machines

Article

Formation Control of Mobile Robots Based on Pin Control of Complex Networks

Jorge D. Rios [1], Daniel Ríos-Rivera [1], Jesus Hernandez-Barragan [1], Marco Pérez-Cisneros [2] and Alma Y. Alanis [1,]*

[1] Departamento de Innovación Basada en la Información y el Conocimiento, Centro Universitario de Ciencias Exactas e Ingenierías, Universidad de Guadalajara, Guadalajara 44430, Mexico

[2] Departamento de Electro-Fotónica, Centro Universitario de Ciencias Exactas e Ingenierías, Universidad de Guadalajara, Guadalajara 44430, Mexico

* Correspondence: alma.alanis@academicos.udg.mx

Abstract: Robot formation control has several advantages that make it interesting for research. Multiple works have been published in the literature using different control approaches. This work presents the control of different groups of robots to achieve a desired formation based on pinning control of complex networks and coordinate translation. The implemented control law comprises complex network bounding, proportional, and collision avoidance terms. The tests for this proposal were performed via simulation and experimental tests, considering different networks of differential robots. The selected robots are Turtlebot3® Waffle Pi robots. The Turtlebot3® Waffle Pi is a differential mobile robot with the Robot Operating System (ROS). It has a light detection and ranging (LiDAR) sensor used to compute the collision avoidance control law term. Tests show favorable results on different formations testing on various groups of robots, each composed of a different number of robots. From this work, implementation on other devices can be derived, as well as trajectory tracking once in formation, among other applications.

Keywords: formation control; differential robots; complex networks; pinning control

check for updates

Citation: Rios, J.D.; Ríos-Rivera, D.; Hernandez-Barragan, J.; Pérez-Cisneros, M.; Alanis, A.Y. Formation Control of Mobile Robots Based on Pin Control of Complex Networks. *Machines* **2022**, *10*, 898. https://doi.org/10.3390/machines10100898

Academic Editor: Dan Zhang

Received: 7 September 2022
Accepted: 29 September 2022
Published: 6 October 2022

1. Introduction

Robot formation control research has attracted interest due to advantages such as flexibility, adaptability, robustness, and cost reduction found in the idea of multiple robots working together on applications in areas such as rescue, transportation, mine sweeping, geographic exploration, surveillance, agriculture, and more [1–4]. One of the objectives behind this kind of collaborative control is to imitate the work of some biological systems [2–5].

Multiple works have been reported in the literature mainly based on leader-follower, behavior-based, and virtual structure schemes [1,2,5]. Moreover, robot formation can be implemented with centralized or decentralized control [2,5]. The authors of [1] propose a leader-follower formation control of multiple robots and transforming the prescribed performance into an unconstrained one. In [5], the control of nonholonomic mobile robots with a leader-follower scheme is presented with simulation results. The study [5] is based on linear approximation, feedback linearization, and known position of the leader, and the collision algorithm is based on priorities. Study [6] shows a behavior-based algorithm for forming patterns based on swarm intelligence. In [7], a leader-follower approach and collision avoidance scheme based on a virtual leader is presented. In [8], a distributed control law is used to achieve circular and non-circular formation for multiple robots.

The works mentioned above and our proposed scheme use the robot's different sensors to access information such as the position of the robot, detection of obstacles, and communication. We encourage the reader to look for the references mentioned in those works.

On the other hand, a complex network can be defined as a system made of many single units which, from the perspective of dynamical systems, are systems on their own, interacting in such a way that the behavior of the whole system is not just the sum of the behaviors of the single units [3,4,9]. Complex networks can be found in many fields, such as engineering, physics, technology, biology, and social sciences, to name a few.

Moreover, a complex network has characteristics such as a better representation of real-world complex systems with nontrivial structures. They are dynamic, and their nodes can be dynamic systems. Moreover, they can be structurally complex, among other things [3,4,9].

In previous works [10,11], pinning control methodology was used to stabilize complex networks; this approach consists of controlling just some of the nodes in the network. The results of these previous works motivate the idea of pinning control for the formation of mobile robots. In this way, it is only necessary to design the position control for just one of the node robots of the network, meaning that only one knows the final position. This is an advantage when adding more nodes or if the position control strategy is thought to be changed in the future.

It is important to remark that the selected node is known as the pinned node; this is the reason why it is called pin control. It has a similar function as a leader in a leader-follower approach. However, it is not a leader; it may receive information from other nodes in the network, but in the leader-follower approach, the leader does not receive information from other nodes. The proposed formation control based on pin control of complex networks has the following characteristics: the formation is achieved through synchronizing the designed complex network, where a node is selected as the pinned control node, and the formation is achieved on a set point which is only known by a selected robot as a pinned node, and just a set number of the remaining robots are connected to it. Collison avoidance is assured against other robots and objects, not only between robots; a mounted LiDAR (light detection and ranging) sensor obtains the distances between a robot and any object. Simulation and experimental test results are included in this work using a different number of robots per test. Tests ended at the desired position in the desired formation. In contrast to previous works that use the robot's known position to avoid a collision, this work uses the mounted LiDAR on the selected Turtlebot3® robots.

The remainder of this work is organized as follows. Section 2 briefly introduces the complex network model and control methodology, along with the model of the mobile robots used in the implementation. In Section 3, the proposed control scheme based on complex networks and pinning control is presented. The implementation of the proposal and the results of the tests are presented in Section 4. Section 5, the results discussion, gives a brief analysis of the obtained results and includes a comparison against other techniques. Finally, respective conclusions are given in Section 6.

2. Mathematical Background

This section introduces the main mathematical concepts used in this work. First, an introduction to complex networks and pinning control is provided, then a presentation of differential mobile robots selected as our case study is given.

2.1. Complex Networks and Pinning Control

Each one of the N nodes of a dynamical complex network can be described by the next model [9]:

$$\dot{X}_i = f_i(X_i) + \sum_{j=1}^{N} c_{ij} a_{ij} B X_j + U_i, \tag{1}$$

where $\dot{X}_i = \begin{bmatrix} x_{i1} & x_{i2} & \dots & x_{in} \end{bmatrix}^\top$ is the state of the i-th node of dimension n, $f_i : \mathbb{R}^n \to \mathbb{R}^n$, c_{ij} is the connection strength between nodes i and j, $a_{ij} = 1$ if there is a connection between node i and j, and otherwise $a_{ij} = 0$ if there is no such connection, $a_{ii} = -\sum\limits_{j=1, j\neq i}^{N} a_{ij}$ which follows the diffusive condition, and consequently $c_{ii} = \frac{1}{a_{ii}} \sum\limits_{j=1, j\neq i}^{N} c_{ij}a_{ij}$, $B \in \mathbb{R}^{n\times n}$ is a real matrix that describes the connections between the different state variables of the different nodes, and U_i is a control input.

The pinning control technique for complex networks can be defined in its simplest form as [10–13]:

$$U_{ei} = \kappa_i e_i, \tag{2}$$

where $e_i = X_i - r$ is the error signal between the measured state and the reference signal r, and κ_i is a real negative constant if node i is selected as a pinned node, if not, $\kappa_i = 0$.

2.2. Differential Drive Robots

Our case study considers multiple mobile robot platforms, each one composed of two independent driving wheels, as illustrated in Figure 1. The pose of a mobile robot is described with a position $\langle x, y\rangle$ and an orientation $\langle \theta \rangle$ in the world frame $\{F_W\}$. Moreover, a base frame $\{F_B\}$ is attached to the middle of the two wheels, and θ is an angle measured from $\langle x, y\rangle$ pointing to $\langle x_h, y_h\rangle$ relative to the x-axis of the world frame $\{F_W\}$, with the coordinate $\langle x_h, y_h\rangle$ defined as a Cartesian position for control purposes. L is the distance from $\langle x, y\rangle$ to $\langle x_h, y_h\rangle$, and finally, v and ω are, respectively, the linear and angular velocities relative to the base frame $\{F_B\}$.

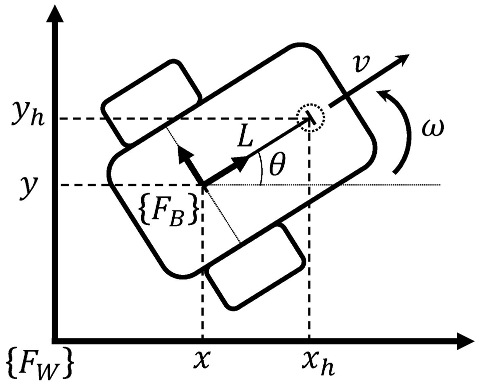

Figure 1. Differential drive robot diagram.

The velocity kinematic equations of a set of N differential drive robots can be described with the next set of equations [14–16]:

$$\begin{aligned} \dot{x}_i &= v_i \cos\theta_i, \\ \dot{y}_i &= v_i \sin\theta_i, \\ \dot{\theta}_i &= \omega_i, \end{aligned} \tag{3}$$

where v_i is the linear velocity and ω_i is the angular velocity of the i-th robot.

The nonholonomic constrains in differential drive robots makes the system difficult to design v_i and ω_i, such that x_i and y_i approach zero state. For this, we define:

$$\begin{aligned} x_{hi} &= x_i + L_i \cos\theta_i, \\ y_{hi} &= y_i + L_i \sin\theta_i, \end{aligned} \tag{4}$$

where x_{hi} and y_{hi} represent the Cartesian position to control robot i. From (3) and (4) we obtain the derivatives of x_h and y_h as follows:

$$\begin{aligned} \dot{x}_{hi} &= v_i \cos\theta_i - \omega_i L_i \sin\theta_i, \\ \dot{y}_{hi} &= v_i \sin\theta_i + \omega_i L_i \cos\theta_i, \end{aligned} \tag{5}$$

which has the next matrix representation

$$\begin{bmatrix} \dot{x}_{hi} \\ \dot{y}_{hi} \end{bmatrix} = \begin{bmatrix} \cos\theta_i & -L_i \sin\theta_i \\ \sin\theta_i & L_i \cos\theta_i \end{bmatrix} \begin{bmatrix} v_i \\ \omega_i \end{bmatrix}. \tag{6}$$

Then, we can design v_i and ω_i as

$$\begin{bmatrix} v_i \\ \omega_i \end{bmatrix} = \begin{bmatrix} \cos\theta_i & -L_i \sin\theta_i \\ \sin\theta_i & L_i \cos\theta_i \end{bmatrix}^{-1} \begin{bmatrix} u_{xi} \\ u_{yi} \end{bmatrix}, \tag{7}$$

and by substituting (7) in (6) we find

$$\dot{X}_{hi} = \begin{bmatrix} \dot{x}_{hi} \\ \dot{y}_{hi} \end{bmatrix} = \begin{bmatrix} u_{xi} \\ u_{yi} \end{bmatrix} = U_{Gi}, \tag{8}$$

which can be designed as needed for our control purposes, detailed in Section 4.

