation (7) can be expressed as a function of the velocity vector  $\mu(t)$  and we can obtain the reference velocity equation  $\mu_{ref}(t)$  representing the dynamic model of the aerial manipulator robot, defined as:

$$\mu_{ref}(t) = M(q(t))\dot{\mu}(t) + C(q(t), \dot{q}(t))\mu(t) + G(q(t)) \tag{8}$$

### 3. Proposed Controllers

The problem of motion control of the aerial manipulator robot is solved with the control scheme proposed in Figure 2, which solves the problem of trajectory tracking control in an autonomous way.

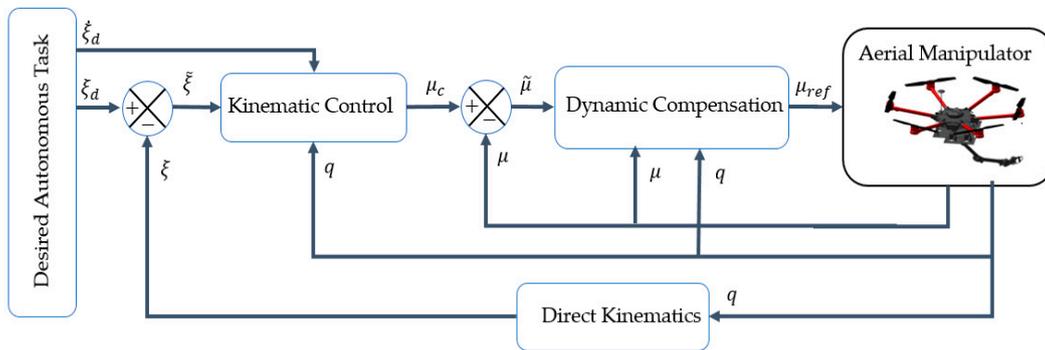


Figure 2. Proposed controller scheme.

Two cascaded subsystems are considered for the controller design:

**Kinematic control:** A minimum-norm control with saturation at maneuverability velocities is considered, for which  $\xi_d(t)$  and  $\dot{\xi}_d(t)$  describe the desired position and velocity, respectively, of the operating end of the aerial manipulator robot. The control error is defined by  $\zeta(t) = \xi_d(t) - \xi(t)$ . Therefore, the control target is defined as

$$\lim_{t \rightarrow \infty} \zeta(t) = 0 \in R^m \tag{9}$$

**Dynamic Compensation:** The main objective is to compensate the dynamics of the aerial manipulator robot, thus reducing the velocity tracking error. For this purpose, this work considers the implementation of four control techniques: (i) Kinematic, (ii) Inverse Dynamics; (iii) SMC Sliding Mode; and (iv) LAMDA, in order to evaluate the behavior of the controllers. The implemented controllers receive the desired maneuverability velocities  $\mu_c(t)$  as inputs calculated by the kinematic controller and generates velocity references  $\mu_{ref}(t)$  or the motion control of the aerial manipulator robot. The velocity control error is defined as  $\tilde{\mu}(t) = \mu_c(t) - \mu(t)$ . Therefore, the control objective is to ensure that,

$$\lim_{t \rightarrow \infty} \tilde{\mu}(t) = 0 \in R^n \tag{10}$$

#### 3.1. Kinematic Controller

The kinematic controller receives the desired position and velocity of the aerial manipulator robot end-effector  $\xi_d(t) \in R^m$  and  $\dot{\xi}_d(t) \in R^m$ , respectively, and generates the maneuverability velocities of the aerial manipulator  $\mu_{ref}(t) \in R^n$ . In other words, the desired operational motion of the robot is an application of  $(\xi_d(t) | t \in [t_0, t_f])$ . Therefore, the motion control problem is to determine the maneuverability vector  $(\mu_{ref}(t) | t \in [t_0, t_f])$  to achieve the desired operational motion of the robot. The corresponding evolution of the entire aerial manipulator robot is given by the generalized coordinate vector of the aerial manipulator robot  $(q(t) | t \in [t_0, t_f])$ .

The proposed controller is based on a minimum norm solution, which means that, at any given time, the aerial manipulator robot will achieve its navigation and manipulation goals with as few movements as possible. In addition, the redundancy of the aerial

manipulator robot is used to achieve secondary control objectives, i.e., the control of the internal configuration of the robotic arm. Therefore, the following control law is proposed,

$$\mu_c(t) = J(q(t))^\# \left( \xi_d^{-1}(t) + W \tanh(\tilde{\xi}(t)) \right) + \left( I_{n \times n} - J(q(t))^\# J(q(t)) \right) \mu_0(t) \quad (11)$$

where  $J(q(t))^\# = J(q(t))^T \left( J(q(t))J(q(t))^T \right)^{-1} \in R^{n \times m}$  represents the pseudoinverse matrix;  $W \in R^{n \times m}$  is a positive diagonal matrix weighing the control errors  $\tilde{\xi}(t)$ . In the proposed control law (11), the first term on the right describes the main task of the end-effector, which minimizes the term  $\frac{1}{2} \|\mu(t)\|_2^2$ . The second term defines the motion of the aerial manipulator robot configuration in which the matrix  $\left( I_{n \times n} - J(q(t))^\# J(q(t)) \right)$  projects an arbitrary vector  $\mu_0(t)$  onto the null space of the Jacobian matrix  $J(q(t))$  such that the secondary control objectives do not affect the main task of the end-effector. Therefore, any value given to  $\mu_0(t)$  will affect only the internal structure of the aerial manipulator robot, but not the final control of the end-effector. The proposed control law (12) considers an analytical saturation of the maneuverability velocities of the aerial manipulator robot. Therefore, the function tanh (hyperbolic tangent) bounds the control errors between  $[-1, 1]$ . The redundancy of the aerial manipulator robot can be effectively used for the achievement of additional performances such as avoiding obstacles in the workspace and singular configurations, or to optimize various motion criteria. In this work, it is considered to control the internal configuration of the robotic arm so that the center of mass of the aerial manipulator robot does not affect the motion of the robotic system. Therefore,  $\mu_0(t)$  is defined as,

$$\mu_0(t) = K \tanh(q_d(t) - q(t)) \quad (12)$$

where  $q_d$  is the desired joint positions of the robotic arm, considering that the center of gravity is located at the center of the aerial manipulator robot, and  $K \in R^{n \times n}$  is a diagonal matrix weighing the errors of the joint positions of the robotic arm, defined as  $\tilde{q}(t) = q_d(t) - q(t)$ .

### 3.2. Inverse Dynamics Controller

The inverse dynamics controller is applied to systems with known dynamics, for which it is essential to obtain the system model and its dynamics be as close as possible to the real system so that it can be controlled. In addition, it is important to mention that it is not a robust controller because it depends directly on the changes that may exist in the dynamics of the system. The proposed controller scheme is shown in Figure 3.

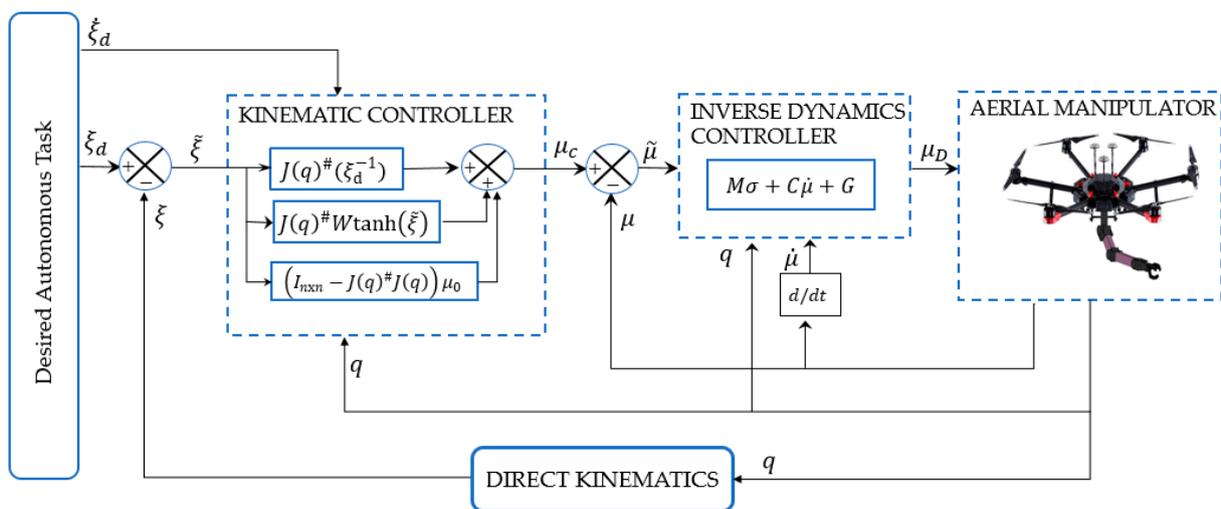


Figure 3. Proposed inverse dynamics controller scheme.

