sensors

*Article*

# Multirobot Heterogeneous Control Considering Secondary Objectives

**Julio F. Acosta** [1,*], **Guillermo González de Rivera** [2], **Víctor Hugo Andaluz** [1] **and Javier Garrido** [2]

[1] Universidad de las Fuerzas Armadas–ESPE, Sangolquí 171103, Ecuador; vhandaluz1@espe.edu.ec
[2] Escuela Politécnica Superior, Universidad Autónoma de Madrid, 28049 Madrid, Spain; guillermo.gdrivera@uam.es (G.G.d.R.); javier.garrido@uam.es (J.G.)
* Correspondence: jfacosta@espe.edu.ec; Tel.: +593-984533158

Received: 30 August 2019; Accepted: 7 October 2019; Published: 9 October 2019

**Abstract:** Cooperative robotics has considered tasks that are executed frequently, maintaining the shape and orientation of robotic systems when they fulfill a common objective, without taking advantage of the redundancy that the robotic group could present. This paper presents a proposal for controlling a group of terrestrial robots with heterogeneous characteristics, considering primary and secondary tasks thus that the group complies with the following of a path while modifying its shape and orientation at any time. The development of the proposal is achieved through the use of controllers based on linear algebra, propounding a low computational cost and high scalability algorithm. Likewise, the stability of the controller is analyzed to know the required features that have to be met by the control constants, that is, the correct values. Finally, experimental results are shown with different configurations and heterogeneous robots, where the graphics corroborate the expected operation of the proposal.

**Keywords:** heterogeneous robots; control based on linear algebra; cooperative control; control considering secondary objectives

## 1. Introduction

Recent advances in robotics have allowed the migration from the operation of a single large vehicle to the integration of many ones in search of a common goal. This migration is the result of optimizing various processes, which require a robot with great capabilities to a set of robots with distinctive features. Applications such as the search and rescue of objectives in natural disasters [1], displacement of extremely heavy or very long objects, mapping of land [2], assembly of pieces, and so on, are an example of the need to bring together small-scale robots and lead them on a common task [3]. In this type of terrestrial applications, the mobile robots based on wheels predominate over robots composed of legs or caterpillars, subdividing them in the way in how the wheels are distributed in the mechanism [4,5].

The usual configuration of mobile robots consists of two wheels joined by a common axis, where the stability of the mechanism is achieved by placing a castor-type wheel. This type of vehicle has had a considerable amount of analysis given the ease of its construction, despite its limitations, such as the non-holonomicity (a common feature of wheeled robots in general) or the loss of traction in sharp turns [6]. However, the limitations presented by this type of differential robot can be complemented by a different configuration, e.g., an Ackerman or an omnidirectional type. A robotic mechanism in the Ackerman configuration is the one found in conventional vehicles, where one motor drives the mobile, and another modifies its orientation. Although this configuration does not provide holonomicity to the mechanism, its application can solve tasks which require traction, since all the wheels have a fixed contact with the floor of the environment. On the other hand, the configuration, which allows moving the mobile in any of the directions, is known as omnidirectional. Depending on the type of wheels used, an omnidirectional robot can be placed at any point in the workspace while its orientation can be simultaneously modified [7,8]. In this way, the scalability of a set of cooperatively working robots can be maximized by using mobile robots of different characteristics, where the control methods can facilitate the inclusion of omnidirectional, differential, car-like, and so forth vehicles [9]. In the same context, heterogeneous robots can be used depending on the applications. Either they are working in structured or semi-structured environments, outdoor or indoor, and so on [10,11].

In related research, the concept of cooperation is generally interpreted in two different ways: Based on leader-follower and based on a virtual leader [12–14]. In the leader-follower consideration, one of the robotic mechanisms plays the role of leader complying with the tracking of the established path, while the other ones move relatively to the main robot. This configuration, although it is simple to implement, does not supply robustness when secondary objectives are required and depend directly on the execution of a single robot. In contrast, the virtual leader consideration generates a virtual formation that serves as a pattern for the rest of the vehicles to maintain a desired position and orientation. From a comparative perspective, the first consideration is led to applications where environmental disturbances are minimal, and computational processing is low [15]. On the other hand, the virtual leader approach allows executing secondary tasks such as obstacles avoidance, reshaping, minimizing of movements, and so on, since there is not a single robot in charge of achieving a control objective, but the formation follows a virtual leader, which can be modified due to disturbances in the environment. However, it requires a relatively high computational cost, as well as a resilient and robust communication structure. A cooperative robotics structure must additionally consider the distribution of control, which maintains the robotic set [16]. In the centralized control, one vehicle generates the calculation of all the velocities which the fellow robots must have as a reference to maintain the formation, where that robot has relatively high processing characteristics. On the contrary, the decentralized control considers that the formation characteristics are shared among the participants at a very high level, where the control actions to maintain the desired grouping are calculated for each vehicle individually. Although both configurations require some type of intercommunication to ensure satisfactory execution, the second case requires high reliability of data exchange to achieve the control objectives [17]. A considerable advantage of decentralized control is the integration of different control methods to the group of robots, taking into account that generally for cooperative robotics, the most important aspect is to maintain the position of the robotic set into a trajectory or path.

Depending on the hardware characteristics of the robotic systems, the controllers can be led to the optimization of processing or the fulfillment of the control objectives with the least possible error. In this way, the formation control can be implemented by various methods. For instance, [18] claims that some control methods are able to meet both the formation of a set of robots and to generate the reference velocities for each of the systems. Likewise, [19] proposes a control method for tracking trajectories based on numerical methods. The work presents a double manipulator embedded on a mobile platform, where the point of interest is in the middle of the two operational ends. The asymptotic stability of the control method is demonstrated analytically, where the experimental tests validate the execution of the formation. Similarly, a control method based on linear algebra is proposed by [20], where the kinematic system is approximated through linear algebra to find both the particular solution and the homogeneous solution, allowing to meet secondary objectives [21,22]. Experimental results are shown in order to demonstrate the stability of the proposed method. Another control method is mentioned in [23], where a cooperative control for non-holonomic robots is proposed through a finite-time controller. As the main contribution, the work is led to solve the finite-time consensus problem for multiple high-order nonholonomic mobile robots from a finite-time consensus control algorithm. Through simulation, it is shown that the controller can achieve the objectives in all the agents in the proposed finite time since the convergence of the method used is tested through a suitable Lyapunov function. In addition to the presented methods, [24] implements a pair of algorithms, the first based on a reinforcement learning algorithm and the second based on a particle filter. The main objective of the work is searching and tracking people with cooperative mobile robots, which locate goals, calculate their distance to them, and follow them, considering they work in highly unstructured spaces. The results of the proposed control algorithms are validated through an extensive set of simulations considering up to five agents, in addition to the hours of real-life experiments with two social robots.

Furthermore, the control methods for the formation of robots have a similar criterion as for a traditional robotic system, where the way in which the points of interest of all the participants are unified differs in many works. In this aspect, [25] can be considered, where the aim is controlling both the center and the angles of rotation and elevation of the triangle formed by a robotic set, a group that is greater than or equal to three members. The triangle formed supposes to be an irregular object held up by the operative end of all the mobile manipulators, where additionally, the redundancy of the systems is considered to carry out the evasion of obstacles or other secondary objectives. In contrast, the shape and orientation of the whole set can be configured through projections between each robot. Reference [26] takes into account this consideration, where the objective is to transport an object by two mobile anthropomorphic robots. The proposed controller to carry out the problem is based on the Lyapunov method, where it is analytically demonstrated that it is asymptotically stable. Additionally, experimental results are shown where three-dimensional platforms are used in order to evaluate the performance of the proposal.

In this work, the fulfillment of the primary and secondary objectives of a set of robots with heterogeneous characteristics and different modes of control and operation, working cooperatively, is analysed assuming that the position of each robot, the path to follow, and the shape of the formations are known as the initial condition, while taking into account that the robotic set knows the environment from external information or previously stored data. Through the proposal of a control algorithm based on linear algebra, which makes the computational cost less and becomes easily programmable, that are a great advantage in the implementation of the system, the set of robots can accomplish a task in a coordinated way by following a desired path as the primary task, while the formation of them is altered depending on the environment or the mission requirements (secondary tasks); this is the main novelty of the proposed system. Different to traditional ways of controlling a group of terrestrial robots, in this work the robotic set is moved along a path with a desired velocity not parameterized in time, which can depend on various purposes (evasion of obstacles encountered along the way, optimization of energy, and so on), while simultaneously it can alter the formation of the whole set through the modification of distances and orientations of the main and secondary projections. Contrary to works presented in [2,3,5,6], the formation scheme of heterogeneous robots is analyzed from projections between a pair of robots (being physical or virtual ones depending on the requirements), providing flexibility and scalability at the moment of adding up the robotic systems. Likewise, the controller based on linear algebra is designed, simulated, and implemented on the set of heterogeneous robots, further analyzing their stability. Another great advantage of the proposed system is that being a set of heterogeneous robots that operate, makes the kinematic and dynamic control of each residing in them, thus reducing the processing load of the system and giving the possibility of increasing robots of any operational feature, including their kinematic and dynamic control in it without the need to modify any other part of the system at all. It is not known of any work that has done something similar. Compared to references [8,9,14], the proposed controller uses a hierarchical control structure, in order to provide scalability to the system and at the same time, not saturate the processing unit, thus merging both the centralized and decentralized information processors. At a very high hierarchy level, a centralized computer is responsible for generating the control actions to achieve the primary and secondary objectives, while, at a local level, each member of the heterogeneous robotic set includes its own processing unity to achieve kinematic and dynamic control and also to provide feedback of odometric information, which is shared through wireless communications based on a data frame (assuming no constrains given by the environment). Additionally, another relevant advantage of the proposal is the low computational cost obtained through the control method based on linear algebra, avoiding saturating both the centralized computer and those ones located in each robot.

In [21] and many more, Gustavo Scaglia presents works based on Linear Algebra theories for collaborative control of mobile robots focused on the leading-follower technique; however, he does not consider the scalability of the robots nor the execution of the secondary control objectives, i.e., modification of robot formation, and obstacle avoidance, among other criteria. We, in our work, have done all of this, which makes it a substantial improvement.

This current work is presented in six sections, including the introduction and conclusions. Thus, Section 2 presents the multilayer control scheme, superficially describing the proposed control structure, while Section 3 presents the development of controllers based on numerical methods considering secondary tasks. Section 4 presents the controllers to execute the primary task, and the secondary task raised, analyzing the stability of both proposed methods. Section 5 shows the proposed kinematic control based on numerical methods, as well as a dynamic compensator for each robotic system. Finally, Section 6 shows three real experimental results to validate the proposed controllers, denoting the control responses of both the formation and each of the robots.

## 2. Multilayer Control Scheme

The multilayer scheme defines how the appropriate variables that make up the cooperative control of heterogeneous mobile robots are distributed (Figure 1). Taking into account that each of the layers has different functionalities, the scheme is divided into six sections: (1) Acquisition of the characteristics of the environment allows planning the path to follow, maintaining the integrity of the robotic systems as well as fulfilling the pre-programmed task. This section has two layers, and in the off-line planning layer, the initial parameters are configured, i.e., the desired path to be followed, the starting locations of all the robotic systems, and the desired structure of formation. The on-line planning layer allows the execution of the secondary objectives, reacting to the unforeseen environmental changes in the off-line layer, produced by the lack of space to fulfill the desired parameters of formation, objects to avoid along the path, environmental conditions that require to change the formation performed, among other conditions, that can alter the fulfillment of the task. This is also very important because, through this scheme, you can navigate between the structured and unstructured environments. When the path is known, or the operating environment has already been obtained previously, this information is entered in this layer, if not, and as the robots are multi-sensors, you can easily install a radar and make the path planning and follow those trajectories found in real-time.

The objective of the (2) formation control layer is focused on calculating the control actions so that the vehicles maintain a positioning that satisfies the desired formation, generating the control signals for each of the robots. The (3) kinematic control layer takes as a reference the velocities generated by the formation control (upper level), i.e., the local controllers receive reference velocities to follow the profile guided by the formation, so these controllers are responsible for delivering maneuverability velocities that correspond of each heterogeneous robot involved (omnidirectional, car-like, unicycle). The (4) dynamics of each robot acting as an individual mechanism is solved by the adaptive dynamic compensation layer, within the control and operation scheme this layer is of paramount importance and provides, as mentioned above, the advantage that if robots of other characteristics are increased, each one has its own controller on top, which means that the only major modification that has to be made to the system is to enter the data from this robot into the communications frame, completely eliminating any change outside of this. The (5) Robots layer represents the set of heterogeneous robots considered in the formation. In addition, two virtual robots are used for the calculation of each of the projections. Finally, the (6) environment layer represents the structural space where the robot interacts, considering obstacles and other physical objects.

From what has been described, it can be noted that the work presented by us differs from the collaborative control literature in the following items: (1) Proposal of a modular multilayer control scheme, that is, it is possible to implement different control techniques without modifying the general structure of the collaborative control between n robots; (2) the control scheme considers both the centralized control technique, and the decentralized control technique, which allows generalization of the collaborative control to n heterogeneous robots; (3) the proposed control algorithms are based on the theory of linear algebra, which has as an advantage a low computational cost and, therefore, can be implemented in robots built with low-cost technology. In addition, when considering the linear algebra technique for the design of control algorithms, it is possible to incorporate redundancy in the transformation matrices of the formation, thus it is feasible to execute the main task (path follow-up of predefined tasks) and several secondary tasks, for example, modification of the formation, evasion of obstacles, fewer movements, among other control criteria; (4) the proposed algorithm considers path follow-up, that is, following the desired profile without being parameterized in time. Most of the works in the literature perform trajectory tracking, which is not feasible to implement in real applications since the trajectory tracking defines the desired position at a certain time. That is, it has a tracking profile parameterized in time; and finally (5) in the article presented there is a formal analysis of the stability of each one of the proposed control algorithms, in which mathematically asymptotic stability is demonstrated, that is, the control errors tend to zero.