3. Proposed Scheme for Formation Control of Differential Robots

The objective of the control is to make formations with distinct sets of differential robots. For this, we define U_{Gi} as a sum of three terms, that is:

$$U_{Gi} = U_{ci} + U_{ei} + U_{ri}. \tag{9}$$

These terms will be defined as follows. First, U_{ci} is:

$$U_{ci} = \begin{bmatrix} c_{i1}a_{i1}b_1 x_{h1} + \ldots + c_{iN}a_{iN}b_1 x_{hN} \\ c_{i1}a_{i1}b_1 y_{h1} + \ldots + c_{iN}a_{iN}b_1 y_{hN} \end{bmatrix}, \tag{10}$$

where the terms are defined as in Section 2.1. This has the form

$$U_{ci} = \sum_{j=1}^{N} c_{ij}a_{ij}BX_j, \tag{11}$$

which is the term describing the connections of the complex network model (1). After this, we can take the pinning control concept of complex networks and apply it here by adding the control term U_{ei} defined in (2).

As seen in multiple works [10–13], with pinning control, all nodes synchronize to the same state; however, robots cannot physically be in the same space as another robot, so a translation of coordinates is necessary to simulate that they are synchronizing in the same state as the other robots, and U_{ci} becomes

$$U_{ci} = \sum_{j=1}^{N} c_{ij}a_{ij}B\left(X_j + T_{ij}\right), \tag{12}$$

where $T_{ij} = \left[T_{xij}\; T_{yij}\right]^{\top}$ with T_{xij} and T_{yij} is the distance between robots i and j in the x and y axis, respectively, in a given final formation.

To avoid robot collisions, a LiDAR sensor is attached to the base frame $\{F_B\}$ at the position $\langle x, y\rangle$ of each mobile robot as seen in Figure 2. This sensor provides the current range data d_{si} at the angular position θ_{si}, both measured from base frame $\{F_B\}$. Moreover, d_{max} is the maximum range value for detection. From these measurements, the repulsion term U_{ri} is proposed as follows:

$$U_{ri} = \kappa_{ri} \begin{bmatrix} (d_{si} - d_{max})\cos(\theta_i + \theta_{si}) \\ (d_{si}\_d_{max})\sin(\theta_i + \theta_{si}) \end{bmatrix}, \tag{13}$$

where κ_r is a positive gain. The orientation θ_i of robot i is considered to compute repulsion U_{ri} relative to the world frame $\{F_W\}$.

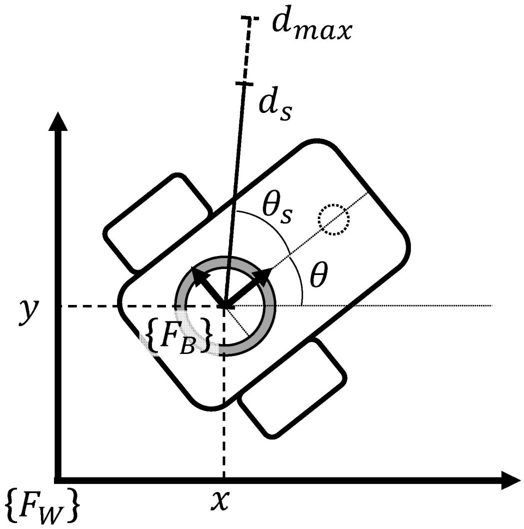

Figure 2. Object detection for differential drive robot diagram.

Then, defining

$$U_i = U_{ei} + U_{ri}, \tag{14}$$

we find

$$\dot{X}_{hi} = \sum_{j=1}^{N} c_{ij} a_{ij} B\left(X_{hj} + T_{ij}\right) + U_i, \tag{15}$$

which is the complex network model of (1) with $f_i(X_i) = 0$ and a coordinate translation in the connections.

4. Simulation and Physical Implementations

This section presents the details of implementing the proposed collaborative control scheme for mobile robots at the simulation and experimental levels using Gazebo software and Turtlebot3® Waffle Pi robots. Such implementation in both scenarios was held using Python and ROS (Robot Operating System) environments.

4.1. Experiment Description

To implement the proposed scheme, we used Turtlebot3® Waffle Pi robots. Among the hardware elements of a Turtlebot3® Waffle Pi robot, it has a Raspberry Pi 3, actuators XL430-W20, LiDAR sensor 360 laser Distance Sensor LDS-01, IMU with gyroscope 3 axis, and accelerometer 3 axis. The robot pose is calculated by the robot giving an odometry ROS message with its position and orientation. It is important to note that the accuracy of this measurement varies in function of the terrain conditions and wear of the robot equipment.

The LiDAR sensor is used to detect near objects. The LiDAR sensor obtains the necessary values to compute U_{ri} of Equation (13). This scheme was tested first in the Gazebo simulation environment using arrays of 4, 6, and 12 robots. Only one of the robots was a pinned node in all configurations, and all coordinates and distances are in meters.

4.1.1. Simulation Test: 4 Robots

The initial positions in the 4-robot simulation and their objectives are illustrated in Figure 3.

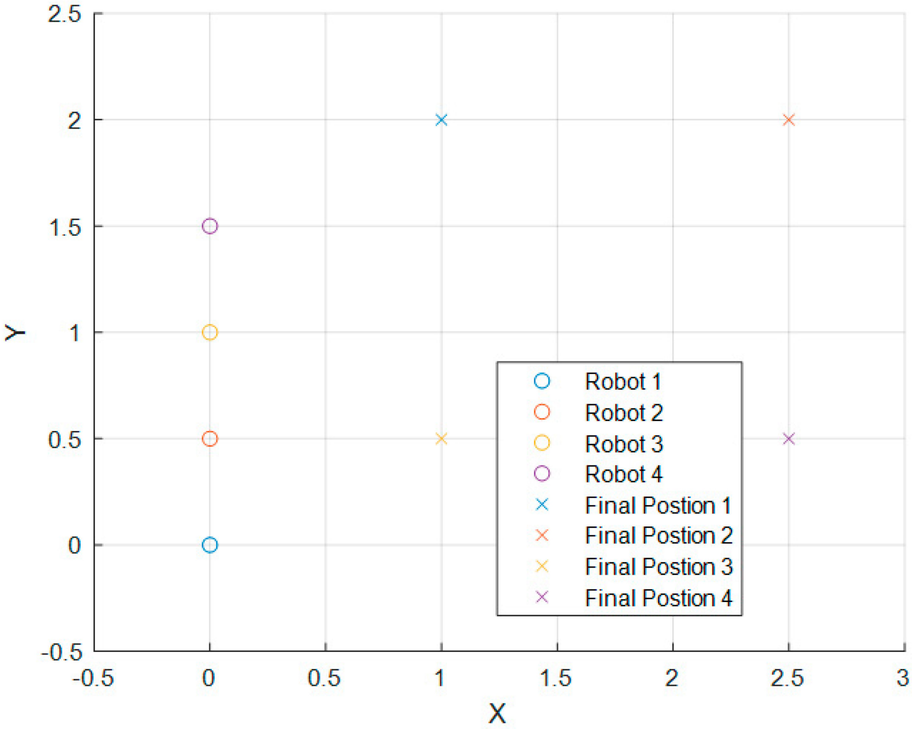

Figure 3. Initial positions of the 4-robot experiment and the desired objective for each robot.

If we define $g_{ij} = c_{ij}a_{ij}$, then, in the 4-robot simulation, g_{ij} can be obtained from the matrix:

$$G = \begin{bmatrix} -0.003 & 0.001 & 0.001 & 0.001 \\ 10 & -10.002 & 0.001 & 0.001 \\ 10 & 0.001 & -10.002 & 0.001 \\ 10 & 0.001 & 0.001 & -10.002 \end{bmatrix}, \tag{16}$$

where values were heuristically determined, B is an identity matrix, and T_{ij} can be obtained from Table 1, where $D = 1.5$.

Table 1. Final formation distance between robots in the 4-robot experiment.

j	**1**		**2**		**3**		**4**	
i	T_x	T_y	T_x	T_y	T_x	T_y	T_x	T_y
1	0	0	$-D$	0	0	D	$-D$	D
2	D	0	0	0	D	D	0	D
3	0	$-D$	$-D$	$-D$	0	0	$-D$	0
4	D	$-D$	0	$-D$	D	0	0	0

Using this, we can compute U_{ci}, which will be scaled as $g_i U_{ci}$ with

$$g_i = \begin{cases} 1 & \text{if} \quad d_{si} \geq d_{max}, \\ 0.05\frac{d_{si}}{d_{max}} & \text{if} \quad d_{si} < d_{max}. \end{cases} \tag{17}$$

For U_{ei}

$$\kappa_i = -10 g_i, \tag{18}$$

and for U_{ri}, $d_{max} = 0.5$ and $\kappa_r = 5$. The g_i condition exists so that the control algorithm gives priority to evading obstacles. Finally, the computed v_i and ω_i are scaled or saturated under certain conditions. Using a $v_{max} = 0.2$, if computed that $v_i > v_{max}$ then it is multiplied by a factor $g_l = 0.1$; if by doing this v_i still exceeds v_{max} then we equate them. Similarly, with computed ω_i, using a $\omega_{max} = 0.9$ and a factor $g_\omega = 0.01$. This was done in order to not exceed the physical limitations of the robot and to not saturate the control signal of the robots all the time. The value of parameter L of Equation (7) was set as $L = 0.01$. Figure 4 shows the Gazebo simulation environment and the robots' initial position for the 4-robot simulation tests.

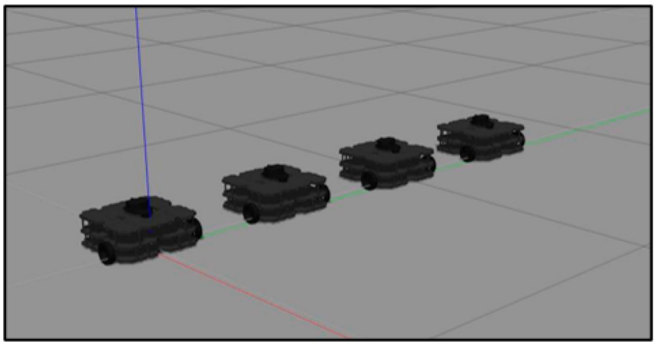

Figure 4. Gazebo 7 of the Open Source Robotics Foundation, used in simulations with four Turtlebot3® Waffle Pi robots aligned at the beginning of the test. The red, green, and blue lines indicate the X, Y, and Z axes, respectively.

4.1.2. Simulation Test: 6 Robots

The initial positions for the 6-robot simulation and their objectives are illustrated in Figure 5.