The control law is given by:

$$\mu_D(t) = M(q(t))\sigma + C(q(t), \dot{q}(t))\mu(t) + G(q(t)) \tag{13}$$

where  $\mu_D(t)$  is the control action and  $\sigma$  is defined as:

$$\sigma = \dot{\mu}_c(t) + Q_D \tanh(Q_D^{-1} K_D \tilde{\mu}(t)) \tag{14}$$

where  $\dot{\mu}_c(t)$  is the derivative of the velocity of the Kinematic controller;  $Q_D$  and  $K_D$  are the positive definite calibration matrices;  $\tilde{\mu}(t) = \mu_c(t) - \mu(t)$  represents the velocity error;  $\mu(t)$  is the velocity vector of the aerial manipulator robot, and  $\mu_c(t)$  is the velocity vector of the Kinematic controller.

To analyze the stability of the controller, the system is considered in a closed loop, where it is established that  $\dot{\mu}(t) = \sigma$  and replacing in Equation (14), we obtain:

$$0 = \dot{\tilde{\mu}}(t) + Q_D \tanh(Q_D^{-1} K_D \tilde{\mu}(t)) \tag{15}$$

From (15) it can be verified that  $\tilde{\mu} \rightarrow 0$  When  $t \rightarrow \infty$ .

### 3.3. Sliding Mode Controller (SMC)

The Sliding Mode algorithm considers the dynamics of a system that allows the fulfilment of the control objective by taking the system to a state space through a sliding surface, minimizing the disturbances that may occur. The scheme of this controller is shown in Figure 4.

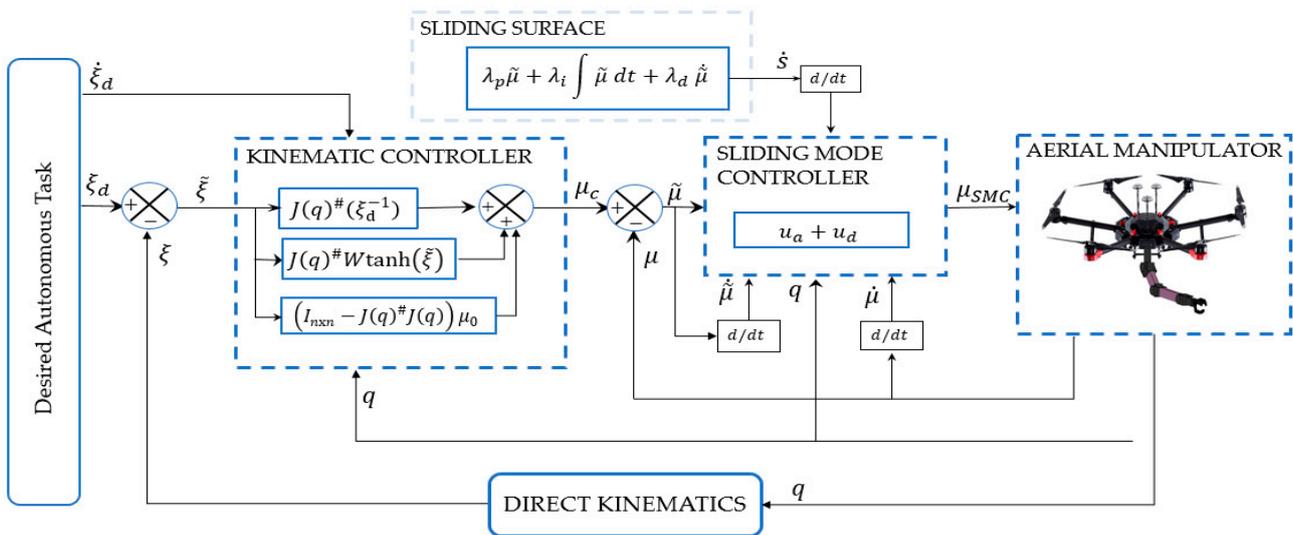


Figure 4. Sliding Mode Controller scheme.

The sliding surface  $s$  of the SMC is described below:

$$s = \lambda_p \tilde{\mu}(t) + \lambda_i \int \tilde{\mu}(t) dt + \lambda_d \dot{\tilde{\mu}}(t) \tag{16}$$

Thus,  $\tilde{\mu}(t) = \mu_c(t) - \mu(t)$  is the error vector of velocities of the aerial manipulator robot;  $\dot{\tilde{\mu}}(t) = \dot{\mu}_c(t) - \dot{\mu}(t)$  is the derivative of the error with respect to time;  $\lambda_p$ ,  $\lambda_i$  y  $\lambda_d$  is the positive definite calibration matrices of the proportional, integral, and derivative part of the controller, respectively.

Obtaining the first derivative of the surface (16) and replacing the errors calculated above, we obtain:

$$\dot{s} = \lambda_p(\dot{\mu}_c(t) - \dot{\mu}(t)) + \lambda_i\tilde{\mu}(t) + \lambda_d\ddot{\mu}(t) \quad (17)$$

where  $\ddot{\mu}(t)$  is the second derivative of  $\tilde{\mu}(t)$ . The SMC control scheme is described with Equation (18)

$$u_{SMC}(t) = u_a(t) + u_d(t) \quad (18)$$

where  $u_a(t)$  is the continuous part and  $u_d(t)$  is the discontinuous part of the control action  $u_{SMC}(t)$ .

Considering  $\dot{s} = 0$  and  $u_d(t) = 0$  and replacing in Equation (17) we obtain:

$$u_a(t) = M(q(t))\dot{\mu}_c(t) + C(q(t), \dot{q}(t))\mu(t) + G(q(t)) + M(q(t))\lambda_i\lambda_p^{-1}\tilde{\mu}(t) + M(q(t))\lambda_d\lambda_p^{-1}\ddot{\mu}(t) \quad (19)$$

Subsequently, the discontinuous part is determined by selecting the Lyapunov candidate  $\mathcal{L}$  described in Equation (20), as a function of the area  $s$ .

$$\mathcal{L} = \frac{1}{2}s^T s \quad (20)$$

Deriving  $\mathcal{L}$  with respect to time we obtain,

$$\dot{\mathcal{L}} = s^T \dot{s} \quad (21)$$

By replacing Equations (17) and (8) in Equation (21) and developing the expressions, we determine the derivative of  $\mathcal{L}$ .

$$\dot{\mathcal{L}} = s^T \left( -\lambda_p(M(q(t)))^{-1}u_d(t) \right) \quad (22)$$

For Equation (22) to satisfy  $\dot{\mathcal{L}} \leq 0$ ,  $u_d(t)$  is defined by:

$$u_d(t) = M(q(t))k_2 \text{sig}(s) \quad (23)$$

where  $k_2$  is a positive defined constant. By replacing Equation (23) in Equation (22) we obtain:

$$\dot{\mathcal{L}} = s^T \left( -\lambda_p k_2 \text{sig}(s) \right) \quad (24)$$

It is verified that if  $k_2 > 0$  y  $\lambda_d > 0$ , then  $\dot{\mathcal{L}} < 0$ , therefore,  $s \rightarrow 0$  with  $t \rightarrow \infty$ , hence, the error  $\tilde{v} \rightarrow 0$  with  $t \rightarrow \infty$ . Then, the proposed controller is:

$$u_{SMC}(t) = M(q(t))\dot{\mu}_c(t) + C(q(t), \dot{q}(t))\mu(t) + G(q(t)) + M(q(t))\lambda_i\lambda_p^{-1}\tilde{\mu}(t) + M(q(t))\lambda_d\lambda_p^{-1}\ddot{\mu}(t) + M(q(t))k_2 \text{sig}(s) \quad (25)$$

To improve the chattering condition, we propose that  $k_2 \text{sig}(s) \cong \frac{s}{|s|+\delta}$  with  $\delta > 0$ ; finally, we obtain the control law given by:

$$u_{SMC}(t) = M(q(t))\dot{\mu}_c(t) + C(q(t), \dot{q}(t))\mu(t) + G(q(t)) + M(q(t))\lambda_i\lambda_p^{-1}\tilde{\mu}(t) + M(q(t))\lambda_d\lambda_p^{-1}\ddot{\mu}(t) + M(q(t))k_2 \frac{s}{|s|+\delta} \quad (26)$$

### 3.4. LAMDA Controller

The LSMC is an intelligent Sliding Mode controller based on LAMDA. The sliding surface defined in Equation (16) and its derivative defined in Equation (17) are composed of vectors with dimensions  $7 \times 1$ , therefore  $s$  and  $\dot{s}$  have the form:  $s = [s_1 \ s_2 \ \dots \ s_7]^T$  and  $\dot{s} = [\dot{s}_1 \ \dot{s}_2 \ \dots \ \dot{s}_7]^T$ .

