drones

Article

Robust Neural Network Consensus for Multiagent UASs Based on Weights' Estimation Error

Alejandro Morfin-Santana [1], Filiberto Muñoz [1,2,*], Sergio Salazar [1] and José Manuel Valdovinos [1]

[1] Laboratory LAFMIA-UMI, Center for Research and Advanced Studies of the National Polytechnic Institute (CINVESTAV), Mexico City 07360, Mexico
[2] Mexican National Council of Science and Technology (CONACYT), Mexico City 03940, Mexico
* Correspondence: filiberto.munoz@cinvestav.mx

Abstract: We propose a neural network consensus strategy to solve the leader–follower problem for multiple-rotorcraft unmanned aircraft systems (UASs), where the goal of this work was to improve the learning based on a set of auxiliary variables and first-order filters to obtain the estimation error of the neural weights and to introduce this error information in the update laws. The stability proof was conducted based on Lyapunov's theory, where we concluded that the formation errors and neural weights' estimation error were uniformly ultimately bounded. A set of simulation results were conducted in the Gazebo environment to show the efficacy of the novel update laws for the altitude and translational dynamics of a group of UASs. The results showed the benefits and insights into the coordinated control for multiagent systems that considered the weights' error information compared with the consensus strategy based on classical σ-modification. A comparative study with the performance index ITAE and ITSE showed that the tracking error was reduced by around 45%.

Keywords: multiagent system; neural network; unmanned aircraft systems; estimation error information

Citation: Morfin-Santana, A.; Muñoz, F.; Salazar, S.;Valdovinos, J.M. Robust Neural Network Consensus for Multiagent UASs Based on Weights' Estimation Error. *Drones* **2022**, *6*, 300. https://doi.org/10.3390/drones6100300

Academic Editor: Andrey V. Savkin

Received: 2 September 2022
Accepted: 7 October 2022
Published: 13 October 2022

1. Introduction

In the last couple of decades, exhaustive research and development began in the field of unmanned aerial vehicles in search and rescue, surveillance, monitoring and mapping the environment, and delivery and transportation tasks [1,2]. Moreover, it is widely recognized both in nature and in robotic systems that the development of a task or mission in a cooperative way offers different advantages, among which are a reduction in time, a robustness or tolerance to failures since a member that presents problems can easily be replaced by some other agent [3,4]. In this sense, consensus methods for multiagent systems (MAS) have a extensive use in the field of intelligent autonomous vehicles, such as unmanned aircraft systems vehicles (UASs), ground vehicles, and sea applications. In [5], for UAV multiagent systems with external disturbances, a finite-time distributed-formation-tracking strategy was used. The authors in [6] applied in sliding mode a consensus algorithm to track the leader in multi-UAS systems and maintain the desired formation at the same time.

With the aim of solving the problem of formation control, more attention was given to the consensus control of multiagent systems, because of their extensive application in several areas [7,8]. Mainly, consensus control is divided into two approaches [9,10], the leaderless and the leader–follower consensus. Firstly, the consensus protocols were designed for linear systems, ranging from first-order integrator systems to higher-order integrator systems [11,12]. Because of the applicability of the consensus algorithm, further developments were conducted. The consensus strategy has been developed for nonlinear systems [10,13], with time-delay [14,15], with the input and actuator saturated [11,16], with sliding-mode techniques [17,18], as well as regarding algorithms dealing with external uncertainties within systems with unknown dynamics [19]. With the development of computational resources, strategies focused on machine learning have been proposed, where, based on the observations of both the states in multiagent systems and the environment, they have to improve both the performance and communication within the consensus algorithm [20,21].

An extended survey of consensus algorithms can be founded in [22,23].

In recent years, in order to solve the problem of lumped disturbances, i.e., unmodelled dynamics and external disturbances, several neural network nonlinear consensus approaches have been developed. Moreover, these protocols are efficient when the model parameters change while the system is working, as is the case with UAVs performing logistics tasks [24]. In this sense, further research on consensus protocols based on neural networks has been conducted in [25,26]. For nonlinear multiagent systems with uncertainties, a leader–follower consensus with state and output feedback was presented in [27], where the uncertain dynamics were approximated through neural networks. An adaptive neural consensus strategy for nonaffine nonlinear multiagent systems with uncertainties, unknown control directions, and subject to switching topologies was developed in [28], where a radial basis neural network was used to approximate the unknown dynamics. A neuronal dynamic surface control based on a predictor was developed for nonlinear systems subject to uncertain dynamics in [29]; in contrast to existing methodologies where the neural weights are updated with information from the tracking errors, a predictor was used and the prediction errors were employed in the adaptation law for the neural weights. In [30], a robust consensus strategy was developed for high-order nonlinear multiagent systems subject to unknown dynamics, affected by unknown actuator failures and unknown control gains. In this approach, a backstepping was combined with a neural network to ensure that the tracking error was restricted to a small region. In [31], an output-feedback formation tracking control strategy with modeling uncertainties subject to communication constraints was developed, where a radial basis neural network was used to identify the unknown dynamic.

In the aforementioned works, a major limitation is that the only signal available that reflects the difference between estimated parameters and the real parameters is the tracking or prediction error signal. To increase the robustness of the adaptive laws, the neural network consensus protocols were improved by introducing variants in the adaptive laws for the neural weights, such as the modification $-\sigma$ [32] and the modification $-e$ [33]. However, these algorithms induced a longer convergence time and affected the response of the tracking error. As stated in [34,35] for adaptive control systems, a fast and accurate neural network weight estimation convergence is valuable for the stability and robustness properties in closed-loop systems.

Since the optimal weights are unknown, the difference between the estimated weights and the optimal weights is not available for measurement. Based on the previous observations, an adaptive estimation strategy for unknown parameters was first introduced in [36], where, unlike conventional adaptation mechanisms, the adaptation of parameters included information on the estimation error of the parameter to increase the convergence of the tracking errors.

Motivated by the use of the parameter estimation error in the development of adaptive laws, further research was conducted by the scientific community. The development of robust adaptive controllers for the online estimation of the mass parameter for robotic systems was reported in [37]. In [38,39], an adaptive parameter estimation strategy was developed to estimate parameters in vehicle systems, mainly the road gradient and the mass of the vehicle. Moreover, an adaptive control strategy was developed for vehicles' active suspensions subject to unknown nonlinearities in [40]; in that approach, the authors proposed an adaptive law based on the parameter estimation error. This novel parameter estimation strategy was used in aerial systems as well. In [41], the inertia and mass parameters of a quadrotor aerial vehicle were estimated by introducing the parameter estimation error. Furthermore, following the same methodology, the inertia and mass parameters of a six-

DOF spacecraft were estimated in [42]. Recently, in addition to being used to estimate unknown parameters, this parameter estimation methodology was employed to estimate the optimal weights for systems based on a neural network [43]. In [44], an adaptive control strategy for robot manipulators was developed by combining a neural controller and a robust controller. The proposed scheme guaranteed that the estimated weights converged in finite time to the optimal weights. Furthermore, for robotic systems, in [45], a sliding mode controller enhanced with a neural network was developed. In that approach, an adaptive law for the neural weights was proposed, which was based on the estimation error. For a dual-arm robot system with dynamic uncertainties, an adaptive command-filtered control strategy was designed in [46]. Moreover, a radial basis neural network was used to approximate the uncertainties of the system, where the information of the estimated error was used in the adaptive law for the neural weights.

Main Contributions

We propose a robust, adaptive-consensus cooperative strategy for a group of quadrotor aerial vehicles with lumped disturbances (i.e., partially unknown nonlinearities and external disturbances). In order to compensate for the lumped disturbances, a radial basis neural network is introduced in the consensus protocol. To calculate the information of the weights' estimation error, a set of first-order intermediate variables are computed. Finally, the adaptive laws for the neural network weights incorporate the information of the estimation error. The main contributions of this research work are listed below:

- The development of a leader–follower consensus algorithm enhanced with a radial basis neural network for multiagent quadrotor unmanned aircraft systems subject to lumped disturbances is presented.
- The algorithm used to update the weights of the neural network is based on a first-order filter and auxiliary matrices to obtain the weight error information and use it in the adaptation law. To the best our knowledge, the consensus algorithms with neural network compensation reported until now have not taken into account the weight error information in the adaptation law.
- A stability proof based on Lyapunov's theory is developed for the consensus algorithm and neural network compensation developed, where the uniformly ultimate boundedness of the (i) formation errors and (ii) neural weights' estimation errors, is guaranteed.
- Simulation experiments for multiagent UAV systems are performed over the ROS platform [47] and ROS-based Gazebo environment.

The document is organized as follows: Some preliminaries and the dynamic model for the UAV are reviewed in Section 2. The consensus control strategy is introduced in Section 3. Section 4 provides the simulation results with examples in Gazebo, and Section 5 concludes this paper.

2. Preliminaries and Dynamic Model of the UAV

In this section, we introduce the graph theory necessary for the development of the consensus control strategy. After that, the general mathematical model for the quadrotor vehicle is presented. However, the quadrotor aerial vehicle simulated in Gazebo has an internal controller for the rotational dynamics. Simulations are conducted to identify the dynamics of the internal controller and finally a mathematical model that includes the internal controller dynamics is presented.

2.1. Graph Theory

Formally, the agents and a leader in a multiagent system are simply nodes, denoted by N, and the connectivity/structure among the agents is define by a communication graph $\mathcal{G} = \{\succeq, \digamma\}$ where $\succeq = \{n_1, \ldots, n_N\}$ is a node set and $\digamma = \{(n_i, n_j) \in \succeq \times \succeq$ is an edge set composed by the elements (n_i, n_j) that represent the communication link with i as the origin node and j as the final node. The so-called adjacency matrix is defined as

$\mathcal{A} = (a_{ij}) \in \mathbb{R}^{N\times N}$, where the element a_{ij} of $\mathcal{A}$ is selected as $a_{ij} = 1$ if node n_i is connected to node n_j and $a_{ij} = 0$ if is disconnected. A very important and practical feature is the degree of each node, which is simply the number of nodes that it is connected to; the in-degree matrix is $\mathcal{D} = diag(d_i) \in \mathbb{R}^{N\times N}$, it provides a single value of each node, with $d_i = \sum_{j\in\mathcal{N}_i} a_{ij}$. It is also used for the computation of the most important graph operator, the graph's Laplacian $\mathcal{L} = (l_{ij}) \in \mathbb{R}^{N\times N}$, defined as $\mathcal{L} = \mathcal{D} - \mathcal{A}$. If $a_{ij} = a_{ji}$, for $i, j = 1, \ldots, N$, then the communication graph $\mathcal{G}$ is defined as an undirected graph. If there is a path connecting every pair of nodes, then an undirected graph is connected. Furthermore, the leader's adjacency matrix is defined as $\mathcal{A}_0 = \mathrm{diag}(a_{i0}) \in \mathbb{R}^{N\times N}$, where $a_{i0} > 0$ implies that agent ith has access to the state of the leader agent and $a_{i0} = 0$ in another case. If the undirected graph is connected and at least one of diagonal entries of the leader's adjacency matrix is not zero, then the matrix $\mathcal{H}$ defined as $\mathcal{H} = \mathcal{L} + \mathcal{A}_0 \in \mathbb{R}^{N\times N}$ is symmetric and positive definite [48].