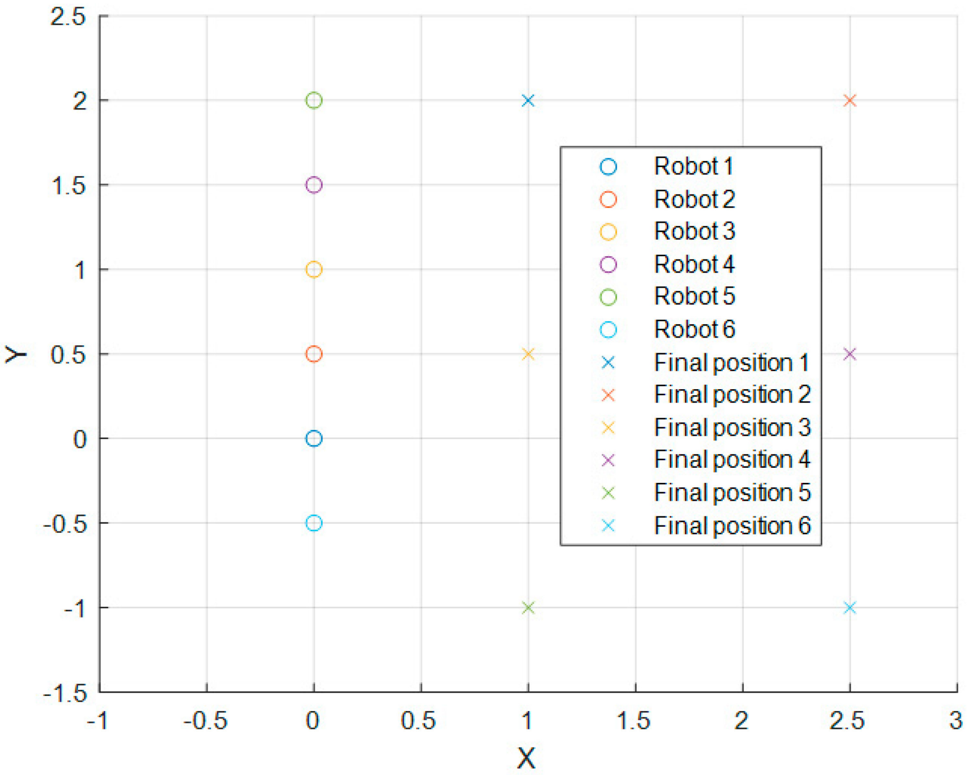

Figure 5. Initial positions of the 6-robot experiment and the desired objective for each robot.

In this simulation, G is defined as follows for a six-robot configuration:

$$G = \begin{bmatrix} -0.005 & 0.001 & 0.001 & 0.001 & 0.001 & 0.001 \\ 10 & -10.004 & 0.001 & 0.001 & 0.001 & 0.001 \\ 10 & 0.001 & -10.004 & 0.001 & 0.001 & 0.001 \\ 10 & 0.001 & 0.001 & -10.004 & 0.001 & 0.001 \\ 10 & 0.001 & 0.001 & 0.001 & -10.004 & 0.001 \\ 10 & 0.001 & 0.001 & 0.001 & 0.001 & 0.001 \end{bmatrix}, \tag{19}$$

and T_{ij} can be obtained from Table 2.

Table 2. Final formation distance between robots in the 6-robot experiment.

j	1		2		3		4		5		6	
i	T_x	T_y	T_x	T_y	T_x	T_y	T_x	T_y	T_x	T_y	T_x	T_y
1	0	0	$-D$	0	0	D	$-D$	D	0	$2D$	$-D$	$2D$
2	D	0	0	0	D	D	0	D	D	$2D$	0	$2D$
3	0	$-D$	$-D$	$-D$	0	0	$-D$	0	0	D	$-D$	D
4	D	$-D$	0	$-D$	D	0	0	0	D	D	0	D
5	0	$-2D$	$-D$	$-2D$	0	$-D$	$-D$	$-D$	0	0	$-D$	0
6	D	$-2D$	0	$-2D$	D	$-D$	0	$-D$	D	0	0	0

Any other parameters stayed as in the previous simulation for 4 robots.

4.1.3. Simulation Test: 12 Robots

The initial positions for the 12-robot simulation and their objectives are illustrated in Figure 6.

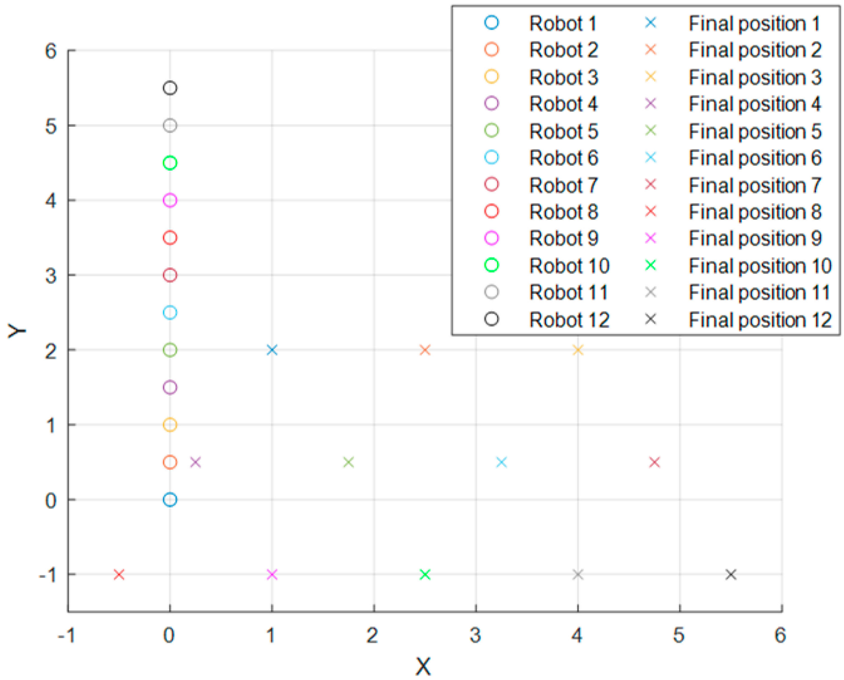

Figure 6. Initial positions of the 12-robot experiment and the desired objective for each robot.

In this simulation, G changes to

$$G = \begin{bmatrix} G_1 & G_{12} \\ G_{21} & G_2 \end{bmatrix}, \tag{20}$$

with

$$G_1 = \begin{bmatrix} -10.002 & 10 & 0 & 0.001 & 0.001 & 0 \\ 0.001 & -0.004 & 0.001 & 0 & 0.001 & 0.001 \\ 0 & 10 & -10.002 & 0 & 0 & 0.001 \\ 10 & 0 & 0 & -10.002 & 0 & 0 \\ 10 & 10 & 0.001 & 0 & -20.002 & 0 \\ 0 & 10 & 10 & 0 & 0 & -20.002 \end{bmatrix}, \tag{21}$$

$$G_{12} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0.001 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.001 & 0.001 & 0 & 0 & 0 \\ 0 & 0 & 0.001 & 0.001 & 0 & 0 \\ 0 & 0 & 0 & 0.001 & 0.001 & 0 \end{bmatrix}, \tag{22}$$

$$G_{21} = \begin{bmatrix} 0 & 0 & 10 & 0 & 0 & 0 \\ 0 & 0 & 0 & 10 & 0 & 0 \\ 0 & 0 & 0 & 10 & 10 & 0 \\ 0 & 0 & 0 & 0 & 10 & 10 \\ 0 & 0 & 0 & 0 & 0 & 10 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \tag{23}$$

$$G_2 = \begin{bmatrix} -10.002 & 0 & 0 & 0 & 0.001 & 0.001 \\ 0 & -10 & 0 & 0 & 0 & 0 \\ 0 & 0 & -20 & 0 & 0 & 0 \\ 0 & 0 & 0 & -20 & 0 & 0 \\ 10 & 0 & 0 & 0 & -20 & 0 \\ 10 & 0 & 0 & 0 & 0 & -10 \end{bmatrix}, \tag{24}$$

and T_{ij} can be obtained from Tables 3–6.

Any other parameters stayed as in the previous presented simulations.

Note that all the tables presented in this section are in accordance with the formation distances shown in Figures 3, 5 and 6, respectively.

Table 3. Final formation distance in the 12-robot experiment of robots 1–6 for robots 1–6.

j	**1**		**2**		**3**		**4**		**5**		**6**	
i	T_x	T_y	T_x	T_y	T_x	T_y	T_x	T_y	T_x	T_y	T_x	T_y
1	0	0	$-D$	0	$-2D$	0	$\frac{D}{2}$	D	$\frac{-D}{2}$	D	$-3\frac{D}{2}$	D
2	D	0	0	0	$-D$	0	$3\frac{D}{2}$	D	$\frac{D}{2}$	D	$\frac{-D}{2}$	D
3	$2D$	0	$-D$	0	0	0	$5\frac{D}{2}$	D	$3\frac{D}{2}$	D	$\frac{D}{2}$	D
4	$\frac{-D}{2}$	$-D$	$-3\frac{D}{2}$	$-D$	$-5\frac{D}{2}$	$-D$	0	0	$-D$	0	$-2D$	0
5	$\frac{D}{2}$	$-D$	$\frac{-D}{2}$	$-D$	$-3\frac{D}{2}$	$-D$	D	0	0	0	$-D$	0
6	$3\frac{D}{2}$	$-\mathrm{D}$	$\frac{D}{2}$	$-D$	$\frac{-D}{2}$	$-D$	$-2D$	0	D	0	0	0

Table 4. Final formation distance in the 12-robot experiment of robots 1–6 for robots 7–12.

j	**7**		**8**		**9**		**10**		**11**		**12**	
i	T_x	T_y	T_x	T_y	T_x	T_y	T_x	T_y	T_x	T_y	T_x	T_y
1	$-5\frac{D}{2}$	D	D	$2D$	0	$2D$	$-D$	$2D$	$-2D$	$2D$	$-3D$	$2D$
2	$-3\frac{D}{2}$	D	$2D$	$2D$	D	$2D$	0	$2D$	$-D$	$2D$	$-2D$	$2D$
3	$\frac{-D}{2}$	D	$3D$	$2D$	$2D$	$2D$	D	$2D$	0	$2D$	$-D$	$2D$
4	$-3D$	0	$\frac{D}{2}$	D	$\frac{-D}{2}$	D	$-3\frac{D}{2}$	D	$-5\frac{D}{2}$	D	$-7\frac{D}{2}$	D
5	$-2D$	0	$3\frac{D}{2}$	D	$\frac{D}{2}$	D	$\frac{-D}{2}$	D	$-3\frac{D}{2}$	D	$-5\frac{D}{2}$	D
6	$-D$	0	$5\frac{D}{2}$	D	$3\frac{D}{2}$	D	$\frac{D}{2}$	D	$\frac{-D}{2}$	D	$-3\frac{D}{2}$	D

Table 5. Final formation distance in the 12-robot experiment of robots 7–12 for robots 1–6.