The LAMDA controller (see Figure 5) is combined with the fundamentals of SMC for the calculation of continuous and discontinuous control actions [24]. For this purpose, the sliding surface  $s_i$  and its derivative  $\dot{s}_i$  are required, which describe an object  $O_i$ , such as:  $O_i = [s_i \dot{s}_i]; \forall i = 1, \dots, 7$ , for all the variables to be controlled. The predefined classes  $C = \{C_1; \dots; C_k; \dots; C_m\}$ , are used to detect the current functional state of the system [23]. If we change the variables  $s_1 = o_{i1}$  and  $\dot{s}_i = o_{i2}$ , then the object  $O_i$  is defined as:

$$O_i = [o_{i1} \ o_{i2}] \tag{27}$$

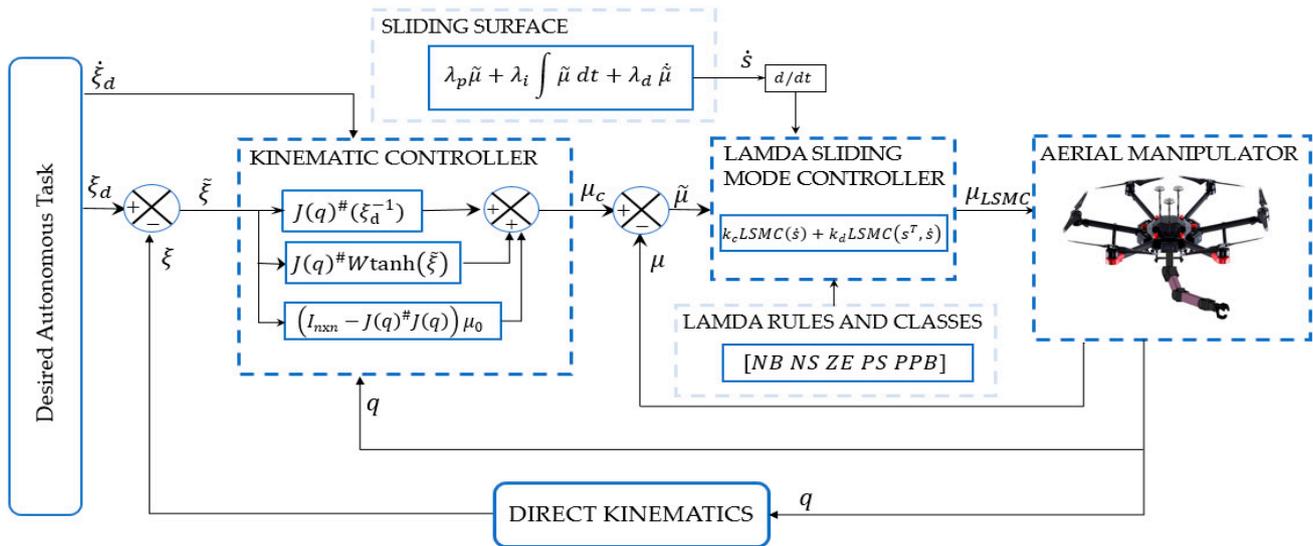


Figure 5. LSMC controller scheme.

Now, the algorithm computes the Marginal Adequacy Degree (MAD), which is a parameter that measures the similarity of a descriptor with the same descriptor in a class using probability density functions, which require the average value of the descriptor  $j$  belonging to the class ( $\rho_{k,ij}$ ) and its standard deviation set as  $\sigma_{k,ij} = 0.25$  [25].

$$MAD_{k,ij} = e^{-\frac{1}{2} \left( \frac{o_{ij} - \rho_{k,ij}}{\sigma_{k,ij}} \right)^2} \tag{28}$$

$$\rho_{k,ij} = \frac{1}{n_{k,ij}} \sum_{t=1}^{n_{k,ij}} o_{ij}(t) \tag{29}$$

where  $n_{k,ij}$  is the amount of data of the descriptor  $ij$  in the class  $k$ .

The Global Adequacy (GAD) measures the membership degree of the object to each class and it is computed by mixing the MADs using fuzzy logic operators, and the Dombi operator [26] has been used in this work. The GADs are linear interpolations between the T-norm “ $T(a, b)$ ” and the S-norm “ $S(a, b)$ ” of the Dombi operator.

The parameters  $a, b$  are the MADs in the class  $k$  to be operated with the T-norm and the S-norm, and  $p \geq 1$  is used to modify the sensitivity.

The GAD for the object  $O_i$  to each class  $k$  is computed by the combination of T-norm and the S-norm as:

$$GAD_{k,O_i}(MAD_{k,i1}, MAD_{k,i2}) = \alpha T(MAD_{k,i1}, MAD_{k,i2}) + (1 - \alpha) S(MAD_{k,i1}, MAD_{k,i2}) \tag{30}$$

where  $\alpha \in [0, 1]$  is the exigency parameter. If  $\alpha$  increases, then the algorithm is stricter, and if  $\alpha$  decreases, then the algorithm is permissive.

The inference method presented in [24] proposes the use of LAMDA classes to evaluate the current state of the system and take corrective action to bring the system to the desired state. The analytical expression that describes the decision-making process is defined as:

$$R^{(k)}_i : IF o_{i1} \text{ is } F_1^p \text{ and } o_{i2} \text{ is } F_j^q \text{ THEN } y_{ki} \text{ is } G_{ki} \tag{31}$$

where  $R^{(k)}_i$  is the rule applied for the functional state (class)  $k$ ,  $o_{ij}; \forall j = 1, 2$  is the descriptor  $j$  of the object  $O_i$  which takes on values of the universe of discourse  $U_j$ . The output linguistic variable  $y_{ki}$  is defined on a universe of discourse  $V$ .  $F_j = \{F_j^q : q = 1, 2, \dots, Q\}$  is a fuzzy set on  $U_j$  where  $Q$  is the number of linguistic values and  $G_{ki}$  is a fuzzy set on  $V$ .

The inference mechanism added to LAMDA is based on each GAD. The first-order TSK (Takagi-Sugeno-Kang) inference method [27] is employed, where  $G_{ki} = \gamma_{ki}$ . The parameter  $\gamma_{ki}$  is a constant value specified for each class to bring the system to the desired functional state, which is  $o_{i1} = 0$  and  $o_{i2} = 0$ . Then, the crisp output is computed as follows:

$$u_i = \zeta_i \sum_{k=1}^m \gamma_{ki} GAD_{k,O_i} \tag{32}$$

where  $u_i$  is the controller output for the object  $O_i$ ,  $\gamma_{ki}$  is its weight applied in the  $k$ -th rule, and  $\zeta_i$  is the adjustment parameter for saturation of the output of the controller. The value of  $\zeta_i$  is computed by:

$$\zeta_i = \frac{\text{argmax}(\gamma_{ki})}{\sum_{k=1}^m \gamma_{ki} GAD_{k, \text{argmax}(O_i)}} \tag{33}$$

The procedure shown in Equations (27)–(33) is used to compute the control actions of for each variable (each controller). Now, to compute  $u_{c_L}$  and  $u_{d_L}$  of the SMC control action, the same surface defined in Equation (16) and its derivative defined in Equation (17) are used.

Replacing Equation (8) in Equation (17) we obtain:

$$\begin{aligned} \dot{s} &= \lambda_p \left( \dot{\mu}_c(t) - (M(q(t)))^{-1} [u_{c_L} - C(q(t), \dot{q}(t))\mu(t) - G(q(t))] \right) + \lambda_i \tilde{\mu}(t) + \lambda_d \ddot{\mu}(t) \\ \dot{s} &= \lambda_p \left( \dot{\mu}_c(t) - (M(q(t)))^{-1} u_{c_L} + (M(q(t)))^{-1} C(q(t), \dot{q}(t))\mu(t) - (M(q(t)))^{-1} G(q(t)) \right) \\ &\quad + \lambda_i \tilde{\mu}(t) + \lambda_d \ddot{\mu}(t) \end{aligned} \tag{34}$$

$M(q(t))$  is defined as positive, thus, analyzing Equation (34), we can note that  $\dot{s}$  decreases as  $u_{c_L}$ . Now, the algorithm computes the Marginal Adequacy Degree (MAD), which is a parameter that measures the similarity of a descriptor with the same descriptor in a class using probability density functions, which require the average value of the descriptor  $j$  belonging to the  $\dot{s} = 0$ , which is the desired condition in SMC schemes.

The number of classes defined in each descriptor depends on the knowledge of the expert (designer). However, in the sensitivity analysis of Morales et al. [24], it is stated that with five classes per descriptor, very good results can be obtained without affecting the computational load. Thus, five classes are considered in this work  $C \in [-1, 1]$  for  $\dot{s}$ , as is detailed in [28]. The fuzzy sets for the classes of  $\dot{s}$  are: Negative Big ( $NB = -1$ ), Negative Small ( $NS = -0.5$ ), zero ( $ZE = 0$ ), Positive Small ( $PS = 0.5$ ), and Positive Big ( $PB = 1$ ). These classes are used to define the rules to compute the normalized continuous control action  $u_{nc_L}$ . The scaling gain  $k_{c_L}$  is used for tuning at the input  $\dot{s}$ , and the scaling gain  $k_c$  at the output as:

$$u_{c_L} = k_{c_L} u_{nc_L} \Rightarrow u_{c_L} = k_c LSMC(\dot{s}); k_{c_L} > 0 \tag{35}$$

As a practical example, we can note that in Equation (35), if  $\dot{s}$  is  $PB$ , then large positive control action  $u_c$  is required to decrease quickly  $\dot{s}$  in order to obtain  $\dot{s} = 0$ . If  $\dot{s} = ZE$ , the desired condition is satisfied, then no control action is needed, thus  $u_{c_L} = ZE$ . The rule table corresponding to the continuous control action is shown in Table 1.