2.2. Dynamic Model of the Quadrotor UAV

A brief description of the mathematical model is shown below [49,50]. In Figure 1, we observe a three-dimensional UAV, where $\Gamma_B = [X_B, Y_B, Z_B]$ represents the body-fixed frame located in the center of gravity of the vehicle. The inertial frame is represented by $\Gamma_I = [X_I, Y_I, Z_I]$, and the position vector for the aerial vehicle in the inertial frame is denoted as $\xi = (x, y, z) \in \Gamma_I$. In this work, the rotation matrix used is given by

$$\mathbf{R}(\eta) = \begin{bmatrix} c\psi c\theta & c\psi s\theta s\phi - s\psi c\phi & c\psi s\theta c\phi + s\psi s\phi \\ s\psi c\theta & s\psi s\theta s\phi + c\psi c\phi & s\psi s\theta c\phi - c\psi s\phi \\ -s\theta & c\theta s\phi & c\theta c\phi \end{bmatrix} \tag{1}$$

where $\mathbf{R}(\eta) : \Gamma_I \to \Gamma_B$ and $\mathbf{R}(\eta) \in SO(3)$. ψ, θ, and ϕ are referred to as the Euler angles representing yaw, pitch, and roll, respectively. Moreover $s(\cdot)$ and $c(\cdot)$ stand for $\sin(\cdot)$ and $\cos(\cdot)$, respectively.

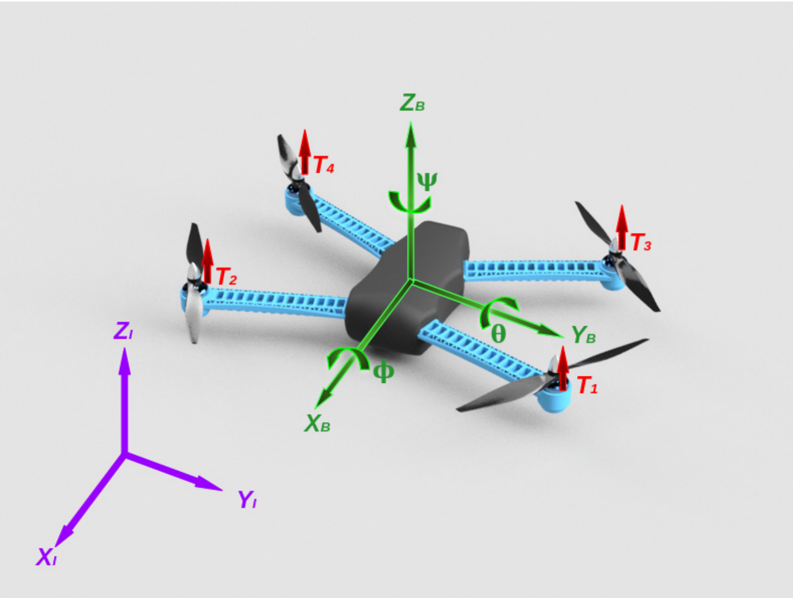


Figure 1. Structure of a quadrotor aerial vehicle with the body frame and inertial frame.

Let $\dot{\xi}$ and $\dot{\eta}$ be defined as the linear and angular velocities of the three-dimensional vehicle expressed in the inertial frame. Moreover, the linear and angular velocities represented in the body-fixed frame are denoted by $\dot{\xi}_b$ and $\dot{\eta}_b$, respectively. Then, $\eta = [\phi \quad \theta \quad \psi]^T \in \Gamma_I$ represents the vector of Euler angles, and $\dot{\eta}_b = [p \quad q \quad r]^T$, as the body angular velocities are related to $\dot{\eta} = \mathbf{W}\dot{\eta}_b$, where the matrix $\mathbf{W}$ is defined as

$$\mathbf{W} = \begin{bmatrix} 1 & s\phi \tan\theta & c\phi \tan\theta \\ 0 & c\phi & -s\phi \\ 0 & \frac{s\phi}{c\theta} & \frac{c\phi}{c\theta} \end{bmatrix} \tag{2}$$

Using the Newton–Euler equations, we can represent the equations of motion of the UAV in the body frame as follows

$$m\ddot{\xi} = \mathbf{R}(\eta)v_z F - mgv_z \tag{3a}$$
$$\mathbf{J}(\eta)\ddot{\eta} = \tau - \mathbf{C}(\eta, \dot{\eta})\dot{\eta} \tag{3b}$$

Here, $\mathbf{J}(\eta) = \mathbf{W}^\top \mathbf{I}\mathbf{W}$ acts as the inertia matrix, where $\mathbf{I} = diag(I_x, I_y, I_z)$ and $\mathbf{C}(\eta, \dot{\eta})\dot{\eta} = \left(\dot{\mathbf{J}} - \frac{\partial(\dot{\eta}_b^\top \mathbf{J})}{2\partial\eta}\right)\dot{\eta}_b$ represents the Coriolis term, $v_z = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^\top$ is a unitary vector along the z-axis in the inertial reference frame, F is the thrust force applied to the aerial vehicle, and $\tau \in \mathbb{R}^3$ denotes the roll, pitch, and yaw moments. The multirotor UAV model given previously is a highly coupled nonlinear system. Under some assumptions, it is possible to redefine the above model. Now, to simplify the moments τ, the auxiliary vector $\tilde{\tau}$ is defined as follows

$$\tilde{\tau} = \begin{bmatrix} u_2/I_x \\ u_3/I_y \\ u_4/I_z \end{bmatrix} = \mathbf{I}^{-1}\mathbf{W}^{-1}\left(-\mathbf{I}\dot{\mathbf{W}}\dot{\eta} - \mathbf{W}\dot{\eta} \times \mathbf{I}\mathbf{W}\dot{\eta} + \tau\right) \tag{4}$$

Using (3) and (4), the UAV dynamical model can be represented by

$$\ddot{x} = \frac{u_1}{m}(\cos\psi \sin\theta \cos\phi + \sin\psi \sin\phi) \tag{5a}$$
$$\ddot{y} = \frac{u_1}{m}(\sin\psi \sin\theta \cos\phi - \cos\psi \sin\phi) \tag{5b}$$
$$\ddot{z} = \frac{u_1}{m}(\cos\theta \cos\phi) - g \tag{5c}$$
$$\ddot{\theta} = \frac{u_2}{I_x} \tag{5d}$$
$$\ddot{\phi} = \frac{u_3}{I_y} \tag{5e}$$
$$\ddot{\psi} = \frac{u_4}{I_z} \tag{5f}$$

where $m \in \mathbb{R}$ represents the mass of the aerial vehicle and $g \in \mathbb{R}$ is the force due to the gravity, and $u_1 = \sum_{i=1}^4 \mathbf{T}_i$ is the thrust, which is generated by the rotors of the vehicle. For simplicity, the generalized torques are obtained as

$$\begin{bmatrix} u_2 \\ u_3 \\ u_4 \end{bmatrix} = \begin{bmatrix} -l\mathbf{T}_1 - l\mathbf{T}_2 + l\mathbf{T}_3 + l\mathbf{T}_4 \\ -l\mathbf{T}_1 + l\mathbf{T}_2 + l\mathbf{T}_3 - l\mathbf{T}_4 \\ -C_M\mathbf{T}_1 + C_M\mathbf{T}_2 - C_M\mathbf{T}_3 + C_M\mathbf{T}_4 \end{bmatrix} \tag{6}$$

where C_M is a constant defined from the rotor characteristics and l is the distance from the center of mass of the vehicle to the center of the rotor.

2.3. Modeling UAV Multirotor in Gazebo

The quadrotor available in Gazebo considers the presence of a low-level attitude controller provided by the PX4 autopilot. The internal controller stabilizes the angular dynamics for the roll, pitch, and yaw angles as well as the altitude in the z-axis direction. In a effort to consider the dynamics of the internal controller, we assumed that those dynamics were represented by the following differential equations

$$\ddot{z} = -a_1\dot{z} - a_2 z + a_3 u_z \tag{7a}$$
$$\ddot{\theta} = -b_1\dot{\theta} - b_2\theta + b_3 u_\theta \tag{7b}$$
$$\ddot{\phi} = -c_1\dot{\phi} - c_2\phi + c_3 u_\phi \tag{7c}$$
$$\ddot{\psi} = -d_1\dot{\psi} - d_2\psi + d_3 u_\psi \tag{7d}$$

where u_z, u_θ, u_ϕ, and u_ψ denotes the control inputs to the quadrotor aerial vehicle. In order to identify the parameters a_i, b_i, c_i, and d_i of Equation (7), a set of simulations was conducted in Gazebo by using step functions as control inputs and recording the input–output data. With the collected data, we used the System Identification Toolbox of MATLAB, which estimated the parameters of the dynamic model (7) by using the least squares method. Table 1 summarizes the parameters obtained.

Table 1. Parameters identified for the dynamics of the internal controller.

$a_1 = 0.1862$	$b_1 = 25.353$	$c_1 = 22.747$	$d_1 = 9.769$
$a_2 = 0.1093$	$b_2 = 179.523$	$c_2 = 170.635$	$d_2 = 29.289$
$a_3 = 1.2737$	$b_3 = 177.246$	$c_3 = 167.685$	$d_3 = 29.283$

The model was further validated by comparing the results obtained through simulations in Matlab with the data obtained from Gazebo. We can observe in Figure 2a the output from the experimental test for the z dynamics using as control input step function and the result for the transfer function. This procedure was similar within the attitude dynamics, where the control inputs of the *Iris drone* were related with the Euler angles as shown in Figures 2b and 3a,b.

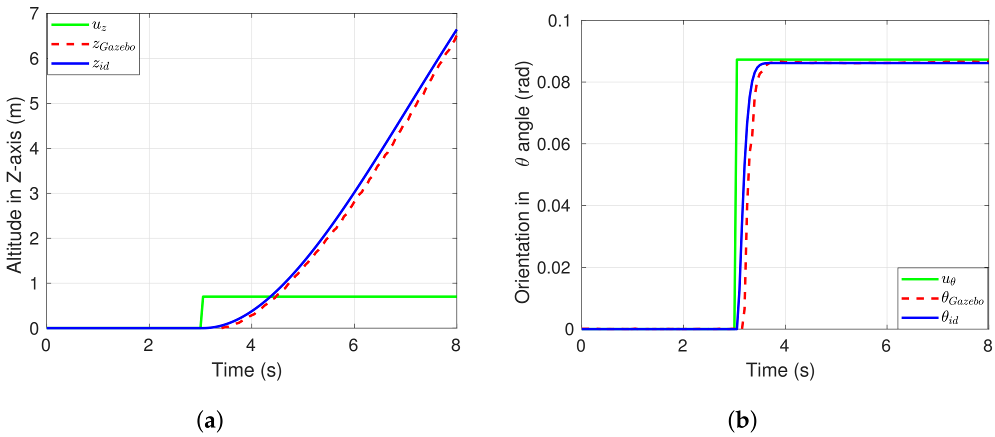


Figure 2. Experiments for the dynamics in Gazebo. (**a**) Response and identification for the dynamics of z within the Gazebo environment. (**b**) Response and identification for the dynamics of θ within the Gazebo environment.