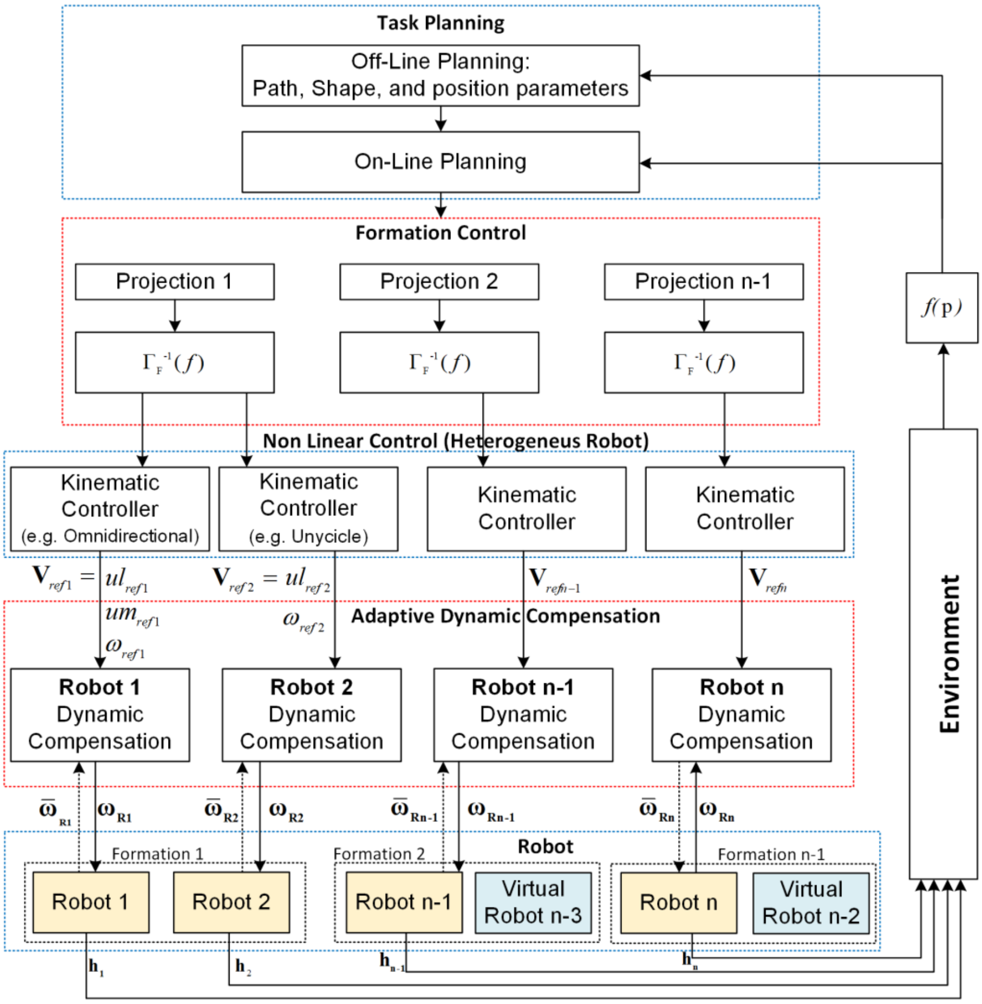
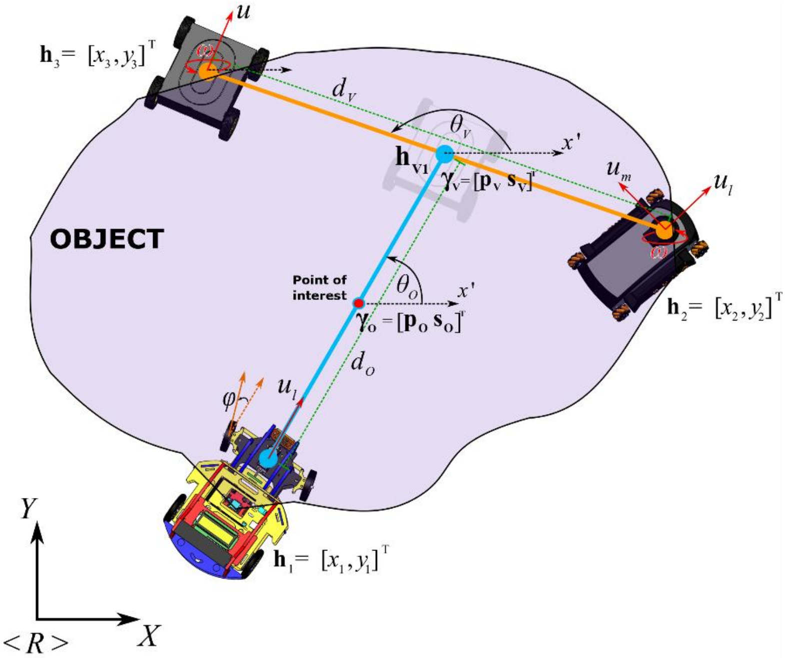

Task Planning
Off-Line Planning:
Path, Shape, and position parameters
On-Line Planning
Formation Control
Projection 1
Projection 2
Projection n-1
$\Gamma_F^{-1}(f)$
$\Gamma_F^{-1}(f)$
$\Gamma_F^{-1}(f)$
$f(\mathrm{p})$
Non Linear Control (Heterogeneous Robot)
Kinematic Controller (e.g. Omnidirectional)
Kinematic Controller (e.g. Unycicle)
Kinematic Controller
Kinematic Controller
$\mathbf{V}_{ref1} = ul_{ref1}$
$um_{ref1}$
$\omega_{ref1}$
$\mathbf{V}_{ref2} = ul_{ref2}$
$\omega_{ref2}$
$\mathbf{V}_{refn-1}$
$\mathbf{V}_{refn}$
Adaptive Dynamic Compensation
Robot 1 Dynamic Compensation
Robot 2 Dynamic Compensation
Robot n-1 Dynamic Compensation
Robot n Dynamic Compensation
$\bar{\boldsymbol{\omega}}_{R1}$
$\boldsymbol{\omega}_{R1}$
$\bar{\boldsymbol{\omega}}_{R2}$
$\boldsymbol{\omega}_{R2}$
$\bar{\boldsymbol{\omega}}_{Rn-1}$
$\boldsymbol{\omega}_{Rn-1}$
$\bar{\boldsymbol{\omega}}_{Rn}$
$\boldsymbol{\omega}_{Rn}$
Robot
Formation 1
Formation 2
Formation n-1
Robot 1
Robot 2
Robot n-1
Virtual Robot n-3
Robot n
Virtual Robot n-2
Environment
$\mathbf{h}_1$
$\mathbf{h}_2$
$\mathbf{h}_{n-1}$
$\mathbf{h}_n$

**Figure 1.** Multilayer control scheme.

## 3. Kinematic Transformation

Since the formations are composed of projections between a pair of robots, an additional formation is necessary to include a third robot, requiring the use of a virtual robot to generate the additional projection. It is assumed that the object to be transported or the shape that the robotic set should have does not have a defined geometry—it may have an unequal weight distribution, or it could be moving in a semi-structured space. In this aspect, the fulfillment of tasks with this type of objective can present better results if the robotic mechanisms have heterogeneous characteristics such as car-like, omnidirectional, or differential robots. The location of each of the robots depends totally on the requirements of the tasks, being able to alter the configurations at any moment of time if the workspace requires it.

Based on the formation presented for the analysis (Figure 2), the main projection to transport an object by three mobile vehicles is described as $\boldsymbol{\gamma_O} = \begin{bmatrix} \boldsymbol{p_O} & \boldsymbol{s_O} \end{bmatrix}^{\boldsymbol{T}} \in \mathbb{R}^4$, where $\boldsymbol{p_O} = [x_O y_O]^{\boldsymbol{T}} \in \mathbb{R}^2$ represents the position of the main intermediate point and $\boldsymbol{s_O} = \begin{bmatrix} \theta_O & d_O \end{bmatrix}^{\boldsymbol{T}} \in \mathbb{R}^2$ indicates the shape of the projection, both with respect to the axis of reference $\langle R \rangle$. Thus, $\boldsymbol{p_O}$ is located between the position of the first physical robot $\mathbf{h}_1$ and the first virtual robot $\mathbf{h_{V1}}$. On the other hand, for the second projection $\boldsymbol{\gamma_V} = \begin{bmatrix} \boldsymbol{p_V} & \boldsymbol{s_V} \end{bmatrix}^{\boldsymbol{T}} \in \mathbb{R}^4$, the point $\boldsymbol{p_V} = [x_V y_V]^{\boldsymbol{T}} \in \mathbb{R}^2$ is located between the position of the second $\boldsymbol{h}_2$ and the third physical robot $\boldsymbol{h}_3$ (the point which is also represented as the virtual robot $\boldsymbol{h_{V1}}$), while $\boldsymbol{s_V} = \begin{bmatrix} d_V & \theta_V \end{bmatrix}^{\boldsymbol{T}}$ describes the secondary shape. The characteristics of the transported object are defined by the distances and the desired angles formed between each of the robots, defined by $\boldsymbol{s_O} = [d_O \theta_O]^{\boldsymbol{T}}$ and $\boldsymbol{s_V} = [d_V \theta_V]^{\boldsymbol{T}}$, where $d_O$ represents the separation between the first robot and the virtual position, $d_V$ represents the separation between the other two physical robots $\boldsymbol{h}_2$ and $\boldsymbol{h}_3$, while $\theta_O$ and $\theta_V$ represent the orientation of both projections with respect to Y-axis and X-axis from the reference plane <R>.



u
$\mathbf{h}_3 = [x_3, y_3]^T$
ω
$d_V$
$\theta_V$
x'
$\mathbf{h}_{V1}$
$\boldsymbol{\gamma}_V = [\mathbf{p}_V\ \mathbf{s}_V]^T$
$u_m$
$u_l$
ω
OBJECT
Point of interest
$\theta_O$
x'
$\boldsymbol{\gamma}_O = [\mathbf{p}_O\ \mathbf{s}_O]^T$
$\mathbf{h}_2 = [x_2, y_2]^T$
$d_O$
φ
$u_l$
$\mathbf{h}_1 = [x_1, y_1]^T$
Y
X
<R>

**Figure 2.** Formation analysis of heterogeneous robots.

The relationship between the formation pose-shape of each projection and the point-of-interest positions of each of the vehicles is given by the direct and inverse kinematic transformation, i.e., $\gamma_k = f\big(g_k\big)$ and $g_k = f^{-1}(\gamma_k)$, where $\gamma_k = \begin{bmatrix} p_k & s_k \end{bmatrix}^T$ and $g_k = \begin{bmatrix} h_{1st}^T & h_{2nd}^T \end{bmatrix}^T$, with $k = O, V$ (main and secondary formations, respectively) for this case, while $h_{1st}^T$ and $h_{2nd}^T$ are the first and second robots of the projection, respectively. The direct kinematic transformations $f(.)$ for both projections are given by:

$$\gamma_V = \begin{bmatrix} \boldsymbol{p_V} \\ \boldsymbol{s_V} \end{bmatrix}; \boldsymbol{p_V} = \begin{bmatrix} x_V \\ y_V \end{bmatrix} = \begin{bmatrix} \frac{1}{2}(x_3 + x_2) \\ \frac{1}{2}(y_3 + y_2) \end{bmatrix}; \boldsymbol{s_V} = \begin{bmatrix} d_V \\ \theta_V \end{bmatrix} = \begin{bmatrix} \sqrt{(x_3 - x_2)^2 + (y_3 - y_2)^2} \\ \tan^{-1}\left(\frac{y_3 - y_2}{x_3 - x_2}\right) \end{bmatrix}, \tag{1}$$

$$\gamma_O = \begin{bmatrix} \boldsymbol{p_O} \\ \boldsymbol{s_O} \end{bmatrix}; \boldsymbol{p_O} = \begin{bmatrix} x_O \\ y_O \end{bmatrix} = \begin{bmatrix} \frac{1}{2}(x_V + x_1) \\ \frac{1}{2}(y_V + y_1) \end{bmatrix}; \boldsymbol{s_O} = \begin{bmatrix} d_O \\ \theta_O \end{bmatrix} = \begin{bmatrix} \sqrt{(x_V - x_1)^2 + (y_V - y_1)^2} \\ \tan^{-1}\left(\frac{y_V - y_1}{x_V - x_1}\right) \end{bmatrix}. \tag{2}$$

In turn, the inverse kinematic transformation $f(.)$ is denoted by

$$g_V = \begin{bmatrix} \boldsymbol{h_3} \\ \boldsymbol{h_2} \end{bmatrix} = \begin{bmatrix} x_V + \frac{1}{2}d_V \cos\theta_V \\ y_V + \frac{1}{2}d_V \sin\theta_V \\ x_V - \frac{1}{2}d_V \cos\theta_V \\ y_V - \frac{1}{2}d_V \sin\theta_V \end{bmatrix} \text{and} \tag{3}$$

$$g_O = \begin{bmatrix} \boldsymbol{p_V} \\ \boldsymbol{h_1} \end{bmatrix} = \begin{bmatrix} x_O + \frac{1}{2}d_O \cos\theta_O \\ y_O + \frac{1}{2}d_O \sin\theta_O \\ x_O - \frac{1}{2}d_O \cos\theta_O \\ y_O - \frac{1}{2}d_O \sin\theta_O \end{bmatrix} \tag{4}$$

**Remark 1:** *$x_O, y_O, x_V, y_V$ are located on the frame of the inertial reference.*

The relationship between the variation of $g(t)$ and $\gamma(t)$ can be obtained through the time derivative of the forward and the inverse kinematics transformations, represented by the Jacobian matrix $\mathbf{\Gamma_F(g)}$ defined as:

$$\dot{\gamma}(t) = \mathbf{\Gamma}_{\boldsymbol{F}}(g)\dot{g}(t) \tag{5}$$

and in the inverse way is given by:

$$\dot{g}(t) = \mathbf{\Gamma}_{\boldsymbol{F}}^{-1}(\gamma)\dot{\gamma}(t), \tag{6}$$

where

$$\mathbf{\Gamma}_{\boldsymbol{F}}(g) = \frac{\partial \gamma_{fx1}}{\partial g_{\mathrm{ex1}}} \text{ and } \mathbf{\Gamma}_{\boldsymbol{F}}^{-1}(\gamma) = \frac{\partial g_{\mathrm{ex1}}}{\partial \gamma_{fx1}} \text{ with } e, f = 4.$$

## 4. Controllers Design

### *4.1. Controller Analysis Considering Secondary Objectives*

The design of the kinematic controllers proposed in this paper was based on numerical methods tools. To facilitate the search for the solution of a set of equations, a system can be represented in a matrix structure, where theorems and axioms of linear algebra are applied. In this way, it is considered the first-order differential equation:

$$\dot{\eta}(t) = f\big(\eta(t), \dot{\zeta}(t)\big), \tag{7}$$

where $\eta(t)$ represents the output of the system to be the controller with initial conditions $\eta(0) = \eta_0$, $\dot{\eta}(t)$ is the first derivative with respect to time, and $\dot{\zeta}(t)$ is the control action. Furthermore, $\eta(t)$ becomes $\eta(k)$ in the discrete time with $t = kT_0$, where $T_0$ represents the proposed sampling time respecting the Nyquist theorem, and $k$ are the samples of the continuous response.

Given that the state and the control action on the time instant $t(k-1)$ are known, the system's state at instant $t(k)$ can be approximated by Euler's method [27] as

$$\frac{\boldsymbol{\eta}(k)-\boldsymbol{\eta}(k-1)}{T_0} = f\Big(\boldsymbol{\eta}(t), \dot{\boldsymbol{\zeta}}(t)\Big). \tag{8}$$

The design of the kinematic controller was based on the kinematic transformation of the mobile vehicles cooperation. In order to design a formation controller or a path following, the kinematic transformation can be approximated as:

$$\frac{\boldsymbol{\eta}(k)-\boldsymbol{\eta}(k-1)}{T_0} = \boldsymbol{\Gamma}(\zeta(k))\dot{\zeta}(k) \tag{9}$$

where $\boldsymbol{\Gamma}(\boldsymbol{\zeta}(k))$ contains the characteristics of both the positioning and the formation of all the robotic systems in the case of the formation controller or the positioning information of the object in the case of the path following.