j	**1**		**2**		**3**		**4**		**5**		**6**	
i	T_x	T_y	T_x	T_y	T_x	T_y	T_x	T_y	T_x	T_y	T_x	T_y
7	$5\frac{D}{2}$	$-D$	$3\frac{D}{2}$	$-D$	$\frac{D}{2}$	$-D$	$-3D$	0	$2D$	0	D	0
8	$-D$	$-2D$	$-2D$	$-2D$	$-3D$	$-2D$	$\frac{-D}{2}$	$-D$	$-3\frac{D}{2}$	$-D$	$-5\frac{D}{2}$	$-D$
9	0	$-2D$	$-D$	$-2D$	$-2D$	$-2D$	$\frac{D}{2}$	$-D$	$\frac{-D}{2}$	$-D$	$-3\frac{D}{2}$	$-D$
10	D	$-2D$	0	$-2D$	$-D$	$-2D$	$3\frac{D}{2}$	$-D$	$\frac{D}{2}$	$-D$	$\frac{-D}{2}$	$-D$
11	$2D$	$-2D$	D	$-2D$	0	$-2D$	$5\frac{D}{2}$	$-D$	$3\frac{D}{2}$	$-D$	$\frac{D}{2}$	$-D$
12	$3D$	$-2D$	$2D$	$-2D$	D	$-2D$	$7\frac{D}{2}$	$-D$	$5\frac{D}{2}$	$-D$	$3\frac{D}{2}$	$-D$

Table 6. Final formation distance in the 12-robot experiment of robots 7–12 for robots 7–12.

j	**7**		**8**		**9**		**10**		**11**		**12**	
i	T_x	T_y	T_x	T_y	T_x	T_y	T_x	T_y	T_x	T_y	T_x	T_y
7	0	D	$-7\frac{D}{2}$	D	$5\frac{D}{2}$	D	$3\frac{D}{2}$	D	$\frac{D}{2}$	D	$\frac{-D}{2}$	D
8	$-7\frac{D}{2}$	0	0	0	$-D$	0	$-2D$	0	$-3D$	0	$-4D$	0
9	$-5\frac{D}{2}$	0	$-D$	0	0	0	$-D$	0	$-2D$	0	$-3D$	0
10	$-3\frac{D}{2}$	0	$-2D$	0	D	0	0	0	$-D$	0	$-2D$	0
11	$\frac{-D}{2}$	0	$-3D$	0	$2D$	0	D	0	0	0	$-D$	0
12	$\frac{D}{2}$	0	$-4D$	0	$3D$	0	$2D$	0	D	0	0	0

After simulations, the 4- and 6-robot arrays were tested with real equipment with the same set of parameters.

4.2. Experiment Results

This section shows the results of the simulation and experimental test using Turtlebot3® Waffle Pi robots. The results of tests with 4 and 6 robots are compared to the experimental implementation on the physical hardware. The results of the 12 robot tests were left at simulation level.

First, we present the data results of the Gazebo implementations. Figure 7 shows the trajectory followed by each robot on a 4-robot array in the Gazebo simulation test in the XY plane.

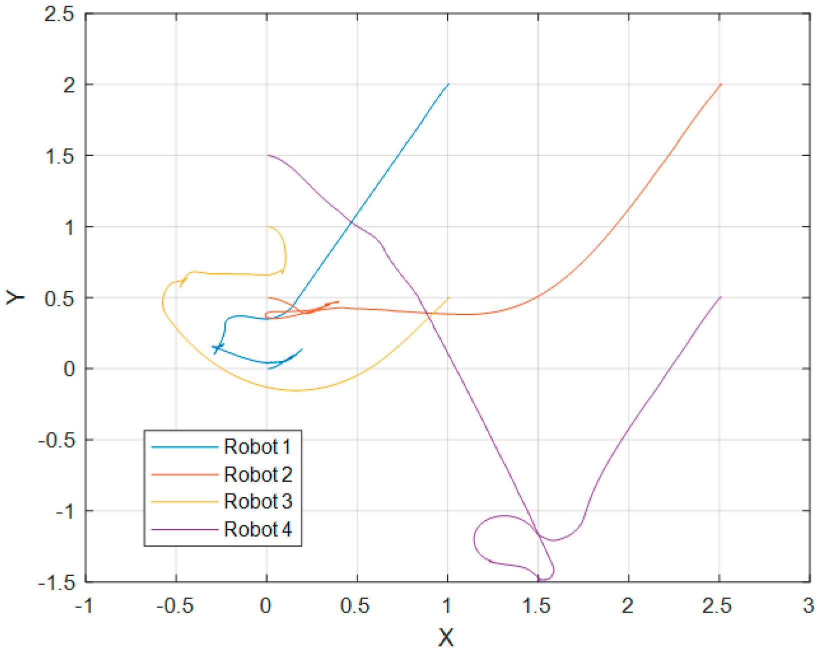

Figure 7. Trajectories in the 4-robot simulation.

The values of x_{hi}, y_{hi}, v_i, and ω_i through the iterations are plotted in Figures 8 and 9.

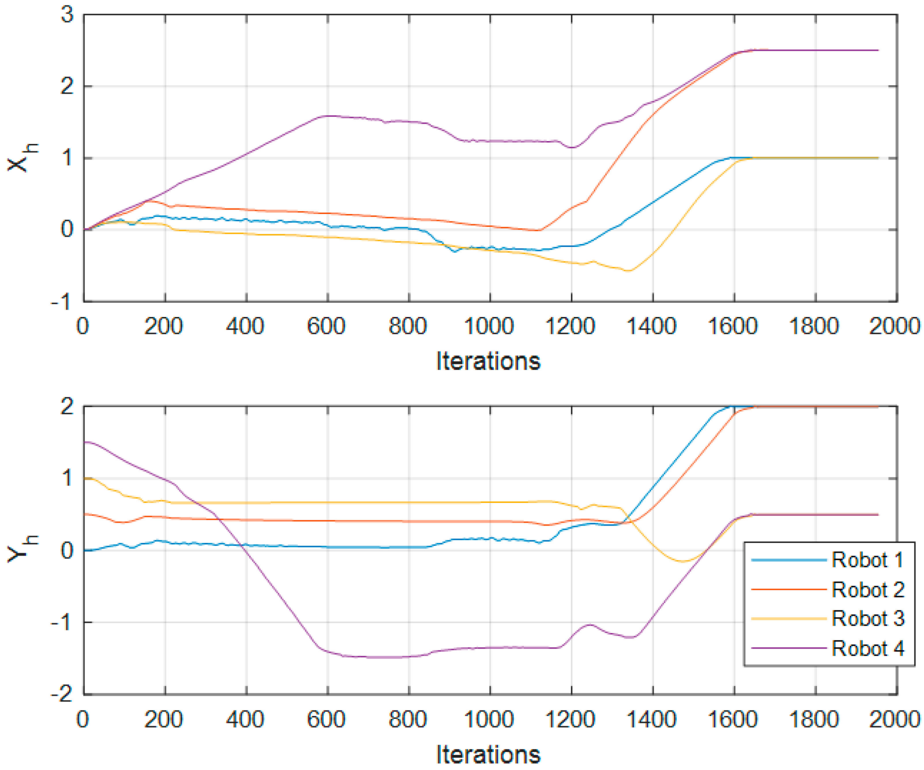

Figure 8. Trajectories along iterations in the 4-robot simulation.

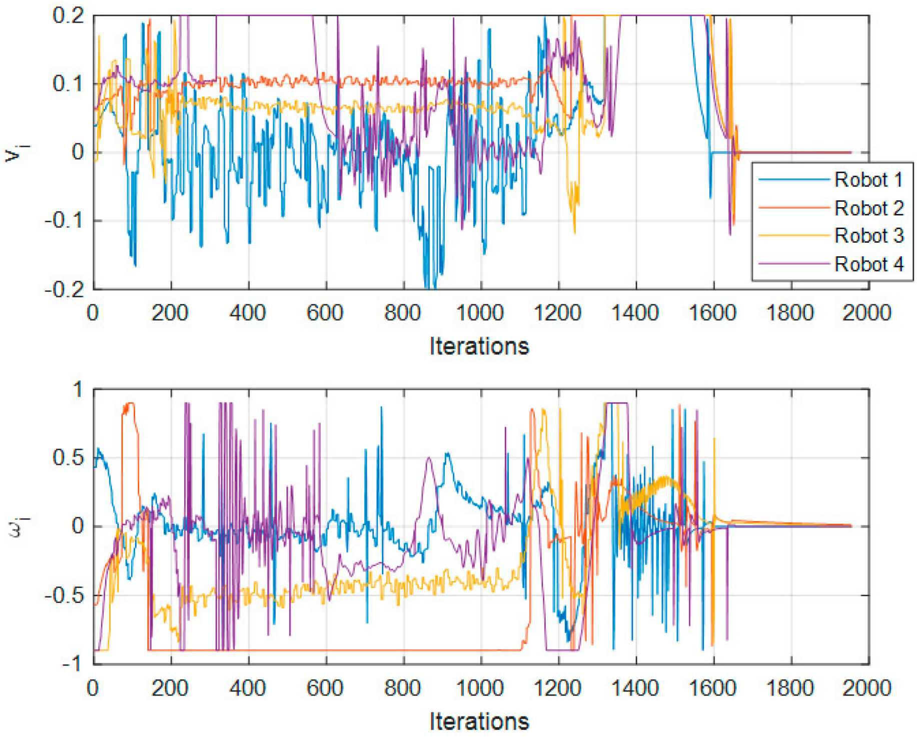

Figure 9. Input values for velocities along iterations in the 4-robot simulation.

In this test, 1000 iterations are roughly equivalent to 40 s with computer equipment with the following characteristics: Intel® Core™ i7-4770 CPU @3GHz under Ubuntu 16.04 operative system.

In Figure 7 it is observed that the robots go to the desired position, trying to maintain the formation as they advance to the objective; that is why Robot 4 goes far down the Y axis before positioning itself in its objective, as that is its place in respect to the other robots. We can also see a few oscillatory movements which are the result of the repulsion input term. In Figure 8, we can observe no collision between robots. These observations can also be made in the corresponding Figures for each test.

Next, for the 6-robot simulation, Figure 10 shows the trajectory followed by the robots in the simulation environment.

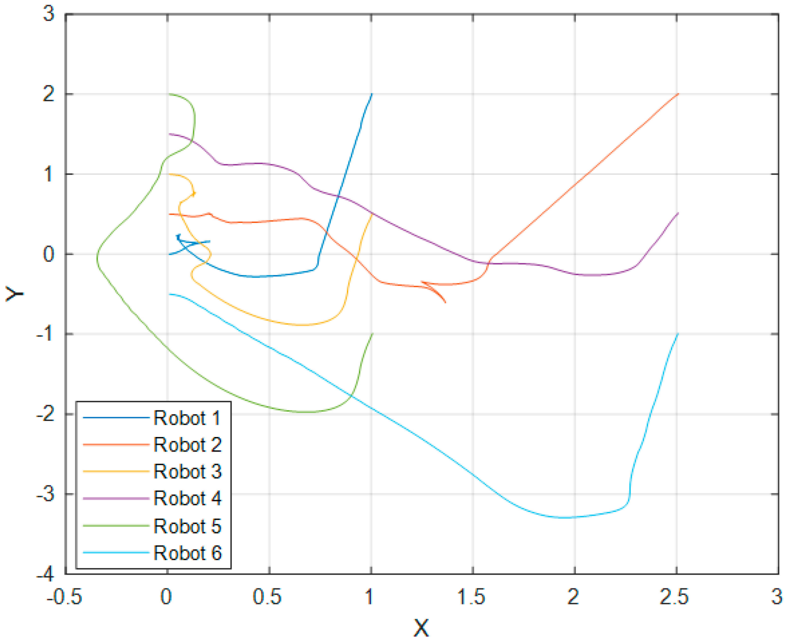

Figure 10. Trajectories in the 6-robot simulation.