**Table 1.** Rule table of LSMC for  $\dot{s}$ .

	$\dot{s}$				
	NB	NS	ZE	PS	PB
$M > 0$	$C_1 = NB$	$C_2 = NS$	$C_3 = ZE$	$C_4 = PS$	$C_5 = PB$

Consider that the values presented in Table 1 for each of the classes is the one corresponding to the weight applied to the control output  $\gamma_k$ .

To compute the discontinuous control action  $u_{dL}$ , which is in charge of attracting the system to the sliding surface, we use the derivative of the Lyapunov function  $\dot{\mathcal{L}}$  defined in Equation (21).

Replacing Equation (34) in Equation (21), and considering only the discontinuous control action  $u_{LSMC} = u_{dL}$ , we obtain:

$$s^T \dot{s} = s^T \lambda_p \left( \dot{\mu}_c(t) - (M(q(t)))^{-1} u_{aL} + (M(q(t)))^{-1} C(q(t), \dot{q}(t)) \mu(t) - (M(q(t)))^{-1} G(q(t)) \right) + s^T \lambda_i \tilde{\mu}(t) + s^T \lambda_d \ddot{\mu}(t) \tag{36}$$

To guarantee stability, Lyapunov theory establishes to satisfy the condition  $\dot{\mathcal{L}} < 0$ . Thus, considering Equation (36), it is necessary to satisfy  $s^T \dot{s} < 0$ . As in the case of continuous control action, five classes are set for each input  $\dot{s}$  and  $s^T$ . Due to the normalization of the classes that is computed  $u_{ndL}$ , the scaling gain  $k_2$  is placed at the input  $s$ , and the scaling gain  $k_d$  at the discontinuous control output as:

$$u_{dL} = k_d u_{ndL} \Rightarrow u_{dL} = k_d LSMC(s^T, \dot{s}); k_d > 0 \tag{37}$$

The rule table for  $u_{dL}$  in which  $\dot{s}$  and  $s^T$  is required, considering that  $M(q(t))$  is defined positive, is shown in Table 2. The rule definition is detailed in [24].

**Table 2.** Rule table of LSMC for  $s(t)$  and  $\dot{s}$ .

		$\dot{s}$				
		NB	NS	ZE	PS	PB
$s^T$	PB	$C_5 = ZE$	$C_{10} = ZE$	$C_{15} = PS$	$C_{20} = PB$	$C_{25} = PB$
	PS	$C_4 = ZE$	$C_9 = ZE$	$C_{14} = PS$	$C_{19} = PB$	$C_{24} = PB$
	ZE	$C_3 = NB$	$C_8 = NS$	$C_{13} = ZE$	$C_{18} = PS$	$C_{23} = PB$
	NS	$C_2 = NB$	$C_7 = NB$	$C_{12} = NS$	$C_{17} = ZE$	$C_{22} = ZE$
	NB	$C_1 = NB$	$C_6 = NB$	$C_{11} = NS$	$C_{16} = ZE$	$C_{21} = ZE$

Consider that the values presented in Table 2 for each of the classes are the ones corresponding to the weight applied to the control output  $\gamma_k$ . Table 2 can be analyzed as follows: For example, in the class  $C_1$ , where  $s^T = NB$  and  $\dot{s} = NB$ , the product  $s^T \dot{s}$  is  $PB$ . Therefore, based on Equation (36) and when  $s^T < 0$ , the negative control input is required ( $\gamma_1 = NB$ ) to quickly decrease  $s(t)\dot{s}(t)$ . The same situation is presented in the classes  $C_2, C_6, C_7$ . In the class  $C_5$ , where  $s^T = PB$  and  $\dot{s} = NB$ , the product  $s^T \dot{s}$  is  $NB$ . Therefore, no change in the control action  $u_d$  is required, thus  $\gamma_5 = ZE$ . The other classes in which no change in the control action is required because the condition is met are  $C_4, C_9, C_{10}, C_{16}, C_{17}, C_{21}$ , and  $C_{22}$ .

As it can be seen in the controller, the  $sig()$  function used in conventional SMC controllers (see Equation (23)) is not required, since LAMDA replaces this function, making it a chattering-free controller. The overall control action is represented in Figure 5, and defined as:

$$u_{LSMC} = k_c LSMC(\dot{s}) + k_d LSMC(s^T, \dot{s}) \tag{38}$$

where:

$$\begin{aligned}
 LSMC(\dot{s}) &= \frac{\text{argmax}(\gamma_{k1})}{\sum_{k=1}^m \gamma_{k1} GAD_{k, \text{argmax}(O)}} \sum_{k=1}^m \gamma_{k1} GAD_{k,O} \text{ and;} \\
 LSMC(s^T, \dot{s}) &= \frac{\text{argmax}(\gamma_{k2})}{\sum_{k=1}^m \gamma_{k2} GAD_{k, \text{argmax}(O)}} \sum_{k=1}^m \gamma_{k2} GAD_{k,O}
 \end{aligned}
 \tag{39}$$

The stability of the controller is guaranteed if selected:

$$(k_c + k_d) > \beta_{vc} - (\beta_A + \beta_d)
 \tag{40}$$

With:

$$\left\{ \begin{aligned}
 &\|-(M(q(t)))^{-1}C(q(t), \dot{q}(t))\| \leq \beta_A \\
 &\|\dot{\mu}_c(t)\| \leq \beta_{vc} \\
 &\|d\| \leq \beta_d
 \end{aligned} \right.
 \tag{41}$$

where  $\beta_{vc}$ ,  $\beta_A$ , and  $\beta_d$  are the limits of the bounded functions corresponding to the change of the reference, the state matrix, and the disturbance, respectively. The expression (40) guarantees  $\dot{\mathcal{L}} < 0$ , therefore,  $s \rightarrow 0$  with  $t \rightarrow \infty$  as proven in [26].

#### 4. Experimental Results

This section presents the experimental results to validate the proposed control algorithms for the aerial manipulator robot. Four experiments are described on a real aerial manipulator robot with three degrees of freedom: (i) Experiment 1: Kinematic controller implementation, (ii) Experiment 2: Inverse Dynamics controller implementation, (iii) Experiment 3: SMC controller implementation, and (iv) Experiment 4: LAMDA controller implementation.

The tests consisted of the tracking of a lemniscate trajectory by the aerial manipulator robot, whose desired trajectory parameters were:  $\xi_d = [\xi_{xd} \ \xi_{yd} \ \xi_{zd}]$ ,  $\xi_{xd} = 2.5 \cos(0.02t)$  m,  $\xi_{yd} = 2.5 \sin(0.04t)$  m,  $\xi_{zd} = (\sin(0.1t) + 0.6)$  m. The time derivative of the desired trajectory was:  $\dot{\xi}_d = [\dot{\xi}_{xd} \ \dot{\xi}_{yd} \ \dot{\xi}_{zd}]$ ,  $\dot{\xi}_{xd} = (-2.5 * 0.2 \sin(0.02t))$  m/s,  $\dot{\xi}_{yd} = (2.5 * 0.04 \cos(0.04t))$  m/s,  $\dot{\xi}_{zd} = (0.1 \sin(0.1t) + 12)$  m/s.

The parameters of the aerial manipulator robot were length of joints 1, 2, and 3 defined by  $l_1 = 0.4$  m,  $l_2 = 0.26$  m,  $l_3 = 0.12$  m, and  $r = 0.12$  m, which is the height of the hexacopter.  $q(t) = [u_l \ u_m \ u_n \ \omega \ q_1 \ q_2 \ q_3] = [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]$ ;  $[q_{10} \ q_{20} \ q_{30}] = [0 \ 0 \ 0]^\circ$ . The initial conditions of position of the end-effector were  $\zeta = [-1.30 \ -0.05 \ 1.70]$  m and the initial conditions of position of the desired trajectory were  $\xi_{d0} [2.5 \ 2 \ 12]$  m.