It is important to mention that following a path consists of maintaining the position and orientation of the object to be transported within a predefined route without parameterization in time. In this way, the control objective is to position the object at the closest point of the marked path P at the desired velocity $\boldsymbol{v}_d$. To reach this goal, the following expression is considered,

$$\frac{\boldsymbol{\eta}(k)-\boldsymbol{\eta}(k-1)}{T_0} = \boldsymbol{v_d} + \frac{\boldsymbol{W}(\boldsymbol{\eta_d}(k-1)-\boldsymbol{\eta}(k-1))}{T_0} \tag{10}$$

where $\boldsymbol{\eta_d}(k-1)$ is a point of the desired path with the desired shape at the previous instant of $k$ and $\boldsymbol{W}(\widetilde{\boldsymbol{\eta}}(k-1))$ is a diagonal matrix that weights control errors $\widetilde{\boldsymbol{\eta}}(k-1) = \boldsymbol{\eta_d}(k-1) - \boldsymbol{\eta}(k-1)$, defined as:

$$\boldsymbol{W}(\widetilde{\eta}_m(k-1)) = \frac{w_m}{1+\left|\widetilde{\eta}_m(k-1)\right|}, \tag{11}$$

with $\widetilde{\boldsymbol{\eta}}$ represents the error vector at the outputs of the controller (see Figure 3), and $m$ that represents each of the states of formation of the cooperative control. Now, Equations (9) and (10) can be considered to generate the system of equations

$$\boldsymbol{\Gamma}(\zeta(k))\dot{\zeta}(k) = \boldsymbol{v_d} + \frac{\boldsymbol{W}(\boldsymbol{\eta_d}(k-1)-\boldsymbol{\eta}(k-1))}{T_0}, \tag{12}$$

which let us rewrite the system as $\boldsymbol{A}\boldsymbol{u} = \boldsymbol{b}$, where $\boldsymbol{A} = \boldsymbol{\Gamma}(\zeta(k)) \in \mathbb{R}^{m\times n}$, $\boldsymbol{u} = \dot{\zeta}(k) \in \mathbb{R}^{m}$, and $\boldsymbol{b} = \boldsymbol{v_d} + \frac{\boldsymbol{W}(\boldsymbol{\eta_d}(k-1)-\boldsymbol{\eta}(k-1))}{T_0}$, with $\boldsymbol{b} \in \mathbb{R}^{m}$. In this way, the control actions are defined as $\boldsymbol{u} = \boldsymbol{A}^{-1}\boldsymbol{b}$. In case $\boldsymbol{A}$ is quadratic (with $\det(\boldsymbol{A}) \neq 0$), $\boldsymbol{A}$ has a direct inverse solution, otherwise, it must be necessary to use a method to solve the pseudoinverse problem, given by:

$$\underbrace{\dot{\zeta}(k)}_{\boldsymbol{u}} = \underbrace{\boldsymbol{\Gamma}(\zeta(k))^{\boldsymbol{T}}\left(\Gamma(\zeta(k))\Gamma(\zeta(k))^{\boldsymbol{T}}\right)^{-1}}_{\boldsymbol{A}^{-1}}\underbrace{\left(\dot{\boldsymbol{\eta}}_{\boldsymbol{d}}(k) + \frac{\boldsymbol{W}(\boldsymbol{\eta_d}(k-1)-\boldsymbol{\eta}(k-1))}{T_0}\right)}_{\boldsymbol{b}}. \tag{13}$$

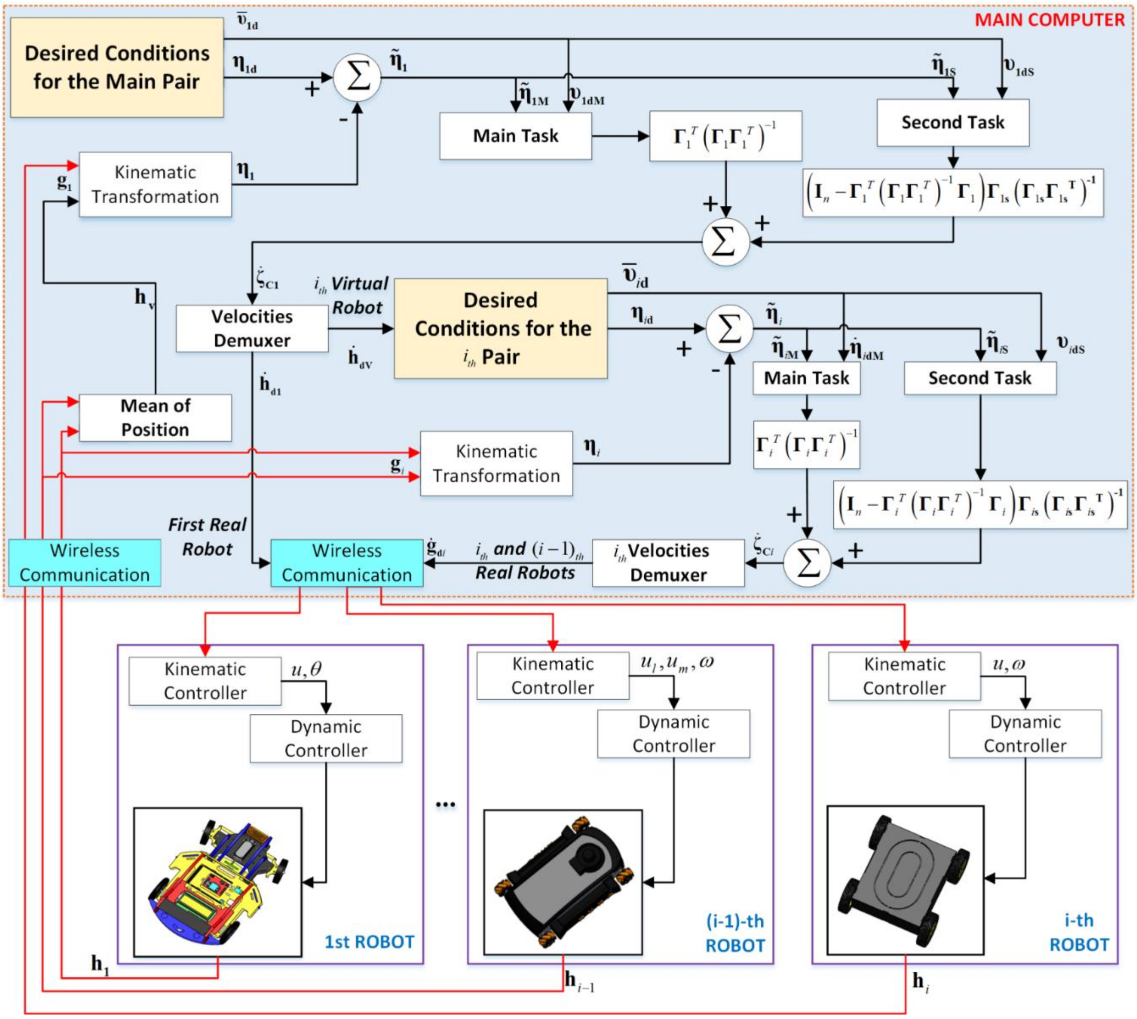


**Figure 3.** Implemented control structure.

**Remark 2:** *The pseudo-inverse here applied was that introduced by Moore and Penrose. It has the property that it thus distributes the coefficients of the redundant columns in the solution that the sum of squares of these coefficients was minimized [28]. Unfortunately, in cases where loses rank, it is known that the pseudo-inverse approach will not always avoid singular configurations. In singularities, the original task is replaced by a solution of which does not precisely correspond to the original one. By means of the use of the Gram-Schmidt algorithm or the Singular Value Decomposition, pseudoinverses can be constructed in which the singularities can be avoided, but that is beyond the scope of this work.*

Furthermore, it is said that a system of linear equations is homogeneous if it can be written in the form $\boldsymbol{Au} = \mathbf{0}$. Now, assuming that the configuration of the robotic system (13) is redundant, the Jacobian matrix $\boldsymbol{\Gamma}(\boldsymbol{\zeta}(k)) \in \mathbb{R}^{m \times n}$ has more unknowns than equations ($m < n$) with range $r = n$ for each $\boldsymbol{b}$, and taking into account that the homogenous equation has a not trivial solution, the system could have infinite solutions. In this case, suppose the equation $\boldsymbol{Au} = \boldsymbol{b}$ is consistent for a $\boldsymbol{b}$ given and letting $\boldsymbol{v}_p$ be a particular solution, the solution is the set of all the vectors of the form

$$\boldsymbol{v} = \boldsymbol{v}_p + \boldsymbol{v}_h, \tag{14}$$

with $\boldsymbol{v}_h$ as any solution of the homogeneous system $\boldsymbol{\Gamma}(\boldsymbol{\zeta}(k))\boldsymbol{v}_h = \mathbf{0}$.

A viable solution method is to formulate the problem as a constrained linear optimization problem

$$\frac{1}{2}\|v\|_2{}^2 = min, \tag{15}$$

yielding the particular solution

$$\boldsymbol{v_p} = \boldsymbol{\Gamma}(\boldsymbol{\zeta}(k))^{\boldsymbol{T}}(\Gamma(\boldsymbol{\zeta}(k))\Gamma(\boldsymbol{\zeta}(k))^{\boldsymbol{T}})^{-1}\boldsymbol{b}. \tag{16}$$

On the other hand, the null space of $\boldsymbol{\Gamma}(\zeta(\mathrm{k}))$ in $\mathbb{R}^{\mathrm{n}}$ is the set of velocities that do not produce any effect over the actions of the heterogeneous robotic set. In that case, it is necessary to rethink the cost function, expressing it as

$$\frac{1}{2}\|v - \boldsymbol{v}_0\|_2{}^2 = min. \tag{17}$$

that yields the homogeneous solution

$$v_h = \Big(\boldsymbol{I}_n - \boldsymbol{\Gamma}(\zeta(k))^{\boldsymbol{T}}\Big(\boldsymbol{\Gamma}(\zeta(k))\boldsymbol{\Gamma}(\zeta(k))^{\boldsymbol{T}}\Big)^{\boldsymbol{-1}}\boldsymbol{\Gamma}(\zeta(k))\Big)v_0. \tag{18}$$

Thereby, inserting Equations (16) and (18) into (14), it yields the proposed law of control

$$v_c = \underbrace{\boldsymbol{\Gamma}(\zeta(k))^T\Big(\boldsymbol{\Gamma}(\zeta(k))\boldsymbol{\Gamma}(\zeta(k))^T\Big)^{-1}\boldsymbol{b}}_{v_p} + \underbrace{\Big(\boldsymbol{I}_n - \boldsymbol{\Gamma}(\zeta(k))^T\Big(\boldsymbol{\Gamma}(\zeta(k))\boldsymbol{\Gamma}(\zeta(k))^T\Big)^{-1}\boldsymbol{\Gamma}(\zeta(k))\Big)\boldsymbol{v_0}}_{v_h}, \tag{19}$$

where the first term on the left-hand side is the particular solution $\boldsymbol{v_p}$ and second term $\boldsymbol{v_h}$ of this equation belong to the null space of $\boldsymbol{\Gamma}(\zeta(k))$.

For the purposes of this work, the second term in Equation (19) represents the projection on the null space of the robotic systems, where $\boldsymbol{v_0}$ is an arbitrary vector that contains the velocities associated with the shape and orientations of the robotic set. Therefore, any value given to $\boldsymbol{v_0}$ will affect the internal structure of the formation only, but not the final control of the first objective at all. The null space created by the high-level task matrix allows each velocity to be projected onto that space, where the sub-tasks compete to solve the problem in different ways. However, the velocities of the second task need to be calculated and included in $\boldsymbol{v_0}$, being

$$\boldsymbol{v_0} = \boldsymbol{\Gamma}_s(\zeta(k))\Big(\boldsymbol{\Gamma}_s(\zeta(k))\boldsymbol{\Gamma}_s(\zeta(k))^{\boldsymbol{T}}\Big)^{\boldsymbol{-1}}\Big(\dot{\boldsymbol{\eta}}_{\boldsymbol{d}}(k) + \frac{\boldsymbol{W_S}(\boldsymbol{\eta_{dS}}(k-1) - \boldsymbol{\eta_S}(k-1))}{T_0}\Big), \tag{20}$$

where $\boldsymbol{\Gamma}_s(\zeta(k))$ is the Jacobian matrix, which contains the secondary objectives.

**Remark 3:** *The consideration of having more than one robot working on the same purpose allows achieving secondary tasks at the same time collectively. To achieve these objectives, a behavior-based controller can be implemented to split tasks into sub-objectives, solving problems separately, and finally combining them to obtain the final solution.*

### *4.2. Formation and Following Control*

The cooperative control of heterogeneous mobiles considers that the type of robots incorporated into the formation system includes its own kinematic controller and a dynamic compensator, therefore, for the formation control, the type of system included is transparent. This section briefly describes the basic control scheme to maintain the formation of many robotic systems, a basis to integrate more robots in a scalable way. Figure 3 shows the control structure implemented for this work, denoting both the controllers which maintain the formation, orientation, and following of the desired path and the controllers that correct the final actions of each robot.

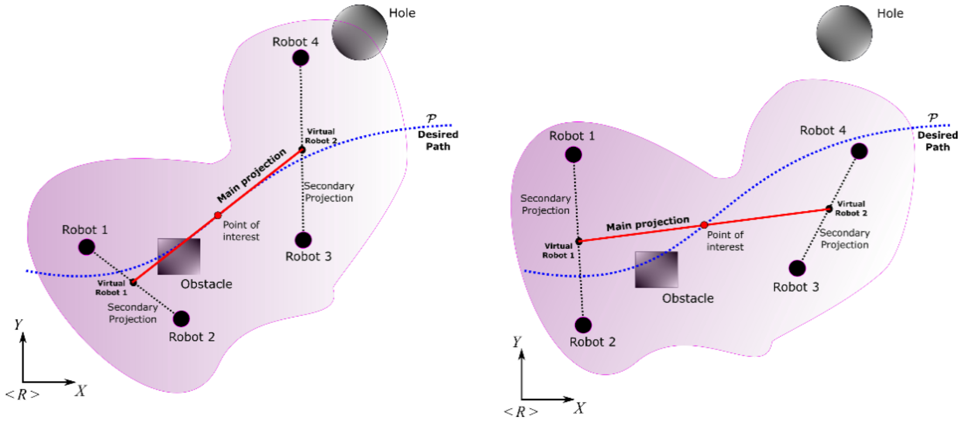
The adoption of the null-space approach was due to the possibility of treating tasks and subtasks separately, which were unified at the end to get the control actions. Since the main task does not conflict with the secondary task, the designer of the controllers selected the task hierarchy. By means of this, different control objectives can be achieved, e.g., maximum manipulability, energy saving, obstacle avoidance, and so on [29]. In this case, the main task was to reach that the set of robots followed a path, while the secondary task was to maintain the geometric shape and orientation of every projection formed by the vehicles.

The unified Jacobian matrix in Equation (9) contains both the first derivatives corresponding to the positions of the center point of a formation, as well as the distance and the orientation between the two robots forming the projection. For the calculation of different control actions, it is taken into account that the Jacobian matrix in Equation (9) is divided into two parts:

$$\begin{bmatrix}\overbrace{\begin{matrix}\dot{x}_O\\ \dot{y}_O\end{matrix}}^{\dot{\boldsymbol{\eta}}_{1M}}\\ \underbrace{\begin{matrix}\dot{d}_O\\ \dot{\theta}_O\end{matrix}}_{\dot{\boldsymbol{\eta}}_{1S}}\end{bmatrix}=\begin{bmatrix}\overbrace{\begin{matrix}\frac{\partial x_O}{\partial x_1} & \frac{\partial x_O}{\partial y_1} & \frac{\partial x_O}{\partial x_{V1}} & \frac{\partial x_O}{\partial y_{V1}}\\ \frac{\partial y_O}{\partial x_1} & \frac{\partial y_O}{\partial y_1} & \frac{\partial y_O}{\partial x_{V1}} & \frac{\partial y_O}{\partial y_{V1}}\end{matrix}}^{\boldsymbol{\Gamma}_{1M}}\\ \underbrace{\begin{matrix}\frac{\partial d_O}{\partial x_1} & \frac{\partial d_O}{\partial y_1} & \frac{\partial d_O}{\partial x_{V1}} & \frac{\partial d_O}{\partial y_{V1}}\\ \frac{\partial \theta_O}{\partial x_1} & \frac{\partial \theta_O}{\partial y_1} & \frac{\partial \theta_O}{\partial x_{V1}} & \frac{\partial \theta_O}{\partial y_{V1}}\end{matrix}}_{\boldsymbol{\Gamma}_{1S}}\end{bmatrix}\begin{bmatrix}\dot{x}_1\\ \dot{y}_1\\ \dot{x}_{V1}\\ \dot{y}_{V1}\end{bmatrix} \tag{21}$$

where $\boldsymbol{\Gamma}_{1M}(\zeta(k)) \in R^{2\times4}$ are the first derivatives of the midpoint positions of the projection, $\boldsymbol{\Gamma}_{1S}(\zeta(k)) \in R^{2\times4}$ are the first derivatives of the distance and orientation angle, $\dot{\boldsymbol{\eta}}_{\boldsymbol{1M}} \in R^{2\times1}$ corresponds to the first variable to be calculated and $\dot{\boldsymbol{\eta}}_{1S} \in R^{2\times1}$ the second variable to calculate.