From Equations (5c) and (7a), we could obtain for the control signal u_1 the following

$$u_1 = \frac{m(-a_1\dot{z} - a_2 z + a_3 u_z + g)}{\cos\theta\cos\phi} \tag{8}$$

Substituting (8) in (5a) and (5b) and combining with the dynamics identified, we obtained a complete model that took into account the dynamics of the internal controller, and was given as

$$\ddot{x} = f(z, u_z)(\cos\psi \tan\theta + \frac{\sin\psi}{\cos\theta} \tan\phi) \tag{9a}$$

$$\ddot{y} = f(z, u_z)(\sin\psi \tan\theta - \frac{\cos\psi}{\cos\theta} \tan\phi) \tag{9b}$$

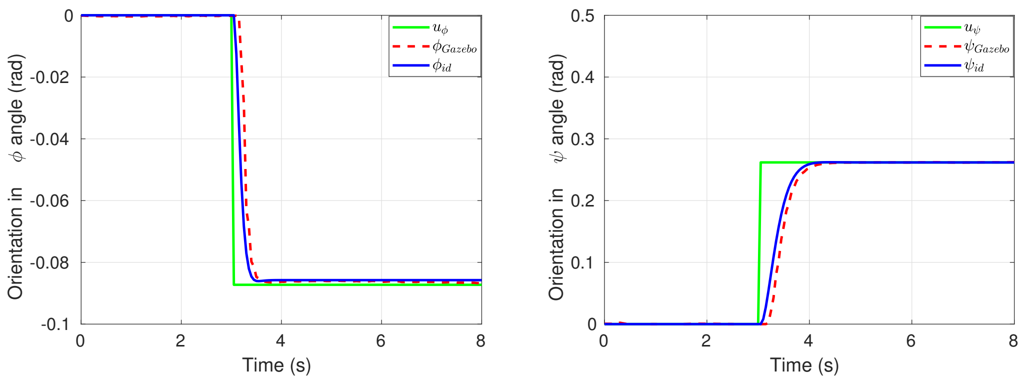
$$\ddot{z} = -a_1\dot{z} - a_2 z + a_3 u_z \tag{9c}$$

$$\ddot{\theta} = -b_1\dot{\theta} - b_2\theta + b_3 u_\theta \tag{9d}$$

$$\ddot{\phi} = -c_1\dot{\phi} - c_2\phi + c_3 u_\phi \tag{9e}$$

$$\ddot{\psi} = -d_1\dot{\psi} - d_2\psi + d_3 u_\psi \tag{9f}$$

where $f(z, u_z) = (-a_1\dot{z} - a_2 z + a_3 u_z + g)$.



(**a**) (**b**)

Figure 3. Experiments for attitude dynamics in Gazebo. (**a**) Response and identification for the dynamics of ϕ within the Gazebo environment; (**b**) Response and identification for the dynamics of ψ within the Gazebo environment.

3. Control Strategy Development

In this section, we introduce the development of the consensus cooperative strategy for the multiagent system. In view of the underactuated property of the quadrotor vehicle, we rewrite the mathematical model of the quadrotor vehicle presented in Equation (9). After that, the consensus strategy for the translational dynamics of the team of quadrotor aerial vehicles is developed.

3.1. Mathematical Model for the Consensus Development

Before the development of the consensus control strategy for the multiagent quadrotor system, we rewrite Equation (9) to obtain a general equation to be used in the development of the consensus control strategy. From (9a) and (9b), we can define the virtual control inputs u_x and u_y as follows

$$\begin{bmatrix} u_x \\ u_y \end{bmatrix} = (-a_1\dot{z} - a_2 z + a_3 u_z + g)\begin{bmatrix} \tan\theta & \frac{\tan\phi}{\cos\theta} \\ -\frac{\tan\phi}{\cos\theta} & \tan\theta \end{bmatrix}\begin{bmatrix} \cos\psi \\ \sin\psi \end{bmatrix} \tag{10}$$

where u_x and u_y can be translated to the desired roll and pitch angles. Therefore, in this matrix representation, we can obtain the desired references angles ϕ_d and θ_d that enable the system to follow the desired positions x_d and y_d; the desired angles are given as

$$\phi_d = \arctan\left[\frac{(u_x\cos\psi + u_y\sin\psi)}{f(z, u_z)}\right] \tag{11a}$$

$$\theta_d = \arctan\left[-\frac{(u_x\sin\psi + u_y\cos\psi)\cos\phi_d}{f(z, u_z)}\right] \tag{11b}$$

We assume that the Euler angles (roll, pitch, and yaw) are bounded in the following ranges: $(-\pi/2 < \theta < \pi/2)$, $(-\pi/2 < \phi < \pi/2)$ and $(-\pi < \psi < \pi)$, respectively.

With the definitions of u_x and u_y, the dynamics of translation for each quadrotor vehicle are given by

$$\ddot{x} = u_x \tag{12a}$$
$$\ddot{y} = u_y \tag{12b}$$
$$\ddot{z} = -a_1\dot{z} - a_2 z + a_3 u_z \tag{12c}$$

The control inputs u_x, u_y, and u_z are computed for each vehicle through the consensus control strategy developed in Section 3.2. Once u_x and u_y are calculated, the desired angles ϕ_d and θ_d are obtained by using (11). Moreover, we assume that the yaw reference angle is set to zero ($\psi_d = 0$). Finally, the desired angles ϕ_d, θ_d and ψ_d are the inputs for the internal controller of each aerial vehicle in the ROS environment.

3.2. Consensus Control Strategy

Now, the consensus cooperative strategy enhanced by a radial basis neural network is developed. First, it is developed for the altitude subsystem given in Equation (12c). For the horizontal plane, given by Equations (12a) and (12b), it is worth mentioning that the development of the cooperative strategy follows the same structure as that for the altitude subsystem.

3.2.1. Consensus Control Strategy for Altitude Dynamics

The control strategy designed assumes that the aforementioned low-level altitude controller has as its input a velocity command given by u_z. Therefore, we need to define the altitude controller such that it converges to the altitude desired and it has the capability to counteract the unknown disturbances. For this reason, we need to begin with the vertical dynamics given in (12c), which can be rewritten for each agent as

$$\dot{z}_i = A_z z_i + B_z(u_{z_i} + d_{z_i}), \quad i = 1, \cdots, N, \tag{13}$$

where $z_i = [z_{1_i} \quad z_{2_i}]^\top$, $A_z = \begin{bmatrix} 0 & 1 \\ -a_2 & -a_1 \end{bmatrix}$, $B_z = \begin{bmatrix} 0 \\ a_3 \end{bmatrix}$, and d_{z_i} denotes external uncertainties. Moreover, we introduce the leader's dynamics as follows

$$\dot{z}_0 = A_z z_0 + B_z r(z_0, t) \tag{14}$$

where $r(z_0, t)$ is a bounded input for the leader agent. The following assumption is stated in order to deal with the external disturbances d_{z_i}

Assumption 1. *The matched uncertainty d_{iz} for the altitude dynamics defined in (13) is approximated by a radial basis neural network as*

$$d_{z_i} = W_{z_i}^{*\top} \Psi_{z_i}(z_i) + \epsilon_{z_i} \quad \forall z_i \in \mathbf{D} \tag{15}$$

where $W_{z_i}^* \in \mathbb{R}^s$ is an unknown constant optimal weight vector. $\Psi_{z_i}(\cdot) : \mathbb{R}^n \to \mathbb{R}^s$ represents a vector function as $\Psi_{z_i}(z_i) = [\Psi_{1_i}(z_i), \Psi_{2_i}(z_i), \ldots, \Psi_{s_i}(z_i)]^T$ where $\Psi_i(x)$ is a radial basis activation function, which can be represented as: if $\Psi_i(x) = exp(-\frac{||x-c||^2}{2b^2})$, $i = 1, 2, \ldots, N$, where the center and width of the Gaussian functions are represented by $c \in \mathbf{R}^r$ and b, respectively. Moreover, the activation function satisfies $||\Psi_{z_i}|| \leq \Psi_{M_i}$ with $\Psi_{M_i} > 0$, the approximation error ϵ_{z_i} fulfills $|\epsilon_{z_i}| \leq \epsilon_{z_i}^+$ where $\epsilon_{z_i}^+ > 0$, and $\mathbf{D} \subset \mathbb{R}^n$ is a sufficiently large domain [29].

Now, we define the tracking error for each follower agent as $\delta_{z_i} = z_i - z_0$, whose time derivative is along (13), (14), and using the approximation (15), we obtain

$$\begin{aligned}\dot{\delta}_{z_i} &= A_z z_i + B_z(u_{z_i} + W_{z_i}^{*\top}\Psi_i(z_i) + \epsilon_{z_i}) - A_z z_0 - B_z r \\ &= A_z\delta_{z_i} + B_z(u_{z_i} + W_{z_i}^{*\top}\Psi_i(z_i) + \epsilon_{z_i} - r)\end{aligned} \tag{16}$$

With the aim to develop the adaptation procedure for the neural weights $\hat{W}_{z_i}$, inspired by the approach presented in for single systems, a set of first-order filters is introduced. It is worth mentioning that with this approach, we can use information about the weight error $\tilde{W}_{z_i}$, in contrast to classical adaptation mechanisms. To conduct this strategy, the tracking error dynamics (16) are parameterized as

$$\begin{aligned}\dot{\delta}_{z_i} &= A_z\delta_{z_i} + B_z u_{z_i} + B_z\Psi_i(z_i)^T W_{z_i}^* + B_z(\epsilon_{z_i} - r) \\ &= \varphi_{z_i}(\delta_{z_i}, u_{z_i}) + \Phi_{z_i}W_{z_i}^* + \varepsilon_{z_i}\end{aligned} \tag{17}$$

where

$$\varphi_{z_i}(\delta_{z_i}, u_{z_i}) = A_z\delta_{z_i} + B_z u_{z_i}, \quad \Phi_{z_i} = B_z\Psi_i(z_i)^T, \quad \varepsilon_{z_i} = B_z(\epsilon_{z_i} - r), \tag{18}$$

with $\varphi_{z_i}(\delta_{z_i}, u_{z_i}) \in \mathbb{R}^2$ and $\Phi_{z_i} \in \mathbb{R}^{2\times s}$ is defined as the "regressor" matrix.

In this sense, the filtered variables δ_{z_if} ,φ_{z_if} and Φ_{z_if} are defined as

$$\begin{aligned}k\dot{\delta}_{z_if} + \delta_{z_if} &= \delta_{z_i}, \quad \delta_{z_if}(0) = 0 \\ k\dot{\Phi}_{z_if} + \Phi_{z_if} &= \Phi_{z_i}, \quad \Phi_{z_if}(0) = 0 \\ k\dot{\varphi}_{z_if} + \varphi_{z_if} &= \varphi_{z_i}, \quad \varphi_{z_if}(0) = 0 \\ k\dot{\varepsilon}_{z_if} + \varepsilon_{z_if} &= \varepsilon_{z_i}, \quad \varepsilon_{z_if}(0) = 0\end{aligned} \tag{19}$$

where $k > 0$ is a filter parameter. In order to obtain from (17) and (19) that

$$\dot{\delta}_{z_if} = \frac{\delta_{z_i} - \delta_{z_if}}{k} - \varphi_{z_if} = \Phi_{z_if}W_{z_i}^* + \varepsilon_{z_if}. \tag{20}$$

let us define two auxiliary matrices $P_{z_i} \in \mathbb{R}^{s\times s}$ and $Q_{z_i} \in \mathbb{R}^s$ as

$$\begin{aligned}\dot{P}_{z_i} &= -lP_{z_i} + \Phi_{z_if}^T\Phi_{z_if}, & P_{z_i}(0) = 0 \\ \dot{Q}_{z_i} &= -lQ_{z_i} + \Phi_{zf}^T[\frac{\delta_{z_i} - \delta_{z_if}}{k} - \varphi_{z_if}], & Q_{z_i}(0) = 0\end{aligned} \tag{21}$$

where the positive scalar l is a parameter to be designed. The solution of (21) is derived as

$$\begin{aligned}P_{z_i} &= \int_0^t e^{-l(t-r)}\Phi_{z_if}^T(r)\Phi_{z_if}(r)dr \\ Q_{z_i} &= \int_0^t e^{-l(t-r)}\Phi_{z_if}^T(r)\left[\frac{\delta_{z_i}(r) - \delta_{z_if}(r)}{k} - \varphi_{z_if}(r)\right]dr\end{aligned} \tag{22}$$

Lemma 1. *The auxiliary matrix P_{z_i} defined in (22) is positive definite, $\lambda_{\min}(P_{z_i}) > \sigma_\Phi > 0$, if the regressor Φ_{z_i} presented in (18) fulfills the PE condition [45]. A function Φ_{z_i} satisfies the PE condition if there exist positive constants T and γ satisfying the condition $\int_t^{t+T}\Phi_{z_i}(r)^T\Phi_{z_i}(r)dr \geq \gamma I$, $\forall t \geq 0$.*

Notice that from (20) and (22), we can rewrite Q_{z_i} as

$$Q_{z_i} = P_{z_i}W_{z_i}^* - \varrho_{z_i} \tag{23}$$

where $\varrho_{z_i} = -\int_0^t e^{-l(t-r)}\Phi_{z_if}(r)\varepsilon_{z_if}(r)dr \in \mathbb{R}^s$. As ε_{z_i} and the regressor Φ_{z_i} are bounded, we obtain that the filtered variables Φ_{z_if} and ε_{z_if} are also bounded. Therefore, ϱ_{z_i} is bounded by $\varepsilon^+_{z_if} > 0$.