The values of x_{hi}, y_{hi}, v_i, and ω_i for this test are plotted in Figures 11 and 12.

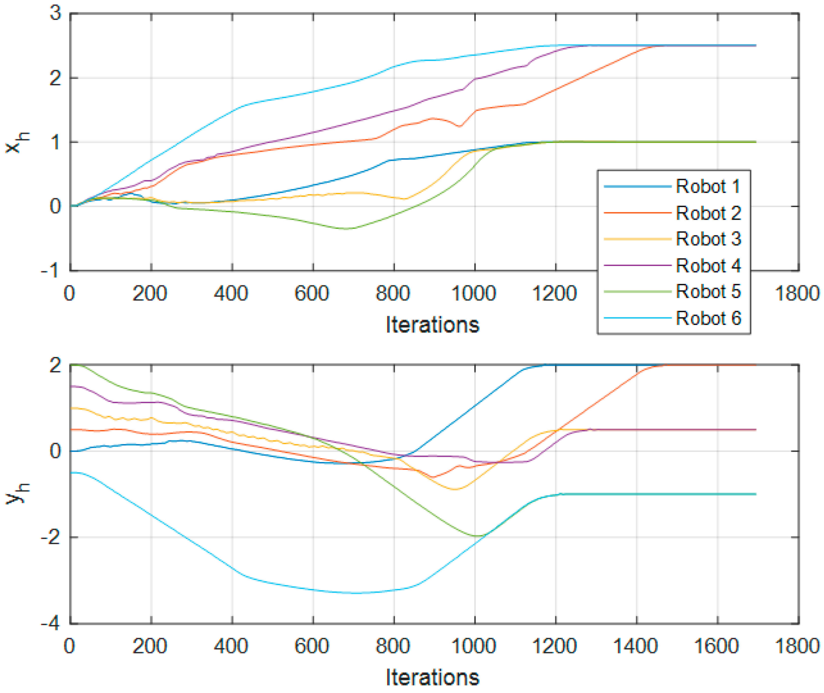

Figure 11. Trajectories along iterations in the 6-robot simulation.

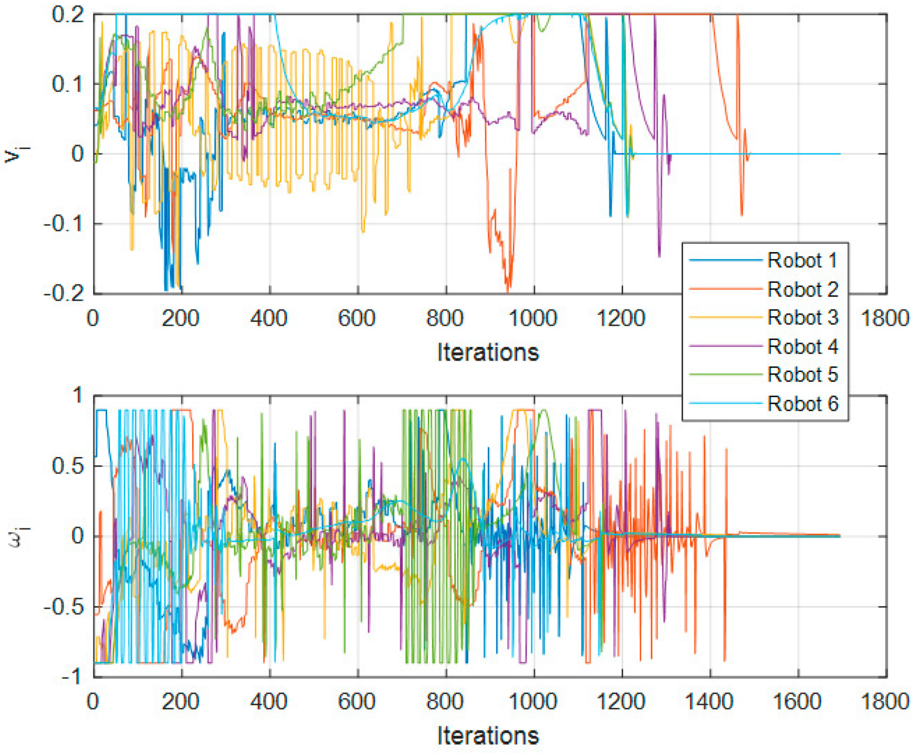

Figure 12. Input values for velocities along iterations in the 6-robot simulation.

In respect of the 12-robot test, Figure 13 shows the trajectory followed by the robots in the simulation environment.

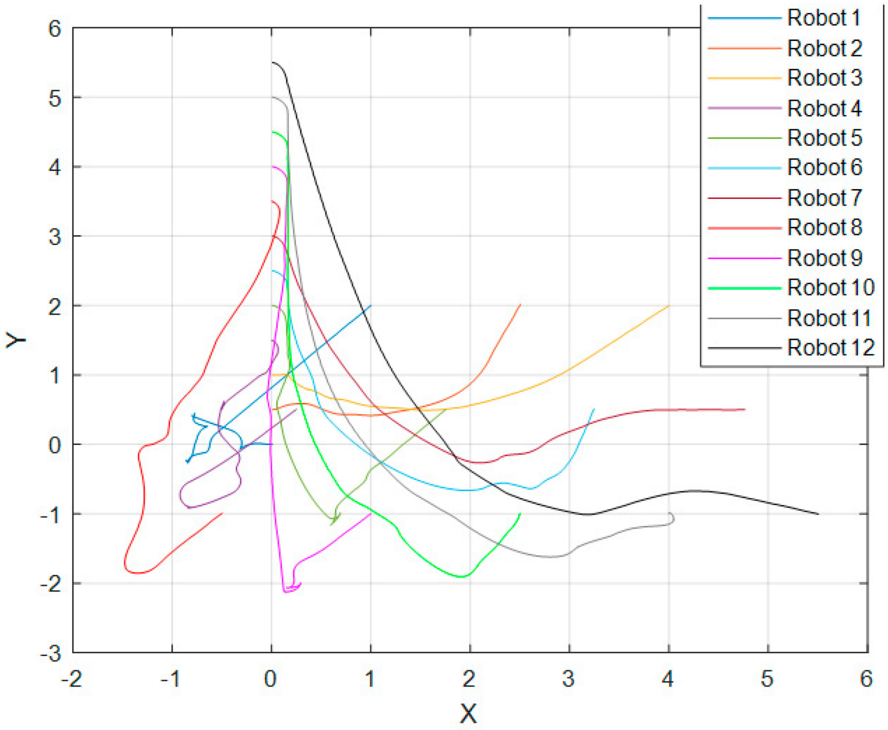

Figure 13. Trajectories in the 12-robot simulation.

The values of x_{hi}, y_{hi}, v_i, and ω_i are plotted in Figures 14 and 15.

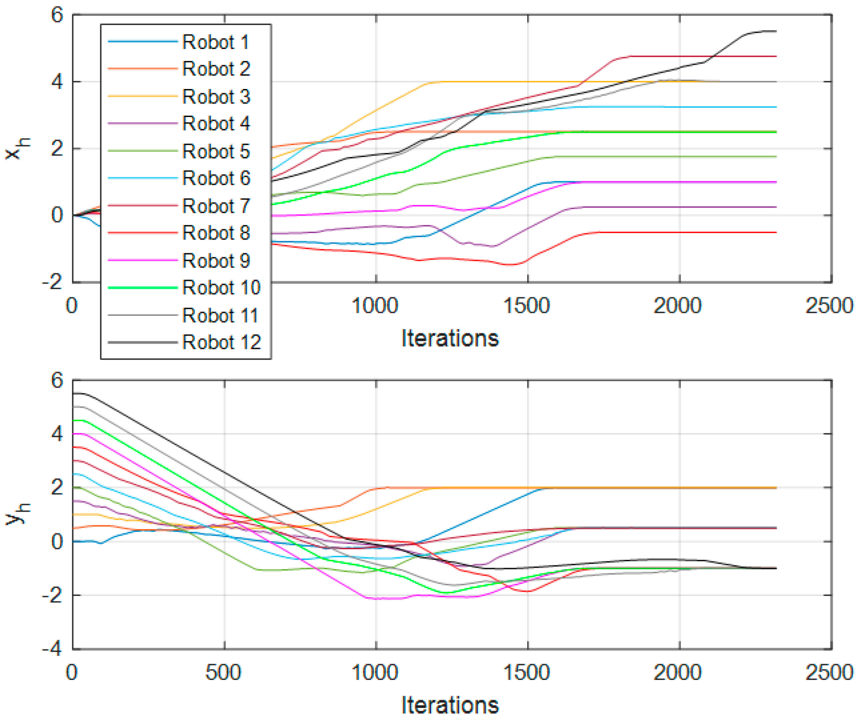

Figure 14. Trajectories along iterations in the 12-robot simulation.

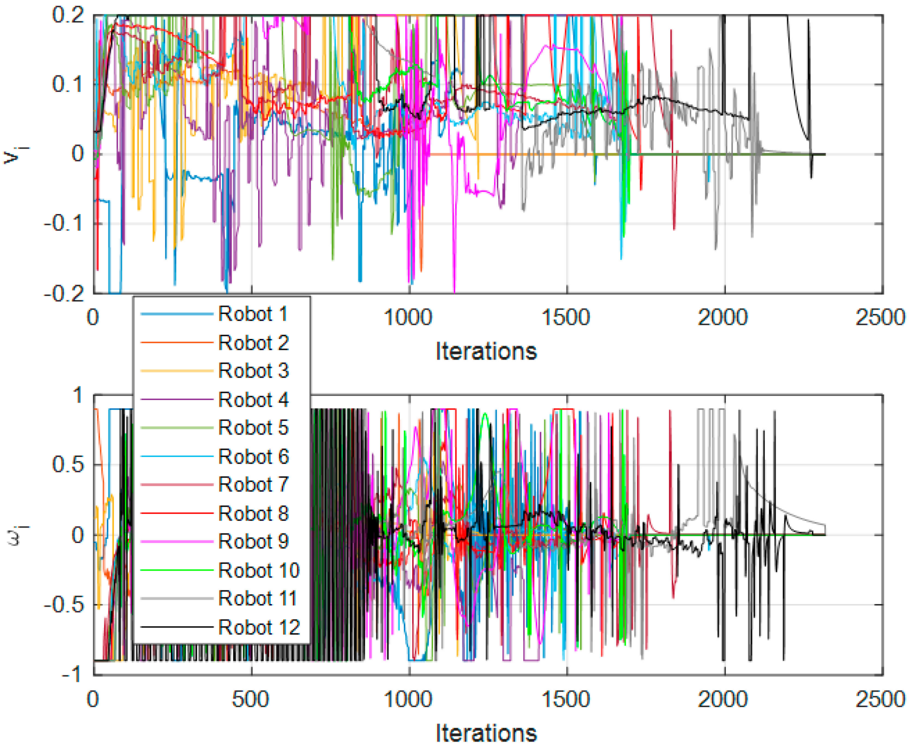

Figure 15. Input values for velocities in the 12-robot simulation.