Considering the proposed splitting in Equation (21), differentiated controls can be proposed to achieve both the primary and secondary objectives. The work proposes as the main objective of the point of interest $\boldsymbol{p_O}$ (redefined as $\boldsymbol{\eta_{1M}}$ for the application of the proposed controller) follows a profile not parameterized in time, with a velocity $\boldsymbol{\upsilon_d}$ imposed by the designer. On the other hand, the secondary objective was to modify the formation of robotic systems, as long as they did not conflict with satisfying the first objective. Having a formation based on projections, the main projection $\boldsymbol{\gamma_O}$ plays an important role since it is the basis for calculating the velocities that the rest of the projections will adopt to fulfill the tasks. Additionally, having a main straight line that can be projected and oriented according to the requirements of the controller designer allows locating the robotic systems to be agreed upon. Contrast the possibilities (a) and (b) of Figure 4; the first projection has a different length and orientation to the second projection. However, the main objective was achieved in both, where the variation in distances and secondary orientations did not conflict with the main task. The following paragraphs describe the characteristics mentioned in more detail.



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

**Figure 4.** Control possibilities considering the path following and formation of the robotic set.

(a) Path following controller

To achieve that the center of the projection $\boldsymbol{\eta}_{iM}$ follows the desired path, the proposed controller is defined by

$$\dot{\zeta}_{iM}(k) = \mathbf{\Gamma}_{iM}(\zeta(k))^T\Big(\mathbf{\Gamma}_{iM}(\zeta(k))\mathbf{\Gamma}_{iM}(\zeta(k))^T\Big)^{-1}\Big(\bar{\boldsymbol{v}}_{idM} + \boldsymbol{W}_{iM}(\widetilde{\boldsymbol{\eta}}_{iM}(k-1))\Big), \tag{22}$$

where $\mathbf{\Gamma}_{i\mathbf{M}}(\zeta(k))$ is the first part of the general Jacobian that determines the characteristics of the robotic set to follow the desired path, with $\widetilde{\boldsymbol{\eta}}_{iM}(k-1) = \boldsymbol{\eta}_{diM}(k-\mathbf{1}) - \boldsymbol{\eta}_{iM}(k-1)$. The obtained response in $\dot{\zeta}_{iM}(k)$ are two pairs of velocities that will be applied to the robots located at the ends of the projection, whether physical or virtual. These velocities are due to maintain the point of interest in the desired path but leaving aside the errors of the distance between the robots and the angle formed with respect to $\langle R\rangle$. It is important to note that the main projection (where the point of interest of the transported object is located) can have the desired velocity, in this case, $\bar{\boldsymbol{v}}_{idM} = \boldsymbol{v}_{dM}$, while the rest of projections will have velocities generated by the main controller, i.e., $\bar{\boldsymbol{v}}_{idM} = \dot{\boldsymbol{\eta}}_{idM}$.

(b) Formation and Guidance Control

This part of the control allows the establishment of the desired formation $\boldsymbol{\eta}_{iS}$ as a secondary objective, providing flexibility to projection modifications between the robots in applications where the tasks of the tracking targets were required, e.g., given the presence of the obstacles on the road. The second part of the Jacobian matrix is considered in this controller, which provides the characteristics of formation and orientation $\mathbf{\Gamma}_{iS}(\zeta(k))$ of both robotic systems. In this way, the controller is defined by

$$\dot{\zeta}_{\mathbf{2}}(k) = \mathbf{\Gamma}_{iS}(\zeta(k))^{\boldsymbol{T}}\Big(\mathbf{\Gamma}_{iS}(\zeta(k))\mathbf{\Gamma}_{iS}(\zeta(k))^{\boldsymbol{T}}\Big)^{-\mathbf{1}}(\boldsymbol{v}_{dS} + \boldsymbol{W}_{\mathbf{2}}(\widetilde{\boldsymbol{\eta}}_{iS}(k-1))). \tag{23}$$

Contrary to a), this controller aims to correct the shape of the $i_{th}$ projection, zeroing distance errors between the robots, and general orientation of the robotic pair.

(c) Unified Controller

This section defines the priority in the control tasks that the group of robots will have. The work proposes the positioning control of the point of interest (in this case, the midpoint of the main projection). Therefore, the task that does not conflict with the main control objective is the form and orientation of the projection. Based on (19), the proposed final controller is presented in the form

$$\begin{gathered}\dot{\zeta}_{\boldsymbol{C}}(k) = \mathbf{\Gamma}_{iM}(\zeta)^T\Big(\mathbf{\Gamma}_{iM}(\zeta)\mathbf{\Gamma}_{iM}(\zeta)^T\Big)^{-1}\Big(\bar{v}_{id\boldsymbol{M}} + \frac{\boldsymbol{W}(\widetilde{\boldsymbol{\eta}}_{iM}(k-1))}{T_0}\Big) + \\ \Big(\boldsymbol{I}_n - \mathbf{\Gamma}_{iM}(\zeta)^T\Big(\mathbf{\Gamma}_{iM}(\zeta)\mathbf{\Gamma}_{iM}(\zeta)^T\Big)^{-1}\mathbf{\Gamma}_{iM}(\zeta)\Big)\mathbf{\Gamma}_{iS}(\zeta)\Big(\mathbf{\Gamma}_{iS}(\zeta)\mathbf{\Gamma}_{iS}(\zeta)^{\boldsymbol{T}}\Big)^{-\mathbf{1}}\Big(v_{id\boldsymbol{S}} + \frac{\boldsymbol{W}_{\boldsymbol{S}}(\widetilde{\boldsymbol{\eta}}_{iS}(k-1))}{T_0}\Big),\end{gathered} \tag{24}$$

where the second part of this controller shows the attempt to correct the errors so long as it does not interfere with the main task, which is carried out by the first part of the controller. This fusion allows having the desired velocities for the pair of robots that form the $i_{th}$ projection.

**Remark 4:** *Cases (a), (b), and (c) are applicable to one of the i possible projections, requiring i controllers to be programmed as projections exist. This facilitates the scalability of the system, while at the same time making the robotics set more flexible by giving desired parameters to each of the projections at any time.*

**Remark 5:** *In the case study, $\boldsymbol{v}_{idS}$ can be imposed by the controller designer, but it must be directly related to the error reduction, i.e., a velocity can be configured with which the shape is corrected to desired parameters, but it must have a null value in the event the objectives are achieved, otherwise, errors caused by the same correction action can appear. One way to solve this is to force it to be immersed within the weight given by $\boldsymbol{W_S}$.*

(d) Stability Analysis

Rewriting the proposed control law (24) in a way

$$\dot{\zeta}_C(k) = \boldsymbol{\Gamma}^{\#}_{\boldsymbol{M}}\left(\bar{v}_{dM} + \frac{\boldsymbol{W_M}(\widetilde{\boldsymbol{\eta}}_{\boldsymbol{M}}(k-1))}{T_0}\right) + \left(\boldsymbol{I} - \boldsymbol{\Gamma}^{\#}_{\boldsymbol{M}}\boldsymbol{\Gamma_M}\right)\boldsymbol{\Gamma}^{\#}_{\boldsymbol{S}}\left(v_{dS} + \frac{\boldsymbol{W_S}(\widetilde{\boldsymbol{\eta}}_{\boldsymbol{S}}(k-1))}{T_0}\right), \tag{25}$$

where $\boldsymbol{\Gamma}^{\#}_{\boldsymbol{M}} = \boldsymbol{\Gamma}_M(\zeta)^T\left(\boldsymbol{\Gamma}_M(\zeta)\boldsymbol{\Gamma}_M(\zeta)^T\right)^{-1}$ and $\boldsymbol{\Gamma}^{\#}_{S} = \boldsymbol{\Gamma}_S(\zeta)^T\left(\boldsymbol{\Gamma}_S(\zeta)\boldsymbol{\Gamma}_S(\zeta)^T\right)^{-1}$ represent a pseudo-inverse matrix notation, the analysis can be developed in a simpler manner.

Now, the analysis is done for the control of the main objective, in addition, it is considered that the secondary objective of the formation is not affected when Equation (25) is multiplied by $\boldsymbol{\Gamma_M}$.

$$\begin{aligned}\boldsymbol{\Gamma_M}\dot{\zeta}_C(k) = \boldsymbol{\Gamma_M} \quad & \boldsymbol{\Gamma}^{\#}_{\boldsymbol{M}}\left(\bar{v}_{dM} + \tfrac{\boldsymbol{W_M}(\widetilde{\boldsymbol{\eta}}_{\boldsymbol{M}}(k-1))}{T_0}\right) \\ & + \boldsymbol{\Gamma_M}\left(\boldsymbol{I} - \boldsymbol{\Gamma}^{\#}_{\boldsymbol{M}}\boldsymbol{\Gamma_M}\right)\boldsymbol{\Gamma}^{\#}_{\boldsymbol{S}}\left(v_{dS} + \tfrac{\boldsymbol{W_S}(\widetilde{\boldsymbol{\eta}}_{\boldsymbol{S}}(k-1))}{T_0}\right)\end{aligned} \tag{26}$$

then, as $\boldsymbol{\Gamma_M}\boldsymbol{\Gamma}^{\#}_{\boldsymbol{M}} = \boldsymbol{I}$ and $\boldsymbol{\Gamma_M}\left(\boldsymbol{I} - \boldsymbol{\Gamma}^{\#}_{\boldsymbol{M}}\boldsymbol{\Gamma_M}\right) = 0$, Equation (26) is reduced to

$$\boldsymbol{\Gamma_M}\dot{\zeta}_C(k) = \left(\bar{v}_{dM} + \frac{\boldsymbol{W_M}(\widetilde{\boldsymbol{\eta}}_{\boldsymbol{M}}(k-1))}{T_0}\right) \tag{27}$$

The stability analysis for Equation (27) is done by means of the feedback, i.e., if it is considered that Equation (21) be perfect tracking then, $\dot{\boldsymbol{\eta}}_{\boldsymbol{M}}(k) \cong \boldsymbol{\Gamma_M}\dot{\zeta}_C(k)$, and taking into account that the error for the primary objective is defined as $\boldsymbol{\eta_{dM}}(k-1) - \boldsymbol{\eta_M}(k-1)$ such that

$$\dot{\boldsymbol{\eta}}_{\boldsymbol{M}}(k) = \bar{v}_{dM} + \boldsymbol{W_M}\left(\frac{\boldsymbol{\eta_{dM}}(k-1) - \boldsymbol{\eta_M}(k-1)}{T_0}\right); \tag{28}$$

therefore, replacing the velocity for the main objective $\dot{\boldsymbol{\eta}}_{\boldsymbol{M}}(k) = \frac{\Delta\dot{\eta}_M(k)}{T_0} = \frac{\boldsymbol{\eta_M}(k) - \boldsymbol{\eta_M}(k-1)}{T_0}$ in Equation (28) and also considering that the desired velocity for the path is $\bar{v}_{dM} = \dot{\boldsymbol{\eta}}_{\boldsymbol{dM}} - \dot{\boldsymbol{\gamma}}_{\boldsymbol{M}}$:

$$\frac{\boldsymbol{\eta_M}(k) - \boldsymbol{\eta_M}(k-1)}{T_0} = \dot{\boldsymbol{\eta}}_{\boldsymbol{dM}} - \dot{\boldsymbol{\gamma}}_{\boldsymbol{M}} + \boldsymbol{W_M}\left(\frac{\boldsymbol{\eta_{dM}}(k-1) - \boldsymbol{\eta_M}(k-1)}{T_0}\right). \tag{29}$$

Next, the differential derivative is applied to obtain $\dot{\boldsymbol{\eta}}_{\boldsymbol{dM}}(k) = \frac{\boldsymbol{\eta_{dM}}(k) - \boldsymbol{\eta_{dM}}(k-1)}{T_0}$ and the same way to $\dot{\boldsymbol{\gamma}}_{\boldsymbol{M}}(k) = \frac{\Delta\boldsymbol{\gamma_M}}{T_0}$, then the expression

$$\frac{\boldsymbol{\eta_M}(k) - \boldsymbol{\eta_M}(k-1)}{T_0} = \frac{\boldsymbol{\eta_{dM}}(k) - \boldsymbol{\eta_{dM}}(k-1)}{T_0} - \frac{\Delta\boldsymbol{\gamma_M}}{T_0} + \boldsymbol{W_M}\left(\frac{\boldsymbol{\eta_{dM}}(k-1) - \boldsymbol{\eta_M}(k-1)}{T_0}\right) \tag{30}$$

Simplifying $T_0$ and grouping Equation (30) with the current states $(k)$ and the previous state $(k-1)$ there is

$$\begin{aligned}(\boldsymbol{\eta}_{\boldsymbol{dM}}(k-1)- \quad & \boldsymbol{\eta}_{\boldsymbol{M}}(k-1)) \\ & = (\boldsymbol{\eta}_{\boldsymbol{dM}}(k)-\boldsymbol{\eta}_{\boldsymbol{M}}(k))-\Delta\boldsymbol{\gamma}_{\boldsymbol{M}}+\boldsymbol{W}_{\boldsymbol{M}}(\boldsymbol{\eta}_{\boldsymbol{dM}}(k-1)-\boldsymbol{\eta}_{\boldsymbol{M}}(k-1))\end{aligned} \tag{31}$$

Now, taking account that $\boldsymbol{\eta}_{\boldsymbol{d_M}}k-\boldsymbol{\eta}_{\boldsymbol{M}}k=\widetilde{\boldsymbol{\eta}}_{\boldsymbol{M}}k$ and $\boldsymbol{\eta}_{\boldsymbol{d_M}}(k-1)-\boldsymbol{\eta}_{\boldsymbol{M}}(k-1)=\widetilde{\boldsymbol{\eta}}_{\boldsymbol{M}}(k-1)$ represent the current error and the previous error, respectively, Equation (31) is reduced to

$$\widetilde{\boldsymbol{\eta}}_{\boldsymbol{M}}(k-1)=\widetilde{\boldsymbol{\eta}}_{\boldsymbol{M}}(k)-\Delta\boldsymbol{\gamma}_{\boldsymbol{M}}+\boldsymbol{W}_{\boldsymbol{M}}(\widetilde{\boldsymbol{\eta}}_{\boldsymbol{M}}(k-1)), \tag{32}$$

and grouping similar terms, it results that

$$\Delta\boldsymbol{\gamma}_{\boldsymbol{M}}=\widetilde{\boldsymbol{\eta}}_{\boldsymbol{M}}(k)+\boldsymbol{W}_{\boldsymbol{M}}(\widetilde{\boldsymbol{\eta}}_{\boldsymbol{M}}(k-1))-\widetilde{\boldsymbol{\eta}}_{\boldsymbol{M}}(k-1). \tag{33}$$

The primary objective error $\widetilde{\boldsymbol{\eta}}_{\boldsymbol{M}}(k)$ is separated to have the following expression

$$\Delta\boldsymbol{\gamma}_{\boldsymbol{M}}=\widetilde{\boldsymbol{\eta}}_{\boldsymbol{M}}(k)+\widetilde{\boldsymbol{\eta}}_{\boldsymbol{M}}(k-1)(\boldsymbol{W}_{\boldsymbol{M}}-\boldsymbol{I}), \tag{34}$$

therefore, stability in discrete time is realized by the $Z$ transform, and taking account the analysis for each diagonal matrix value $W_M$ so

$$\begin{aligned}\left(1-z^{-1}\right)\boldsymbol{\gamma}_{\boldsymbol{M}}(z) &= \widetilde{\boldsymbol{\eta}}_{\boldsymbol{M}}(z)+\widetilde{\boldsymbol{\eta}}_{\boldsymbol{M}}(z)z^{-1}(\boldsymbol{W}_{\boldsymbol{M}}-\boldsymbol{I}) \\ \widetilde{\boldsymbol{\eta}}(z) &= \tfrac{1-z^{-1}}{1+z^{-1}(\boldsymbol{W}_{\boldsymbol{M}}-\boldsymbol{I})}\gamma(z).\end{aligned} \tag{35}$$

Finally, the poles of the function obtained in the following way are analyzed:

$$1+z^{-1}(\boldsymbol{W}_{\boldsymbol{M}}-1)=0$$

$$z=1-\boldsymbol{W}_{\boldsymbol{M}} \tag{36}$$

Thus, for the control to be stable, the $Z$ plane pole must be inside the unit circle. Therefore, the values of the diagonal matrix must be considered as $0<\boldsymbol{diag}(\boldsymbol{W}_{\boldsymbol{M}})<1$, hence the control error $\widetilde{\boldsymbol{\eta}}_{\boldsymbol{M}}(k)$ come close to being zero when $k\rightarrow\infty$, then the system is asymptotically stable.