Based on the solutions of P_{z_i} and Q_{z_i}, we define an auxiliary vector H_{z_i} as:

$$H_{z_i} = P_{z_i}\hat{W}_{z_i} - Q_{z_i} \tag{24}$$

Then, based on (22) and the parameter estimation error $\tilde{W}_{z_i} = W^*_{z_i} - \hat{W}_{z_i}$, we can verify that

$$\begin{aligned} H_{z_i} &= P_{z_i}\hat{W}_{z_i} - Q_{z_i} \\ &= P_{z_i}\hat{W}_{z_i} - P_{z_i}W^*_{z_i} + \varrho_{z_i} \\ &= -P_{z_i}\tilde{W}_{z_i} + \varrho_{z_i}. \end{aligned} \tag{25}$$

To guarantee the stability of the tracking error δ_{z_i}, we design the following distributed control law with augmented adaptive control term

$$u_{z_i} = c_z K_z e_{z_i} - \hat{W}^T_{z_i}\Psi_{z_i}(z_i), \tag{26}$$

where $\hat{W}_{z_i}$ is an estimate of the unknown weights $W^*_{z_i}$, the scalar c_z is a coupling gain, and e_{z_i} is a neighborhood error represented as

$$e_{z_i} = \sum_{j\in N} a_{ij}(z_i - z_j) + a_{i0}(z_i - z_0), \tag{27}$$

Define $K_z \in \mathbb{R}^{1\times 2}$ as $K_z = -B_z^T M_z$, where M_z is the solution of the Riccati inequality defined as

$$A_z^T M_z + M_z A_z + N_z - M_z B_z B_z^T M_z \leq 0, \tag{28}$$

Notice that M_z and $N_z \in \mathbb{R}^{2\times 2}$ are positive definite matrices. If the consensus control law (26) is substituted into (16), we arrive at

$$\begin{aligned} \dot{\delta}_{z_i} &= A_z\delta_{z_i} + B_z(c_zK_ze_{z_i} - \hat{W}^T_{z_i}\Psi_{z_i}(z_i) + W^{*\top}_{z_i}\Psi_{z_i}(z_i) + \epsilon_{z_i} - r) \\ &= A_z\delta_{z_i} + B_z(c_zK_ze_{z_i} + \tilde{W}^\top_{z_i}\Psi_{z_i}(z_i) + \epsilon_{z_i} - r). \end{aligned} \tag{29}$$

It should be noted from (25) that the information about the weights' estimation error $\tilde{W}_{z_i}$ is contained in H_{z_i}. Then, the adaptation law, for the weights of the ith follower agent in the altitude dynamics is proposed as

$$\dot{\hat{W}}_{z_i} = \Gamma_{z_i}(\Psi_{z_i}(z_i)e^T_{z_i}M_zB_z - k_{z_i}H_{z_i}). \tag{30}$$

where Γ_{z_i} and k_{z_i} are positive constants. Then, we state the theorem for the consensus cooperative control for the altitude dynamics of the quadrotor multiagent system as follows.

Theorem 1. *Consider the altitude dynamics for the ith follower agent given in Equation (13). If the regressor Φ_{z_i} defined in (18) fulfills the PE condition, the cooperative control strategy u_{z_i} is proposed as in Equation (26), the gain K_z is obtained from solving the Riccati Equation (28), the coupling gain c_z fulfills condition (58), and the neural weights are updated by the adaptive law given by Equation (30), where the auxiliary matrix H_{z_i} is defined in Equation (24). Then, the altitude tracking error δ_{z_i} and the estimation error of the neural weights $\tilde{W}_{z_i}$ for the ith agent converge to a bounded region defined by (66).*

Proof. The proof of this Theorem is presented in Section 6. □

3.2.2. Consensus Control Strategy for Position Control in the Horizontal Plane

To obtain the consensus control strategy for the X–Y plane, it is necessary to carry out the development of a consensus strategy for each axis. In this sense, we present the development for the X axis and notice that for the Y axis the procedure is similar. Consider the dynamics for the X axis presented in Equation (12a), which is rewritten as

$$\dot{x}_i = A_x x_i + B_x(u_{x_i} + d_{x_i}) \tag{31}$$

where $x_i = \begin{bmatrix} x_{1_i} & x_{2_i} \end{bmatrix}^\top$, $A_x = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$, $B_x = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$, and d_{x_i} denotes the external uncertainties. Moreover, the leader dynamics are given as

$$\dot{x}_0 = A_x x_0 + B_x r_x(x_0, t). \tag{32}$$

Let us define the tracking errors for the ith agent in the X axis as $\delta_{x_i} = x_i - x_0$. The distributed control strategy is proposed as

$$u_{x_i} = c_x K_x e_{x_i} - \hat{W}_{x_i}^T \Psi_{x_i}(x_i), \tag{33}$$

where c_x is a scalar coupling gain, $\hat{W}_{x_i}$ is an estimate of the unknown weights $W_{x_i}^*$, and the neighborhood synchronization error in the X axis is defined as

$$e_{x_i} = \sum_{j \in N} a_{ij}(x_i - x_j) + a_{i0}(x_i - x_0) \tag{34}$$

Similar to Equation (30), the updated law for the weights of the neural network is given by

$$\dot{\hat{W}}_{x_i} = \Gamma_{x_i}(\Psi_{x_i}(x_i) e_{x_i}^T M_x B_x - k_x H_{x_i}), \tag{35}$$

where we can observe that the knowledge of the estimation error of the neural weights is incorporated in the term H_{x_i}, which is defined as

$$H_{x_i} = P_{x_i} \hat{W}_{x_i} - Q_{x_i} \tag{36}$$

The auxiliary filtered matrices P_{x_i} and Q_{x_i} are defined as in (21). The filtered variables $\delta_{x_i f}$, $\varphi_{x_i f}$, and $\Phi_{x_i f}$ are defined as in Equations (18) and (19). For the purpose of guaranteeing the convergence of the tracking errors δ_{x_i} and the weights' estimation error $\tilde{W}_{x_i}$, Theorem 1 can be applied for the consensus control strategy in the X and Y axes.

4. Simulation Results

The simulation results for the robust cooperative control strategy developed are presented in this section. First, we introduce the simulation environment, where the required steps to carry out the simulations trials in ROS and Gazebo are summarized in Algorithm 1. After that, we present two simulations scenarios. The first one is for the altitude control of a team of four quadrotor aerial vehicles, with a comparative study with a classical distributed control and adaptive distributed control. In the second one, the simulation results for the translational cooperative control in the plane X–Y are presented.

4.1. Simulation Environment

In this section, we present the implementation of the algorithm within the Gazebo robotics simulator, it allows the testing of the behavior of multirotor vehicles without the need to use physical systems, due to the complexity of the tests and the fact it requires a greater number of components due to the number of vehicles that we defined as follower agents in this simulator. The environment can be constituted from the tools and packages developed by *PX4* [51] , which supports the connection with an embedded computer through a standardized protocol called *Mavlink*. We can observe that in the following Algorithm 1, we have to set the windy environment with the *gazebo plugin* that introduces external disturbances, followed by the number of agents to be used, then the topics and services are created for each vehicle. The Algorithm 1 summarizes the program developed.

Algorithm 1 Algorithm to set the multiagent system tests.

Step 1 Download and install the PX4-Autopilot package from its repository.
Step 2 Compile and run the iris model in the "empty" world to *Gazebo*.
Step 3 Add the path to source the environment.
Step 4 Run the launch file multi_uav_mavros_sitl.launch. ; `/* Modify the number of agents and use the windy.world file */`
Step 5 Define the communication matrix and set the initial states and the bounded reference for the virtual leader agent.
Step 6 Define the execution time: $exec\_time = 90$ seconds.
Step 7 Run the main node.
Step 8 **if** $time \leq 10$ **then**
  Set the agents in the initial states ; `/* Use the internal simple PID control law. */`
  if $time > 10$ & $time \leq exec\_time$ **then**
    Compute the neighbor error and send the information to each agent program;
    Get the input control signal and the approximation function from the neural network;
    Publish the robust neural network consensus;
    Save the states and control signals for each UAV agent in a .csv file;
  end
else
  Set the autolanding mode for all UAV agents
end

4.2. Simulation Scenarios

In these simulation results, we considered the configuration of the multiagent system with $N = 4$ multirotor vehicles. Each aircraft had a mass m = 1.5 kg and the communication graphs utilized in the simulations are represented in Figure 4.

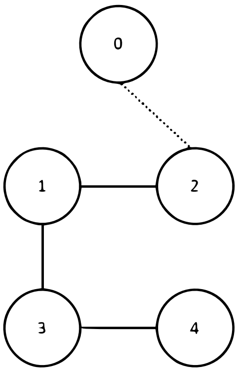


Figure 4. Communication graph proposed for the multiagent systems.

For the communication graph of Figure 4, the adjacency and in-degree matrices were defined as

$$\mathcal{A} = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad \mathcal{D} = \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{37}$$

Moreover, the Laplacian and leader's adjacency matrices were defined as

$$\mathcal{A}_0 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad \mathcal{L} = \begin{bmatrix} 2 & -1 & -1 & 0 \\ -1 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & 0 & -1 & 1 \end{bmatrix} \tag{38}$$

In order to show the performance of our proposed approach, we present a comparative analysis with two other consensus control strategies presented in the literature.

- Classical nominal distributed control. In this approach, a static control strategy is used, given by
$$u_{z_i} = c_z K_z e_{z_i}, \tag{39}$$
which uses only the coupling gain c_z, the feedback gain K_z, and the neighborhood error e_{z_i} [52].
- Adaptive neural network consensus control. In this strategy, a radial basis neural network is employed to compensate the lumped disturbances. The consensus cooperative strategy and the update law for the neural weights are presented below, where the weights are updated using the conventional method [53]
$$\begin{aligned} u_{z_i} &= c_z K_z e_{z_i} - \hat{W}_{z_i}^T \Psi_{z_i}(z_i) \\ \dot{\hat{W}}_{z_i} &= \Gamma_{z_i}(\Psi_{z_i}(z_i) e_{z_i}^T M_z B_z - k_{z_i} \hat{W}_{z_i}) \end{aligned} \tag{40}$$
In comparison with our approach, notice that for the neural weights, the adaptation law (40) includes the estimation of the weights given by the term $\hat{W}_{z_i}$.