Figures 9, 12 and 15 show the control signals v_i and ω_i of each of the nodes; these figures can be compared against Figures 8, 11 and 14, respectively. Figures 8, 11 and 14 show the nodes changing their position through time (iterations). In Figures 9, 12 and 15, the saturation of signals is noticeable, resulting in a saturation imposed by us to protect the equipment; however, even with this saturation, the proposed approach achieved the desired reference point and formation.

We continued with the experimental tests using the Turtlebot3® Waffle Pi Robots. First, we present the results for the 4-robot test, where Figure 16 shows the trajectory of the robots.

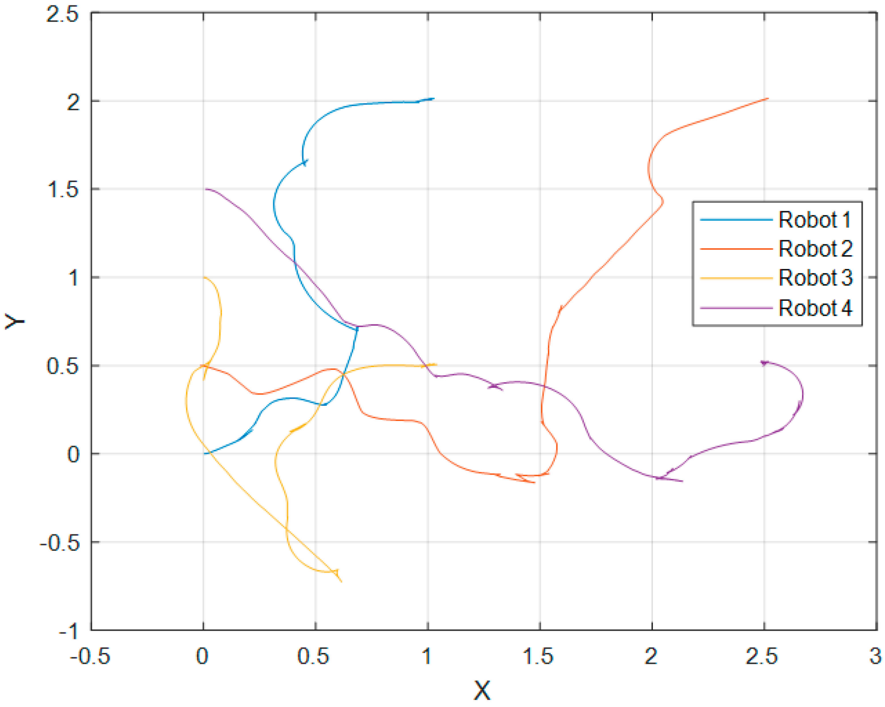

Figure 16. Trajectories in the 4-robot real-world test.

The values x_{hi}, y_{hi}, v_i, and ω_i for this test are plotted in Figures 17 and 18.

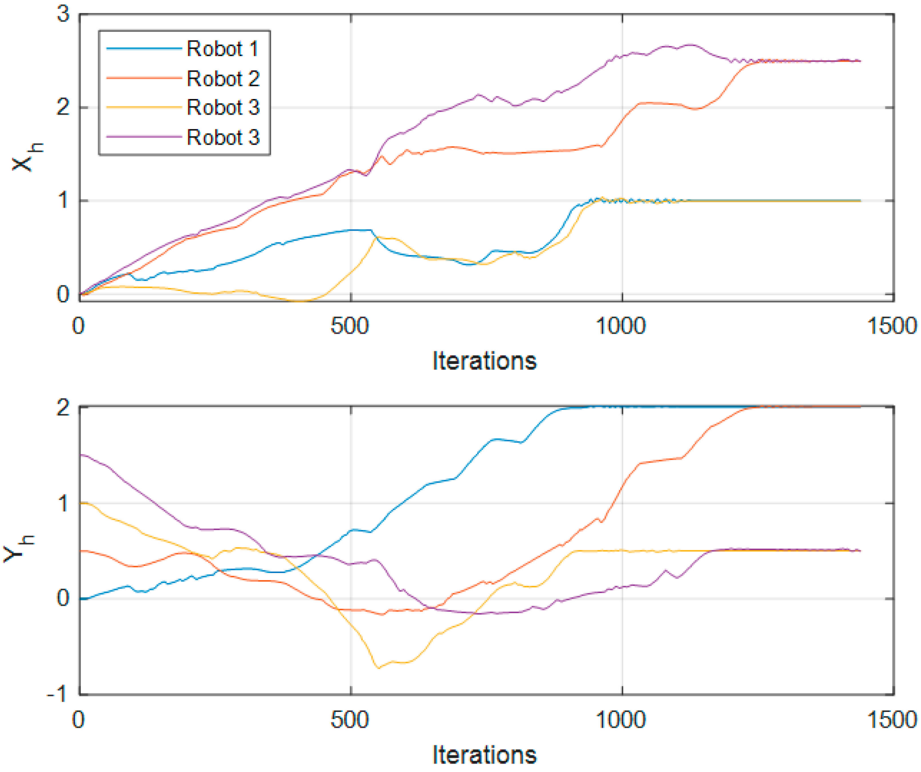

Figure 17. Trajectories along iterations in the 4-robot real-world test.

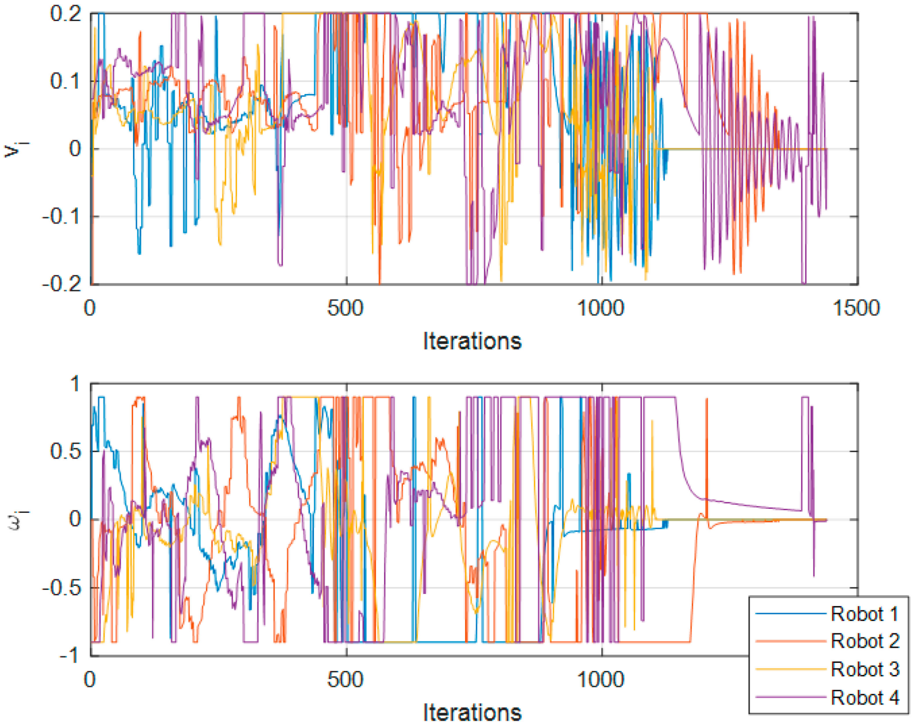

Figure 18. Input values for velocities along iterations in the 4-robot real-world test.

For the 6-robot real experimental test, Figure 19 shows the trajectory followed by the 6 robots.

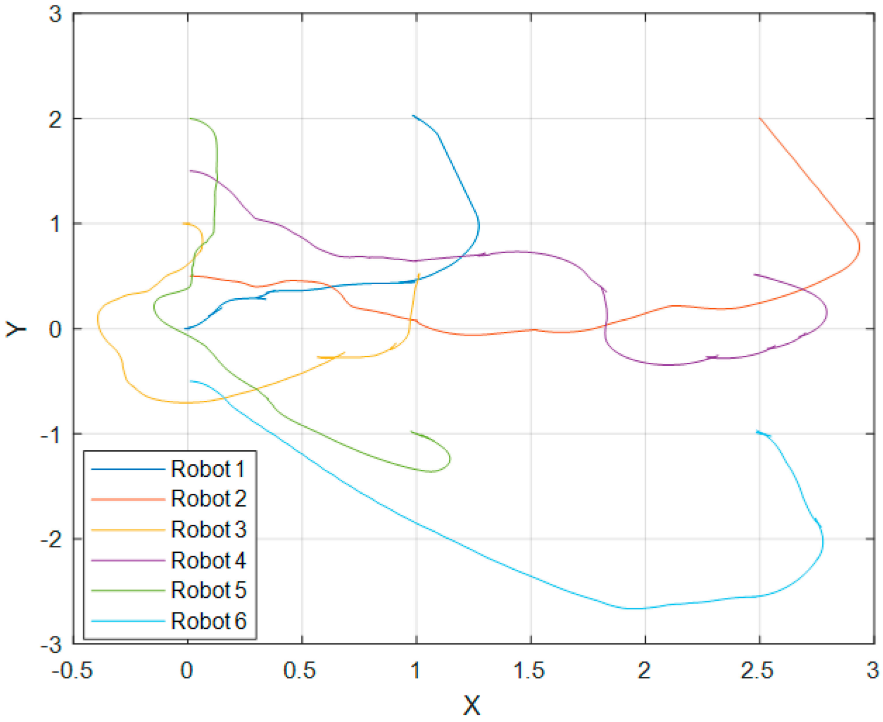

Figure 19. Trajectories in the 6-robot real-world test.

Real test values x_{hi}, y_{hi}, v_i, and ω_i are plotted in Figures 20 and 21.