Now, as a second step, we proceed to effectuate the analysis for the secondary objective of formation. In a similar way to the primary objective, multiply both members of the control law by $\boldsymbol{\Gamma}_{\boldsymbol{S}}$, then Equation (25) it is written as:

$$\boldsymbol{\Gamma}_{\boldsymbol{S}}\dot{\zeta}_{C}(k)=\boldsymbol{\Gamma}_{\boldsymbol{S}}\boldsymbol{\Gamma}_{\boldsymbol{M}}^{\#}\left(\bar{v}_{dM}+\frac{\boldsymbol{W}_{\boldsymbol{M}}(\widetilde{\boldsymbol{\eta}}_{\boldsymbol{M}}(k-1))}{T_0}\right)+\boldsymbol{\Gamma}_{\boldsymbol{S}}\left(\boldsymbol{I}-\boldsymbol{\Gamma}_{\boldsymbol{M}}^{\#}\boldsymbol{\Gamma}_{\boldsymbol{M}}\right)\boldsymbol{\Gamma}_{\boldsymbol{S}}^{\#}\left(v_{dS}+\frac{\boldsymbol{W}_{\boldsymbol{S}}(\widetilde{\boldsymbol{\eta}}_{\boldsymbol{S}}(k-1))}{T_0}\right). \tag{37}$$

According to [28], where an important property of Jacobian matrices is analyzed when one does not have a conflict between them, i.e., that it is feasible to fulfill the two tasks simultaneously and completely, it results that

$$\boldsymbol{\Gamma}_{\boldsymbol{S}}\boldsymbol{\Gamma}_{\boldsymbol{M}}^{\#}=0. \tag{38}$$

Therefore, if Equations (36) in (37) are replaced, the below expression is obtained:

$$\boldsymbol{\Gamma}_{\boldsymbol{S}}\dot{\zeta}_{C}(k)=\boldsymbol{\Gamma}_{\boldsymbol{S}}\boldsymbol{\Gamma}_{\boldsymbol{S}}^{\#}\left(v_{dS}+\frac{\boldsymbol{W}_{\boldsymbol{S}}(\widetilde{\boldsymbol{\eta}}_{\boldsymbol{S}}(k-1))}{T_0}\right). \tag{39}$$

Additionally, considering Equation (21) to be perfect tracking, there is $\dot{\eta}_S(k)\cong\Gamma_S\dot{\zeta}_C(k)$. Performing a similar procedure of analysis that Equations (27) in (39), it is concluded in the same way that $z=I-W_S$, therefore the values of the diagonal matrix must be considered as 0 $<diag(W_S) < 1$, hence the control error $\widetilde{\eta}_S(k)$ come close to being zero when $k \to \infty$, then the system is asymptotically stable.

## 5. Kinematics Control and Dynamic Compensation of Each Robot

The velocities delivered by the position and formation controller are inputs for each of the robots, where the local controllers interpret independently the velocities as the profile $\dot{\boldsymbol{h}}_i$ to follow. This section describes the control of each of the heterogeneous robots that could be considered in the set, both for the kinematic control and for the correction of errors given by the dynamics of each system.

(a) Kinematics control

The kinematic model of a mobile robot gives the location of a point of interest to any part of the robotic system. The instantaneous kinematic model of a mobile robot gives the derivative of its point of interest as a function of the derivatives of the whole system itself

$$\dot{\boldsymbol{h}}_i(t) = \boldsymbol{J}_i(\boldsymbol{q})\boldsymbol{v}_i(t), \tag{40}$$

where $\dot{\boldsymbol{h}}_i = \begin{bmatrix} \dot{h}_{xi} & \dot{h}_{yi} \end{bmatrix}^T$ is the vector of the point of interest velocities, $\boldsymbol{v}_i = \begin{bmatrix} u_{li} & u_{mi} & \omega_i & \ldots \end{bmatrix}^T$ is the vector of the mobile robot velocities, which contains the linear and angular velocities of the mobile platform, $\boldsymbol{J}_i(\boldsymbol{q})$ is the Jacobian matrix, which defines a linear mapping between the vector of the mobile robot velocities $\boldsymbol{v}_i(t)$ and the vector of the point of interest velocities, and $i$ is the $i_{th}$ robot of the set.

Through the Euler's approximation of the kinematic model of whatever mobile robot, the following kinematic model discrete is obtained by:

$$\boldsymbol{h}(k+1) = \boldsymbol{h}(k) + \boldsymbol{T}_0\boldsymbol{J}(\boldsymbol{q}(k))\boldsymbol{v}(k), \tag{41}$$

where, values of $\mathbf{h}$ at the discrete time $t = kT_0$ will be denoted as $\mathbf{h}(k)$, with $T_0$ as the sample time, and $k \in \{0, 1, 2, 3, \ldots\}$. Next, by the Markov property and to adjusting the performance of the proposed control law, the states vector $\mathbf{h}(k+1)$ are replaced by:

$$\boldsymbol{h}(k+1) = \boldsymbol{h_d}(k+1) - \boldsymbol{W}(\boldsymbol{h_d}(k) - \boldsymbol{h}(k)), \tag{42}$$

where $\boldsymbol{W}$ is a diagonal matrix, which must satisfy 0 $<\boldsymbol{diag}\left(W_{hx}, W_{hy}\right) < 1$, allowing to reduce the variations in state variables, and $\boldsymbol{h_d}$ is the desired trajectory.

Then, from Equations (41) and (42), the following system of linear equations is obtained, which allows at each sampling instant to calculate the control actions

$$\boldsymbol{J}(\boldsymbol{q})\boldsymbol{v} = \boldsymbol{b}, \tag{43}$$

where $\boldsymbol{v} = \begin{bmatrix} u_l(k) & u_m(k) & \omega(k) \end{bmatrix}^T$ and:

$$\boldsymbol{b} = \frac{1}{T_0}\begin{bmatrix} h_{xd}(k+1) - w_{hx}(e_{hx}(k)) - h_x(k) \\ h_{yd}(k+1) - w_{hy}\left(e_{hy}(k)\right) - h_y(k) \end{bmatrix} \tag{44}$$

From Equation (43), its solution by least squares is obtained by solving the normal equations

$$\boldsymbol{v}_{ref} = \boldsymbol{J}(\boldsymbol{q})^T\left(\boldsymbol{J}(\boldsymbol{q})\boldsymbol{J}(\boldsymbol{q})^T\right)^{-1}\frac{1}{T_0}\begin{bmatrix} h_{xd}(k+1) - w_{hx}(e_{hx}(k)) - h_x(k) \\ h_{yd}(k+1) - w_{hy}\left(e_{hy}(k)\right) - h_y(k) \end{bmatrix} \tag{45}$$

(b) Dynamics compensation

This section details how the controller was designed to compensate the robot's dynamics in the workspace, achieving greater robustness for each robot. A dynamics controller was programed within each mobile platform, i.e., robotic platforms have an internal control that corrects linear and angular velocity errors.

Linear or angular errors can be caused by external forces or torques that affect the dynamics of the platforms. Given that all the robots received the maneuverability commands as input (linear and angular ones), the number of velocities depended on each mobile robot to meet the formation requirements. Figure 5 describes how the internal controller is structured to compensate for the dynamics of robots. In this case, the controller generates a control output defined as:

$$\mu_{\omega i}(k) = Kp_i\Big\{\widetilde{\omega}_i(k) + Ki_i T_o \textstyle\sum_{j=1}^{k}\widetilde{\omega}_i(j-1) + K_{Di}\dfrac{\widetilde{\omega}_i(k)-\widetilde{\omega}_i(k-1)}{T_o}\Big\}, \tag{46}$$

where, $\widetilde{\omega}_i(k) = \omega_i(k) - \bar{\omega}_i(k)$ represents the angular velocity error between the desired angular velocity $\omega_i(k)$ and the actual angular velocity $\bar{\omega}_i(k)$ with $i = 1,\ 2,\ 3,\dots$ ($i$ depends on the number of controlled traction wheels of each robot), the variables $Kp_i, Ki_i, K_{Di}$ are the gains that weigh the control error, and $T_o$ is the sampling time considered for the discrete PID (Proportional, Integral and Derivative control). Direct and reverse transformation matrices depend on the configuration of each mobile robot to achieve linear velocities, angular, and, eventually, the position and orientation of the mobile robot to feedback the control loop of the proposed cooperative scheme.

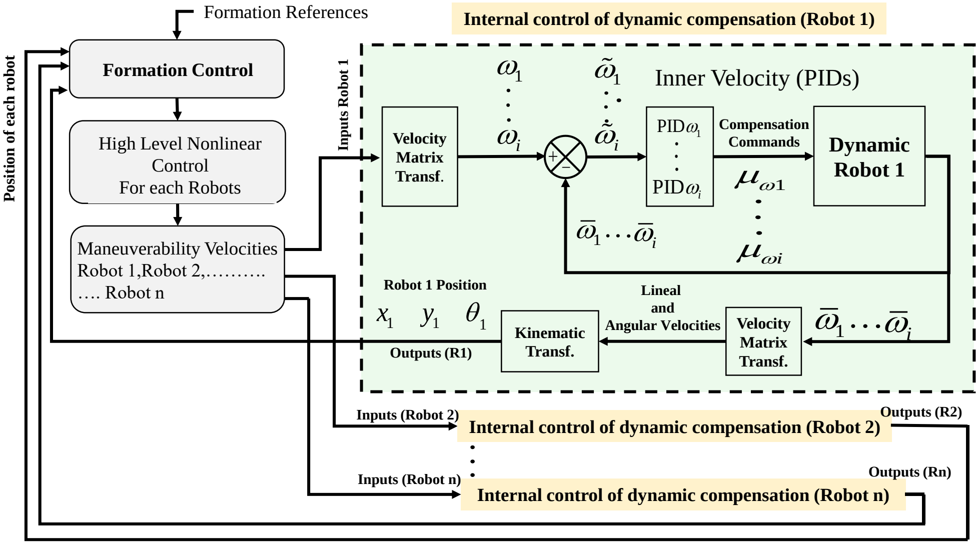


**Figure 5.** Control operation scheme for dynamic compensation.

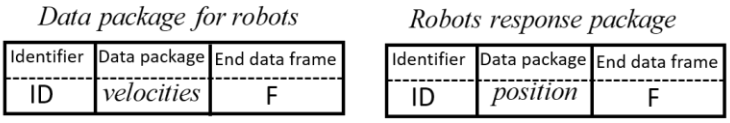
## 6. Robots Instrumentation

### 6.1. Sensors for Relative Position

Within the multiple robotic systems, electronic components were used to power the embedded processing units, know the state of the charge of the batteries, share information with remote receivers wirelessly, know the odometry of the system, and so on. Some of these sensors provided information on the state of each robotic system, allowing the robot to locate itself within their workspace in order to avoid collisions. The robots used for the experiments of this work have built-in sensors to estimate the position given a relative location, i.e., the wheels of each of the robots contain encoders that measure the velocity and the direction of rotation. These sensors determined the angular displacement, and thus the linear displacement of the robots, showing limitations when there were landslides or the resolution of the encoders was not adequate. The proposed control algorithm considers the knowledge of the current position of each one of the mobile robots at all times of the task; therefore, when starting the desired task, the initial condition was the position and orientation of each robot that was part of the collaborative task.

## *6.2. Communication System*

The communication structure proposed for the set of robots was based on the IEEE 802.15.4 protocol. This protocol allows the creation of multipoint networks, and it is also capable of transmitting a large amount of data (bytes) with very low latency and achieving a predictable communication synchronization. The XBEE S2 modules allow the implementation of the established protocol, configured in such a way that they are all within the same wireless network. The topology used in this work was the mesh configuration, i.e., all devices can communicate with each other in such a way that each robot has an identifier to know which data package corresponds to it. This provides the advantage if, at some instant of time the communication fails or is lost, communication can continue between all the other nodes (mobile robots) because the connectivity is forced to be restored. The coordinator of the network sends the maneuverability commands to each robot (linear velocities and angular velocities), where the velocities are packaged within a data frame. Figure 6 shows how the data packet sent by the master-robot and the package that responds to the slave-robots.



*Data package for robots*

| Identifier | Data package | End data frame |
|---|---|---|
| ID | *velocities* | F |

*Robots response package*

| Identifier | Data package | End data frame |
|---|---|---|
| ID | *position* | F |

**Figure 6.** Master-slave data package frame.

The proposed communication scheme complies with a sampling rate of 175 ms, i.e., send-receive and process data from robots within the proposed network. Figure 7 shows how the communication among the robots is structured into the cooperative system. Communications in this type of MESH configuration can eventually be made between 65,000 devices without losing their characteristics, according to the manufacturers data. In this work, the cost of increasing robots makes the frames larger, and that implies a reduction in the velocity of communications, which is established can not be less than 100 ms. Increasing the number of robots that exceed this margin would imply an application of a massive use of robots, but if this were the case, the restriction of communications in the network would have to be increased, reducing the velocity of the system itself: e.g., in an application of object transport, the whole would move at lower velocities.

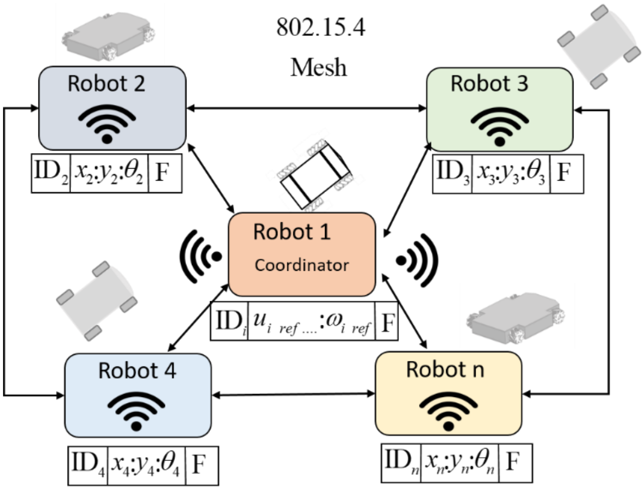

802.15.4
Mesh
Robot 2
$ID_2$ $x_2$:$y_2$:$\theta_2$ F
Robot 3
$ID_3$ $x_3$:$y_3$:$\theta_3$ F
Robot 1
Coordinator
$ID_i$ $u_{i\ ref}$....:$\omega_{i\ ref}$ F
Robot 4
$ID_4$ $x_4$:$y_4$:$\theta_4$ F
Robot n
$ID_n$ $x_n$:$y_n$:$\theta_n$ F

**Figure 7.** The mesh structure of communication used.

**Remark 6:** *The encoding and size of packets sent to each robot depend on the type of mobile platform, while the feedback information has similar sizes. As a MESH configuration was used in this work, the robots were able to exchange all the information between them, whether they were the destination or not; they received it to process it or only as a rebroadcast node. This means that the robots exchange all the information of the system as transport nodes but only process the information of which they are the addressees.*

## 7. Experimental Results

Experimental results are shown in this Section. To demonstrate the scalability of the proposed system, the fulfillment of tasks with two, three, and four robots were proposed. In order to include more robots, it was necessary to add the recognition of communication frames and a control segment for the new projection. At the computational level, this did not significantly increase the processing time since dynamic and kinematic controllers were carried out in each robot. An additional simulated result was included at the end, in order to validate the scalability of the control system. The control scheme presented in Figure 3 allows meeting the objectives of shape and position for a system of multiple heterogeneous robots. For the validation of the proposed controller, robots of different dynamic configurations were used in order to experience the scalability of the cooperative control. Figure 8 shows the robots used for the experiments performed. It can be seen that the set of robots was heterogeneous, with two holonomic omnidirectional robots of different characteristics: Type A was an omnidirectional robot, while the Type B robot was very similar but contained components of greater performance that made it able to carry heavy weights; the Type C robot was a normal unicycle robot, with two rear-wheel drive, and finally the Type D robot had a four-wheel drive.