4.2.1. Altitude Control

The mathematical model for the altitude dynamic of the ith agent can be written as follows
$$\dot{z}_i = \begin{bmatrix} 0 & 1 \\ -0.1093 & -0.1862 \end{bmatrix} z_i + \begin{bmatrix} 0 \\ 1.2737 \end{bmatrix} (u_{z_i} + d_{z_i}) \tag{41}$$
For the leader agent, we selected a bounded input given as $r(t) = 6 - 3\cos\left(\frac{t}{3.5}\right)$.

Table 2 presents the parameters used for this simulation scenario, including the control gains, the gains of the robust estimator, and the gains for updating the neural weights.

Table 2. Parameters and initial conditions used in the simulation results.

Parameter	Value	Parameter	Value
c_z	2.7	k_{wz}	0.17
K_z	$[-0.75 \quad -0.35]$	$z_1(0)$	$[3.0 \quad 0.0]^\top$
λ	13	$z_2(0)$	$[2.7 \quad 0.0]^\top$
Γ	0.13	$z_3(0)$	$[2.5 \quad 0.0]^\top$
k_{fz}	0.3	$z_4(0)$	$[3.7 \quad 0.0]^\top$

A radial-based neural network was used, where the output of this neural network is given in Equation (15). The neural network proposed had 25 neurons, where the centers of the neural weights were chosen as
$$cen_z = \begin{bmatrix} 2 & 2 & 2 & 2 & 2 & 3.5 & 3.5 & 3.5 & 3.5 & 3.5 & 5 & 5 & 5 & 5 & 5 & 7.5 & 7.5 & 7.5 & 7.5 & 7.5 & 9 & 9 & 9 & 9 & 9 \\ -0.6 & -0.3 & 0 & 0.3 & 0.6 & -0.6 & -0.3 & 0 & 0.3 & 0.6 & -0.6 & -0.3 & 0 & 0.3 & 0.6 & -0.6 & -0.3 & 0 & 0.3 & 0.6 & -0.6 & -0.3 & 0 & 0.3 & 0.6 \end{bmatrix} \tag{42}$$
with $b = 0.7$ as the width of the hidden neuron. Furthermore, the initial weights of the network were selected randomly.

As we can see in Figure 5, a classic controller was used that was stationary so the tracking error performed by the agents was constant at 20 and 60 s. Furthermore, in the zoomed rectangle, we can see that the vehicles oscillated more in comparison to the reference virtual agent. In the second scenario in Figure 6, we can see the effects of the adaptive term as the tracking error began to decrease. However, the update law to estimate the parameters generated oscillations, specifically in the lower part, at 40 and 80 s. The robust estimator was introduced in the third scenario and we can observe in Figure 7 how the tracking error converged and the oscillations were reduced. In this work, to verify the effectiveness of the strategy proposed, we used two performance indices based on the tracking error between each one of the follower agents and the leader agent.

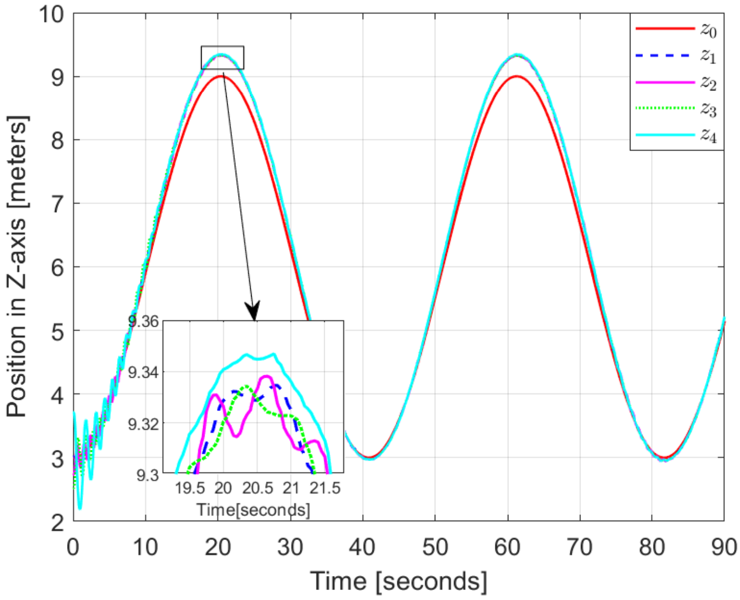


Figure 5. Consensus with a traditional nominal distributed controller for 4 agents z_n tracking a virtual leader z_0.

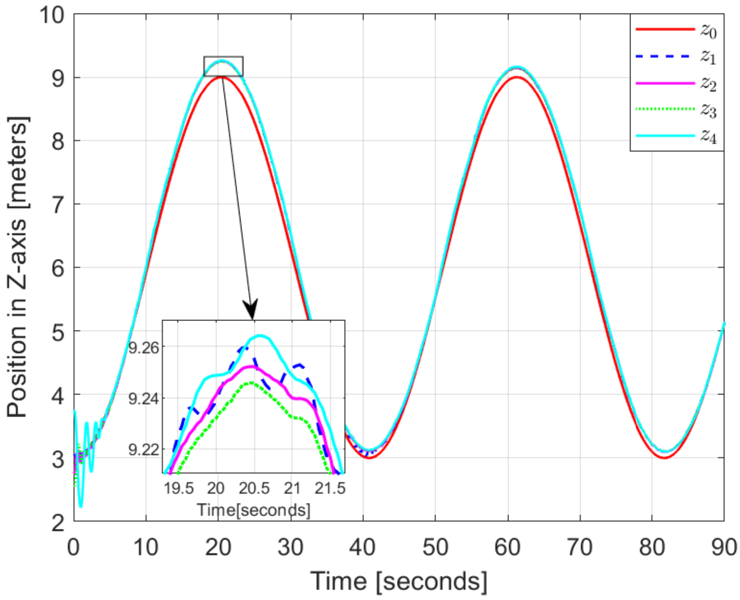


Figure 6. Consensus with adaptive term for 4 agents z_n tracking a virtual leader z_0.

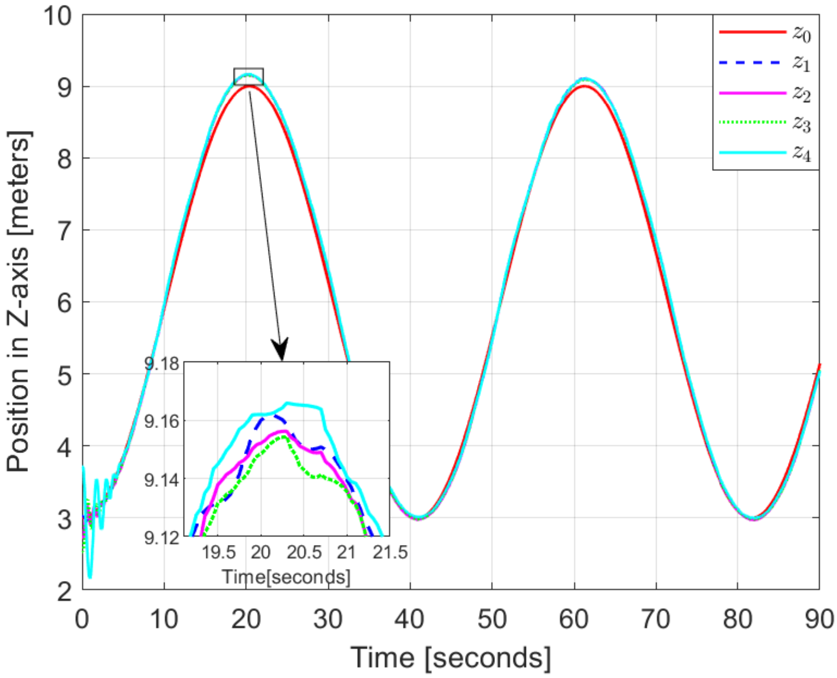


Figure 7. Consensus with our algorithm proposed for 4 agents z_n tracking a virtual leader z_0.

The indices used were the integral-time absolute error *(ITAE)* and integral-time square error *(ITSE)*. The performance values are shown in Table 3 where as can be seen, the values in each index decrease and improve in our approach compared with the other methods.

Table 3. Table of comparison of the error-integral performance indexes.

Performance Index List			
Controller	**UAV**	**ITAE**	**ITSE**
Classical nominal distributed control	Agent 1	0.568	0.141
	Agent 2	0.567	0.139
	Agent 3	0.575	0.143
	Agent 4	0.598	0.157
Classical nominal distributed control with adaptive term	Agent 1	0.465	0.083
	Agent 2	0.463	0.081
	Agent 3	0.467	0.085
	Agent 4	0.494	0.091
Our proposed algorithm	Agent 1	0.259	0.025
	Agent 2	0.251	0.023
	Agent 3	0.263	0.027
	Agent 4	0.267	0.029

4.2.2. Translational Cooperative Control

The dynamic model in the horizontal plane formed by the X and Y axes can be written as follows:

$$\begin{aligned} \dot{x}_i &= \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} x_i + \begin{bmatrix} 0 \\ 1 \end{bmatrix} (u_{x_i} + d_{x_i}) \\ \dot{y}_i &= \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} y_i + \begin{bmatrix} 0 \\ 1 \end{bmatrix} (u_{y_i} + d_{y_i}) \end{aligned} \tag{43}$$

It is worth mentioning that from the consensus control signals u_{x_i} and u_{y_i}, the desired reference angles ϕ_{d_i} and θ_{d_i} were obtained and used as control inputs for the quadrotor vehicles.

In order to follow a circular trajectory, the reference for the leader agent in the X and Y axes were given as

$$\begin{aligned} r_x(t) &= (-3\cos\left(\frac{t\pi}{20}\right) + 3) - 3 \\ r_y(t) &= -3\sin\left(\frac{\pi}{20}t\right) \end{aligned} \tag{44}$$

The number of neurons was similar to the neural network for the height dynamics and the width $b = 0.7$, and the centers of the neural weights were

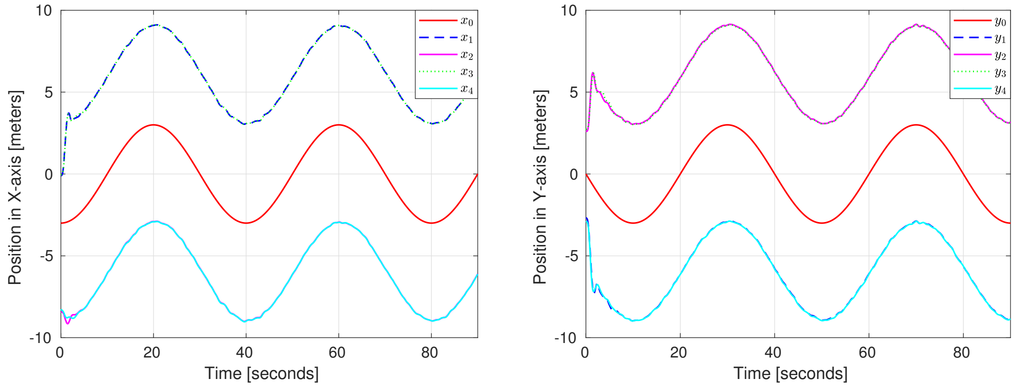
$$cen_x = \begin{bmatrix} -6 & -6 & -6 & -6 & -6 & -3 & -3 & -3 & -3 & -3 & 0 & 0 & 0 & 0 & 0 & 3 & 3 & 3 & 3 & 3 & 6 & 6 & 6 & 6 & 6 \\ -0.6 & -0.3 & 0 & 0.3 & 0.6 & -0.6 & -0.3 & 0 & 0.3 & 0.6 & -0.6 & -0.3 & 0 & 0.3 & 0.6 & -0.6 & -0.3 & 0 & 0.3 & 0.6 & -0.6 & -0.3 & 0 & 0.3 & 0.6 \end{bmatrix} \tag{45}$$

The parameters and the initial conditions changed the effect, when analyzing the performance of the proposed algorithm as we can observe in Table 4. Note that the centroids and parameters described in Table 4 for the X-axis dynamics were the same for the Y-axis dynamics of the system.