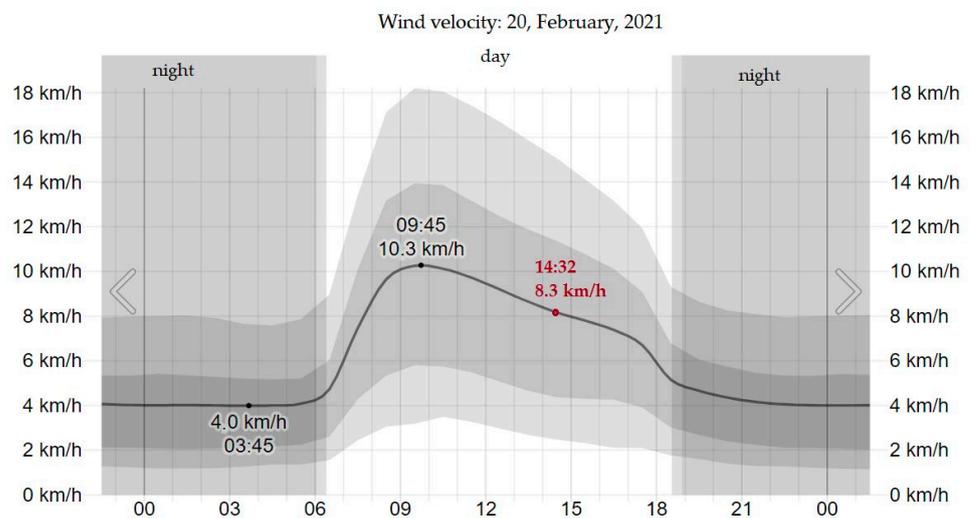
The following parameters were available in the kinematic controller:  $W = I_{3 \times 3}$  identity matrix,  $K = I_{3 \times 3}$  identity matrix, whereas the parameters defined for the Inverse Dynamics controller were:  $K_D = I_{7 \times 7}$  identity matrix,  $Q_D = 2I_{7 \times 7}$  identity matrix. The SMC controller parameters were as follows:  $\lambda_p = 1$ ;  $\lambda_d = 1$ ;  $\lambda_i = 1$ ;  $k_2 = 1$ ;  $\delta = 0.6$ . Finally, the parameters of the LAMDA controller were stated as:  $k_c = 0.1I_{7 \times 7}$ ,  $k_d = \mathcal{B}_{7 \times 7}$ , a diagonal matrix whose diagonal elements are  $[10 \ 10 \ 40 \ 20 \ 20 \ 20 \ 20]$ ,  $\beta_d = 0.5$ ,  $\beta_{vc} = 0.1$ , and  $\beta_A = 20$ .

For the implementation of the four control algorithms, the experimental tests were carried out on the hexacopter-type aerial manipulator robot incorporating a three-degree-of-freedom robotic arm shown in Figure 6.



**Figure 6.** Aerial manipulator robot used in the in experimentation.

The experiments were developed in the city of Ambato, province of Tungurahua, Ecuador, whose latitude and longitude coordinates are  $1^{\circ}16'36.4''$  S  $78^{\circ}35'19.0''$  W. The experiment started at 14:32 on 20 February 2021. Wind velocity at that time and location was 8.3 km/h, approximately, according to [29] as evidenced in Figure 7.



**Figure 7.** Wind velocity during experimentation.

#### 4.1. Experiment 1: Kinematic Controller

In this experiment, the kinematic controller is implemented within the aerial manipulator robot to track the lemniscate trajectory. Figure 8 depicts the described trajectory, observing that there is an expected error in the tracking, due to the fact that the dynamics of the aerial manipulator robot had not been considered.

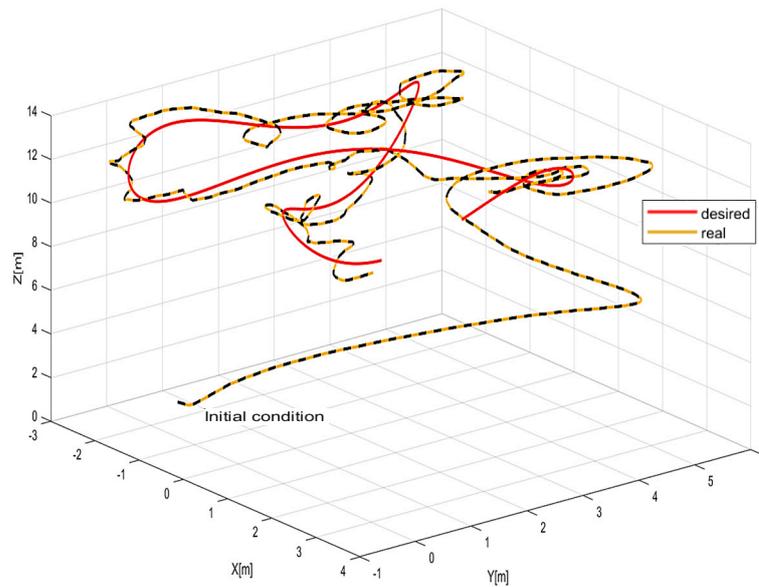


Figure 8. Trajectory tracking: Kinematic controller.

Figure 9 depicts the position errors obtained for the Kinematic controller, showing errors of up to 1 m of amplitude. These errors are because the dynamics of the aerial manipulator robot is not considered. In addition, the presence of wind influences the performance of the controller since it is considered as an external disturbance.

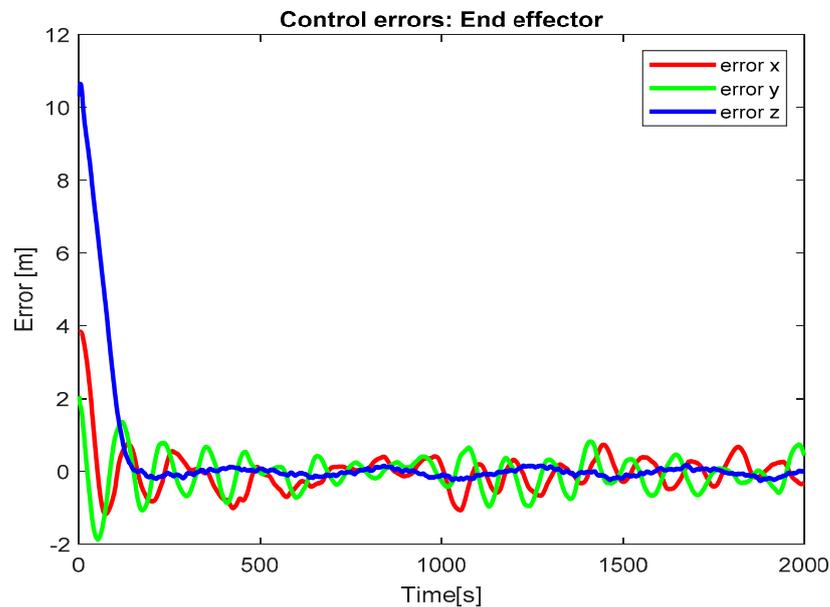


Figure 9. Control errors: Kinematic controller.

4.2. Experiment 2: Inverse Dynamics Controller

In this test, the performance of the controller for trajectory tracking is observed. The trajectory described by the aerial manipulator robot and the desired trajectory are presented in Figure 10. We observed that the end effector of the aerial manipulator robot follows the desired trajectory with a margin of error of approximately 0.4 m as verified in Figure 11. Therefore, a better performance of this controller is observed in comparison with the Kinematic controller, despite the presence of a wind of approximately 8.3 km/h.

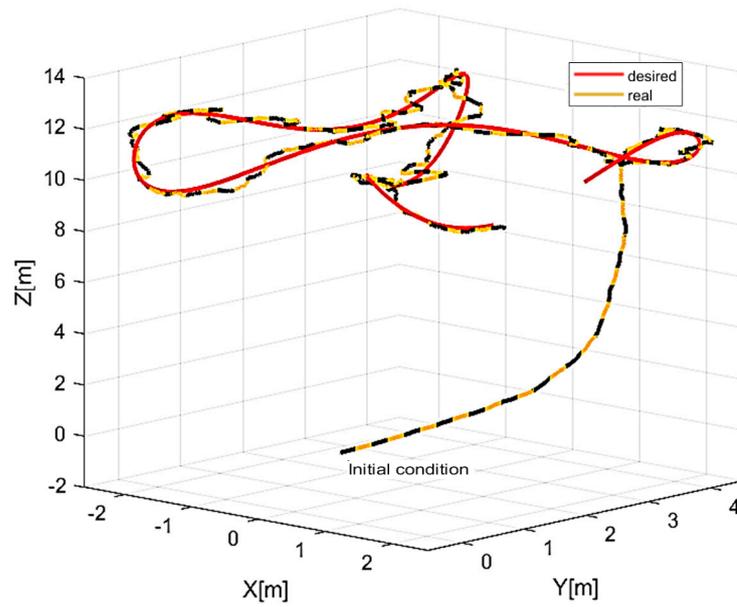


Figure 10. Trajectory tracking: Inverse Dynamics controller.

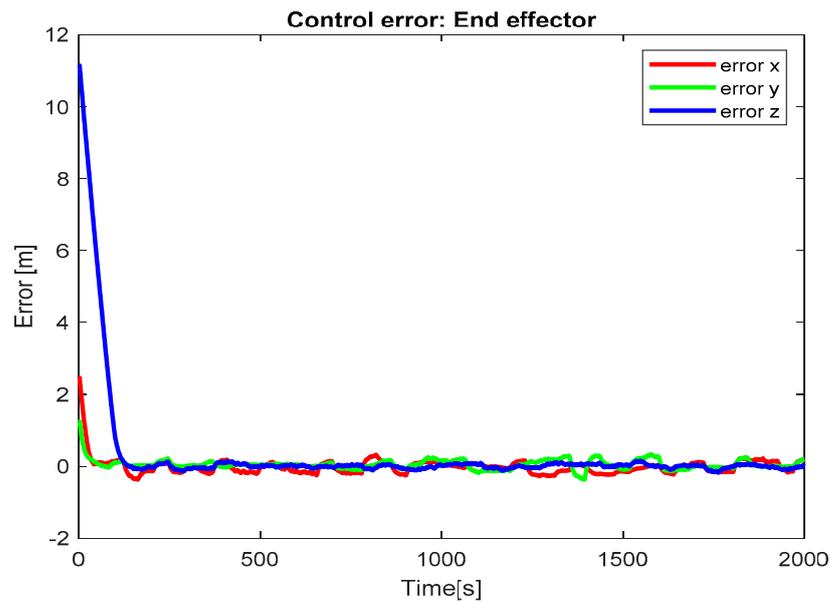


Figure 11. Control errors: Inverse Dynamics controller.