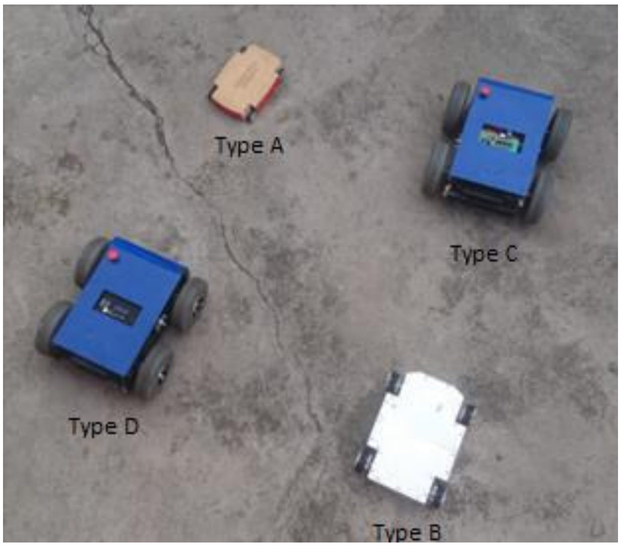


**Figure 8.** Robots considered for experimental results.

### 7.1. First Experiment

The first experiment consisted of executing a formation between a unicycle robot (Type C robot) and an omnidirectional robot (Type A robot). The desired path $\rho(s)$ is described by $x_{dO} = \frac{4}{5}\cos\left(\frac{27}{100}t\right) + \frac{13}{100}t{-}2\ [m]$, $y_{dO} = \frac{3}{10}t - 4\ [m]$, during 40 [sec] with $v_d = 0.2\left[\frac{m}{s}\right]$; take into account that the desired path may be the result of an offline path planning or planned online. The formation parameters defined for the experiment are: $d_O = 1\ [m]$ and $\theta_O = \frac{\pi}{2} + \tan^{-1}\left(\frac{\dot{y}_{dO}}{\dot{x}_{dO}}\right)\ [rad]$. Figure 9 illustrates the path executed by the robots, where the primary objective $(x_O, y_O)$ stays on the desired path. On the other hand, the second objective during the experiment meets the shape defined, i.e., orientation and distance.

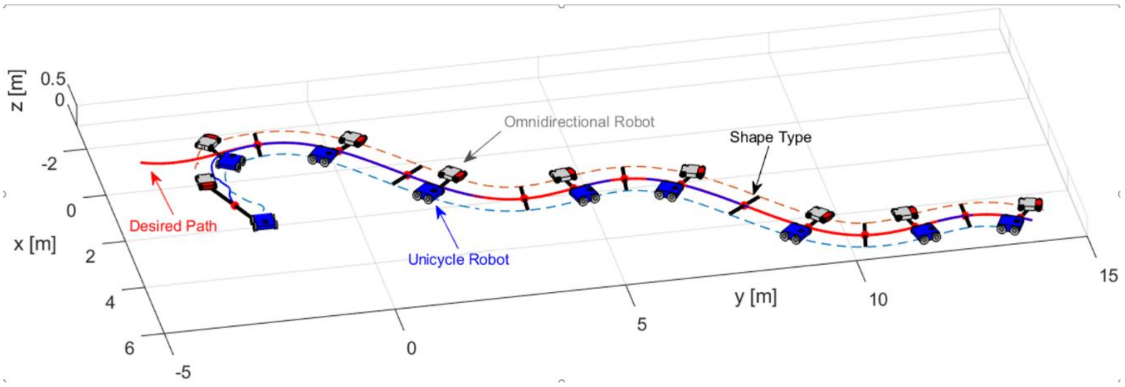


**Figure 9.** Movement executed by the robots.

Figure 10 presents the desired position errors $\widetilde{\boldsymbol{\eta}}(k)$ of the object. It is noted that at the beginning of the experiment, there was a major error that was caused by the initial positions of each robot. Once the point closest to the path was found, the mobile platforms were positioned in such a way that the position of the main objective was on the desired path.

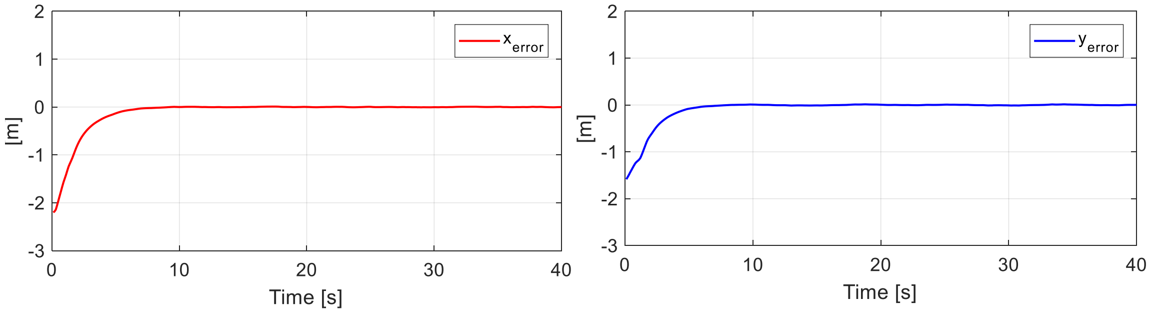


**Figure 10.** Errors of the main objective.

Figure 11 displays the error of the secondary objective, i.e., the distance error and angle error of formation between both the robots.

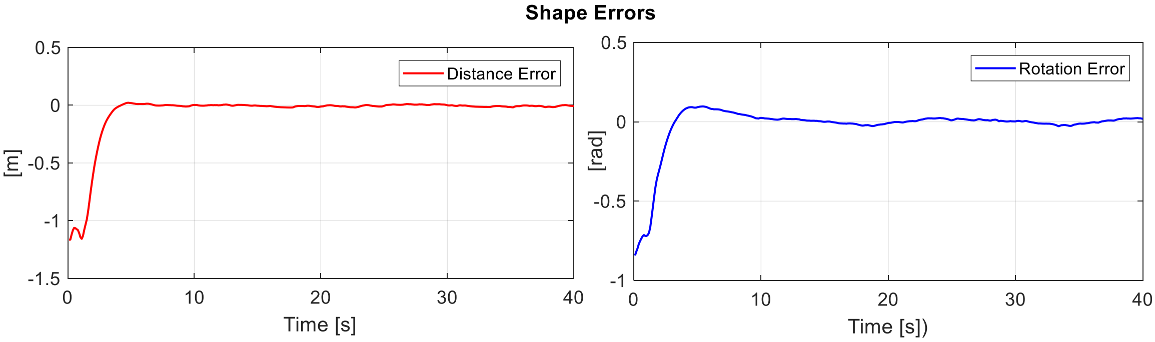


**Figure 11.** Errors of the secondary objective (distance and orientation).

For the planned experiment, the formation controls generated velocities references $\left[\begin{array}{cc}\dot{h}_1 & \dot{h}_2\end{array}\right]$, which by means of the kinematic control, each robot led to follow a trajectory to fulfill the formation expected. Therefore, in Figure 12, the position errors of each robot during the experiment are presented.

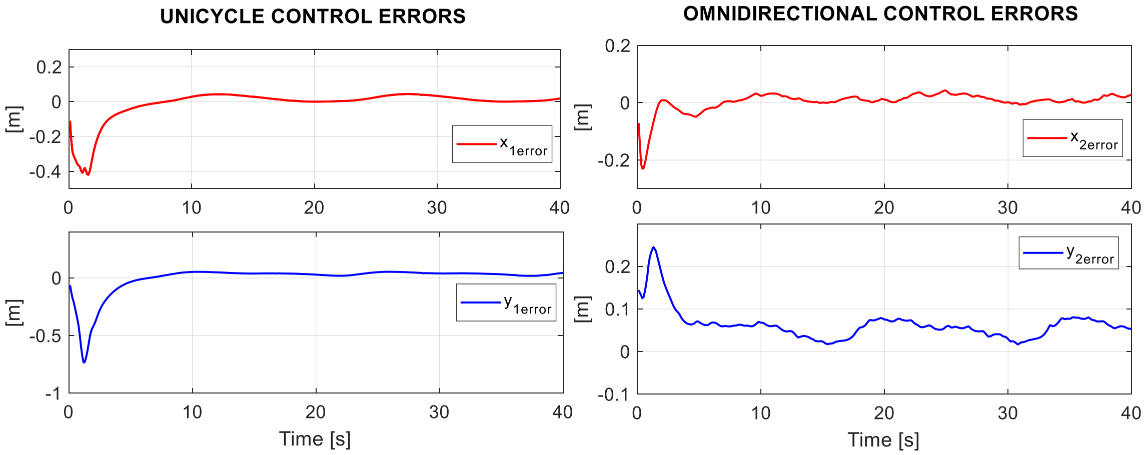


**Figure 12.** Position errors of the Unicycle and Omnidirectional robots during the experiment.

### 7.2. Second Experiment

The second experiment propounds to incorporate four robots for the formation, two unicycle-type mobile platforms (Type C and D robot), and two omnidirectional-type mobile platforms (Type A and B robot). In contrast, with the previous experiment, the desired formation for the four robots varied the shape parameters during the experimentation, the change in shape consisted of varying the angle and the defined virtual distances. Figure 13 indicates the shape parameters for the proposed cooperative control experiment, the number of parameters depends on the type of shape and the number of robots incorporated. Table 1 indicates the parameters desired for this experiment.

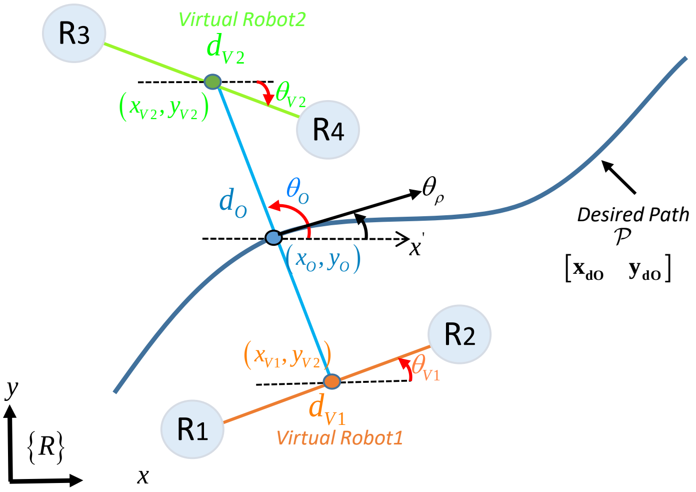


**Figure 13.** Formation parameters scheme for the second experiment.

**Table 1.** Parameters formation for the experiment.

| **Formation Parameter** | **0>$t$<30 [$s$]** | **30≥$t$≤50 [$s$]** |
|---|---|---|
| $\boldsymbol{v_d}$ | $0.15\ [m/s]$ | |
| $\boldsymbol{x_{dO}}$ | $\frac{3}{10}t+2\ [m]$ | |
| $\boldsymbol{y_{dO}}$ | $\frac{2}{5}\sin\left(\frac{1}{5}t\right)+\frac{1}{40}t-4\ [m]$ | |
| $\boldsymbol{\theta_\rho}$ | $\tan^{-1}\left(\frac{\dot{y}_{dO}}{\dot{x}_{dO}}\right)\ [rad]$ | |
| $\boldsymbol{d_O}$ | $1\ [m]$ | $2.5\ [m]$ |
| $\boldsymbol{\theta_O}$ | $\theta_\rho+\frac{\pi}{2}\ [rad]$ | $\tan^{-1}\left(\frac{\dot{y}_{dO}}{\dot{x}_{dO}}\right)\ [rad]$ |
| $\boldsymbol{d_{V1}}$ | $1\ [m]$ | $2.5\ [m]$ |
| $\boldsymbol{\theta_{V1}}$ | $\tan^{-1}\left(\frac{\dot{y}_{dO}}{\dot{x}_{dO}}\right)\ [rad]$ | $\frac{\pi}{6}\ [rad]$ |
| $\boldsymbol{d_{V2}}$ | $1\ [m]$ | $2\ [m]$ |
| $\boldsymbol{\theta_{V2}}$ | $\tan^{-1}\left(\frac{\dot{y}_{dO}}{\dot{x}_{dO}}\right)\ [rad]$ | $\frac{\pi}{6}\ [rad]$ |

Through the changing of shape, it was verified that during the execution of the task, the robots could autonomously modify the positions to move the object from one place to another following a defined path. Figure 14 shows the stroboscopic movement of the executed task, while in Figure 15, it is appreciated how the shape of the robotic set changes gradually.

Figure 16 shows the errors of the formation (main objective - secondary objectives), i.e., the errors of the main point $\left[\widetilde{x}_O, \widetilde{y}_O, \widetilde{d}_O, \widetilde{\theta}_O\right]$. In the same way, the errors of virtual parameters are denoted as $\left[\widetilde{x}_{V1}, \widetilde{y}_{V1}, \widetilde{d}_{V1}, \widetilde{\theta}_{V1}, \widetilde{x}_{V2}, \widetilde{y}_{V2}, \widetilde{d}_{V2}, \widetilde{\theta}_{V2}\right]$.