Table 4. Parameters and initial conditions used in the horizontal example.

Parameter	Value	Parameter	Value
c_x	30	Γ_x	0.3
K_x	$[-0.2 \quad -0.01]$	λ_x	13
$k_{f(x)}$	0.3	$k_{w(x)}$	2.9
$x_1(0)$	$[0.0 \quad 0.0]^\top$	$x_3(0)$	$[0.0 \quad 0.0]^\top$
$x_2(0)$	$[-9.0 \quad 0.0]^\top$	$x_4(0)$	$[-9.0 \quad 0.0]^\top$
$y_1(0)$	$[-3.0 \quad 0.0]^\top$	$y_3(0)$	$[3.0 \quad 0.0]^\top$
$y_2(0)$	$[-3.0 \quad 0.0]^\top$	$y_4(0)$	$[3.0 \quad 0.0]^\top$

In Figure 8, we present the results of the simulations with the conditions mentioned above. The reference of the virtual leader agent was a circular path in the X–Y plane centered on $[0,0]$ and the circular references for each vehicle maintained an offset of their centers of ± 6 m. Last, in Figure 8a,b, we can observe the performance of the position of the UAVs in the X and Y axes, respectively, where we can see two vehicles following the circular trajectory in their respective offsets despite having a wind disturbance $d_{x,y} = 1\frac{m}{s}$ with a positive direction on the horizontal plane X-Y. In Figure 9, we show the point of view of the trajectory of the follower agents, where we can observe how the tracking error converges; a video showing the simulations in the Gazebo environment was shared at https://youtu.be/i6Qp5MMB2Ws (accessed in 22 June 2022).



(**a**) (**b**)

Figure 8. Time development of the positions of the follower agents tracking the circular path. (**a**) Consensus performance in the X-axis. (**b**) Consensus performance in the Y-axis.

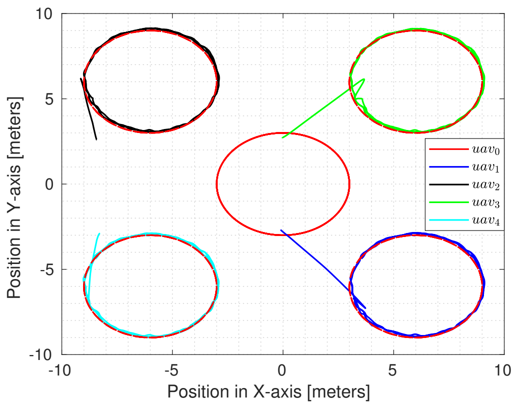


Figure 9. Consensus performance in horizontal plane X–Y.

5. Conclusions

A consensus protocol with a robust neural network estimator for the unknown elements and flock motion was proposed for multi-UAV systems to reduce the tracking error subject to unknown external perturbations and unmodeled dynamics. The closed-loop stability for the whole system was verified through Lyapunov's theory, arriving at a result of a uniformly ultimately bounded stability, guaranteeing that the estimation and tracking errors were bounded. The consensus strategy and the update law for the neuronal weights for each vehicle were obtained through the stability proof as well. In addition, the proposed consensus protocol combined with the neural network allowed us to estimate and compensate the unmodeled dynamics and external perturbations. To demonstrate the performance of the proposed method, simulations were carried out in a virtual environment, where the method was also compared against a nominal distributed controller as well as against a consensus protocol in combination with an adaptive strategy where a traditional estimation method was used and was verified with two performance indexes to measure the tracking error of the multiagent system. The simulation results showed that the update of the neuronal weights with the first-order filters and auxiliary matrices was more robust against unmodeled dynamics, parameter uncertainties, and external disturbances. Future lines of research will consider the implementation of the cooperative strategy developed with multiple vehicles to observe the performance in real time and under external disturbances. Moreover, the strategy proposed can be extended to deal with transportation tasks with multiple vehicles.

6. Stability Proof

Proof. In order to conduct the stability proof, we define the following variables, $\delta_z = [\delta_{z_1}^T, \ldots, \delta_{z_N}^T]^T \in \mathbb{R}^{2N}$, $e_z = [e_{z_1}^T, \ldots, e_{z_N}^T]^T \in \mathbb{R}^{2N}$, $\epsilon_z = [\epsilon_{z_1}, \ldots, \epsilon_{z_N}]^T \in \mathbb{R}^N$, $\Psi_z = [\Psi_{z_1}^T(x_1), \ldots, \Psi_{z_N}^T(x_N)]^T \in \mathbb{R}^{Ns}$, $\tilde{W}_z = diag(\tilde{W}_{z_i}) \in \mathbb{R}^{Ns \times N}$, and $\underline{r_z} = [r_z, \ldots, r_z]^T \in \mathbb{R}^N$. From the previous definitions and the tracking error dynamics for the ith agent given in Equation (16), the dynamics of the global tracking error δ_z can be written as

$$\dot{\delta}_z = \Big(I_N \otimes A_z + c_z \mathcal{H} \otimes B_z K_z\Big)\delta_z + \Big(I_N \otimes B_z\Big)\Big(\tilde{W}_{z_i}^T \Psi_z(z) + \epsilon_z - \underline{r_z}\Big). \tag{46}$$

Now, consider the candidate Lyapunov function defined as

$$V_z = \frac{1}{2}\delta_z^T\Big(\mathcal{H} \otimes M_z\Big)\delta_z + \frac{1}{2}\mathrm{tr}\Big\{\tilde{W}_z^T \Gamma_z^{-1} \tilde{W}_z\Big\} \tag{47}$$

where $\Gamma_z = \mathrm{diag}(\bar{\Gamma}_{z_i}) \in \mathbb{R}^{Ns \times Ns}$, $\bar{\Gamma}_{z_i} = \Gamma_{z_i} I_{x \times s} \in \mathbb{R}^{s \times s}$, and $\Gamma_{z_i} > 0,\ i = 1, \ldots, N$. The time derivative of V_z along trajectories of (46) is obtained as

$$\begin{aligned}\dot{V}_z =& \delta_z^T\Big(\mathcal{H} \otimes M_z\Big)\Big[(I_N \otimes A_z + c_z \mathcal{H} \otimes B_z K_z)\delta_z + (I_N \otimes B_z)(\tilde{W}_z^T \Psi_z(z) + \epsilon_z - \underline{r_z})\Big] \\ &+ \mathrm{tr}\Big\{\tilde{W}_z^T \Gamma_z^{-1} \dot{\tilde{W}}_z\Big\}\end{aligned} \tag{48}$$

which can be rewritten as

$$\begin{aligned}\dot{V}_z =& \frac{1}{2}\delta_z^T\Big[\mathcal{H} \otimes (M_z A_z + A_z^T M_z) - 2c_z \mathcal{H}^2 \otimes (M_z B_z B_z^T M_z)\Big]\delta_z \\ &+ \delta_z^T\Big(\mathcal{H} \otimes M_z B_z\Big)\Big(\tilde{W}_z^T \Psi_z(z) + \epsilon_z - \underline{r_z}\Big) + tr\Big\{\tilde{W}^T \Gamma^{-1} \dot{\tilde{W}}\Big\}\end{aligned} \tag{49}$$

From the trace properties, we can obtain the following

$$\begin{aligned}\dot{V}_z =& \frac{1}{2}\delta_z^T\Big[\mathcal{H} \otimes (M_z A_z + A_z^T M_z) - 2c_z \mathcal{H}^2 \otimes (M_z B_z B_z^T M_z)\Big]\delta_z + \delta_z^T\Big(\mathcal{H} \otimes M_z B_z\Big)\Big(\epsilon_z - \underline{r_z}\Big) \\ &+ \mathrm{tr}\Big\{\tilde{W}_z^T \Psi_z(z)\delta_z^T(\mathcal{H} \otimes M_z B_z) - \tilde{W}_z{}^T \Gamma_z^{-1} \dot{\hat{W}}_z\Big\}\end{aligned} \tag{50}$$

Notice that the adaptation law for the weights of the ith agent presented in Equation (30), can be rewritten in a global form as

$$\dot{\hat{W}}_z = \Gamma_z\Big[\Psi_z(z)\delta_z^T(\mathcal{H} \otimes M_z B_z) - k_z H_z\Big] \tag{51}$$

where $k_z = \mathrm{diag}(\bar{k}_{z_i}) \in \mathbb{R}^{Ns \times Ns}$, $\bar{k}_{z_i} = k_{z_i} I_{s \times s} \in \mathbb{R}^{s \times s}$, and k_{z_i} are positive constants, and $H_z = \mathrm{diag}(H_{z_i}) \in \mathbb{R}^{Ns \times N}$ with H_{z_i} defined in Equation (25). Substituting the global adaptation law (51) in the derivative of the Lyapunov function (52), we obtain

$$\begin{aligned}\dot{V}_z =& \frac{1}{2}\delta_z^T\Big[\mathcal{H} \otimes (M_z A_z + A_z^T M_z) - 2c_z \mathcal{H}^2 \otimes (M_z B_z B_z^T M_z)\Big]\delta_z + \delta_z^T\Big(\mathcal{H} \otimes M_z B_z\Big)\Big(\epsilon_z - \underline{r_z}\Big) \\ &+ \mathrm{tr}\Big\{\tilde{W}_z^T k_z H_z\Big\}\end{aligned} \tag{52}$$

Defining $P_z = \mathrm{diag}(P_{z_i}) \in \mathbb{R}^{Ns \times Ns}$, $Q_z = \mathrm{diag}(Q_{z_i}) \in \mathbb{R}^{Ns \times N}$, and $\varrho_z = \mathrm{diag}(\varrho_{z_i}) \in \mathbb{R}^{Ns \times N}$, we can express H_z as

$$\begin{aligned}H_z =& P_z \hat{W}_z - Q_z \\ =& - P_z \tilde{W}_z + \varrho_z.\end{aligned} \tag{53}$$

Then, substituting H_z in Equation (52), we obtain the following

$$\dot{V}_z = \frac{1}{2}\delta_z^T\Big[\mathcal{H}\otimes(M_zA_z + A_z^TM_z) - 2c_z\mathcal{H}^2\otimes(M_zB_zB_z^TM_z)\Big]\delta_z + \delta_z^T\Big(\mathcal{H}\otimes M_zB_z\Big)\Big(\epsilon_z - \underline{r_z}\Big) \\ - \mathrm{tr}\Big\{\tilde{W}_z^Tk_zP_z\tilde{W} + \tilde{W}_z^Tk_z\varrho_z\Big\} \tag{54}$$