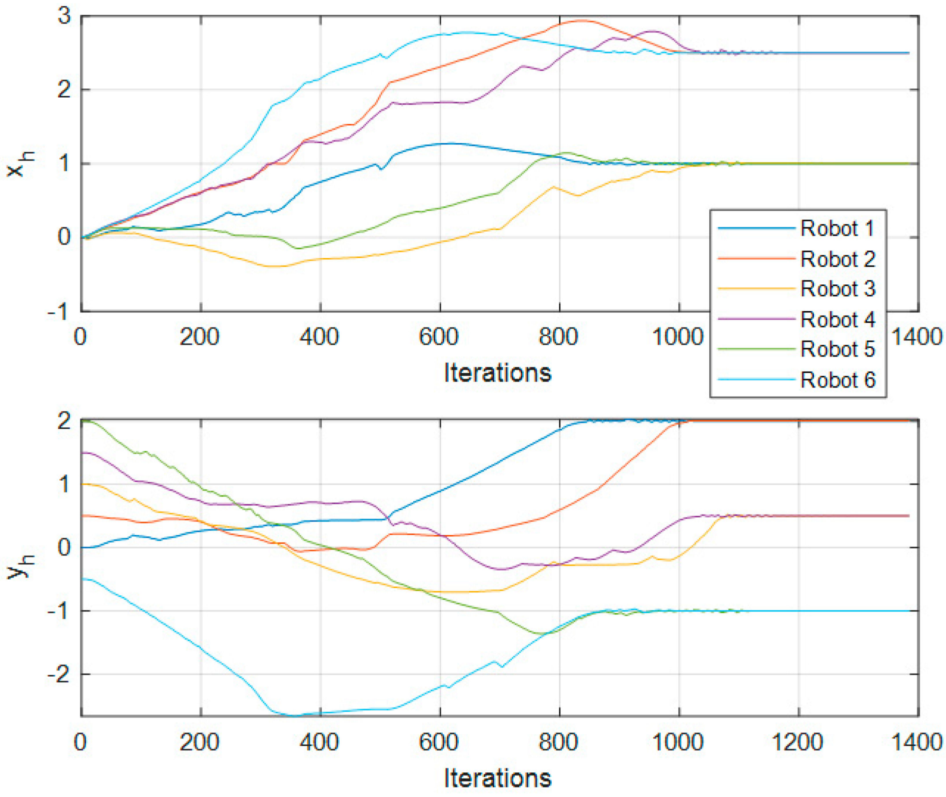

Figure 20. Trajectories along iterations in the 6-robot real-world test.

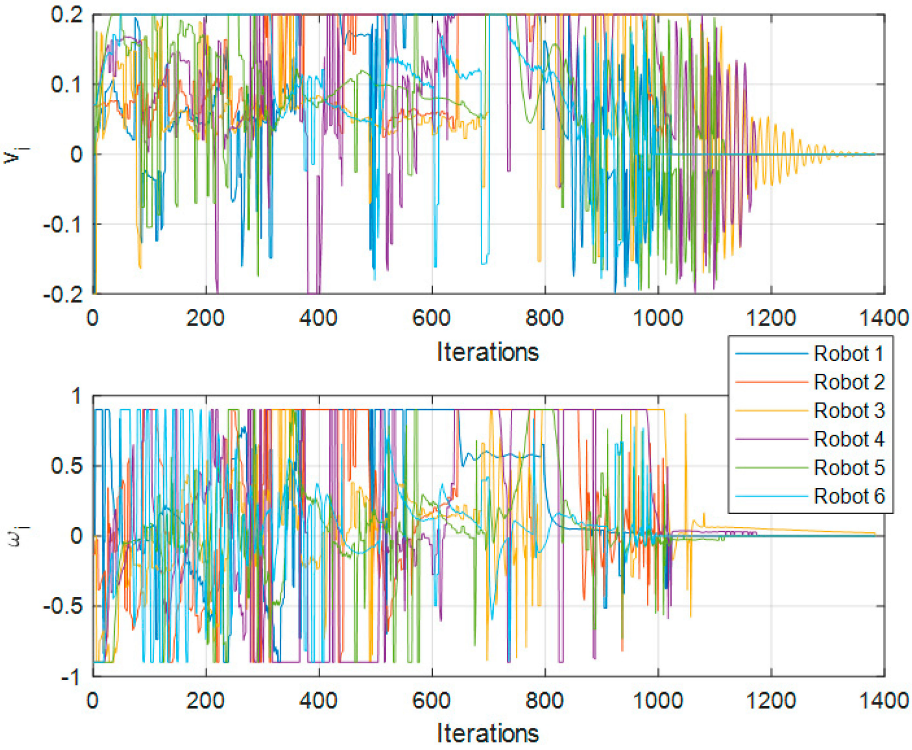

Figure 21. Input values for velocities in the 6-robot real-world test.

Even though pinning control is seen as a decentralized control in the complex network field, the implementation of the proposed scheme is presented as a centralized approach, considering that all computation is left to one computer; however, this can be changed by letting each robot carry out its own control computation, with the only requirement of having each other's information. More than one robot can be selected for pinning control, and not necessarily all the robots need information from all others, not even from the leader, as seen in the 12-robot simulation.

5. Results Discussion

There are visible differences between the performance of simulation tests under the Gazebo environment and experimental tests with the actual hardware. Many motives explain these differences. Let us begin by discussing the Gazebo platform, a robot simulator environment used in industry and academia. Gazebo computes the robot's physical and sensory responses in a scenario similar to a real-world one. Even if Gazebo does not consider some real-world factors, it gives results that give us a general idea of the performance of the robots and the algorithms in a real-world application.

Next, the main difference between our simulation and the real-world test results. Gazebo simulations were performed locally; all the ROS packages never left localhost. In the experimental test case, the package passes through a local network via Wi-Fi communications involving loss of packages and delays affecting the performance. Moreover, the scenario where the robots moved is not a perfect flat surface as in the simulation; it has irregularities. Furthermore, the simulation does not consider robot wear conditions and non-model dynamics.

Another important point to discuss is that all nodes in the proposed work are differential-drive robots. However, using complex networks, it would be possible to combine different kinds of robots, including, for example, unmanned aerial vehicles and omnidirectional robots. Let us consider the studies [10,11] where non-identical chaotic nodes were used in the same complex network. Judging by their results, we could expect similar results for the proposed control scheme using different robotic platforms. However, the relevant variables to each of the different robots must be first defined.

The results in the previous sections contrast with those presented in other related works, in that the desired formation was achieved on a previously set point on the plane; additionally, using the LiDAR sensor ensures the avoidance of collisions between robots and other nonrobotic elements. Furthermore, other works mainly present results at the simulation level and with fewer robots. For instance, in the study [17], the authors proposed a formation control scheme that is not achieved on a specific point in space, collisions are only avoided between robots using the previously known positions of each other, and results were presented in simulation for nine robots. The study [18] is similar to [17], presenting simulation and experimental results, where a camera obtained the robot positions, and their experimental results used three robots. In [19], formation control was achieved without a final position reference, using four robots for experimental results, and OptiTrack cameras obtained their position. Work [20] shows the results of formation control only in simulation, and collision was avoided only between robots. In [21], a group of robots arrived at a certain point, but the robots appeared to overlap in simulation, which is not physically possible. Work [22] presents a cooperative control tracking scheme which, compared to our proposal, differs in that all the robots knew the position of each other to avoid collisions in that study, and it presents the use of a virtual point as a reference for the robots. In contrast, our proposal uses one robot as the reference point.

Comparative Analysis

This section presents comparative results between this work proposal and two different formation control schemes. First, we compare the proposed scheme of [21], where there is no collision avoidance. Second, we compare the proposed scheme of [17], where there is collision avoidance but no final reference set point.

For the first comparison test, Figure 22 shows a visual comparison of our proposed scheme (Figure 22a) and the scheme proposed in work [21] (Figure 22b). Both schemes reach the goal. It is important to note that in favor of the scheme of work [21], a final formation and position where collisions would not be a problem was previously selected for this test.

However, for most final formations and positions, it would be a task impossible to achieve by the scheme of [22]. The lack of collision avoidance in a real scenario is not something to be left unattended. Moreover, in work [22], every robot had a reference controller, which would get more challenging to design if a different, more complex controller is to be implemented in a future application, as it would be necessary to implement one for every robot.

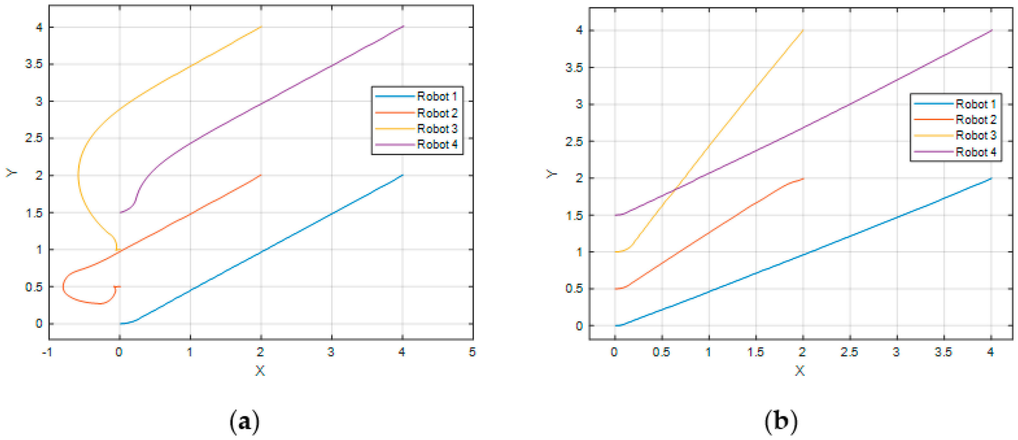

Figure 22. Movement of the robots in the plane in the comparison tests without collision avoidance: (**a**) Scheme proposed in this research; (**b**) scheme proposed in [21].

Figure 23 shows the trajectories of both schemes along the iterations, where it can be noted that there is not much difference in the final number of iterations for both schemes, meaning that they achieved the objective almost at the same time.

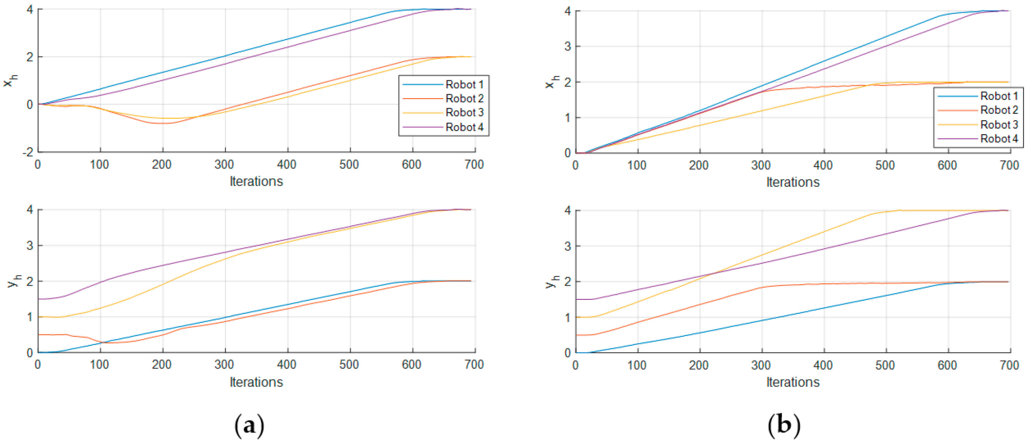

Figure 23. Trajectories along iterations in the comparison tests without collision avoidance: (**a**) Scheme proposed in this paper; (**b**) scheme proposed in [21].