4.3. Experiment 3: SMC Controller

Figure 12 shows the results of the trajectory tracking performed by the aerial manipulator robot for this controller, showing that it performs a lemniscate trajectory with a margin of error similar to the Inverse Dynamics controller, since both controllers incorporate within their control law the compensation of the dynamics of the aerial manipulator robot. Figure 13 depicts the control errors present in this experiment, reaching a maximum value of approximately 0.4 m.

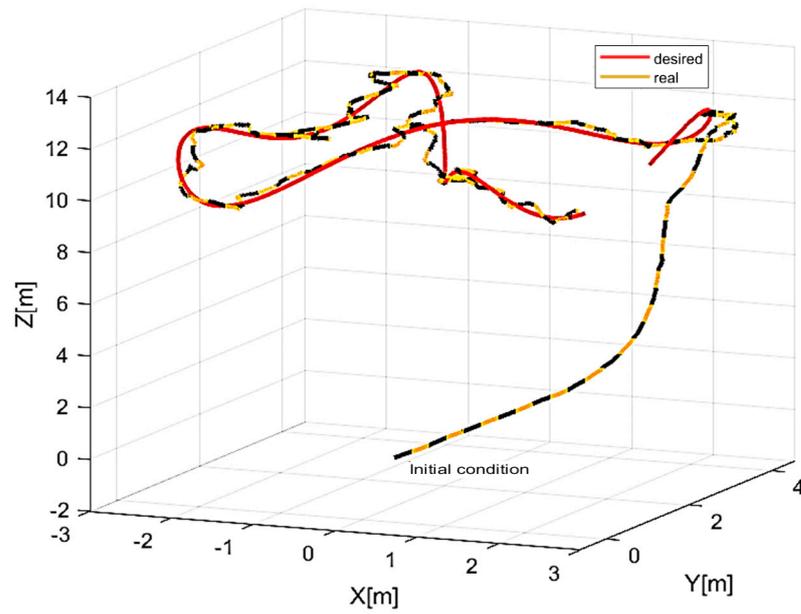


Figure 12. Trajectory tracking: SMC controller.

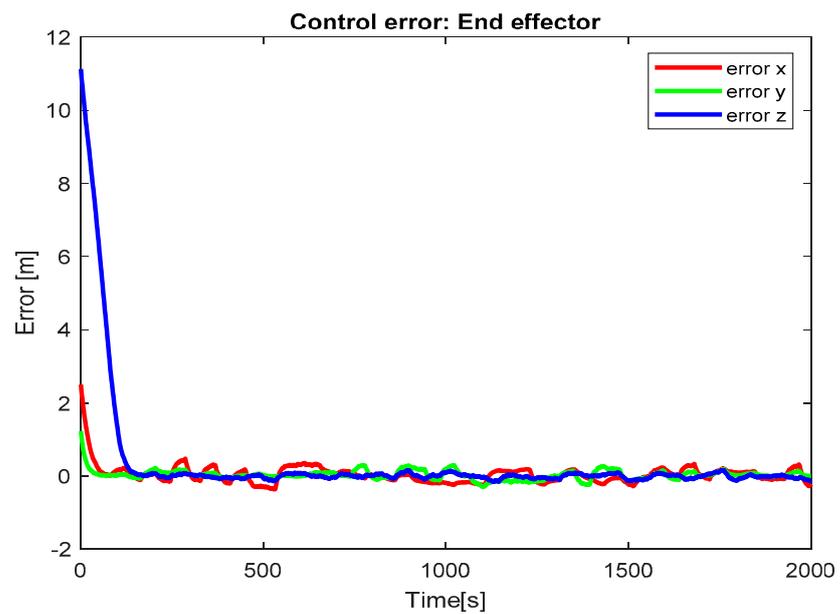


Figure 13. Control errors: SMC controller.

4.4. Experiment 4: LAMDA Controller

The results of the new control strategy applied for the trajectory tracking of an aerial manipulator robot are shown in Figures 14–17. Figure 14 shows the trajectory carried out by the end-effector of the aerial manipulator robot, corroborating that it follows the desired trajectory with an expected margin of error.

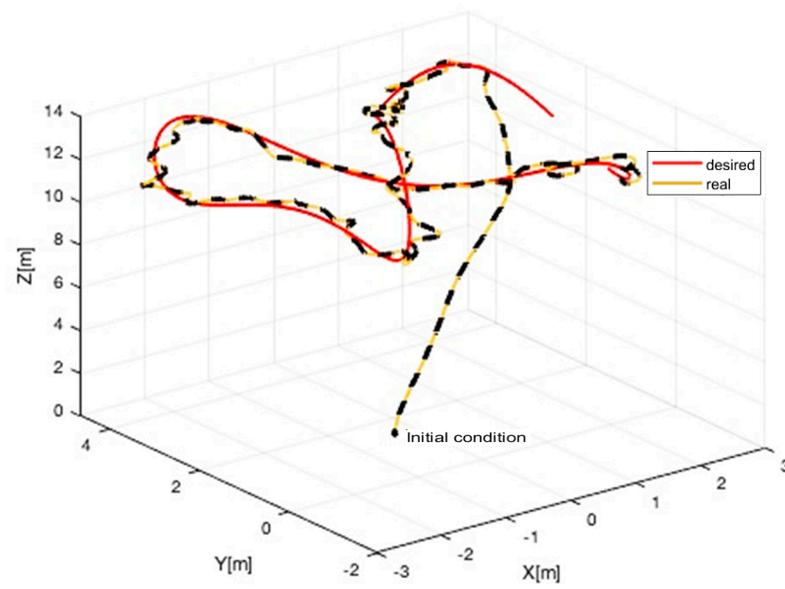


Figure 14. Trajectory tracking: LAMDA controller.

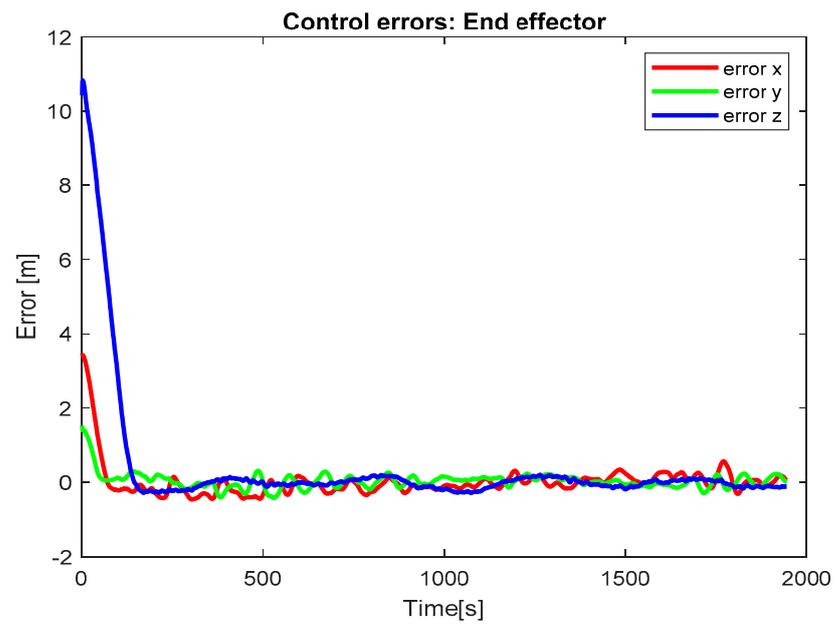


Figure 15. Control errors: LAMDA controller.

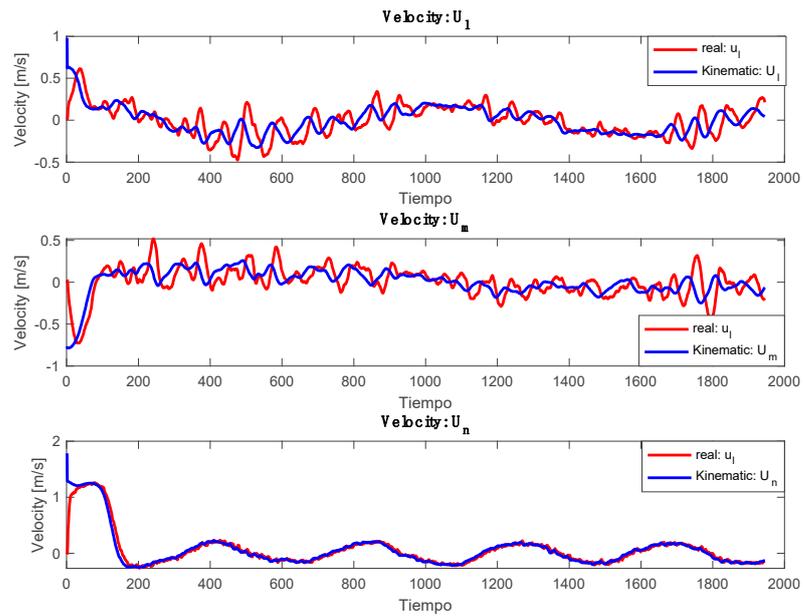


Figure 16. Control signals: LAMDA controller.