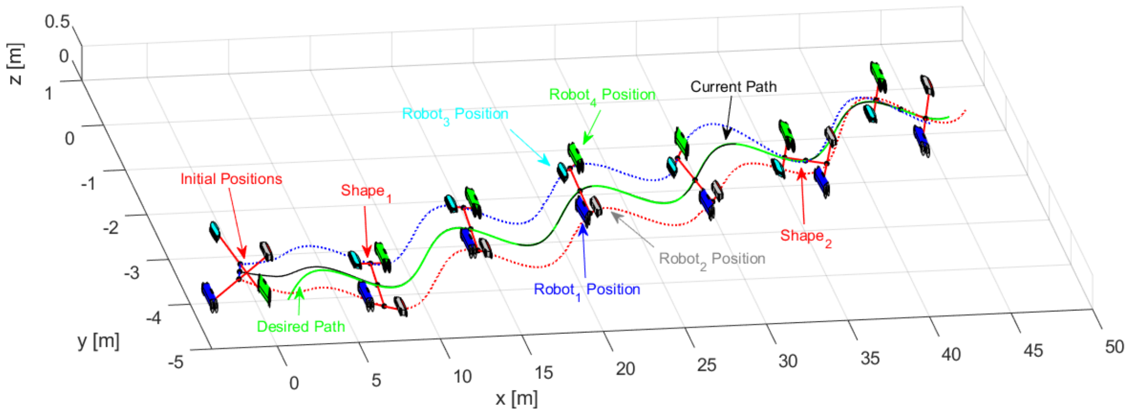


**Figure 14.** Robots movement to complete the task.

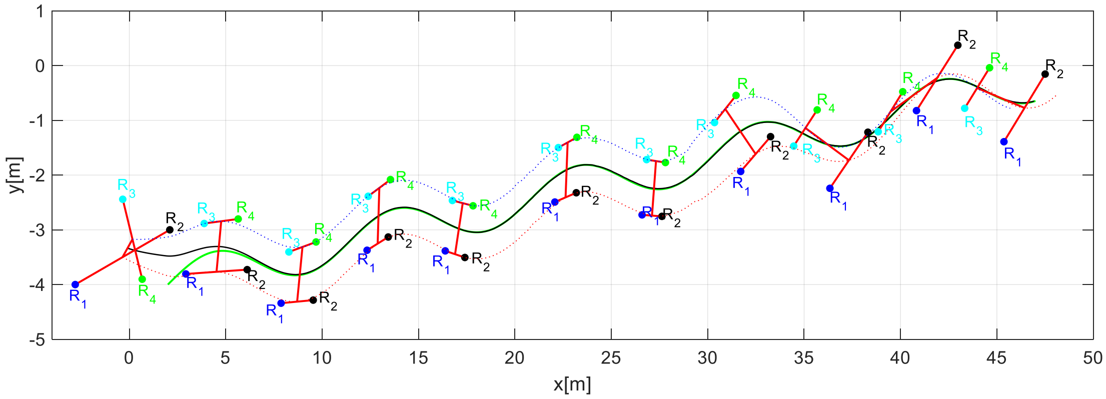
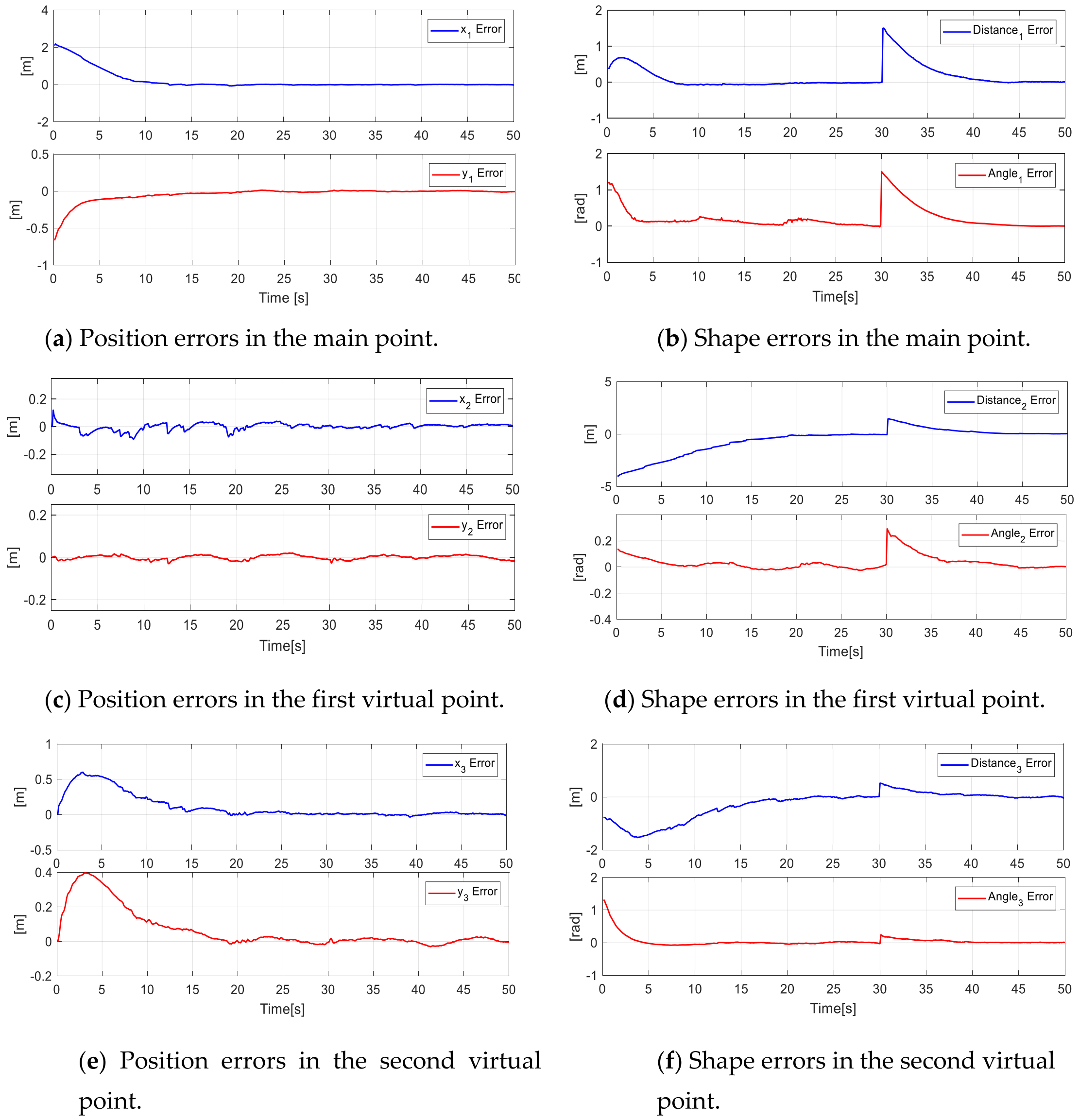


**Figure 15.** Stroboscopic movement of the second experiment.



(**a**) Position errors in the main point.

(**b**) Shape errors in the main point.

(**c**) Position errors in the first virtual point.

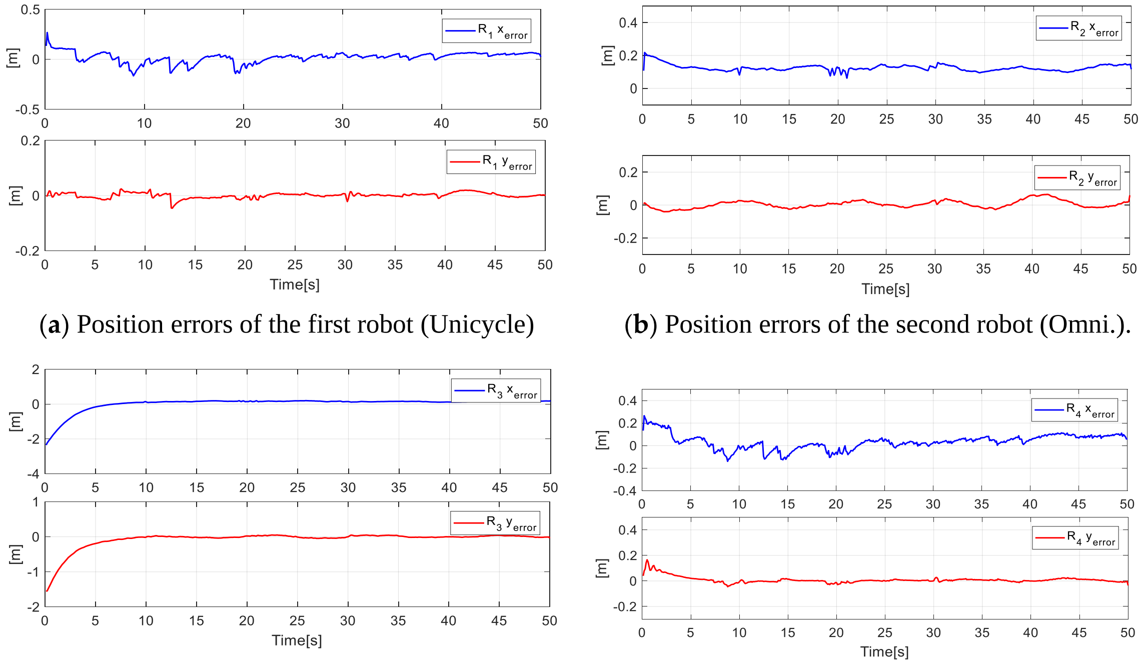
(**d**) Shape errors in the first virtual point.

(**e**) Position errors in the second virtual point.

(**f**) Shape errors in the second virtual point.

**Figure 16.** Main Objective errors and secondary objective errors at the primary point and virtual points of the formation.

Finally, Figure 17 indicates the position errors of each robot during the experimentation, where it is shown that the error is approaching zero of each robot in the executed experimentation.



(**a**) Position errors of the first robot (Unicycle)

(**b**) Position errors of the second robot (Omni.).

(**c**) Position error of the third robot (Omni.)

(**d**) Position error of robot four (Unicycle)

**Figure 17.** Position errors of each of the four robots for the desired shape.

### 7.3. Third Experiment

The third experiment was performed with three robots (Type C-D-A robots) changing the formation conditions of distances and orientations at an instant of time $t$. Table 2 indicates the desired parameters for the formation. Figure 18 shows how the robots comply with the requirements of position and shape.

**Table 2.** Formation parameters for the third experiment.

| **Formation Parameters** | **$0>t<30$ [$s$]** | **$30\geq t<50$ [$s$]** | **$50\geq t\leq 70$ [$s$]** |
|---|---|---|---|
| $\boldsymbol{v_d}$ | | $0.17\ [m/s]$ | |
| $\boldsymbol{x_{dO}}$ | | $\frac{2}{5}t+2\ [m]$ | |
| $\boldsymbol{y_{dO}}$ | | $\frac{1}{2}\sin\left(\frac{1}{5}t\right)+\frac{1}{20}t-4\ [m]$ | |
| $\boldsymbol{\theta_\rho}$ | | $\tan^{-1}\left(\frac{\dot{y}_{dO}}{\dot{x}_{dO}}\right)\ [rad]$ | |
| $\boldsymbol{d_O}$ | $2\ [m]$ | $2\ [m]$ | $2\ [m]$ |
| $\boldsymbol{\theta_O}$ | $\theta_\rho\ [rad]$ | $\theta_\rho+\frac{\pi}{2}\ [rad]$ | $\theta_\rho\ [rad]$ |
| $\boldsymbol{d_{V1}}$ | $2\ [m]$ | $1\ [m]$ | $2\ [m]$ |
| $\boldsymbol{\theta_{V1}}$ | $\frac{\pi}{2}\ [rad]$ | $0\ [rad]$ | $\frac{\pi}{2}\ [rad]$ |

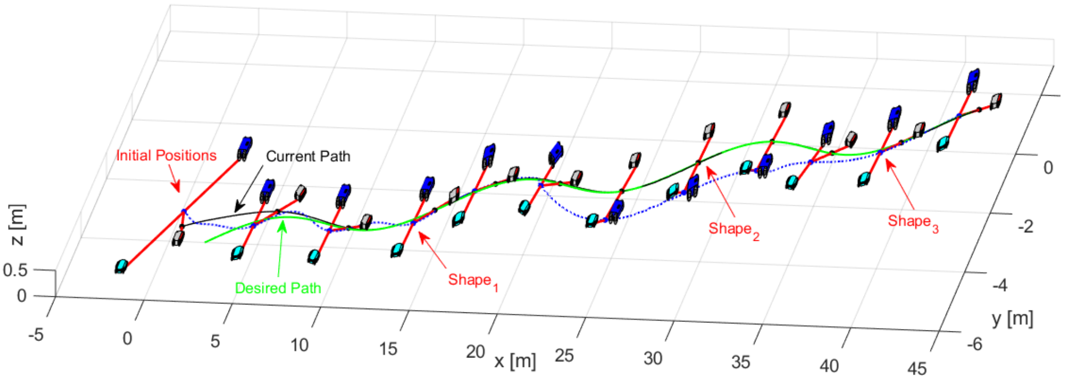


**Figure 18.** Evolution movement by the robots for the third experiment.

Figure 19, shows in detail the formation of robots as they follow the desired path, demonstrating that each robot changed the position according to the required shape.

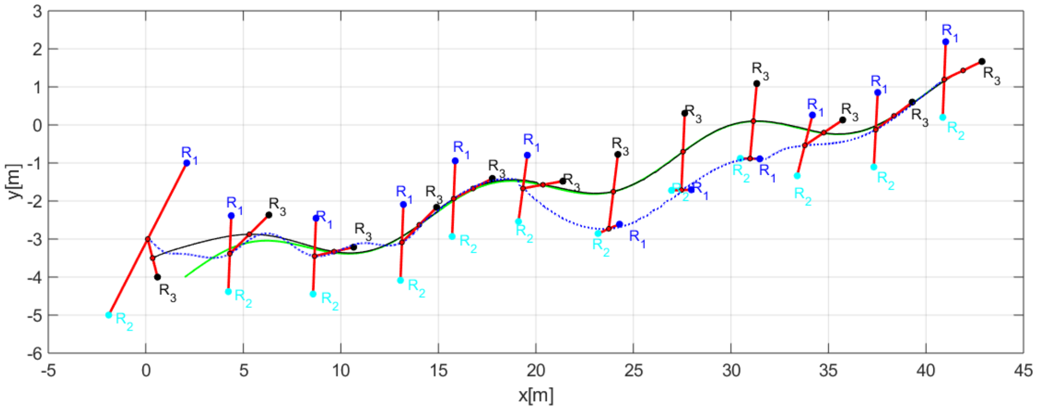


**Figure 19.** Position of each robot to complete the desired shapes.

Figure 20 shows how the movement of each robot evolves to comply with the desired shapes.

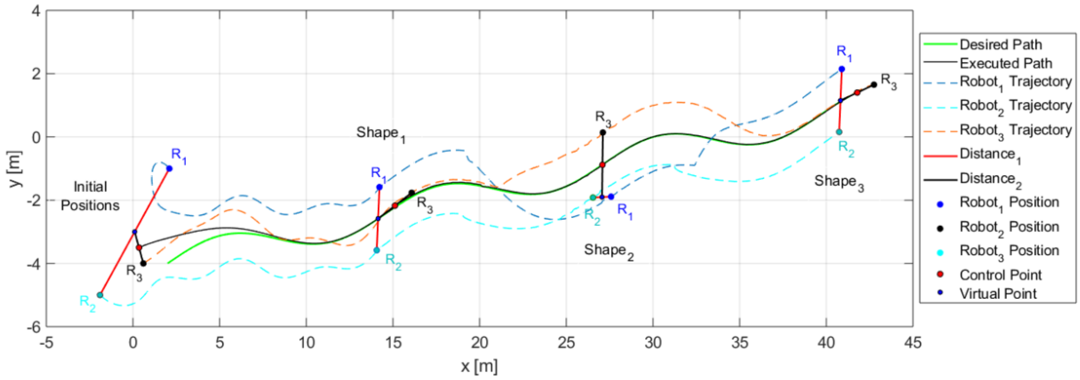
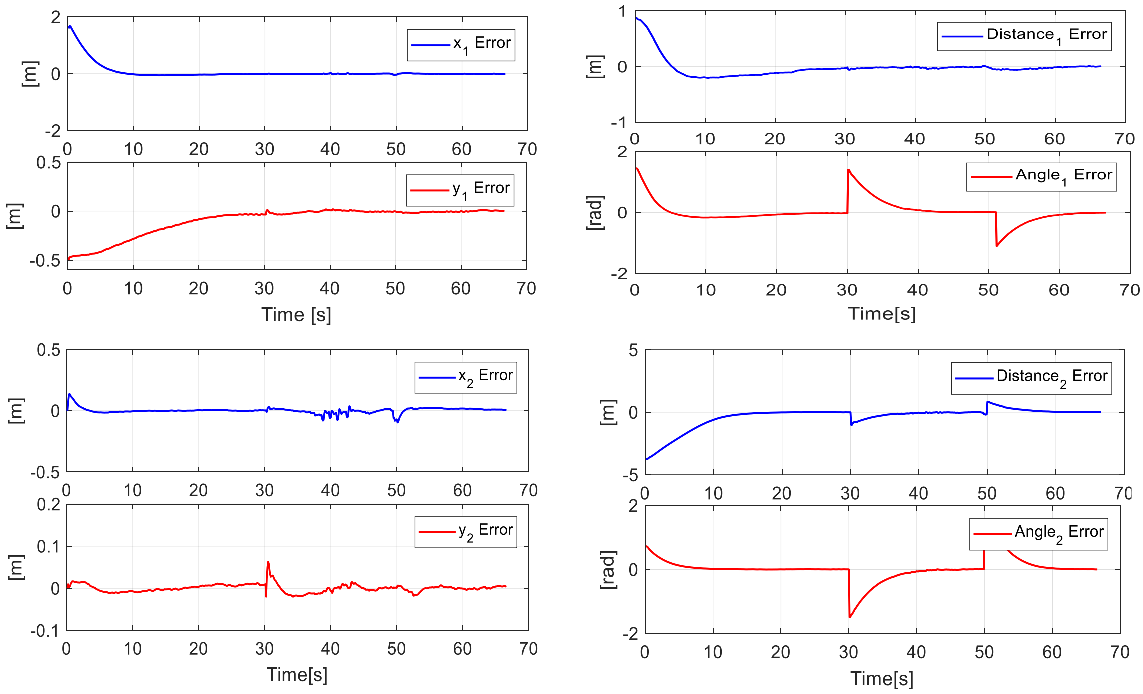


**Figure 20.** Robot positions and trajectories executed to stay on track with the desired shapes for the third experiment.

To validate the correct operation of the controller and verify that the robots complied with the task designed in shape and position, the errors are shown in Figure 21.



(**c**) Position errors in the virtual point.

(**d**) Shape errors in the virtual point.