From Young's inequality, for the second term in (54), we have

$$\begin{aligned}\dot{V}_z =& \frac{1}{2}\delta_z^T\Big[\mathcal{H}\otimes(M_zA_z + A_z^TM_z) - 2c_z\mathcal{H}^2\otimes(M_zB_zB_z^TM_z)\Big]\delta_z \\ &+ \frac{1}{2}\delta_z^T\Big(\mathcal{H}\otimes M_zB_z\Big)\Big(\mathcal{H}^\top\otimes B_z^\top M_z\Big)\delta_z - \mathrm{tr}\Big\{\tilde{W}_z^Tk_zP_z\tilde{W} + \tilde{W}_z^Tk_z\varrho_z\Big\} \\ &+ \frac{1}{2}\Big(\epsilon_z - \underline{r_z}\Big)^\top\Big(\epsilon_z - \underline{r_z}\Big) \\ =& \frac{1}{2}\delta_z^T\Big[\mathcal{H}\otimes(M_zA_z + A_z^TM_z) - (2c_z+1)\mathcal{H}^2\otimes(M_zB_zB_z^TM_z)\Big]\delta_z \\ &- \mathrm{tr}\Big\{\tilde{W}_z^Tk_zP_z\tilde{W}_z + \tilde{W}_z^Tk_z\varrho_z\Big\} + \frac{1}{2}\Big(\epsilon_z - \underline{r_z}\Big)^\top\Big(\epsilon_z - \underline{r_z}\Big)\end{aligned} \tag{55}$$

In order to simplify, we use a unitary matrix U such that $U^T\mathcal{H}U = \Lambda_z = diag(\lambda_{z_1}, \dots, \lambda_{z_N})$, where λ_{z_i} are the eigenvalues of matrix $\mathcal{H}$, which is positive definite. Let us introduce a state transformation given as $\xi_z = (U^T\otimes I_n)\delta_z$ with $\xi_z = [\xi_{z_1}^T, \dots, \xi_{z_N}^T]^T$. Then, introducing the state transformation in Equation (55), we obtain

$$\dot{V}_z = \frac{1}{2}\xi_z^T\Big[\Lambda\otimes(M_zA_z + A_z^TM_z) - (2c_z+1)\Lambda^2\otimes(M_zB_zB_z^TM_z)\Big]\xi_z \\ - \mathrm{tr}\Big\{\tilde{W}_z^Tk_zP_z\tilde{W}_z + \tilde{W}_z^Tk_z\varrho_z\Big\} + \frac{1}{2}\Big(\epsilon_z - \underline{r_z}\Big)^\top\Big(\epsilon_z - \underline{r_z}\Big) \tag{56}$$

which is rewritten in an expanded form as

$$\begin{aligned}\dot{V}_z =& \frac{1}{2}\sum_{i=1}^N\lambda_{z_i}\xi_{z_i}^\top\Big[M_zA_z + A_z^TM_z - (2c_z+1)\lambda_{z_i}M_zB_zB_z^TM_z + N_z\Big]\xi_{z_i} - \frac{1}{2}\sum_{i=1}^N\lambda_{z_i}\xi_{z_i}^\top N_z\xi_{z_i} \\ &- \sum_{i=1}^N k_{z_i}\tilde{W}_{z_i}^TP_{z_i}\tilde{W}_{z_i} - \sum_{i=1}^N k_{z_i}\tilde{W}_{z_i}^T\varrho_{z_i} + \frac{1}{2}\Big(\epsilon_z - \underline{r_z}\Big)^\top\Big(\epsilon_z - \underline{r_z}\Big) \\ \le& \frac{1}{2}\sum_{i=1}^N\lambda_{z_i}\xi_{z_i}^\top\Big[M_zA_z + A_z^TM_z - (2c_z+1)\min(\lambda_{z_i})M_zB_zB_z^TM_z + N_z\Big]\xi_{z_i} \\ &- \frac{1}{2}\min_{i=1,\dots,N}(\lambda_{z_i})\sum_{i=1}^N\xi_{z_i}^\top N_z\xi_{z_i} - \sum_{i=1}^N k_{z_i}\tilde{W}_{z_i}^TP_{z_i}\tilde{W}_{z_i} - \sum_{i=1}^N k_{z_i}\tilde{W}_{z_i}^T\varrho_{z_i} + \epsilon_z^+\end{aligned} \tag{57}$$

where $\frac{1}{2}\| \epsilon_z - \underline{r_z} \|^2 \le \epsilon_z^+$ with ϵ_z^+ a positive scalar. Let us define the coupling gain c_z as

$$c_z \ge \frac{1}{2\min\limits_{i=1,\dots,N}(\lambda_{z_i})} - \frac{1}{2}, \tag{58}$$

such that $(2c_z+1)\min(\lambda_{z_i}) \ge 1$. Since the part (A_z, B_z) is stabilizable, then the existence of the solution matrix M_z is guaranteed, such that

$$M_zA_z + A_z^TM_z + N_z - (2c_z+1)\min(\lambda_{z_i})M_zB_zB_z^TM_z \le M_zA_z + A_z^TM_z + N_z - M_zB_zB_z^TM_z \le 0. \tag{59}$$

Then, from Equation (59), we can rewrite (57) as

$$\dot{V}_z \le -\frac{1}{2}\min_{i=1,\dots,N}(\lambda_{z_i})\sum_{i=1}^N\xi_{z_i}^\top N_z\xi_{z_i} - \sum_{i=1}^N k_{z_i}\tilde{W}_{z_i}^TP_{z_i}\tilde{W}_{z_i} - \sum_{i=1}^N k_{z_i}\tilde{W}_{z_i}^T\varrho_{z_i} + \epsilon_z^+ \tag{60}$$

From Lemma 1, the matrices P_{z_i} are positive definite. Then, defining $\underset{min}{\lambda}(P_{z_i})$ as the minimum eigenvalue of $P_{z_i},\ i = 1, \ldots, N$, we obtain

$$\dot{V}_z \le -\frac{1}{2}\min_{i=1,\ldots,N}(\lambda_{z_i})\sum_{i=1}^{N}\xi_{z_i}^{\top}N_z\xi_{z_i} - k_z^{+}\underset{min}{\lambda}(P_{z_i})\sum_{i=1}^{N}\tilde{W}_{z_i}^{T}\tilde{W}_{z_i} - \sum_{i=1}^{N}k_{z_i}\tilde{W}_{z_i}^{T}\varrho_{z_i} + \epsilon_z^{+} \tag{61}$$

Removing the summations and grouping the terms, we obtain

$$\dot{V}_z \le -\frac{1}{2}\min_{i=1,\ldots,N}(\lambda_{z_i})\xi_z^{\top}(I_N \otimes N_z)\xi_z - k_z^{+}\underset{min}{\lambda}(P_w)\mathrm{tr}\left\{\tilde{W}_z^{\top}\tilde{W}_z\right\} + \left|\mathrm{tr}\left\{\tilde{W}_z^{T}k_z\varrho_z\right\}\right| + \epsilon_z^{+}, \tag{62}$$

From the definition of ξ_z, we obtain by the Cauchy–Schwartz inequality and (62) the following

$$\dot{V}_z \le -\frac{1}{2}\min_{i=1,\ldots,N}(\lambda_{z_i})\delta_z^{\top}(I_N \otimes N_z)\delta_z - k_z^{+}\underset{min}{\lambda}(P_{z_i})\|\tilde{W}\|_F^2 + k_{z_F}^{+}\varrho_z^{+}\|\tilde{W}\|_F + \epsilon_z^{+} \tag{63}$$

where $k_{z_F}^{+} = \| k_z \|_F$, $\| \varrho_z \|_F \le \varrho_{z_F}^{+}$ with $\varrho_{z_F}^{+}$ a positive constant. By Young's inequality $ab \le \frac{a^2}{2\eta} + \frac{\eta b^2}{2}$ with $\eta > 0$, we obtain

$$\begin{aligned}\dot{V}_z &\le -\tfrac{1}{2}\min_{i=1,\ldots,N}(\lambda_{z_i})\lambda_{\min}(N_z)\|\delta_z\|^2 - \left(k_z^{+}\underset{min}{\lambda}(P_{z_i}) - \tfrac{1}{2\eta_z}\right)\|\tilde{W}\|_F^2 + \frac{\eta_z\left(k_{z_F}^{+}\varrho_z\right)^2}{2} + \epsilon_z^{+} \\ &\le -\alpha_z V_z + \beta_z\end{aligned} \tag{64}$$

where

$$\alpha_z = \min\left(\frac{\lambda_{\min}(\mathcal{H})\lambda_{\min}(N_z)}{\lambda_{\max}(\mathcal{H})\lambda_{\max}(M_z)}, \frac{2\left(k_z^{+}\underset{min}{\lambda}(P_{z_i}) - \frac{1}{2\eta_z}\right)}{\sigma_{\max}\left(\Gamma_z^{-1}\right)}\right), \quad \beta_z = \frac{\eta_z(k_{z_F}^{+}\varrho_z)^2}{2} + \epsilon_z^{+} \tag{65}$$

Considering the extended Lyapunov theory, it is possible to conclude that δ_z and $\tilde{W}$ are uniformly ultimately bounded, and converge to a small compact set given by

$$\Omega_z := \left\{\delta_z, \tilde{W} \;\middle|\; \|\delta_z\| \le \sqrt{\frac{2}{\lambda_{\min}(\mathcal{H})\lambda_{\min}(M_z)}\frac{\beta_z}{\alpha_z}},\ \|\tilde{W}\| \le \sqrt{\frac{2}{\sigma_{\min}\left(\Gamma_z^{-1}\right)}\frac{\beta_z}{\alpha_z}}\right\} \tag{66}$$

Therefore, the tracking errors δ_{z_i} for the ith agent are uniformly ultimately bounded. □

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

Funding: This work was partially supported by the Mexican National Council for Science and Technology Project *Laboratorio Nacional en Vehículos Autónomos y Exoesqueletos.*

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

References

1. Dorling, K.; Heinrichs, J.; Messier, G.; Magierowski, S. Vehicle Routing Problems for Drone Delivery. *IEEE Trans. Syst. Man Cybern. Syst.* **2017**, *47*, 70–85. [CrossRef]
2. Bähnemann, R.; Schindler, D.; Kamel, M.; Siegwart, R.; Nieto, J. A decentralized multi-agent unmanned aerial system to search, pick up, and relocate objects. In Proceedings of the 2017 IEEE International Symposium On Safety, Security Furthermore, Rescue Robotics (SSRR), Shanghai, China, 11–13 October 2017; pp. 123–128.