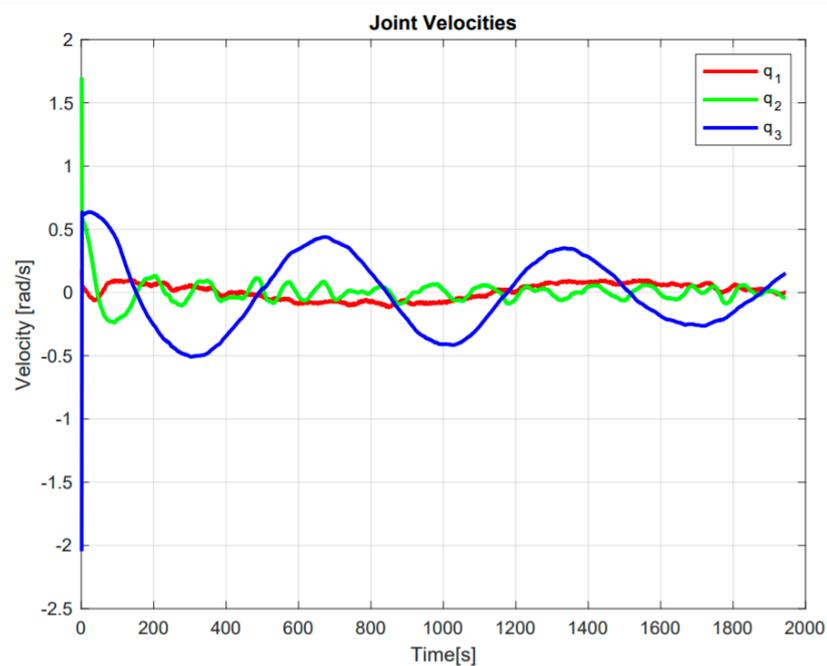


Figure 17. Joint velocities: LAMDA controller.

Figure 15 shows the end-effector position errors, with a maximum error of approximately 0.4 m. This corroborates the good performance of the new proposed LAMDA control system that has not been used yet for aerial manipulators robots. This type of control even responds favorably to the presence of external disturbances such as wind velocity of approximately 8.3 km/h.

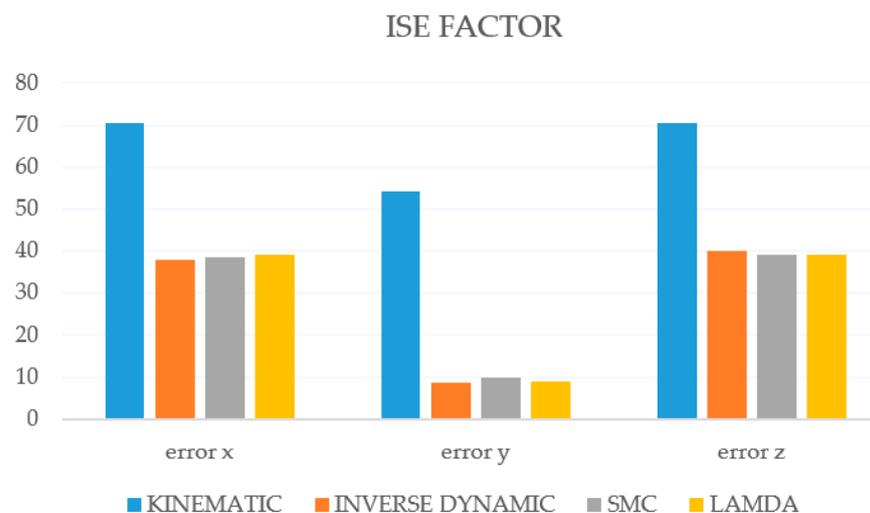
The control signals are shown in Figure 16, where we can observe that there is a good velocity tracking by the aerial manipulator robot, verifying that its control signals are smooth and slightly oscillatory, which allows its actuators to not receive abrupt signals that shorten its useful life.

The velocities of the joints of the robotic arm  $\dot{q}_1$ ,  $\dot{q}_2$  y  $\dot{q}_3$  are shown in Figure 17. We observed that joint 3 generates an angular velocity greater than the rest of joints of robotic

arm since it is easier for the aerial manipulator robot to move this joint of the robotic arm to achieve the control objective. Therefore, minor movements are generated in joint 1 and 2 as verified in this figure.

Taking into account the experiments carried out, it is possible to confirm the good behavior of the LAMDA controller, exhibiting a better performance compared to the Kinematic controller and a similar performance in comparison to the Inverse Dynamics and Sliding Mode control strategies for trajectory tracking.

The obtained Integral square error (ISE) factor is shown in Figure 18, verifying that the new LAMDA control proposal can be applied to aerial manipulators robots, obtaining a good performance that can be comparable to control techniques with dynamic compensation for these systems.



**Figure 18.** ISE factor for the different control techniques used in the experiments.

## 5. Discussion and Conclusions

In this work, experimental tests were carried out for four controllers: Kinematic, Inverse Dynamics, SMC Sliding Mode, and LAMDA, aimed at trajectory tracking by an aerial manipulator robot. The experimental results obtained from the Kinematic controller showed higher errors with respect to the other controllers. This fact was expected since the Kinematic controller does not compensate the dynamics of the system. Furthermore, the presence of wind causes the controller to not exhibit a good performance, as shown in Figure 18. The Inverse Dynamics, SMC, and LAMDA controllers exhibited better results with respect to the Kinematic controller, showing a good performance. The corresponding stability analysis of the proposed controllers have also been presented. The errors obtained in the latter controllers were as expected according to their nature. Furthermore, the presence of wind generates errors in the trajectory tracking.

The proposed LAMDA controller applied to trajectory tracking for the aerial manipulator robot demonstrated its stability in experimental tests under disturbance conditions. Therefore, the LAMDA controller shows a better performance than the Kinematic controller and a similar performance to the rest of the controllers developed in this work, as it can be seen in the experimental tests through the ISE.

Although LAMDA has similar results to the Inverse Dynamics and SMC controllers, its advantage over them is that LAMDA does not depend on the model of the aerial manipulator robot. Thus, it is not necessary to determine and obtain the parameters of the aerial manipulator robot for the implementation of the algorithm, which is usually complex and requires validation prior to use in the different control algorithms that are based on the model. Due to the nature of its structure, the LAMDA controller is based on a set of fixed hidden layers and is considered as an artificial intelligence method, whose

computational performance is not as high compared to similar algorithms [24], as well as the fact that the mathematics turns out to be not so complex.

The LAMDA controller is a technique that has not been used previously for robotic systems such as aerial manipulators robots, although this type of controller has been implemented in other applications such as data analysis, data classification, system identification, and process regulation.

LAMDA is robust against disturbances due to its design based on the fundamentals of SMC, considering the stability of Lyapunov. LAMDA has presented good behavior in the presence of wind whose velocity was around 8.3 km/h, so there is evidence that promising results could be obtained with the application of this controller in the area of robotics, specifically for aerial manipulator robots.

The objective of this work is to demonstrate that the LAMDA controller was experimentally tested in applications not common to which it has been used habitually. Thus, for the first time, it has been used in aerial manipulators robots for trajectory tracking, the results of which confirm that LAMDA controllers can be brought into the field of robotics with promising results. Future work deals with the implementation of this controller for cooperative systems, as well as the application of adaptive fuzzy logic to the LAMDA controller to compensate systems with delays or disturbances that are usually present in robotic systems.