**Figure 21.** Errors of the main and second objectives in the third experiment.

### 7.4. Fourth Experiment

In order to demonstrate the system scalability, this additional experiment was taken into account. For this, a task with eight robots is carried out in a simulated environment, given the hardware limitations. Figure 22 shows the stroboscopic movements of the robotic set, while Figure 23 presents the control errors of the first, second, third, and fourth projections. The rest of the projections are supposed to tend to zero, given the stroboscopic results.

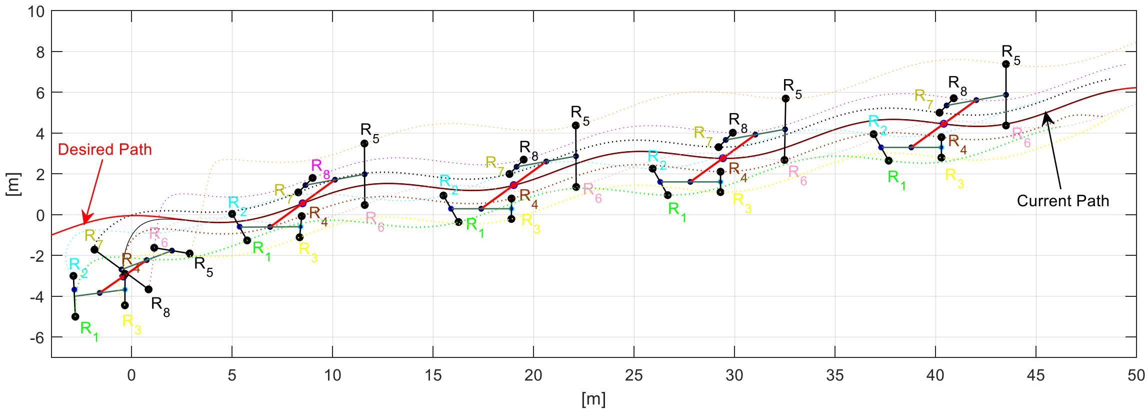
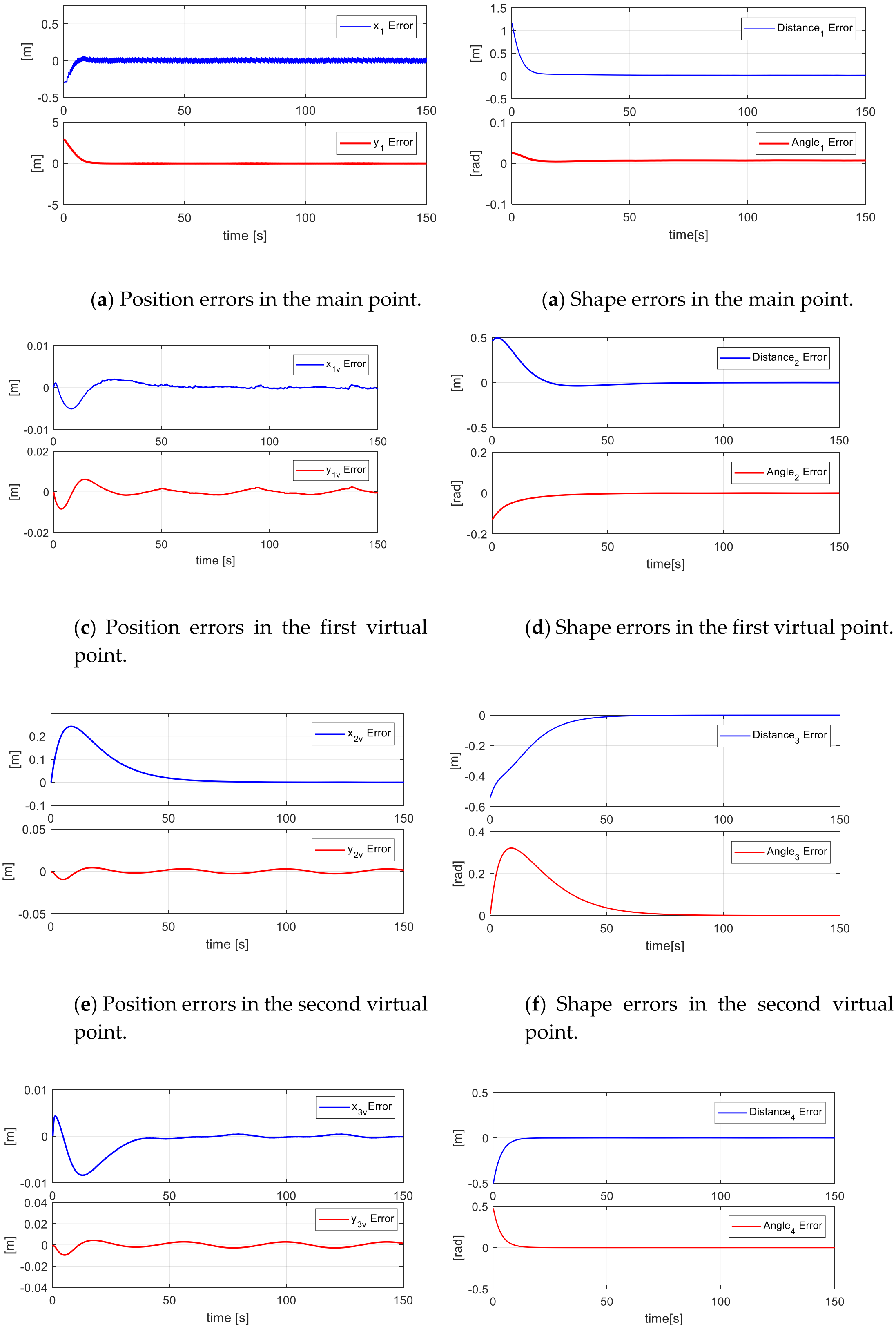


**Figure 22.** Robot positions and trajectories executed to stay on track with the desired shapes for the fourth experiment.



(**a**) Position errors in the main point.

(**a**) Shape errors in the main point.

(**c**) Position errors in the first virtual point.

(**d**) Shape errors in the first virtual point.

(**e**) Position errors in the second virtual point.

(**f**) Shape errors in the second virtual point.

(**g**) Position errors between the 1st–2nd robot

(**h**) Shape errors between the 1st–2nd robot

**Figure 23.** Errors of objectives in the fourth experiment.

## 8. Conclusions

Cooperative control considering primary and secondary objectives is achieved through linear algebra methods in this work, where diverse heterogeneous robots were used to determine the performance of the developed controllers. Compared with other formation control techniques, this proposal proposes to create projections for each robot that is included, providing the system with scalability in an easier way than other ones found in related works. Moreover, this work mathematically analyses the needed controllers to fulfill a primary task, in this case, the path followed by the robotic set, while the shape and orientation of all the robotic systems can be modified depending on the application without altering the main task given the redundancy of the system. Additionally, the stability analysis of the control law is presented to know the maximum values that the calibration parameters must have, as well as to demonstrate that the system is asymptotically stable. Furthermore, this work fuses centralized and decentralized processing of tasks in order to have a computer calculating the primary and secondary objectives, while processing units assure the kinematic and dynamic control of each robot. Finally, the set of robots used to obtain the experimental results is shown, results that demonstrate the proper functioning of the proposed controller through different real experiments where up to four different mechanisms are used. Everything described can be applied in civil construction environments, for transport and placement of materials in strategic sites; in warehouse management for transport of heavy objects of various characteristics and dimensions, and finally also in military applications for placement of dissuasive objectives in training camps.

**Author Contributions:** Conceptualization, J.F.A. and V.H.A.; Data curation, G.G.d.R. and J.G.; Formal analysis, J.F.A. and V.H.A.; Funding acquisition, J.F.A.; Investigation, J.F.A. and V.H.A.; Methodology, G.G.d.R. and V.H.A.; Project administration, V.H.A. and J.G.; Resources, G.G.d.R. and J.G.; Software, J.F.A.; Supervision, J.F.A., G.G.d.R., V.H.A. and J.G.; Validation, J.F.A., G.G.d.R., V.H.A. and J.G.; Visualization, G.G.R.; Writing—Original draft, J.F.A.; Writing—Review & editing, G.G.d.R., V.H.A. and J.G.

**Funding:** This research was funded by Corporación Ecuatoriana para el Desarrollo de la Investigación y Academia–CEDIA

**Acknowledgments:** The authors would like to thanks to the Corporación Ecuatoriana para el Desarrollo de la Investigación y Academia–CEDIA for the financing given to the research, development, and innovation, through the CEPRA projects, especially the project CEPRA XIII-2019-08; Sistema Colaborativo de Robots Aéreos para Manipular Cargas con Óptimo Consumo de Recursos; also to ARSI Research Group, for the support to develop this work.

**Conflicts of Interest:** The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

## References

1. Wang, J.-W.; Guo, Y.; Fahad, M.; Bingham, B. Dynamic Plume Tracking by Cooperative Robots. *IEEE/ASME Trans. Mechatronics* **2019**, *24*, 609–620. [CrossRef]
2. Mansouri, S.S.; Kanellakis, C.; Fresk, E.; Kominiak, D.; Nikolakopoulos, G.; Hutter, M.; Siegwart, R. Cooperative UAVs as a Tool for Aerial Inspection of the Aging Infrastructure. *Field and Service Robotics* **2017**, *5*, 177–189.
3. Cardarelli, E.; Digani, V.; Sabattini, L.; Secchi, C.; Fantuzzi, C. Cooperative cloud robotics architecture for the coordination of multi-AGV systems in industrial warehouses. *Mechatronics* **2017**, *45*, 1–13. [CrossRef]
4. Klancar, G.; Andrej, Z.; Blazic, S.; Skrjanc, I. From Fundamentals Towards Autonomous Systems. In *Wheeled Mobile Robotics*; Butterworth-Heinemann: Oxford, UK, 2017; pp. 397–416.
5. Ghotbi, B.; González, F.; Kövecses, J.; Angeles, J. Mobility evaluation of wheeled robots on soft terrain: Effect of internal force distribution. *Mech. Mach. Theory* **2016**, *100*, 259–282. [CrossRef]
6. Roy, A.G.; Rakshit, P. Motion Planning of Non-Holonomic Wheeled Robots Using Modified Bat Algorithm. In *Advances in Computational Intelligence and Robotics*; IGI Global: Hershey, PA, USA, 2019; pp. 94–123.
7. Chung, W.; Iagnemma, K. Wheeled Robots. In *Springer Handbook of Robotics*; Siciliano, B., Khatib, O., Eds.; Springer: Cham, Switzerland, 2016; pp. 575–594.

8. Vidoni, R.; Bietresato, M.; Gasparetto, A.; Mazzetto, F. Evaluation and stability comparison of different vehicle configurations for robotic agricultural operations on side-slopes. *Biosyst. Eng.* **2015**, *129*, 197–211. [CrossRef]
9. Rosales, C.; Leica, P.; Sarcinelli-Filho, M.; Scaglia, G.; Carelli, R. 3D Formation Control of Autonomous Vehicles Based on Null-Space. *J. Intell. Robot. Syst.* **2016**, *84*, 453–467. [CrossRef]
10. Haider Jafri, S.M.; Kala, R. Path Planning of a Mobile Robot in Outdoor Terrain. In *Intelligent Systems Technologies and Applications*; Springer: Cham, Switzerland, 2015; Volume 385, pp. 187–195.
11. Lee, J.; Lim, J.; Lee, J. Compensated Heading Angles for Outdoor Mobile Robots in Magnetically Disturbed Environment. *IEEE Trans. Ind. Electron.* **2018**, *65*, 1408–1419. [CrossRef]
12. Andaluz, V.H.; Ortiz, J.S.; Sánchez, J.S.; M, F.M.S. Autonomous Cooperation Between Terrestrial and Aerial Robots. *Adv. Sci. Lett.* **2018**, *24*, 8812–8816. [CrossRef]
13. Khan, M.U.; Li, S.; Wang, Q.; Shao, Z. Formation Control and Tracking for Co-operative Robots with Non-holonomic Constraints. *J. Intell. Robot. Syst.* **2016**, *82*, 163–174. [CrossRef]
14. Wang, G.; Du, Q.; Li, L.; Dong, W. Distributed Cooperative Control of Multiple Nonholonomic Mobile Robots. *J. Intell. Robot. Syst.* **2016**, *83*, 525–541. [CrossRef]
15. Chen, J.; Gauci, M.; Li, W.; Kölling, A.; Groß, R.; Gross, R. Occlusion-Based Cooperative Transport with a Swarm of Miniature Mobile Robots. *IEEE Trans. Robot.* **2015**, *31*, 307–321. [CrossRef]
16. Molina, M.F.; Ortiz, J.S. Coordinated and Cooperative Control of Heterogeneous Mobile Manipulators. *Agreement Technol.* **2018**, *11357*, 483–492.
17. Savkin, A.V.; Wang, C.; Baranzadeh, A.; Xi, Z.; Nguyen, H.T. Distributed formation building algorithms for groups of wheeled mobile robots. *Robot. Auton. Syst.* **2016**, *75*, 463–474. [CrossRef]
18. Hausman, K.; Müller, J.; Hariharan, A.; Ayanian, N.; Sukhatme, G.S. Cooperative multi-robot control for target tracking with onboard sensing. *Int. J. Robot. Res.* **2015**, *34*, 1660–1677. [CrossRef]
19. Andaluz, V.H.; Molina, M.F.; Erazo, Y.P.; Ortiz, J.S. Numerical Methods for Cooperative Control of Double Mobile Manipulators. *Intell. Rob. Appl.* **2017**, *10463*, 889–898.
20. Andaluz, V.H.; Sásig, E.R.; Chicaiza, W.D.; Velasco, P.M. Linear Algebra Applied to Kinematic Control of Mobile Manipulators. *IT Convergence Secur.* **2017**, *449*, 297–306.
21. Rosales, A.; Scaglia, G.; Mut, V.; Di Sciascio, F. Formation control and trajectory tracking of mobile robotic systems – A Linear Algebra approach. Formation control and trajectory tracking of mobile robotic systems – A Linear Algebra approach. *Robotica* **2010**, *29*, 335–349. [CrossRef]
22. Acosta, J.; Andaluz, V.H.; Glez-de-Rivera, G.; Garrido, J. Energy-Saver Mobile Manipulator Based on Numerical Methods, Motion Planning and Control for Robotics. *Electronics* **2019**, *8*, 1100. [CrossRef]
23. Du, H.; Wen, G.; Cheng, Y.; He, Y.; Jia, R. Distributed Finite-Time Cooperative Control of Multiple High-Order Nonholonomic Mobile Robots. *IEEE Trans. Neural Networks Learn. Syst.* **2017**, *28*, 2998–3006. [CrossRef]
24. Goldhoorn, A.; Garrell, A.; Alquézar, R.; Sanfeliu, A. Searching and tracking people with cooperative mobile robots. *Auton. Robot.* **2017**, *42*, 739–759. [CrossRef]
25. Andaluz, V.H.; Roberto, F.; Toibero, M.; Leica, P.; Carelli, R. Adaptive Coordinated Cooperative Control of Multi-Mobile Manipulators. In *Frontiers in Advanced Control Systems*; IntechOpen: London, UK, 2012; pp. 163–190.
26. Ortiz, J.S.; Aldás, J.V.; Andaluz, V.H. Mobile Manipulators for Cooperative Transportation of Objects in Common. In *Towards Autonomous Robotic Systems. TAROS 2017. Lecture Notes in Computer Science*; Gao, Y., Fallah, S., Jin, Y., Lekakou, C., Eds.; Springer: Cham, Switzerland; pp. 651–660.
27. Paul, R.P. *Robot Manipulators: Mathematics, Programming, and Control*; MIT Press: Cambridge, MA, USA, 1981.
28. Siciliano, B.; Sciavicco, L.; Villani, L.; Oriolo, G. *Robotics: Modelling, Planning and Control (Advanced Textbooks in Control and Signal Processing)*; Springer: London, UK, 2010.
29. Chiaverini, S. Singularity-Robust Task-Priority Redundancy. *IEEE Trans. Robot. Autom.* **1997**, *3*, 398–410. [CrossRef]