3. Messina, F.; Vasilakos, A.; De Meo, P. Introduction to the special section on Recent trends in flocking control and communication for Unmanned vehicles. *Comput. Electr. Eng.* **2019**, *80*, 106495. [CrossRef]
4. Ge, X.; Han, Q.; Ding, D.; Zhang, X.; Ning, B. A survey on recent advances in distributed sampled-data cooperative control of multi-agent systems. *Neurocomputing* **2018**, *275*, 1684–1701. [CrossRef]
5. Hua, C.; Jiang, A.; Li, K. Adaptive neural network finite-time tracking quantized control for uncertain nonlinear systems with full-state constraints and applications to QUAVs. *Neurocomputing* **2021**, *440*, 264–274. [CrossRef]
6. Zhang, M.; Liu, H.T. Cooperative Tracking a Moving Target Using Multiple Fixed-wing UAVs. *J. Intell. Robot. Syst.* **2016**, *81*, 505–529. [CrossRef]
7. Tong, S. Event-triggered adaptive fuzzy bipartite consensus control of multiple autonomous underwater vehicles. *IET Control. Theory Appl.* **2020**, *14*, 3632–3642.
8. Ponniah, J.; Dantsker, O. Strategies for Scaleable Communication and Coordination in Multi-Agent (UAV) Systems. *Aerospace* **2022**, *9*, 488. [CrossRef]
9. Bai, J.; Wen, G.; Rahmani, A. Leaderless consensus for the fractional-order nonlinear multi-agent systems under directed interaction topology. *Int. J. Syst. Sci.* **2018**, *49*, 954–963. [CrossRef]
10. Wang, C.; Wang, J.; Wu, P.; Gao, J. Consensus Problem and Formation Control for Heterogeneous Multi-Agent Systems with Switching Topologies. *Electronics* **2022**, *11*, 2598. [CrossRef]
11. Abdessameud, A.; Tayebi, A. On consensus algorithms design for double integrator dynamics. *Automatica* **2013**, *49*, 253–260. [CrossRef]
12. Tian, B.; Zuo, Z.; Wang, H. Leader–follower fixed-time consensus of multi-agent systems with high-order integrator dynamics. *Int. J. Control* **2017**, *90*, 1420–1427. [CrossRef]
13. Liu, Z. Neural-network-based adaptive leader-following consensus control for second-order non-linear multi-agent systems. *IET Control. Theory Appl.* **2015**, *9*, 1927–1934.
14. Neto, A.; Mozelli, L.; Souza, F. Control of air-ground convoy subject to communication time delay. *Comput. Electr. Eng.* **2019**, *76*, 213–224. [CrossRef]
15. Ma, L.; Wang, Z.; Liu, Y.; Alsaadi, F. A note on guaranteed cost control for nonlinear stochastic systems with input saturation and mixed time-delays. *Int. J. Robust Nonlinear Control* **2017**, *27*, 4443–4456. [CrossRef]
16. Sakthivel, R.; Sakthivel, R.; Kaviarasan, B.; Alzahrani, F. Leader-following exponential consensus of input saturated stochastic multi-agent systems with Markov jump parameters. *Neurocomputing* **2018**, *287*, 84–92. [CrossRef]
17. Yang, P.; Ding, Y.; Shen, Z.; Feng, K. Integral Non-Singular Terminal Sliding Mode Consensus Control for Multi-Agent Systems with Disturbance and Actuator Faults Based on Finite-Time Observer. *Entropy* **2022**, *24*, 1068. [CrossRef] [PubMed]
18. Yu, S.; Long, X. Finite-time consensus for second-order multi-agent systems with disturbances by integral sliding mode. *Automatica* **2015**, *54*, 158–165. [CrossRef]
19. Liu, C.; Sun, S.; Tao, C.; Shou, Y.; Xu, B. Sliding mode control of multi-agent system with application to UAV air combat. *Comput. Electr. Eng.* **2021**, *96*, 107491. [CrossRef]
20. Zhang, L.; Li, J.; Zhu, Y.; Shi, H.; Hwang, K. Multi-agent reinforcement learning by the actor-critic model with an attention interface. *Neurocomputing* **2022**, *471*, 275–284. [CrossRef]
21. Wu, H.; Zhang, J.; Wang, Z.; Lin, Y.; Li, H. Sub-AVG: Overestimation reduction for cooperative multi-agent reinforcement learning. *Neurocomputing* **2022**, *474*, 94–106. [CrossRef]
22. Qin, J.; Ma, Q.; Shi, Y.; Wang, L. Recent Advances in Consensus of Multi-Agent Systems: A Brief Survey. *IEEE Trans. Ind. Electron.* **2017**, *64*, 4972–4983. [CrossRef]
23. Li, Y.; Tan, C. A survey of the consensus for multi-agent systems. *Syst. Sci. Control Eng.* **2019**, *7*, 468–482. [CrossRef]
24. Sierra, J.; Santos, M. Wind and Payload Disturbance Rejection Control Based on Adaptive Neural Estimators: Application on Quadrotors. *Complexity* **2019**, *2019*, 6460156. [CrossRef]
25. Lee, J.; Choi, Y.; Suh, J. DeConNet: Deep Neural Network Model to Solve the Multi-Job Assignment Problem in the Multi-Agent System. *Appl. Sci.* **2022**, *12*, 5454. [CrossRef]
26. Dong, C.; Ye, Q.; Dai, S. Neural-network-based adaptive output-feedback formation tracking control of USVs under collision avoidance and connectivity maintenance constraints. *Neurocomputing* **2020**, *401*, 101–112. [CrossRef]
27. Peng, Z.; Wang, D.; Zhang, H.; Lin, Y. Cooperative output feedback adaptive control of uncertain nonlinear multi-agent systems with a dynamic leader. *Neurocomputing* **2015**, *149*, 132–141. [CrossRef]
28. Shahvali, M.; Shojaei, K. Distributed adaptive neural control of nonlinear multi-agent systems with unknown control directions. *Nonlinear Dyn.* **2016**, *83*, 2213–2228. [CrossRef]
29. Peng, Z.; Wang, D.; Wang, J. Predictor-Based Neural Dynamic Surface Control for Uncertain Nonlinear Systems in Strict-Feedback Form. *IEEE Trans. Neural Netw. Learn. Syst.* **2017**, *28*, 2156–2167. [CrossRef]
30. Hashemi, M.; Shahgholian, G. Distributed robust adaptive control of high order nonlinear multi agent systems. *ISA Trans.* **2018**, *74*, 14–27. [CrossRef]
31. Fan, L.; Wu, C.; Ji, H. Distributed Adaptive Finite-Time Consensus for High-Order Multi-Agent Systems with Intermittent Communications under Switching Topologies. *Symmetry* **2022**, *14*, 1368. [CrossRef]
32. Ioannou, P.; Kokotovic, P. Robust redesign of adaptive control. *IEEE Trans. Autom. Control* **1984**, *29*, 202–211. [CrossRef]

33. Narendra, K.; Annaswamy, A. A new adaptive law for robust adaptation without persistent excitation. *IEEE Trans. Autom. Control* **1987**, *32*, 134–145. [CrossRef]
34. Adetola, V.; Guay, M.; Lehrer, D. Adaptive Estimation for a Class of Nonlinearly Parameterized Dynamical Systems. *IEEE Trans. Autom. Control* **2014**, *59*, 2818–2824. [CrossRef]
35. Na, J.; Mahyuddin, M.; Herrmann, G.; Xuemei, R.; Barber, P. Robust adaptive finite-time parameter estimation and control for robotic systems. *Int. J. Robust Nonlinear Control* **2014**, *25*, 3045–3071. [CrossRef]
36. Na, J.; Herrmann, G.; Ren, X.; Mahyuddin, M.; Barber, P. Robust adaptive finite-time parameter estimation and control of nonlinear systems. In Proceedings of the 2011 IEEE International Symposium On Intelligent Control, Denver, CO, USA, 28–30 September 2011; pp. 1014–1019.
37. Jing, B.; Na, J.; Gao, G.; Sun, G. Robust Adaptive Control for Robotic Systems with Guaranteed Parameter Estimation. In Proceedings of the 2015 Chinese Intelligent Systems Conference, Yangzhou, China, 8 November 2015; pp. 341–352.
38. Yang, J.; Na, J.; Guo, Y.; Wu, X. Adaptive estimation of road gradient and vehicle parameters for vehicular systems. *IET Control Theory Appl.* **2015**, *9*, 935–943. [CrossRef]
39. Mahyuddin, M.; Na, J.; Herrmann, G.; Ren, X.; Barber, P. Adaptive Observer-Based Parameter Estimation With Application to Road Gradient and Vehicle Mass Estimation. *IEEE Trans. Ind. Electron.* **2014**, *6*, 2851–2863. [CrossRef]
40. Na, J.; Huang, Y.; Wu, X.; Gao, G.; Herrmann, G.; Jiang, J. Active Adaptive Estimation and Control for Vehicle Suspensions With Prescribed Performance. *IEEE Trans. Control. Syst. Technol.* **2018**, *26*, 2063–2077. [CrossRef]
41. Zhao, J.; Wang, X.; Gao, G.; Na, J.; Liu, H.; Luan, F. Online Adaptive Parameter Estimation for Quadrotors. *Algorithms* **2018**, *11*, 167. [CrossRef]
42. Zhao, Q.; Duan, G. Adaptive finite-time tracking control of 6DOF spacecraft motion with inertia parameter identification. *IET Control Theory Appl.* **2019**, *13*, 2075–2085. [CrossRef]
43. Sildir, H.; Sarrafi, S.; Aydin, E. Uncertainty Propagation Based MINLP Approach for Artificial Neural Network Structure Reduction. *Processes* **2022**, *10*, 1716. [CrossRef]
44. Chenguang, Y.; Tao, B.; Bin, X.; Zhijun, L.; Na, J.; Chun-Yi, S. Global Adaptive Tracking Control of Robot Manipulators Using Neural Networks with Finite-time Learning Convergence. *Int. J. Control Autom. Syst.* **2017**, *15*, 1916–1924.
45. Luan, F.; Na, J.; Huang, Y.; Gao, G. Adaptive neural network control for robotic manipulators with guaranteed finite-time convergence. *Neurocomputing* **2019**, *337*, 153–164. [CrossRef]
46. Jiang, Y.; Wang, Y.; Miao, Z.; Na, J.; Zhao, Z.; Yang, C. Composite-Learning-Based Adaptive Neural Control for Dual-Arm Robots With Relative Motion. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**, *33*, 1010–1021. [CrossRef] [PubMed]
47. Kafaf, D.; Kim, D. A web service-based approach for developing self-adaptive systems. *Comput. Electr. Eng.* **2017**, *63*, 260–276. [CrossRef]
48. Zhang, X.; Liu, L.; Feng, G. Leader–follower consensus of time-varying nonlinear multi-agent systems. *Automatica* **2015**, *52*, 8–14. [CrossRef]
49. Bouabdallah, S.; Noth, A.; Siegwart, R. PID vs. LQ control techniques applied to an indoor micro quadrotor. In Proceedings of the 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Sendai, Japan, 28 September–2 October 2004; Volume 3, pp. 2451–2456.
50. Das, A.; Lewis, F.; Subbarao, K. Backstepping Approach for Controlling a Quadrotor Using Lagrange Form Dynamics. *J. Intell. Robot. Syst.* **2009**, *56*, 127–151. [CrossRef]
51. Meier, L.; Honegger, D.; Pollefeys, M. PX4: A node-based multithreaded open source robotics framework for deeply embedded platforms. In Proceedings of the 2015 IEEE International Conference on Robotics Furthermore, Automation (ICRA), Seattle, WA, USA, 26–30 May 2015; pp. 6235–6240.
52. Souza, F.; Santos, S.; Oliveira, A.; Givigi, S. Influence of Network Topology on UAVs Formation Control based on Distributed Consensus. In Proceedings of the 2022 IEEE International Systems Conference (SysCon), Montreal, QC, Canada, 25–28 April 2022; pp. 1–8.
53. Peng, Z.; Wang, D.; Zhang, H.; Sun, G. Distributed Neural Network Control for Adaptive Synchronization of Uncertain Dynamical Multiagent Systems. *IEEE Trans. Neural Netw. Learn. Syst.* **2014**, *25*, 1508–1519. [CrossRef] [PubMed]