**Author Contributions:** Conceptualization, G.M.A., L.M., P.L. and G.P.-N.; methodology, G.M.A., L.M. and P.L.; software, G.M.A., P.L. and V.H.A.; validation, G.M.A., P.L. and V.H.A.; formal analysis, G.M.A., L.M., and P.L.; investigation, G.M.A.; resources, G.M.A. and L.M.; data curation, G.P.-N.; writing—original draft preparation, G.M.A. and L.M.; writing—review and editing, P.L., G.P.-N. and V.H.A.; visualization, G.M.A.; supervision, G.P.-N. and P.L.; project administration, G.P.-N. and P.L.; funding acquisition, G.M.A. and P.L. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Acknowledgments:** This article has been elaborated under support of the Corporación Ecuatoriana para el Desarrollo de la Investigación and Academia CEDIA, through the project CEPRA-XIII-2019-08: “Sistema colaborativo de robots Aéreos para Manipular Cargas con Óptimo Consumo de Recursos” and also supported by project PIS-19-10 of the Escuela Politécnica Nacional. The authors would also like to thank the ARSI and GIECAR research groups for their support in developing this work.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Tognon, M.; Chavez, H.A.T.; Gasparin, E.; Sable, Q.; Bicego, D.; Mallet, A.; Lany, M.; Santi, G.; Revaz, B.; Cortes, J.; et al. A Truly-Redundant Aerial Manipulator System with Application to Push-and-Slide Inspection in Industrial Plants. *IEEE Robot. Autom. Lett.* **2019**, *4*, 1846–1851. [[CrossRef](#)]
2. Hamaza, S.; Georgilas, I.; Richardson, T. 2D Contour Following with an Unmanned Aerial Manipulator: Towards Tactile-Based Aerial Navigation. In Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China, 4–8 November 2019; pp. 3664–3669.
3. Bartelds, T.; Capra, A.; Hamaza, S.; Stramigioli, S.; Fumagalli, M. Compliant Aerial Manipulators: Toward a New Generation of Aerial Robotic Workers. *IEEE Robot. Autom. Lett.* **2016**, *1*, 477–483. [[CrossRef](#)]
4. Lee, S.; Kim, H.; Kim, U.; Lee, H. Concave Wall Surface Tracking for Aerial Manipulator Using Contact Force Estimation Algorithm. In Proceedings of the 2020 20th International Conference on Control, Automation and Systems (ICCAS), Busan, Korea, 13 October 2020; pp. 850–855.
5. Yang, Y.; Liu, J.; Li, Z.; Yang, X.; Yu, X.; Gao, H. Nonlinear Disturbance Observer Based Adaptive Backstepping Control for Trajectory Tracking of Aerial Parallel Manipulator. In Proceedings of the IECON 2020 the 46th Annual Conference of the IEEE Industrial Electronics Society, Singapore, 18 October 2020; pp. 4776–4781.

6. Guayasamin, A.; Leica, P.; Herrera, M.; Camacho, O. Trajectory Tracking Control for Aerial Manipulator Based on Lyapunov and Sliding Mode Control. In Proceedings of the 2018 International Conference on Information Systems and Computer Science (INCISCOS), Quito, Ecuador, 14–16 November 2018; pp. 36–41.
7. Imanberdiyev, N.; Kayacan, E. Redundancy Resolution Based Trajectory Generation for Dual-Arm Aerial Manipulators via Online Model Predictive Control. In Proceedings of the IECON 2020 the 46th Annual Conference of the IEEE Industrial Electronics Society, Singapore, 18 October 2020; pp. 674–681.
8. Tlatelpa-Osorio, Y.E.; Rodriguez-Cortes, H.; Acosta, J.A. A Decentralized Approach for the Aerial Manipulator Trajectory Tracking. In Proceedings of the 2020 International Conference on Unmanned Aircraft Systems (ICUAS), Athens, Greece, 1–4 September 2020; pp. 504–511.
9. Wang, X.; Zhou, Z.; Chen, S.; Wang, R. Trajectory Tracking and Vibration Suppression of Spacecraft Combination Connected by a Space Manipulator. In Proceedings of the 2019 Chinese Control Conference (CCC), Guangzhou, China, 27–30 July 2019; pp. 8055–8060.
10. Camacho, O.; Leica, P.; Antamba, J.; Quinonez, J. Null-Space Based Control Applied to a Formation of Aerial Manipulators in Congested Environment. In Proceedings of the 2019 International Conference on Information Systems and Computer Science (INCISCOS), Quito, Ecuador, 20–22 November 2019; pp. 244–250.
11. Gkoutas, K.; Tzes, A. Leader/Follower Force Control of Aerial Manipulators. *IEEE Access* **2021**, *9*, 17584–17595. [[CrossRef](#)]
12. Morton, K.; McFadyen, A.; Gonzalez, F. Generalized Trajectory Control for Tree-Structured Aerial Manipulators. In Proceedings of the 2018 International Conference on Unmanned Aircraft Systems (ICUAS), Dallas, TX, USA, 12–15 June 2018; pp. 947–956.
13. Jia, P.; Li, E.; Liang, Z.; Qiang, Y. Adaptive Neural Network Control of an Aerial Work Platform's Arm. In Proceedings of the 10th World Congress on Intelligent Control and Automation, Beijing, China, 6–8 July 2012; pp. 3567–3570.
14. Muscio, G.; Pierri, F.; Trujillo, M.A.; Cataldi, E.; Antonelli, G.; Caccavale, F.; Viguria, A.; Chiaverini, S.; Ollero, A. Coordinated Control of Aerial Robotic Manipulators: Theory and Experiments. *IEEE Trans. Contr. Syst. Technol.* **2018**, *26*, 1406–1413. [[CrossRef](#)]
15. Liu, Y.-C.; Huang, C.-Y. DDPG-Based Adaptive Robust Tracking Control for Aerial Manipulators With Decoupling Approach. *IEEE Trans. Cybern.* **2021**, 1–14. [[CrossRef](#)]
16. Naldi, R.; Pounds, P.; De Marco, S.; Marconi, L. Output Tracking for Quadrotor-Based Aerial Manipulators. In Proceedings of the 2015 American Control Conference (ACC), Chicago, IL, USA, 1–3 July 2015; pp. 1855–1860.
17. Mora-Florez, J.; Barrera-Nunez, V.; Carrillo-Caicedo, G. Fault Location in Power Distribution Systems Using a Learning Algorithm for Multivariable Data Analysis. *IEEE Trans. Power Deliv.* **2007**, *22*, 1715–1721. [[CrossRef](#)]
18. Botía Valderrama, J.F.; Botía Valderrama, D.J.L. On LAMDA Clustering Method Based on Typicality Degree and Intuitionistic Fuzzy Sets. *Expert Syst. Appl.* **2018**, *107*, 196–221. [[CrossRef](#)]
19. van der Tak, F.; Lique, F.; Faure, A.; Black, J.; van Dishoeck, E. The Leiden Atomic and Molecular Database (LAMDA): Current Status, Recent Updates, and Future Plans. *Atoms* **2020**, *8*, 15. [[CrossRef](#)]
20. Valderrama, J.F.B.; Valderrama, D.J.L.B. Two Cluster Validity Indices for the LAMDA Clustering Method. *Appl. Soft Comput.* **2020**, *89*, 106102. [[CrossRef](#)]
21. Ricardo Hernandez, H.; Luis Camas, J.; Medina, A.; Perez, M.; Veronique Le Lann, M. Fault Diagnosis by LAMDA methodology Applied to Drinking Water Plant. *IEEE Latin Am. Trans.* **2014**, *12*, 985–990. [[CrossRef](#)]
22. Morales Escobar, L.; Aguilar, J.; Garces-Jimenez, A.; Gutierrez De Mesa, J.A.; Gomez-Pulido, J.M. Advanced Fuzzy-Logic-Based Context-Driven Control for HVAC Management Systems in Buildings. *IEEE Access* **2020**, *8*, 16111–16126. [[CrossRef](#)]
23. Morales, L.; Aguilar, J.; Rosales, A.; Chávez, D.; Leica, P. Modeling and Control of Nonlinear Systems Using an Adaptive LAMDA Approach. *Appl. Soft Comput.* **2020**, *95*, 106571. [[CrossRef](#)]
24. Morales, L.; Aguilar, J.; Camacho, O.; Rosales, A. An Intelligent Sliding Mode Controller Based on LAMDA for a Class of SISO Uncertain Systems. *Inf. Sci.* **2021**. [[CrossRef](#)]
25. Ruiz, F.A.; Isaza, C.V.; Agudelo, A.F.; Agudelo, J.R. A New Criterion to Validate and Improve the Classification Process of LAMDA Algorithm Applied to Diesel Engines. *Eng. Appl. Artif. Intell.* **2017**, *60*, 117–127. [[CrossRef](#)]
26. Morales, L.; Aguilar, J.; Chávez, D.; Isaza, C. LAMDA-HAD, an Extension to the LAMDA Classifier in the Context of Supervised Learning. *Int. J. Inf. Technol. Decis. Mak.* **2020**, *19*, 283–316. [[CrossRef](#)]
27. Wai, R.-J.; Chen, P.-C. Intelligent Tracking Control for Robot Manipulator Including Actuator Dynamics via TSK-Type Fuzzy Neural Network. *IEEE Trans. Fuzzy Syst.* **2004**, *12*, 552–559. [[CrossRef](#)]
28. Li, D.; Zhenqi, G. Trajectory Tracking Control of an Aerial Robot Based on Linear Active Disturbance Rejection Control Approach. In Proceedings of the 2019 Chinese Control And Decision Conference (CCDC), Nanchang, China, 3–5 June 2019; pp. 5004–5009.
29. Weather Spark. Average Wheather on February 20th in Ambato. 2021. Available online: <https://es.weatherspark.com/d/20027/2/20/Tiempo-promedio-el-20-de-febrero-en-Ambato-Ecuador#Sections-Wind> (accessed on 25 March 2021).