Now the results of the second tests against the scheme of work [17] are presented. Figure 24 shows the movement of the robots in both schemes, this study's proposal (Figure 24a) and the scheme of [17] (Figure 24b). Since the scheme of [17] achieves the desired formation in a non-previous specified point, for our scheme, we omitted the final position term for just getting the desired formation.

Both schemes reach the goal; however, there is an oscillation when robots get near to achieving formation in the scheme of [17]. Collision avoidance in [17] is performed by following a particular counterclockwise turning movement. In contrast, the proposal presented in this paper follows a more natural path search in every direction as the LiDAR senses all of the surrounding environment, which makes it more practical for real applications. Figure 25 shows the trajectories where the oscillation at the end can also be observed.

It can also be seen that our proposal is slightly faster, judging by the number of iterations, even if we were to finish the test where oscillations begin.

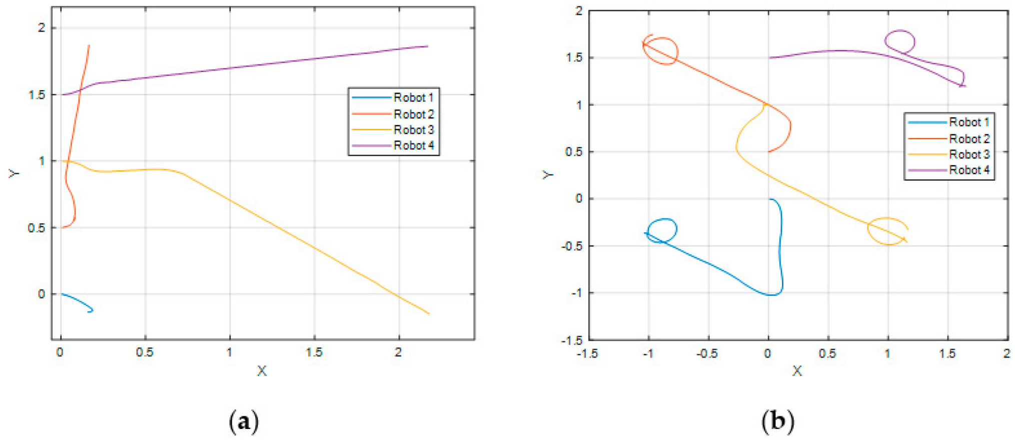

Figure 24. Movement of the robots in the plane in the comparison tests with collision avoidance but without an end point reference: (**a**) Scheme proposed in this paper; (**b**) scheme proposed in [17].

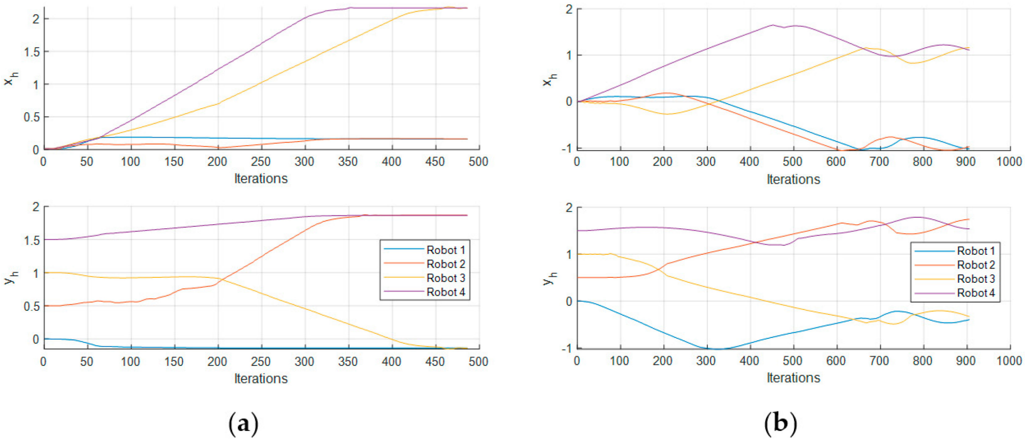

Figure 25. Trajectories along iterations in the comparison tests with collision avoidance but without an end point reference: (**a**) Scheme proposed in this paper; (**b**) scheme proposed in [17].

6. Conclusions

The complex network approach allows us to see the pattern formation of the mobile robot's objective as the couplings of synchronizing a complex network, and such synchronization is achieved using pinning control methodology. In the pinning control method, we only focus on one of the robots to set the desired final position of the formation, where not all the robots are necessarily connected to it, allowing us to combine different numbers of robots. Additionally, as mentioned in Section 5, combining different kinds of robots would be possible based on previous works focusing on complex networks where non-identical chaotic nodes were used in the same complex network. Some extra work would be necessary, and this idea is left for future studies.

Moreover, using pinning control in combination with complex networks presents an advantage if the position control law is thought to be changed to a more complicated one, since it would only be necessary to worry about modifying the control law for one of the robots.

The results of the presented tests show that the control objective was achieved for different formations, different numbers of robots, and different desired final positions, which is presented in both the simulation and experimental results. Comparative results with other works show that the proposed scheme would be a better option for real applications.

Besides using different robots in the complex network, more future studies can be proposed, including trajectory tracking in formation and dynamic topology of the form complex networks.

Author Contributions: Conceptualization, A.Y.A. and D.R.-R.; data curation, D.R.-R., J.D.R. and J.H.-B.; funding acquisition, A.Y.A. and M.P.-C.; investigation and software, D.R.-R. and J.D.R.; methodology, D.R.-R.; project administration and supervision, A.Y.A. and M.P.-C.; validation, J.H.-B.; visualization and writing, D.R.-R., J.D.R. and J.H.-B.; writing—review and editing, D.R.-R., J.D.R., J.H.-B. and A.Y.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by CONACYT Mexico, through Project FOP16-2021-01-319619.

Data Availability Statement: Data is available under request to the corresponding author.

Acknowledgments: The authors also thank Universidad de Guadalajara for its support in this research.

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

References

1. Shi, W.; Wang, Q.; Gong, L. Formation control of multiple mobile robots with prescribed performance. In Proceedings of the 2020 Chinese Control And Decision Conference (CCDC), Hefei, China, 22–24 August 2020. [CrossRef]
2. Issa, B.A.; Rashid, A.T. A Survey of Multi-mobile Robot Formation Control. *Int. J. Comput. Appl.* **2019**, *181*, 12–16. [CrossRef]
3. Latora, V.; Nicosia, V.; Russo, G. *Complex Networks: Principles, Methods and Applications*, 1st ed.; Cambridge University Press: Cambridge, UK, 2017. [CrossRef]
4. Erciyes, K. *Complex Networks: An Algorithmic Perspective*, 1st ed.; CRC Press: Boca Raton, FL, USA, 2017. [CrossRef]
5. Alfaro, A.; Morán, A. Leader-Follower Formation Control of Nonholonomic Mobile Robots. In Proceedings of the 2020 IEEE ANDESCON, Quito, Ecuador, 13–16 October 2020. [CrossRef]
6. Vaishnav, J.; Uday, A.B.; Poulose, T. Pattern Formation in Swarm Robotic Systems. In Proceedings of the 2018 2nd International Conference on Trends in Electronics and Informatics (ICOEI), Tirunelveli, India, 11–12 May 2018. [CrossRef]
7. Kiełczewski, M.; Kowalczyk, W.; Krysiak, B. Differentially-Driven Robots Moving in Formation—Leader–Follower Approach. *Appl. Sci.* **2022**, *12*, 7273. [CrossRef]
8. Xu, P.; Tao, J.; Xu, M.; Xie, G. Practical Formation Control for Multiple Anonymous Robots System with Unknown Nonlinear Disturbances. *Appl. Sci.* **2021**, *11*, 9170. [CrossRef]
9. Chen, G.; Wang, X.; Li, X. *Fundamentals of Complex Networks: Models, Structures and Dynamics*, 1st ed.; Wiley: Singapore, 2015.
10. Ríos-Rivera, D.; Alanis, A.Y.; Sanchez, E.N. Neural-Impulsive Pinning Control for Complex Networks Based on V-Stability. *Mathematics* **2020**, *8*, 1388. [CrossRef]
11. Alanis, A.Y.; Ríos-Rivera, D.; Sanchez, E.N.; Sanchez, O.D. Learning Impulsive Pinning Control of Complex Networks. *Mathematics* **2021**, *9*, 2436. [CrossRef]
12. Xiang, J.; Chen, G. On the V-stability of complex dynamical networks. *Automatica* **2007**, *43*, 1049–1057. [CrossRef]
13. Li, X.; Wang, X.; Chen, G. Pinning a complex dynamical network to its equilibrium. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2004**, *51*, 2074–2087. [CrossRef]
14. Cao, Y.; Yu, W.; Ren, W.; Chen, G. An Overview of Recent Progress in the Study of Distributed Multi-Agent Coordination. *IEEE Trans. Ind. Inform.* **2013**, *9*, 427–438. [CrossRef]
15. Shih, C.-L.; Lin, L.-C. Trajectory Planning and Tracking Control of a Differential-Drive Mobile Robot in a Picture Drawing Application. *Robotics* **2017**, *6*, 17. [CrossRef]
16. Klancar, G.; Zdesar, A.; Blazic, S.; Skrjanc, I. *Wheeled Mobile Robotics: From Fundamentals Towards Autonomous Systems*, 1st ed.; Butterworth-Heinemann: Oxford, UK, 2017.
17. Flores-Resendiz, J.F.; Aranda-Bricaire, E. Cyclic Pursuit Formation Control Without Collisions in Multi-Agent Systems Using Discontinuous Vector Fields. In Proceedings of the XVI Latin American Control Conference (CLCA), Cancún, Mexico, 14–17 October 2014.
18. Hernandez-Martinez, E.G.; Aranda-Bricaire, E. Convergence and Collision Avoidance in Formation Control: A Survey of the Artificial Potential Functions Approach. In *Multi-Agent Systems—Modeling, Control, Programming, Simulations and Applications*, 1st ed.; Alkhateeb, F., Maghayreh, E.A., Doush, I.A., Eds.; IntechOpen: London, UK, 2011; pp. 103–126. [CrossRef]
19. Flores-Resendiz, J.F.; Aranda-Bricaire, E.; González-Sierra, J.; Santiaguillo-Salinas, J. Finite-Time Formation without Collisions for Multiagent Systems with Communication Graphs Composed of Cyclic Paths. *Math. Probl. Eng.* **2015**, *2015*, 948086. [CrossRef]
20. Flores-Resendiz, J.F.; Aranda-Cricaire, E.A. General Solution to the Formation Control Problem without Collisions for First-Order Multi-Agent Systems. *Robotica* **2020**, *38*, 1123–1137. [CrossRef]
21. Cao, Y.; Ren, W. Distributed Coordinated Tracking with Reduced Interaction via a Variable Structure Approach. *IEEE Trans. Autom. Control* **2012**, *57*, 33–48. [CrossRef]
22. Morales, A.; Nijmeijer, H. Merging Strategy for Vehicles by Applying Cooperative Tracking Control. *IEEE Trans. Intell. Transp. Syst.* **2016**, *17*, 3423–3433. [CrossRef]